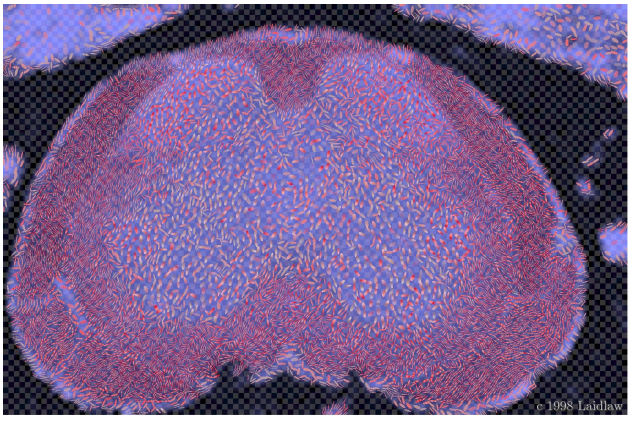
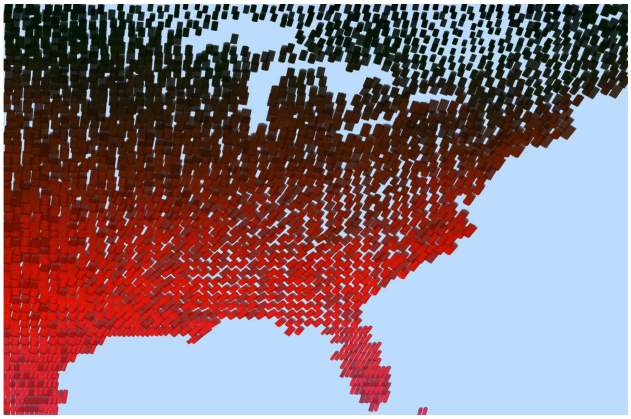
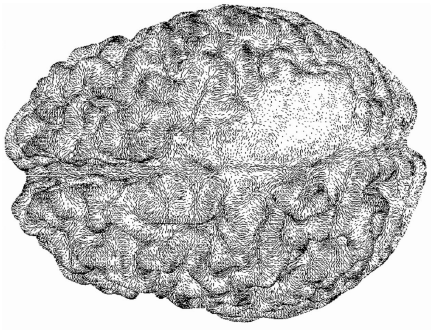
Nonphotorealistic Rendering in Scientific Visualization

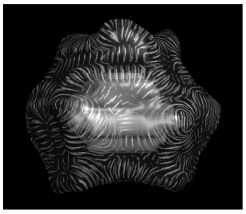
Christopher G. Healey, Victoria Interrante, davidkremers, David Laidlaw, Penny Rheingans

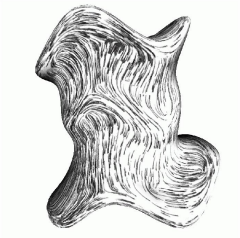
Course #32 SIGGRAPH 2001
Los Angeles, California August 12-17, 2001

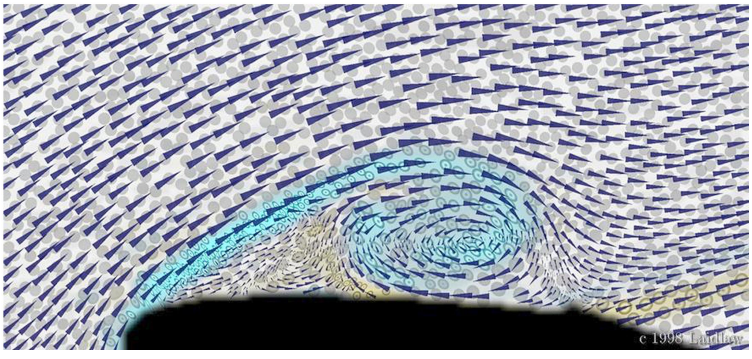












Course Summary

The goal of this course is to introduce participants to the wealth of visualization inspiration available from art and art history. How people perceive an image can have a profound effect on the meaning they attach to that image. A compelling example is the artist's use of painterly techniques that harness our perception to evoke a specific emotional response. This course surveys a number of important issues in nonphotorealistic rendering and visual perception, then discusses their direct relevance to computer graphics and scientific visualization through a series of descriptions, examples, and practical applications. Topics address questions like: Which artistic techniques can we apply during image generation? How can these techniques be used to enhance the expressive power of traditional methods like volume visualization or line integral convolution? How does the correspondence between artistic properties and human perception allow us to produce painterly renditions of complex information spaces? Answers to these questions are important to graphics researchers and practitioners who want to construct nonphotorealistic images the convey an intended meaning or perceptual effect when viewed by their audience.

Course Organizers

Dr. Christopher G. Healey is an Assistant Professor of Computer Science at North Carolina State University. He received a BMath at the University of Waterloo, Canada in 1990 and a MSc and PhD at the University of British Columbia, Canada in 1996. During his PhD, he was supervised by Dr. Kellogg S. Booth (Department of Computer Science) and Dr. James T. Enns (Department of Psychology). Prior to starting at North Carolina, Dr. Healey completed a two year postdoctoral fellowship in computer graphics with Dr. Carlo Séquin at the University of California, Berkeley. Dr. Healey's dissertation studied methods for displaying effectively large, multidimensional datasets during scientific visualization. This work investigated methods for exploiting the low-level human visual system for information representation. His current research focuses on the use of visual features like color, texture, and apparent motion for analyzing multidimensional data. Dr. Healey has taught under-graduate and graduate courses in computer graphics and visualization, and has lectured in course at SIGGRAPH 98 and 99.

Dr. Victoria Interrante is an Assistant Professor in the Department of Computer Science and Engineering at the University of Minnesota. She received her PhD from the University of North Carolina, Chapel Hill in 1996, where her dissertation research, coadvised by Drs. Henry Fuchs and Stephen Pizer, focused on the design of perceptually inspired artistic techniques for improving the comprehensibility of layered transparent surfaces. From 1996–1998 she was a staff scientist at ICASE, a non-profit research center operated by the Universities Space Research Association at NASA Langley, where she worked on developing perceptually based methods for effectively visualizing 3D flow. Her current research focuses on the application of insights from visual perception, art and illustration to the design of more effective techniques for visualizing 3D data, in particular the study of shape and depth perception, shape-based feature extraction, and the representation of 3D shape through texture. Dr. Interrante teaches graduate and undergraduate courses in visualization and computer graphics at the University of Minnesota, and has organized and lectured in tutorials on perceptual issues in graphics and visualization at SIGGRAPH 97, 98, and 99, and IEEE Visualization '96, '97, and '98.

davidkremers is an artist inspired by the confluence of art and science, and has recently been appointed the Caltech Distinguished Conceptual Artist in Biology. He grew the first paintings from genetically engineered bacteria and his work combining living organisms and digital media has evolved into biospace station concepts and visual information systems for biotechnology research. His art has generated numerous gallery and museum exhibitions in the United States, Germany, Belgium, Denmark, and Austria. His artwork may be viewed in real time at public collections of the Denver Art Museum, the San Francisco Museum of Modern Art, the Eli Broad Family Foundation, and the Panza Collection.

David Laidlaw is the Robert Stephen Assistant Professor in the Computer Science Department at Brown University. His research centers around applications of visualization, modeling, computer graphics, and computer science to other scientific disciplines. He is working with researchers in, for example, archaeology, developmental neurobiology, medical imaging, orthopedics, art, cognitive science, remote sensing, and fluid mechanics to develop new computational applications and to understand their strengths and weaknesses. Particular interests include visualization of multivalued multidimensional imaging data, comparisons of virtual and non-virtual environments for scientific tasks, and applications of art and perception to visualization. His PhD in Computer Science is from Caltech, where he also did post-doctoral work in the Division of Biology.

Penny Rheingans is an assistant professor of Computer Science and Electrical Engineering at the University of Maryland, Baltimore County. She received a Ph.D in Computer Science from the University of North Carolina, Chapel Hill and a BS in Computer Science from Harvard University. Dr. Rheingans has also held positions as an assistant professor of Computer and Information Science at the University of Mississippi, a Visualization Specialist for Lockheed Martin at the US EPA Scientific Visualization Center (where she developed tools for the effective display of environmental data), and a research assistant at UNC (where she developed tools for the visualization of molecular surfaces, the design and manipulation of color mappings for bivariate information, and the utilization of virtual reality technology to conduct interactive walkthroughs of very large architectural databases). Dr. Rheingans has over thirty published works in such locations as the Proceedings of the IEEE Visualization Conference, IEEE Computer Graphics and Applications, the Eurographics/IEEE TCVG Symposium on Visualization, the IBM Journal of Research and Development, and the SIGGRAPH Film Show, as well as included as chapters of various books. Her current research interests include the visualization of data with associated uncertainty, multivariate visualization, information visualization, perceptual and illustration issues in visualization, dynamic and interactive representations and interfaces, and the experimental validation of visualization techniques.

Course Schedule

First Session (1 hour, 25 minutes)

8:30 Introduction (Laidlaw, 10 minutes)

1. The scientific method
2. Where visualization fits in
3. Perception and art in visualization
4. A roadmap for this course

8:40 Artistic Enhancement in Scientific Visualization (Interrante, 75 minutes)

1. Motivation: when is a drawing more useful than a photograph?
2. Goals: enhancing the important features; minimizing the extraneous details; hierarchically structuring the attentional focus
3. Historical examples from scientific illustration
4. Recent applications: emphasizing 3D shape with texture

Coffee Break (15 minutes)

Second Session (1 hour, 30 minutes)

10:10 Participant Painting Session, Introduction (Laidlaw, davidkremers, 45 minutes)

1. Introduction to hands-on painting session
2. Purpose and goals of hands-on painting session
3. Presentation and explanation of example vector and scalar fields

10:55 Participant Painting Session (davidkremers and other instructors, 45 minutes)

1. Participant painting session with assistance and real-time feedback from course instructors
2. Presentation and critique of individual participant results

Lunch Break (1 hour, 20 minutes)

Third Session (2 hours)

1:00 Participant Painting Session, cont'd (davidkremers and other instructors, 45 minutes)

1. Participant painting session with assistance and real-time feedback from course instructors
2. Presentation and critique of individual participant results

1:45 Impressionism, Perception, and Multidimensional Visualization (Healey, 75 minutes)

1. Discussion of low-level human vision
2. Perceptual properties of color and texture patterns
3. Painterly styles in Impressionist art
4. Correspondence between low-level visual features and painterly styles
5. Nonphotorealistic techniques for multidimensional visualization

3:00 Artistic Inspiration for Volume Rendering (Rheingans, 35 minutes)

1. Expressive techniques in technical illustrations
2. Artistic techniques for translucent phenomena

Coffee Break (15 minutes)

Fourth Session (1 hour, 20 minutes)

3:50 Expressive Volume Rendering (Rheingans, 40 minutes)

1. Features in volume models
2. Techniques for expressive volume rendering
3. Implementing expressive rendering for volume models

4:30 Beyond Perceptual Psychology with Oil Painting (Laidlaw and davidkremers, 40 minutes)

1. A sequence of painters
2. Deconstructing van Gogh
3. Layering of brush strokes for data representation
4. Textures, icons, and geometry - a lesson in multiple scales
5. Some open problems

5:10 Conclusions and Wrap-up

Table of Contents

Introduction

David Laidlaw
Brown University

This course will explore where scientists can find inspiration for scientific visualization. We will discuss examples from art and perceptual psychology. Art is a natural source for visual inspiration. Perceptual psychology is less obvious, but sensible, considering that it attempts to understand how the human visual system "sees."

Throughout the day we will see examples of art-motivated visualization, look at art and understand how and why it works, experiment with making some art of our own, and talk among ourselves about what works and what doesn't.

The audience for this course is likely to be varied. The approaches that we will talk about are inherently multidisciplinary (and hence fun and exciting!). We hope that folks in the audience who are artists artists will see some new applications for what they have learned works visually. We hope that scientists in the audience will be inspired by some of the art that they see and think about how ideas from the art might be applied to their scientific problems. We hope that software developers in the audience might see new visual representations that they can build into visualization tools. Finally, we hope that some cross fertilization will take place between members of the audience and with the presenters as well. During the day there will be several times when you will be looking at art, thinking about it, making it, and talking to your compatriots. Collaboration is encouraged.

1.1 Scientific Visualization

The organizing principle behind scientific visualization is the scientific method. Broadly speaking, scientists pose hypotheses about how the world works, design tests to evaluate those hypotheses, and then carry out the tests. Occasionally, those tests work out as expected, but, much more often, they lead to surprises. From the resulting information, expected or now, new hypotheses emerge, and new tests are designed and carried out. Through this iterative process, scientists gain insight into the phenomena they are studying.

Visualization can play an important role. Figure [dhl cycle figure] shows an iterative scientific visualization cycle that includes visualization. The cycle typically begins in the lower right with a hypothesis or model of the world. Cycling to the left, a scientist generates measurements of the world, say with a medical imaging device, a camera, or by running a simulation. The resulting data is then analyzed at the top. The analysis may involve comparing simulated results with physical measurements, searching for patterns in measured or simulated data, or comparing measurements across subjects. In all cases, the overarching goal of the visualization process is to evaluate the scientific hypothesis initially posed.

This goal comes with some implications. First, it implies that the visualization needs to show certain things. This constraint makes the visualization more challenging, but it also provides something of an evaluation metric – the performance of researchers on their scientific tasks can be considered a measure of the effectiveness of the visualization methods. This is important, because it colors the entire visualization development and evaluation process.

1.2 Art and Perception for Inspiration

With clearly defined goals, then, we have some hope of developing effective visualization tools. Much of this course will focus on finding ideas and inspiration in art and perceptual psychology and applying them to scientific visualization tasks. Through several centuries, artists have evolved a tradition of techniques to create visual representations for particular communication goals. Art history provides a language for understanding that knowledge.

Beyond inspiration, perceptual psychology also brings a second set of knowledge to bear on scientific visualization problems. Evaluating the effectiveness of visualization methods is difficult because, not only are the goals difficult to define and codify, tests that evaluate them meaningfully are difficult to design and execute. These evaluations are akin to evaluating how the human perceptual system works. While this may seem outside the search for inspiration, it is closely coupled, because the inspiration has to lead to effective methods, which are otherwise impossible to evaluate.

Reprinted from "Illustrating Transparency: Communicating the 3D Shape of Layered Transparent Surfaces via Texture", PhD dissertation, University of North Carolina at Chapel Hill, 1996.

5.2: How artists represent form with line

Line patterns have been used for centuries to represent three dimensional figures in flat drawings. It therefore seems appropriate to begin this investigation into promising techniques for using line to represent shape by briefly reviewing what artists and illustrators have written on this subject over the years in textbooks describing their pen-and-ink methods.

Thickness, spacing and orientation appear to be the three most important and often discussed line characteristics. Sullivan [1922] demonstrates how the relative weight of the line used to surround an object determines whether the outline is perceptually grouped with the object or with the background, or, alternatively, whether the line stands apart from both and demands recognition as an entity in its own right. He goes on to describe how the amount of "depth" conveyed in an image will vary considerably depending on the effectiveness with which the outline is drawn; in the cases where the line assumes its own identity, he says, an extra measure of flatness will be imparted to the image which is, in most cases, undesirable. It should be noted that Sullivan is referring here to outlines which are drawn for the sole purpose of separating figure from ground, lines which have no intrinsic, viewpoint-invariant meaning on the surface itself.

Figure 5.4: An illustration demonstrating the potential significance of stroke characteristics, from [Pitz 1957]. A dramatically different effect is achieved in these two images although little more is changed than the lengths and directions of the lines used to represent the tonal values. Of particular note here is the dynamism or sense of radiant flow in the image on the left in contrast with the relative serenity or stillness in the same scene in the depiction on the right.

When lines are used to represent surfaces areas rather than boundaries of form, the thickness and spacing of the individual strokes is often varied over the image depending on the "tone" desired in each region. Gradations in tone can be represented in many different ways: by changing the spacing between strokes of equal thickness, by changing the thicknesses of strokes whose central axes are more or less equally spaced, or by varying the width of each stroke along its length. Sullivan [1922] emphasizes the importance of proper spacing, a theme that is reiterated in almost every book describing the art of drawing:

> *... lines may be placed so far apart relative to ... the space occupied that they hardly appear as a tone, but as individual lines, independent of each other except insofar as their parallelism is marked. They then take up a position which may challenge the supremacy of the main constructive lines of the drawing, so that the adjective is more forcible than the noun, as in a common and senseless form of swearing, or they may appear, not as belonging to and suggesting surface or an intangible shade or shadow, but as something positive, either as construction or pattern.*

Pitz [1957] cautions that when the lines comprising a tone are spaced 1/16" apart or more, the eye is more likely to be conscious of them as individual elements. Guptill [1976] suggests an even closer spacing, writing that, as a general rule, the strokes used to represent a tone should be placed 1/32" (0.8mm) apart if the drawing is to be viewed from a distance of about 2ft. In the samples of work by medical illustrators shown at the end of this section, we can see that relatively larger line spacings are common.

Although it is clear that styles vary, it is important nevertheless to take somewhat to heart the admonishment against representing strokes, or any texture elements, in such a way that they take on a character of their own. Our primary intention in adding opaque markings to transparent surfaces is to allow the form of the depicted objects to be more easily and intuitively understood. To the extent that the individual texture markings are unduly prominent, they may do more harm than good, distracting the attention of the observer, confusing the appearance of the picture and adding visual "noise" that detracts from rather than enhances the overall effectiveness of the presentation. This may be one reason that the circular texture elements are so rarely employed by scientific and medical illustrators for the purpose of more explicitly depicting surface shape.

While variations in tone are important for representing the patterns of light and shade that are defined by illumination, this is not the only way in which they are commonly used. An important concern in any illustration is emphasis, and Guptill [1976] demonstrates how to define the focus of attention in a drawing by varying the contrast with which different parts of the scene are depicted (and, concomitantly, varying the amount of detail that is represented in each part). The lengths of the individual strokes and the precision with which they are drawn and placed on the page (including the regularity of the patterning and the amount by which the individual strokes fluctuate in direction) also influence the impression conveyed by a drawing.

In terms of the effectiveness with which shape and depth are portrayed, however, the most important line characteristic by far appears to be *orientation*. (This should not be interpreted as meaning that shading somehow plays a lesser role, but only as an acknowledgment of the fact that tonal variations can be equivalently conveyed by a multitude of different particular line styles. The essential shape of an object, both within and apart from the shading, is conveyed most clearly by the orientations of the lines that fill and bound its form.) Sullivan [1922] is particularly adamant about the importance of stroke direction in line drawings, saying

> *...all a fastidious spectator's pleasure in a drawing may be destroyed by a wrong use of direction in a space of modelling, no matter how fine the lines composing it may be, or how pretty the general effect.*

There are a number of different commonly-accepted techniques for defining line orientation. Possibly the simplest approach, and incidentally the one that communicates the least amount of information about shape and depth, is to not vary the line orientation at all but merely to consistently apply the strokes in a uniform vertical or horizontal direction. This is the effect achieved in computer graphics when an image-space texture is applied. Although the orientation of the lines is independent of the three-dimensional space occupied by the depicted objects, our perception of form is nevertheless affected by the choice of line direction. Pitz [1957] illustrates and explains the effects of a variety of different directed stroke techniques; figure 5.5 shows excerpts from this work, demonstrating how the use

of vertical lines can emphasize height, horizontal lines can emphasize width, and lines that follow the edges of the forms can emphasize their volume.

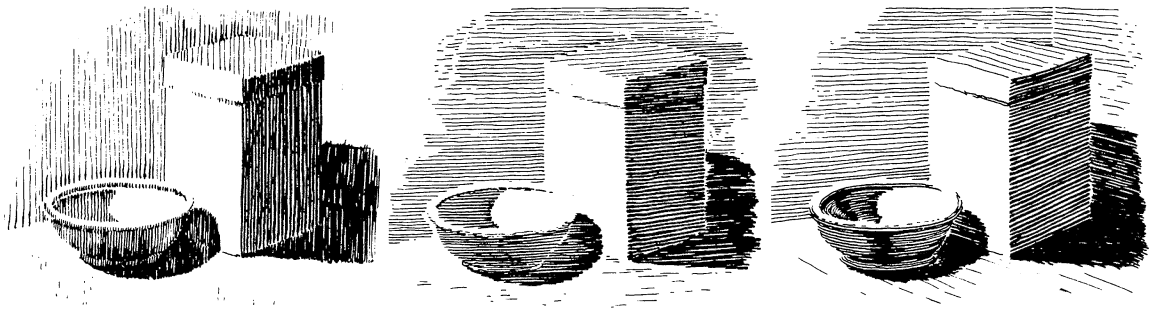


Figure 5.5: An illustration of the biasing effects of various stroke textures, from [Pitz 1957]. Vertical lines emphasize height, horizontal lines emphasize width, and lines that follow the outer edges of the forms emphasize volume.

Guptill [1976] describes how, even for a very simple scene, there seem to be an almost infinite variety of different ways in which lines can "follow the form" of a surface. The upper group of illustrations in figure 5.6 depicts, for example, his demonstration of eight different combinations of edge-following uni-directional line textures that can be applied to the visible faces of a shaded cube.

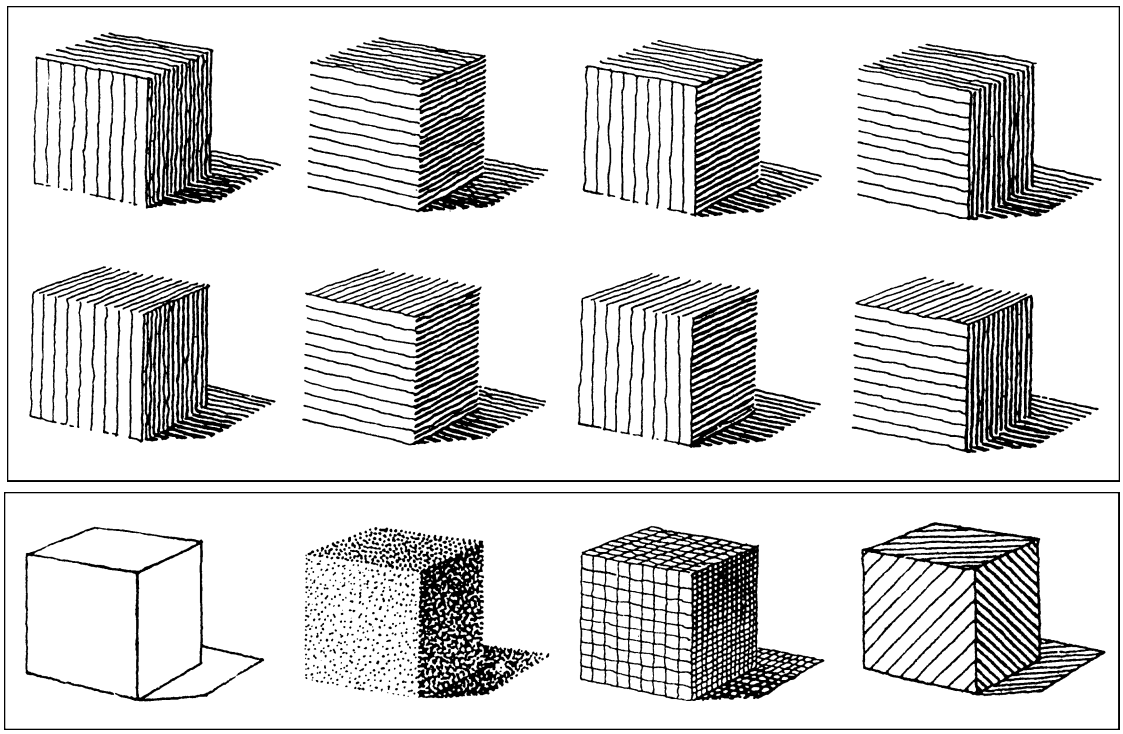


Figure 5.6: Different pen-and-ink techniques for representing the surfaces of a simple object, from [Guptill 1976]. Upper group: Eight different combinations of edge-following textures that can be used to shade the faces of a cube. Lower group: four alternative texturing approaches: outline (no texture), stipple, crosshatch, diagonal lines.

Just as there may be a great many equivalently appropriate techniques for representing the surface of an object with line texture, there are also many ways in which, if inappropriately used, texture lines can misrepresent surface shape, orientation or depth. Guptill [1976] points out, in addition to the above-mentioned effects of vertical and horizontal strokes, that diagonally oriented textures can convey a sense of expansion, making things look relatively larger than they otherwise might. Diagonal lines are also problematic in that, in some instances, they can make parallel lines appear to converge or diverge, as demonstrated by the well-known Zöllner illusion. (Gillam [1980] discusses the implications that many of the classical geometrical illusions have for shape and depth perception.) Guptill advises, however, that "as a general rule, a subject offers some hint as to a natural arrangement of lines, and when this is followed there is little danger of distortion".

Lines, of course, are not the only texture elements that can suggest a distortion of surface shape. Op artists, and in particular Victor Vasarely, demonstrated how surface markings could be successfully manipulated to create stunningly vivid and often incongruous portrayals of depth. Figure 5.7, after Vasarely's *Tupa-3* [1972], shows how the impression of curvature conveyed by a distorted pattern of surface texture can dominate the impression of flatness conveyed by contradictory shape-from-shading and non-generically ambiguous shape-from-contour cues.

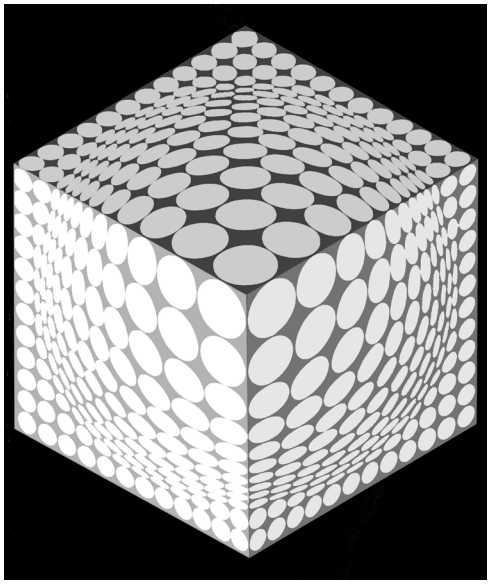


Figure 5.7: An illustration, after *Tupa-3* [Vasarely 1974], showing how texture can be misused to distort the appearance of surface shape. One of the keys to the success of this picture lies in the non-generic positioning of the surface; from this very specific viewpoint, the straight lines are technically consistent with either a flat or distorted cube (from any other angle, the projections of the lines would be different for the two alternative configurations). The effect of the texture gradients is impressive, especially when one notes that the surface illumination can only be consistent with a planar, faceted object.

Sullivan [1922] offers three alternative techniques for defining the orientations of the strokes in a line texture so that they "follow the form" and communicate the surface shape in an intuitively meaningful way, as illustrated in figure 5.8. (The distribution of the strokes is defined, in all three cases, by the patterns of illumination, which he says convey relief best when generated by a proximal light source of relatively low intensity.) The first, and by his admission simplest, of these approaches is to define the direction of the strokes according to the "fall of light upon the object". He suggests two ways of doing this, schematically portrayed in the upper portion of figure 5.8: either along the radiating rays of light or at right angles to them. It inevitably occurs with this approach, however, that at some points the direction of the texture lines closely parallels the direction of bounding contour, and Sullivan says that

when this happens the "turn" of the form is not well-represented and the impression of depth is diminished. The second general approach, which he describes as probably the most difficult and demanding the greatest knowledge of the form itself, is to orient the strokes "at right angles to the length of the form".

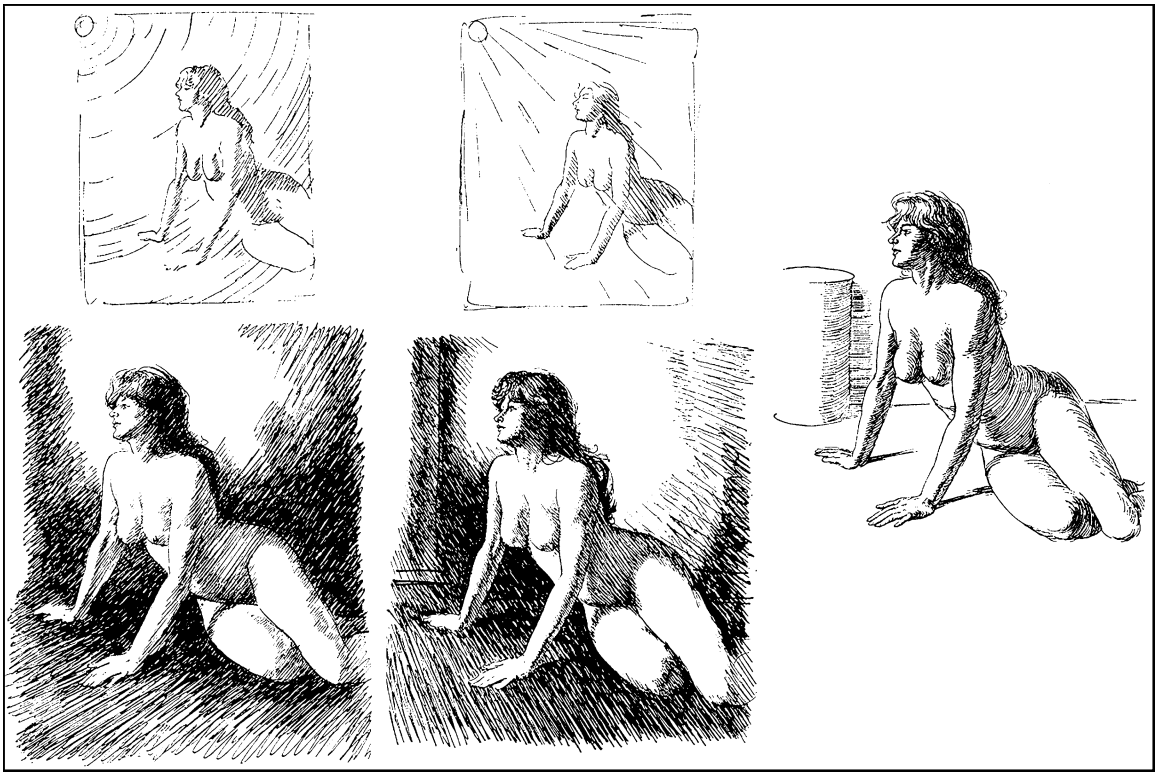


Figure 5.8: An illustration of some alternative conventions for determining stroke direction — following the light or following the form, from [Sullivan 1922]. Left: concentric circles emanating from the light source. Center: radial lines emanating from the light source. Right: strokes "taken through the form at right angles to its length".

Medical illustrators, in particular, seem to favor this "form-following" convention for using line to depict surface shape, as can be observed in the samples of artwork reproduced in figure 5.9 [Sweet (in Bunnell 1944), Loechel 1964, Hagen 1990, Drake 1932]. The basic form of the hand, in the upper-left illustration, is represented primarily by outline and valley lines. Of particular note is how the parallel strokes (which appear to lie in the direction of maximum curvature) are used around the wrist to bring the drawing "out of the plane" and convey a sense of depth distance between the outer skin surface and the underlying structures. In the illustration on the upper right, the artist presents anatomical information educed from an x-ray, along with surgical annotations, in a clearly understandable figural context. Interior strokes are used to represent shape (where they appear to be oriented in the direction of greatest normal curvature), direction (arrows) and texture (in the area below the stomach), as well as outline and tone (note that a neutral line orientation is chosen when the purpose is to represent color rather than shape in the case of the distinguishing breast features). In the illustration on the lower left, lines are used to indicate shape and to differentiate the component elements of the figure. The artist here seems to be following the line drawing style advocated by Drake, in which the strokes are applied in a manner that suggest deformations of a ruled surface. An advantage of this approach is that it permits a consistent line direction to be used for the representation of an undulating structure and also allows separate structures to be easily differentiated (note the difference in the direction of the lines used to represent the main body of the colon and the lines representing the muscle running along the center of it). The illustration in the lower right shows Drake's technique implemented in a drawing by the master

himself. Line orientation conveys shape; width and spacing convey shading; the strong outline separates figure from ground. Describing the pen-and-ink techniques illustrated in figure 5.10, Drake [1987] says

> *...think of [a surface] as a metal sheet with a series of parallel lines and then think of bending this sheet ... it is only natural that the lines on the receding surfaces become closer and closer together as they recede. ... Similarly, if lines are made to converge on the sides as well as receding on the top and bottom, a spherical shape will result. This can be demonstrated by thinking of [the surface] as a rubber sheet. ...if we press a finger into the rubber, the lines at the high spot will become separated, converging on the sides, and become closer together at the top and bottom. This is the foundation of line drawing...*

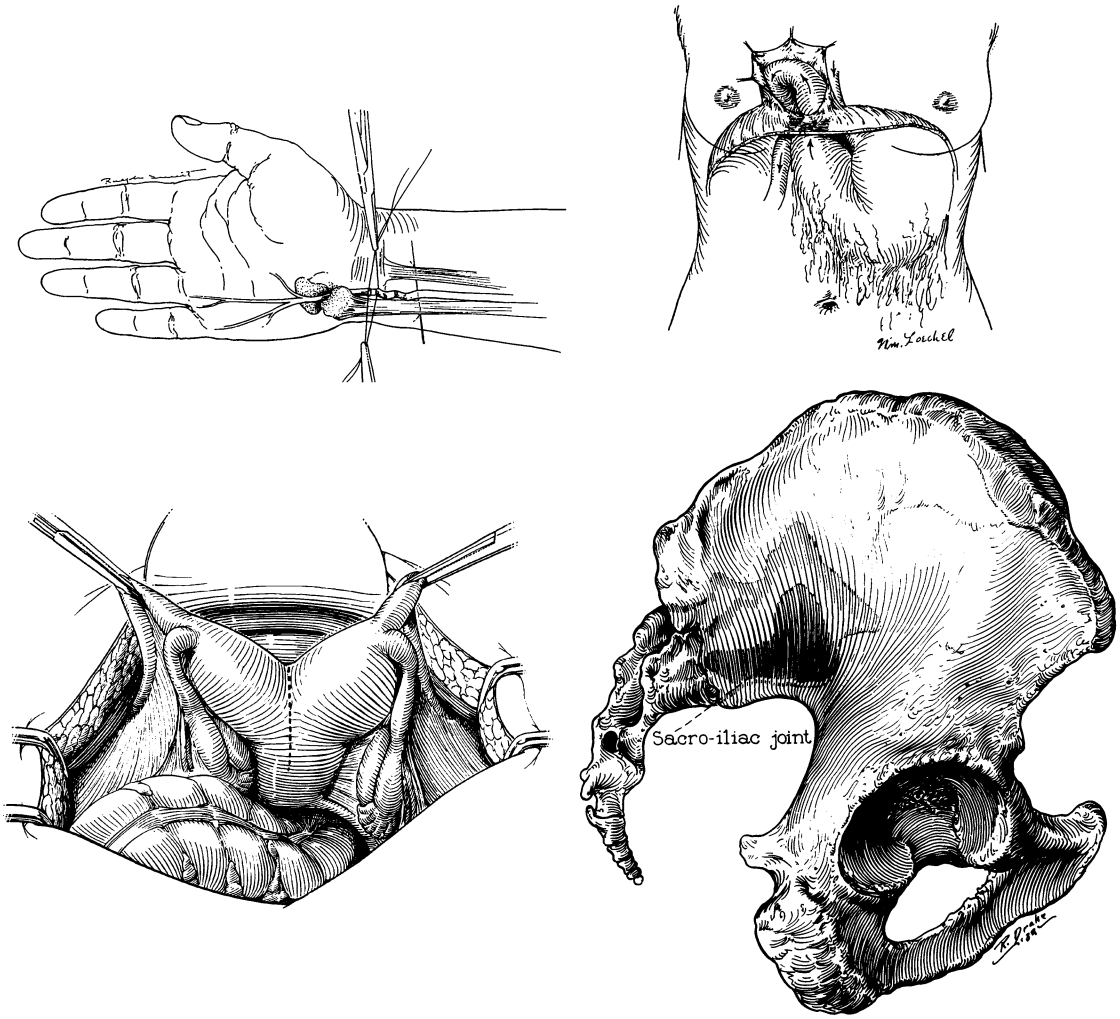


Figure 5.9: A representative sample of medical line illustrations. Upper left: [Ralph Sweet, in Bunnell 1944]; upper right: [Loechel 1964]; lower left: [Hagen 1990], lower right: [Drake 1932].

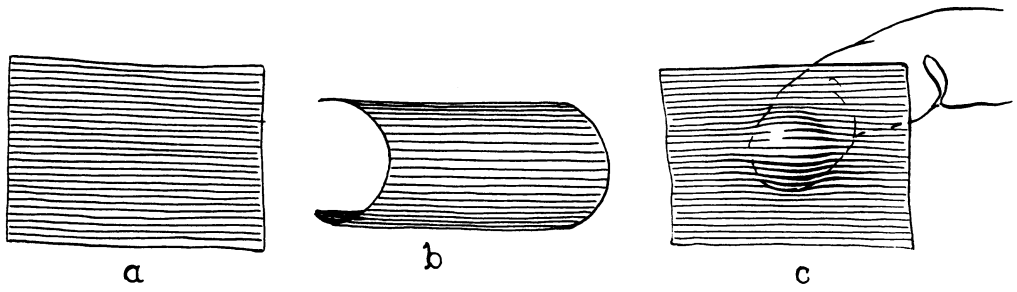


Figure 5.10: An illustration of Drake's system for representing shape with line, from Drake [1987]. Left: a flat surface, aligned with the image plane. Center: this same surface, rotated slightly about the vertical axis in the plane and bent backwards in depth along the top and bottom edges. Right: line techniques for modeling a protrusion in the planar surface.

Drake and other artists also emphasized the different treatment required for representing transparent vs. opaque surfaces with patterns of line. In figure 5.11 [Hodge 1980] we can see that the tone of the lines on the opaque surface vary in basic accordance with the surface reflectance. On the transparent surface, which is not characterized by Lambertian shading, the tones are used instead to emphasize the brilliance of the specular highlight by sharpening its contrast with the surrounding surface.

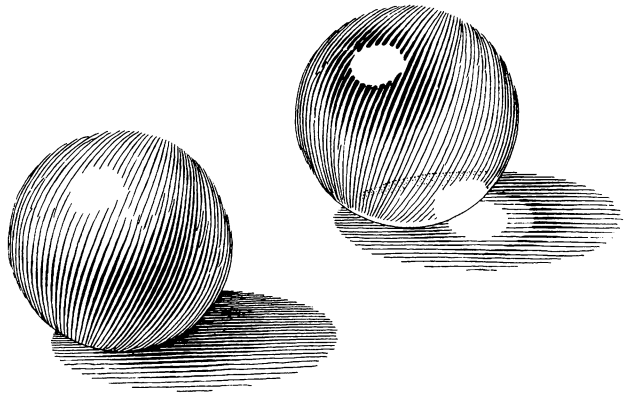


Figure 5.11: A demonstration of the different techniques required to represent transparent and opaque spheres with line texture, by Gerald P. Hodge [©1980] (from [Hodges 1989]).

The use of line in scientific illustration in fields such as biology or entomology, where faithful reproduction of the specimen is the primary task, appears, to me, to be slightly more delicate and subdued than in medical line illustration, where the object is often as much to convey a concept as it is to accurately portray a specific anatomical structure, although in many cases the same conventions are used for defining line orientation. In scientific illustration, great emphasis is put on the detailed representation of patterns of light and dark, and there is a standard convention of portraying the light as coming from above and from the left [Hodges 1989].

Despite its simplicity as a device for representing tone, cross-hatching has been singled out by several authors as a generally inelegant use of line. Sullivan [1922] is particularly adamant, counseling that

> *Except as the rhythmic solution of forces of line or for the establishment of a neutral tone, [cross-hatching] is better avoided, it then having no value, unless as a correction of an error in tone, when of course, it stands as a confession of underlying weakness. This is probably the reason why cross-hatching, unless as the resolution of opposing forces of line, becomes increasingly unpleasant the more elongated the included white "diamond" becomes, as the weakness of intention in the original lines is made more manifest.*

Zweifel [1961] reiterates the benefits of using meaningfully directed lines, which can indicate shape as well as shading (the term "hachure lines" is used in this case to refer to directed short strokes):

> *...hachure lines which follow the contours of the subject may show the form more effectively than stipple in many instances ... A few well-placed hachure lines will show the contours of the subject satisfactorily, whereas it would require several thousand stipple dots to give the same effect.*

A fundamental theme running through this entire section is that line direction is important for showing shape and that there seem to be specific, intrinsic surface shape features that can guide the determination of "appropriate" line directions. Inspired by this insight, I set about to see whether, perhaps, I could define a texture consisting of short, opaque "strokes" locally oriented in the direction of maximum normal curvature, that might when applied to a transparent surface help convey an intuitively meaningful impression of the surface shape as well as provide direct evidence of the surface location in depth.

References:

Stirling Bunnell. Surgery of the Hand, J. B. Lippincott Company, 1944.

Russell L. Drake. "The Single Line System of Shading", *Journal of Biocommunication*, pp. 12-14, Summer 1987.

Barbara Gillam. "Geometrical Illusions", *Scientific American*, vol. 242, pp. 102-110, January 1980.

Arthur L. Guptill. Rendering in Pen and Ink, Watson-Guptill Publications, 1976.

John V. Hagen. *Surgical Repair of the Septate Uterus* (1990), in Raymond A. Lee, Atlas of Gynecologic Surgery, Saunders, 1992.

Elaine R. S. Hodges. The Guild Handbook of Scientific Illustration, Van Nostrand Reinhold, 1989.

William E. Loechel. Medical Illustration; a guide for the doctor-author and exhibitor, Charles C. Thomas, 1964.

Henry C. Pitz. Ink Drawing Techniques, Watson-Guptill Publications, 1957.

Edmund J. Sullivan. Line; an art study, Chapman & Hall, 1922.

Victor Vasarely. *Tupa-3* (1972), in Vasarely III, p. 193, Éditions du Griffon Neuchâtel, 1974.

Frances W. Zweifel. A Handbook of Biological Illustration, University of Chicago Press, 1961.

Painting, Drawing, and Visualization

David Laidlaw
Brown University

During the next hour and a half we will have a chance to work with paint, pastels, and other media to experiment with visually representing multi-valued scientific data. Our goals for this section are to give you an opportunity to experiment with visual representations in ways that you may not have before and to give a new framework for some of the artistic and perception-based guidelines and rules that we have been and will be talking about today. We hope that you'll take away some new ideas about how to visually represent data and/or develop visualization tools. Of course, we hope that you'll have fun, as well!

1.1 Data Example: 2D Fluid Flow

We will work with 2D vector data for a number of reasons. First, it's 2D, so we can concentrate on representing the data instead of on representing 3D shapes. Second, 3D fluid flow is a very important phenomenon that is not understood. Visualization has the potential to help with that understanding. If we can develop ideas in 2D and then generalize them to 3D, they will have wide applicability. Third, there are a number of quantities important to fluid flow that can be derived from 2D vector fields. These quantities have different types and physical interpretations, so there is a significant challenge here and a wealth of combinations with which to experiment. Finally, it's relatively easy to generate random examples, so we have have a lot of data!

We will provide separate 2D images of a number of quantities you can represent in your work today. Each image is located at the same place as all of the others, but represents a different quantity.

The base for this type of data is a 2D vector field. Imagine that at each point on your piece of paper there is an arrow. It's length and direction represent the magnitude and direction of the vector field. This will be one of the quantities that we can represent.

Differentiating this vector field spatially produces a 2x2 second-order tensor, the velocity gradient tensor, at each point on your paper. Fluid researchers often decompose that tensor into simpler quantities.

One quantity, vorticity, is a scalar value. It represents the tendency of a particle at a particular location to be spinning. An example may help make this clearer. Imagine a velocity field where direction is always left to right, but the speed increases as we go from bottom to top. If you pick any point in this vector field, the point will feel a faster velocity above it than below it and will tend to spin clockwise. It would be said to have positive vorticity. Negative vorticity corresponds to counter-clockwise spinning. This is the second quantity that we can represent.

The effect of vorticity can be removed from the velocity gradient tensor to leave a new tensor known as the rate-of-strain tensor. This tensor describes how a fluid would squash and stretch at each point. Imagine a circle located at a single point on your paper. Forces would be squashing and stretching it in different directions. If all the forces were pushing toward the circle, it would shrink. If all the forces were pulling away, it would grow. If they were pushing in one direction and pulling in another, it would become an ellipse. In fact, there is a one-to-one mapping from this tensor into ellipses (including circles) just as described here. This is the third quantity that we can represent.

A vector field where the ellipses all maintain their size (area) is known as incompressible. When flow is compressible, a scalar measure of this tendency to shrink or grow (i.e., increase in pressure or decrease in pressure) at each point is the fourth quantity we can represent.

After factoring out the change in size of these ellipses, the remaining components can be represented with a scalar measure of the eccentricity of the ellipse together with a vector in the direction of the largest radius. These two quantities are the fifth and sixth that we can represent.

Pressure and temperature are two additional quantities that are sometimes available together with velocity data. These will be the seventh and eighth quantities that we can represent.

Finally, two more complicated derived quantities, turbulent charge (a scalar) and turbulent current (a vector) will be the ninth and tenth quantities that we can represent.

Pick and choose from among these ten quantities and represent them together.

1.2 Media

We will have available several media: tempera paint, transparent markers, pastels, tape, scissors, paper, and transparencies. Please experiment in any way you think will work. Use different media, layer things.

1.3 Thought Experiments

At the risk of over-directing, here are some questions that you might ponder as you produce. These are only intended as guidelines. Please also create your own questions!

1. How many quantities can you represent together? Does it depend on the data? What are some tradeoffs?
2. What kinds of abstractions can be used to represent lots of information with a small amount of visual bandwidth?
3. What kind of layering works? Large strokes over small? Discrete strokes over continuous? High contrast over low? Layers that are blended together, or kept separate? Again, what are some of the tradeoffs?
4. Do untraditional items, like taped-on cutouts, expand expressiveness? How?

Expressive Volume Rendering

Penny Rheingans
University of Maryland Baltimore County

The main goal of visualization is to effectively convey information to the user using the wide input channel provided by the human visual system. A visual representation of a large data set can capture both interesting elements and interesting structure in the data. For volume data, one key goal is to convey the structure of the data distribution, for example the shape of the liver, or the extent of ground water contamination, or the density of ozone throughout the atmosphere. In some applications, the boundaries of interesting regions are sharp and well-defined, for instance the boundaries of organs or the places where groundwater contamination exceeds legal limits. In other applications, the boundaries of interesting regions are rather diffuse, for instance the boundaries of tumors or molecules. And in still other applications, the whole structure of the data distribution is of interest, rather than just the boundaries of particular regions. Effective general volume visualization techniques must address all three kinds of applications.

Algorithms for the visualization of volume data can be characterized into two general approaches. Surface algorithms first map the volume data to representative geometry, such as an isosurface of constant value, and then render the geometric representation using standard rendering techniques. The second type of approach, direct volume rendering, generates the image directly from the volume data, without first creating any geometry. For volume models, the key advantage of direct volume rendering over surface rendering approaches is the potential to show the structure of the value distribution throughout the volume, rather than just at selected boundary surfaces of variable value (by isosurface) or coordinate value (by cutting plane). The contribution of each volume sample to the final image is explicitly computed and included. The key challenge of direct volume rendering is to convey that value distribution clearly and accurately. In particular, showing each volume sample with sufficient clarity and opacity that its structure is apparent but not so much that volume samples in the rear of the volume are overly obscured.

Volume illustration is a new approach to volume rendering involving the augmentation of a physics-based rendering process with non-photorealistic rendering (NPR) techniques to enhance the expressiveness of the visualization [Ebert00]. NPR draws inspiration from such fields as art and technical illustration to develop automatic methods to synthesize images with an illustrated look from geometric surface models. Non-photorealistic rendering research has effectively addressed both the illustration of surface shape and the visualization of 2D data, but has virtually ignored the rendering of volume models. Volume illustration introduces a set of NPR techniques specifically for the visualization of volume data, including both the adaptation of existing NPR techniques to volume rendering and the development of new techniques specifically suited for volume models.

The volume illustration approach combines the benefits of the two traditional volume rendering approaches in a flexible and parameterized manner. It provides the ease of interpretation resulting from familiar physics-based illumination and accumulation processes with the flexibility of the transfer function approach. In addition, volume illustration provides flexibility beyond that of the traditional transfer function, including the capabilities of local and global distribution analysis, and light and view direction specific effects. Therefore, volume illustration techniques can be used to create visualizations of volume data that are more effective at conveying the structure within the volume than either of the traditional approaches. As the name suggests, volume illustration is intended primarily for illustration or presentation situations, such as figures in textbooks, scientific articles, and educational video.

Motivation for Volume Illustration

Volumetric illustration differs from surface-based NPR in several important ways. In surface-based NPR, the surfaces (features) are well defined, whereas with volumes, the volumetric features vary continuously throughout three-dimensional space and are not as well defined as surface features. Feature areas within the volume must be determined through analysis of local volumetric properties. Once these volumetric

feature volumes are identified, user selected parametric properties can be used to enhance and illustrate them.

In a surface model, the essential feature is the surface itself. The surface is explicitly and discretely defined by a surface model, making "surfaceness" a boolean quality. Many other features, such as silhouettes or regions of high curvature, are simply interesting parts of the surface. Such features can be identified by analysis of regions of the surface. In a volume model, there are no such discretely defined features. Additional processing is required to first identify interesting features in the volume.

Another difficulty with volumetric models is that few of the usual depth cues are present in traditional rendering of translucent volumes. Obscuration cues are largely missing since there are no opaque objects to show a clear depth ordering. Perspective cues from converging lines and texture compression are also lacking, since few volume models contain straight lines or uniform textures. The dearth of clear depth cues makes understanding spatial relationships of features in the volume difficult. One common approach to this difficulty is the use of hard transfer functions, those with rapidly increasing opacity at particular value ranges of interest. While this may increase depth cues by creating the appearance of surfaces within the volume, it does so by hiding all information in some regions of the volume, sacrificing a key advantage of volume rendering.

Similarly, information about the orientation of features within the volume is also largely missing. Transfer function approaches typically perform no illumination calculations to determine the color at a volume sample. Nor does the transfer function approach usually include the effect of shadows, particularly self-shadows in which the volume shadows itself. Although physics-based volume rendering algorithms do include illumination and shadow effects, the effects are generally subtle and difficult to interpret unambiguously. As a result, the shape of individual structures within the volume is difficult to perceive, as can be seen in this volume rendering of an abdominal CT volume:

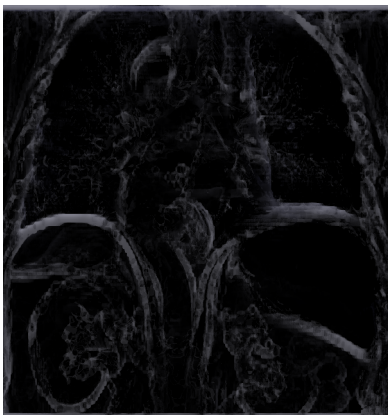
Some Volume Illustration Techniques

A wide variety of illustration techniques can be adapted or invented for use with volume models. Some of these techniques, described more fully in [Ebert00], are:

1. Silhouette Enhancement

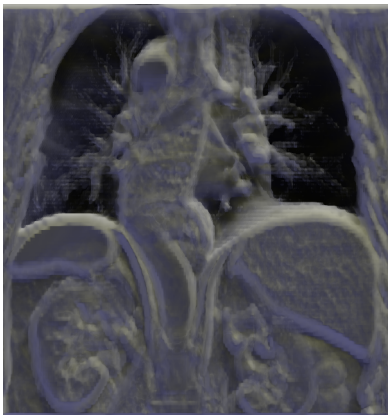
Silhouette lines are particularly important in the perception of surface shape, and have been utilized in surface illustration and surface visualization rendering. Similarly, silhouette volumes increase the perception of volumetric features. In order to strengthen the cues provided by silhouette volumes, we can increase the opacity of volume samples where the gradient is perpendicular to the view direction.

2. Volumetric Sketch Lines



Oriented adjustment of the volume opacity and color can also be used to emphasize volume feature orientation. Decreasing the opacity and/or altering the color of volume features oriented toward the viewer emphasizes feature orientation, and in the extreme cases, can create sketches of the volume. Additionally, oriented adjustment of opacity with respect to the light source can enhance the perception of features and feature orientation.

3. Tone Shading



Another illustrative technique used by painters is to modify the tone of an object based on the orientation of that object relative to the light. This technique can be used to give surfaces facing the light a warm cast while surfaces not facing the light get a cool cast, giving effects suggestive of illumination by a warm light source, such as sunlight. Gooch et al. proposed an illumination model based on this technique [Gooch98]. The parameters define a warm color by combining yellow and the scaled fully illuminated object color and a cool color combining blue and the scaled ambient illuminated object color. The final surface color is formed by interpolation between the warm and cool color based on the signed dot product between the surface normal and light vector. The image at right was calculated using an illumination model similar to Gooch tone shading for use with volume models. As with Gooch tone shading, the tone contribution is formed by interpolation between the warm and cool colors based on the signed dot product between the volume sample gradient and the light vector. Unlike Gooch tone shading, the illuminated object contribution is calculated using only the positive dot product, becoming zero at orientations orthogonal to the light vector. This more closely matches familiar diffuse illumination models.

4. Oriented Fading

The use of surface orientation with respect to the viewer is a common property used in sketching and medical illustration to emphasize object orientation and features of interest. For instance, in medical illustration the portions of anatomical structures oriented toward the viewer are desaturated and structures oriented away from the view are darkened and saturated [Clark99]. We simulated these effects by allowing the volumetric gradient orientation to the viewer to modify the color, saturation, value, and transparency of the given volume sample.

5. Feature Halos

Illustrators sometimes use null halos around foreground features to reinforce the perception of depth relationships within a scene. The effect is to leave the areas just outside surfaces empty, even if an accurate depiction would show a background object in that place. Interrante [Interrante98] used a similar idea to show depth relationships in 3D flow data using Line Integral Convolution (LIC). She created a second LIC volume with a larger element size, using this second volume to impede the view. Special care was required to keep objects from being obscured by their own halos. The resulting halos achieved the desired effect, but the method depended on having flow data suitable for processing with LIC. We used a more general method for creating halo effects during the illumination process using the local spatial properties of the volume. Halos are created primarily in planes orthogonal to the view vector by making regions just outside features darker and more opaque, obscuring background elements which would otherwise be visible. This method produces effects similar to those of Interrante, but can be applied to any type of data or model during the illumination process. Since the halos generated are inherently view dependent, no special processing must be done to keep features from casting a halo on themselves.

Adapting NPR Surface Techniques for Volumes

There has been extensive research for illustrating surface shape using non-photorealistic rendering techniques. Adopting a technique found in painting, Gooch et al. developed a tone-based illumination model that determined hue, as well as intensity, from the orientation of a surface element to a light source [Gooch98]. The extraction and rendering of silhouettes and other expressive lines has been addressed by several researchers [including Saito90, Salisbury94, Gooch99, Interrante95, Girshik00, Hertzmann00]. Expressive textures have also been applied to surfaces to convey surface shape and depth [including Levoy90b, Rheingans96, Salisbury97, Interrante97, Hamel98].

A few researchers have applied NPR techniques to the display of data. Laidlaw used concepts from painting to create visualizations of 2D data, using brushstroke-like elements to convey information [Laidlaw98] and a painterly process to compose complex visualizations [Kirby99]. Treavett has developed techniques for pen-and-ink illustrations of surfaces within volumes [Treavett00].

The key difficulty of adapting surface techniques for NPR is that there are no explicit surfaces in volume models. So, instead of simply calculating illustration appearance on surfaces, volume illustration must first explicitly identify features to be illustrated and then render these features appropriately. These features may be either boolean (as surfaces are) or may be continuously defined to indicate the strength of a feature at a location. Continuously defined features, in particular, generally require the consideration of each voxel in the rendering process.

Although volume data contains no surfaces, the boundaries between regions may still be of interest. The local gradient magnitude at a volume sample can be used to indicate the degree to which the volume sample is a boundary between disparate regions. The direction of the gradient is analogous to the surface normal. Regions of high gradient are similar to surfaces, but now "surfaceness" is a continuous, rather than boolean, quality. This makes gradient a straight-forward feature indicator. There is a rich literature in the estimation of gradients for volume data [including Goss94, Marschner94, Bentum96, Möller97, Möller98, Neumann00].

Additional feature indicators may be used, usually in conjunction with the gradient. These include higher-order derivatives, solid shape in the vicinity of a voxel, location, alignment, and metadata.

Some NPR techniques originally developed for surfaces can be adapted for volumes including a boolean or continuous feature indicator in the appearance model, resulting in strong illustration where there are strong features and fainter illustration for fainter features. In this way, the feature strength scales the illustration weight by modifying voxel opacity, stroke density, or other rendering parameters. The techniques most suitable for this adaption are those which can be characterized by some appearance model evaluated at sample points (for instance, the tone shading model of Gooch et al. [Gooch98] or the 3D textures of Treavett and Chen [Treavett00]) or those in which strokes of differing densities are generated in image space (for instance, Salisbury97 and Deussen00).

Architectures for Expressive Volume Rendering

Two basic types of rendering systems that potentially support expressive direct volume rendering are ray-based volume renderers and systems which represent voxels with one or several stroke-like primitives. Both have been used to create illustration effects. Interrante augmented a raycast volume renderer to convey depth relationships with halos around foreground features in flow data [Interrante98]. Ebert and Rheingans added a variety of illustrative enhancements to a ray-based volume renderer [Ebert00]. Saito converted 3D scalar fields into a sampled point representation and visualized selected points with a simple primitive, creating an NPR look [Saito94], while Treavett et al. also used strokes within a volume [Treavett01]. Morris used strokes to create the appearance of a stipple drawing [Morris01].

In general, ray-based renderers provide obscuration cues more naturally through their accumulation model, while stroke based approaches can better exploit hardware acceleration to allow better interaction with the visualization. Although volume rendering hardware can now provide interactive viewing of volume data, present systems do not allow sufficiently complex real-time control of the mapping from volume densities and features to displayed opacities to allow interactive volume illustration.

Adapting Ray-based Volume Rendering to be Illustrative

Traditional direct volume rendering has relied on the use of transfer functions from scalar value to rendered opacity to produce artificial views of the data which highlight regions of interest [Drebin88]. These transfer functions, however, require in-depth knowledge of the data and need to be adjusted for each data set. The design of effective transfer functions is still an active research area [Fang98, Fujishiro99]. While transfer functions can be effective at bringing out the structure in the value distribution of a volume, they are generally limited by their dependence on voxel value as the sole transfer function domain.

Augmenting a ray-based volume renderer to include illustration effects can be accomplished by modifying the color and opacity of volume samples, either instead of or in addition to a traditional transfer function. Multiple enhancements may be made in sequence to create more complex effects.

Augmenting a ray-based volume renderer to include illustration effects can be accomplished by modifying the color and opacity of volume samples, either instead of or in addition to a tradition transfer function.

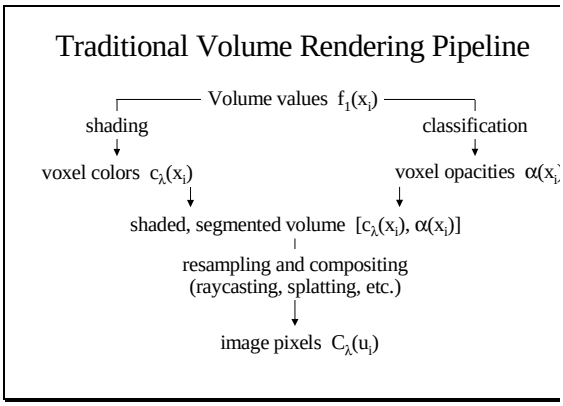


Diagram 1. Traditional Volume Rendering Pipeline. From [Ebert00].

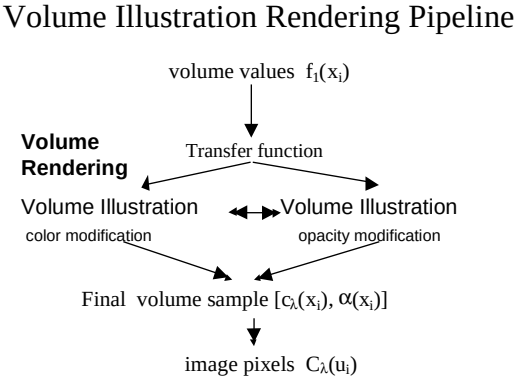


Diagram 2. Volume Illustration Rendering Pipeline. From [Ebert00].

To some extent, volume illustration techniques may be regarded as representing a very complex and flexible multivariate transfer function mechanism. Some previous research has considered both voxel value and gradient and more rarely other quantities, either directly or indirectly, as the domain of a transfer function, starting with Levoy's use of gradient magnitude to increase opacity in boundary regions[Levoy88]. Kindlmann and Durkin used the first and second directional derivatives in the gradient direction to calculate a boundary emphasis to be included in the opacity transfer function [Kindlmann98] More recent work has considered curvature, as well [Hladuvka00, Sato00]..

Using this abstraction, the parameters controlling boundary and silhouette enhancement specify a multivariate opacity transfer function from voxel value, voxel location, gradient, and view vector. Similarly, the parameters controlling oriented fading, distance color blending, and tone shading specify a multivariate color transfer function from voxel value, initial voxel color, voxel location, gradient, view vector, and light position and intensity. The two transfer functions would require ten and seventeen inputs, respectively. Specifying illustration techniques through multiple separable enhancements breaks the problem of designing these potentially very complex multivariate functions into smaller, more manageable design problems.

The transfer function abstraction breaks down when feature halos are considered. Both transfer functions would not only need to take the voxel value, voxel location, gradient, and view vector as inputs, but also the value, location, and gradient of each voxel in a neighborhood of user-controllable size. This would require transfer functions with a number of inputs which was not only large, but variable.

References

[Bentum96] Mark Bentum, Barthold Lichtenbelt, and Tom Malzbender. Frequency Analysis of Gradient Estimators in Volume Rendering, *IEEE Transactions on Visualization and Computer Graphics*, 2(3), (September 1996).

[Clark99] John O.E. Clark. *A Visual Guide to the Human Body, Barnes and Noble Books, 1999.*

[Drebin88] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume Rendering, *Computer Graphics (Proceedings of SIGGRAPH 88)*, 22(4), pp. 65-74 (August 1988, Atlanta, Georgia). Edited by John Dill.

[Duessen00] O. Duessen, S. Hiller, C. van Overveld, and T. Strothotte. "Floating Points: A Method for Computing Stipple Drawings". *Proceedings of Eurographics 2000*. vol. 19. no. 3. The Eurographics Association and Blackwell Publishers. Malden, MA. 2000.

[Ebert90] David S. Ebert and Richard E. Parent. Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-buffer Techniques, *Computer Graphics (Proceedings of SIGGRAPH 90)*, 24 (4), pp. 357-366 (August 1990, Dallas, Texas). Edited by Forest Baskett. ISBN 0-201-50933-4.

[Ebert00] David S. Ebert and Penny Rheingans. Volume Illustration: Nonphotorealistic Rendering of Volume Models, *Proceedings of IEEE Visualization '00*, October 2000, pp. 195-202.

[Fang98] Shiaofen Fang, Tom Biddlecome, and Mihran Tuceryan. Image-Based Transfer Function Design for Data Exploration in Volume Visualization, *IEEE Visualization '98*, pp. 319-326 (October 1998). IEEE. Edited by David Ebert and Hans Hagen and Holly Rushmeier. ISBN 0-8186-9176-X.

[Foley96] James Foley, Andries van Dam, Steven Feiner, and John Hughes, *Computer Graphics: Principles and Practice, Second Edition in C*, Addison Wesley 1996.

[Fujishiro99] Issei Fujishiro, Taeko Azuma, and Yuriko Takeshima. Automating Transfer Function Design for Comprehensible Volume Rendering Based on 3D Field Topology Analysis, *IEEE Visualization '99*, pp. 467-470 (October 1999, San Francisco, California). IEEE. Edited by David Ebert and Markus Gross and Bernd Hamann. ISBN 0-7803-5897-X.

[Girshik00] Ahna Girshik, Victoria Interrante, Steven Haker, and Todd Lemoine. Line Direction Matters: An Argument for the Use of Principal Curvature Directions in 3D Line Drawings, *Proceedings of the First International Symposium on Non Photorealistic Animation and Rendering(NPAR2000)*, pp. 43-52, 2000.

[Gooch98] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A Non-photorealistic Lighting Model for Automatic Technical Illustration. In *Proceedings of SIGGRAPH '98* (Orlando, FL, July 1998), Computer Graphics Proceedings, Annual Conference Series, pp. 447-452, ACM SIGGRAPH, ACM Press, July 1998.

[Gooch99] Bruce Gooch, Peter-Pike J. Sloan, Amy Gooch, Peter Shirley, and Rich Riesenfeld. Interactive Technical Illustration, *1999 ACM Symposium on Interactive 3D Graphics*, pp. 31-38 (April 1999). ACM SIGGRAPH. Edited by Jessica Hodgins and James D. Foley. ISBN 1-58113-082-1 .

[Goss94] Michael Goss. An Adjustable Gradient Filter for Volume Visualization Image Enhancement, *Proceedings of Graphics Interface '94*, pp. 67-74 (Toronto, Ontario).

[Hamel98] Jorg Hamel, Stefan Schlectweg, and Thomas Strothotte. An Approach to Visualizing Transparency in Computer-Generated Line Drawings, *Proceedings of Information Visualization '98*, pp. 151-156.

[Hertzmann00] Aaron Hertzmann and Denis Zorin. Illustrating Smooth Surfaces, *SIGGRAPH 2000 Conference Proceedings*, July 2000, pp. 517-526.

[Hladuvka00] Jiří Hladůvka, Andreas König, Eduard Gröller, Curvature-Based Transfer Functions for Direct Volume Rendering,. Technical Report TR-186-2-00-01, Vienna Univeristy of Technology, January 2000.

[Hodges89] Elaine Hodges. *The Guild Handbook of Scientific Illustration*, John Wiley & Sons, New York, 1989.

[Interrante95] Victoria Interrante, Henry Fuchs, and Stephen Pizer. Enhancing Transparent Skin Surfaces with Ridge and Valley Lines, *IEEE Visualization '95*, pp. 52-59 (October 1995, Atlanta GA). IEEE. Edited by Gregory Nielson and Deborah Silver. ISBN 0-8186-7187-4.

[Interrante97] Victoria Interrante, Henry Fuchs and Stephen M. Pizer. Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture, *IEEE Transactions on Visualization and Computer Graphics*, 3(2), (April - June 1997). ISSN 1077-2626.

[Interrante98] Victoria Interrante and Chester Grosch. Visualizing 3D Flow, *IEEE Computer Graphics & Applications*, 18(4), pp. 49-53 (July - August 1998). ISSN 0272-1716.

[Kindlmann98] Gordon Kindlmann and James Durkin. Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering, *Proceedings of 1998 IEEE Symposium on Volume Visualization*, pp. 79-86.

[Kirby99] R.M. Kirby, H. Marmanis, and D.H. Laidlaw. Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting, *IEEE Visualization '99*, pp. 333-340 (October 1999, San Francisco, California). IEEE. Edited by David Ebert and Markus Gross and Bernd Hamann. ISBN 0-7803-5897-X.

[Laidlaw98] David H. Laidlaw, Eric T. Ahrens, David Kremers, Matthew J. Avalos, Russell E. Jacobs, and Carol Readhead. Visualizing Diffusion Tensor Images of the Mouse Spinal Cord, *IEEE Visualization '98*, pp. 127-134 (October 1998). IEEE. Edited by David Ebert and Hans Hagen and Holly Rushmeier. ISBN 0-8186-9176-X.

[Levoy85] Marc Levoy and T. Whitted. "The Use of Points as a Display Primitive". *UNC-Chapel Hill Computer Science Technical Report #85-022*. January, 1985.

[Levoy88] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, vol. 8, no. 5, pp. 29-37, 1988.

[Levoy90] Marc Levoy. Efficient Ray Tracing of Volume Data, *ACM Transactions on Graphics*, 9 (3), pp. 245-261 (July 1990). ISSN 0730-0301.

[Marschner94] Stephen Marschner and Richard Lobb. An Evaluation of Reconstruction Filters for Volume Rendering, *Proceedings of IEEE Visualization '94*, pp. 100-107.

[Möller97] Torsten Möller, Raghu Machiraju, Klaus Mueller, and Roni Yagel. Evaluation and Design of Filters Using a Taylor Series Expansion, *IEEE Transactions on Visualization and Computer Graphics*, 3 (2), pp. 184-199 (April 1997).

[Möller98] Torsten Möller, Klaus Mueller, Yair Kurzion, Raghu Machiraju, and Roni Yagel. Design of Accurate and Smooth Filters for Function and Derivative Reconstruction, *Proceedings of 1998 IEEE Symposium on Volume Visualization*, pp. 143-151.

[Morris01] Christopher J. Morris. *Non-Photorealistic Direct Volume Rendering Using Stippling Techniques*, MS Thesis, Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, 2001.

[Neumann00] László Neumann, Balázs Csébfalvi, Andreas König, and Eduard Gröller. Gradient Estimation in Volume Data using 4D Linear Regression, *Proceedings of Eurographics 2000*, pp.

[Rheingans96] Penny Rheingans. Opacity-modulating Triangular Textures for Irregular Surfaces, *Proceedings of IEEE Visualization '96*, pp. 219-225 (October 1996, San Francisco CA). IEEE. Edited by Roni Yagel and Gregory Nielson. ISBN 0-89791-864-9.

[Saito90] Takafumi Saito and Tokiichiro Takahashi. Comprehensible Rendering of 3-D Shapes, *Computer Graphics (Proceedings of SIGGRAPH 90)*, 24 (4), pp. 197-206 (August 1990, Dallas, Texas).

[Saito94] Takafumi Saito. Real-time Previewing for Volume Visualization. *Proceedings of 1994 IEEE Symposium on Volume Visualization*, pp. 99-106.

[Salisbury94] Michael P. Salisbury, Sean E. Anderson, and Ronen Barzel and David H. Salesin. Interactive Pen-And-Ink Illustration, *Proceedings of SIGGRAPH 94, Computer Graphics Proceedings, Annual Conference Series*, pp. 101-108 (July 1994, Orlando, Florida). ACM Press. Edited by Andrew Glassner. ISBN 0-89791-667-0.

[Salisbury97] Michael P. Salisbury, Michael T. Wong, John F. Hughes, and David H. Salesin. Orientable Textures for Image-Based Pen-and-Ink Illustration, *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pp. 401-406 (August 1997, Los Angeles, California). Addison Wesley. Edited by Turner Whitted. ISBN 0-89791-896-7.

[Sato00] Y. Sato, C-F Westin, A. Bhalerao, S. Nakajima, N. Shiraga, S. Tamura, and R. Kikinis. Tissue Classification Based on 3D Local Intensity Structures for Volume Rendering, IEEE Transactions on Visualization and Computer Graphics, vol. 6., no. 2, April-June 2000, pp. 160-179.

[Treavett00] S.M.F. Treavett and M. Chen. Pen-and-Ink Rendering in Volume Visualisation, *Proceedings of IEEE Visualization 2000*, pp. 203-209, October 2000.

[Treavett01] S.M.F. Treavett, M. Chen, R. Satherley, and M.W. Jones, Volumes of Expression: artistic modelling and rendering of volume data sets, to appear in *Proceedings of Computer Graphics International (CGI2001)*, Hong Kong, July 2001.

[Winkenbach94] Georges Winkenbach and David H. Salesin. Computer-Generated Pen-And-Ink Illustration, *Proceedings of SIGGRAPH 94, Computer Graphics Proceedings, Annual Conference Series*, pp. 91-100 (July 1994, Orlando, Florida). ACM Press. Edited by Andrew Glassner. ISBN 0-89791-667-0.

Combining Perception and Impressionist Techniques for Nonphotorealistic Visualization of Multidimensional Data

Christopher G. Healey
North Carolina State University

1 Introduction

An important research problem in computer graphics is the visualization of multidimensional data, the conversion of a dataset D containing strings and numbers into a sequence of one or more images. Values in D represent m attributes $A = \{A_1, \ldots, A_m\}$ recorded at n sample points e_i, that is, $D = \{e_1, \ldots, e_n\}$ and $e_i = \{a_{i,1}, \ldots, a_{i,m}\}, a_{i,j} \in A_j$. A data-feature mapping $M(V, \Phi)$ defines a visual feature $V_j \in V$ to use to display values from A_j; it also defines $\phi_j : A_j \rightarrow V_j, \phi_j \in \Phi$ to map the domain of A_j to the range of displayable values in V_j. Visualization in this framework is the construction of a data-feature mapping M together with a viewer's interpretation of the images produced by M.

Although the need to visualize multiple layers of information simultaneously is well documented [43, 61, 63], progress towards this goal has been slow [56]. It has often proven difficult to construct methods that represent multidimensional data in a way is easy to *explore, analyze, verify* and *discover*. The desire to build fundamental techniques that are appropriate for a wide range of visualization environments further complicates this problem.

Previous work has studied methods for harnessing the low-level human visual system during visualization [4, 26, 52, 81]. Certain visual features (*e.g.,* hue, luminance, contrast, and motion) are detected very quickly by the visual system [14, 70, 83]; when combined properly, these same features can be used to construct multidimensional displays that can be *rapidly, accurately,* and *effortlessly* explored and analyzed by a viewer. The application of perception in aid of visualization has shown great promise, and has been explicitly cited as an important area of current and future research [63].

We have recently initiated a study of the use of artistic techniques for multidimensional visualization. This investigation was motivated in large part by work on nonphotorealistic rendering in computer graphics [10, 18, 27, 28, 39, 44, 65], and by the efforts of researchers like Interrante [30], Laidlaw [36, 37], and Ebert and Rheingans [13] to extend this work to a visualization environment. Certain movements and techniques in painting (*e.g.,* impressionism, expressionism, or watercolor) are characterized by a set of fundamental styles. If these styles can be identified and simulated on a computer, we believe they can then be applied to represent individual data attributes in a multidimensional dataset. Consider, for example, a dataset containing weather conditions. A "painting" made up of simulated brush strokes could be used to visualize this data. A brush stroke's color would represent temperature at a given spatial location; stroke direction and length would represent wind direction and strength; stroke density would represent pressure. The result is an image that looks like a painting, not of a real-world scene, but rather of the information contained in the underlying dataset.

Such a technique might initially seem difficult to control and test. An important insight is that many painterly styles correspond closely to perceptual features that are detected by the human visual system. In some sense this is not surprising. Artistic masters understood intuitively which properties of a painting would capture a viewer's gaze, and their styles naturally focused on harnessing these features. Moreover, certain movements used scientific studies of the visual system to help them understand how viewers would perceive their work (*e.g.,* the use of the perceptual color models of Chevruel [9] and Rood [55] in Impressionism). The overlap of artistic styles and perception offers two significant advantages. Most importantly, the body of knowledge on the use of perception during visualization can help us to predict how corresponding painterly styles might perform in the same environment. In addition, psychophysical experiments offer a method for designing controlled studies that can test the fundamental strengths and limitations of a given style, both in isolation and in combination with other styles being shown simultaneously in the same display.

Feature	*Author*
line (blob) orientation	Julész & Bergen (1983); Wolfe (1992)
length	Triesman & Gormican (1988)
width	Julész (1984)
size	Triesman & Gelade (1980)
curvature	Triesman & Gormican (1988)
number	Julész (1985); Trick & Pylyshyn (1994); Healey, Booth, & Enns (1996)
terminators	Julész & Bergen (1983)
intersection	Julész & Bergen (1983)
closure	Enns (1986); Triesman & Souther (1986)
color (hue)	Triesman & Gormican (1988); Nagy & Sanchez (1990); D'Zmura (1991); Healey (1997)
intensity	Beck et al. (1983); Triesman & Gormican (1988)
flicker	Julész (1971)
direction of motion	Nakayama & Silverman (1986); Driver & McLeod (1992)
binocular lustre	Wolfe & Franzel (1988)
stereoscopic depth	Nakayama & Silverman (1986)
3-D depth cues	Enns (1990)
lighting direction	Enns (1990)
texture	Healey & Enns (1998)

Table 1: A list of two-dimensional features that "pop out" during visual search, and a list of authors who describe preattentive tasks performed using the given feature.

2 Low-Level Human Vision

An important requirement for any visualization technique is a method for rapid, accurate, and effortless visual exploration. We address this goal by using what is known about the control of human visual attention as a foundation for our visualization tools. The individual factors that govern what is attended in a visual display can be organized along two major dimensions: bottom-up (or stimulus driven) versus top-down (or goal directed).

Bottom-up factors in the control of attention include the limited set of features that psychophysicists have identified as being detected very quickly by the human visual system, without the need for search. These features are often called preattentive, because their detection occurs rapidly and accurately, usually in an amount of time independent of the total number of elements being displayed[1]. When applied properly, preattentive features can be used to perform different types of exploratory analysis. Examples include searching for data elements with a unique visual feature, identifying the boundaries between groups of elements with common features, tracking groups of elements as they move in time and space, and estimating the number of elements with a specific feature. Preattentive tasks can be performed in a single glance, which corresponds to 200 milliseconds (ms) or less. As noted above, the time required to complete the task is independent of the number of data elements being displayed. Since the visual system cannot choose to refocus attention within this timeframe, users must complete their task using only a "single glance" at the image. Table 1 lists a number of preattentive features, and provides references that describe the tasks that can be performed using these features.

Fig. 1 shows examples of both types of target search. In Fig. 1a-1d the target, a red circle, is easy to find. Here, the target contains a preattentive feature unique from the background distracters: color (red versus blue) or shape (circle versus square). This unique feature is used by the low-level visual system to rapidly identify the presence or absence of the target. Unfortunately, an intuitive combination of these results can lead to visual interference. Fig. 1e and 1f simulate a two-dimensional dataset where one attribute is encoded with color (red or blue), and the other is encoded with shape (circle or square). Although these features worked well in isolation, searching for a red circle target in a sea of blue circles and red squares is significantly more difficult. In fact, experiments have shown that search time is directly proportional to the number of elements in the display, suggesting

[1] Although we now know that these visual features are influenced by the goals and expectations of the observer, the term preattentive is still useful because it conveys the relative ease with which these processes are completed.

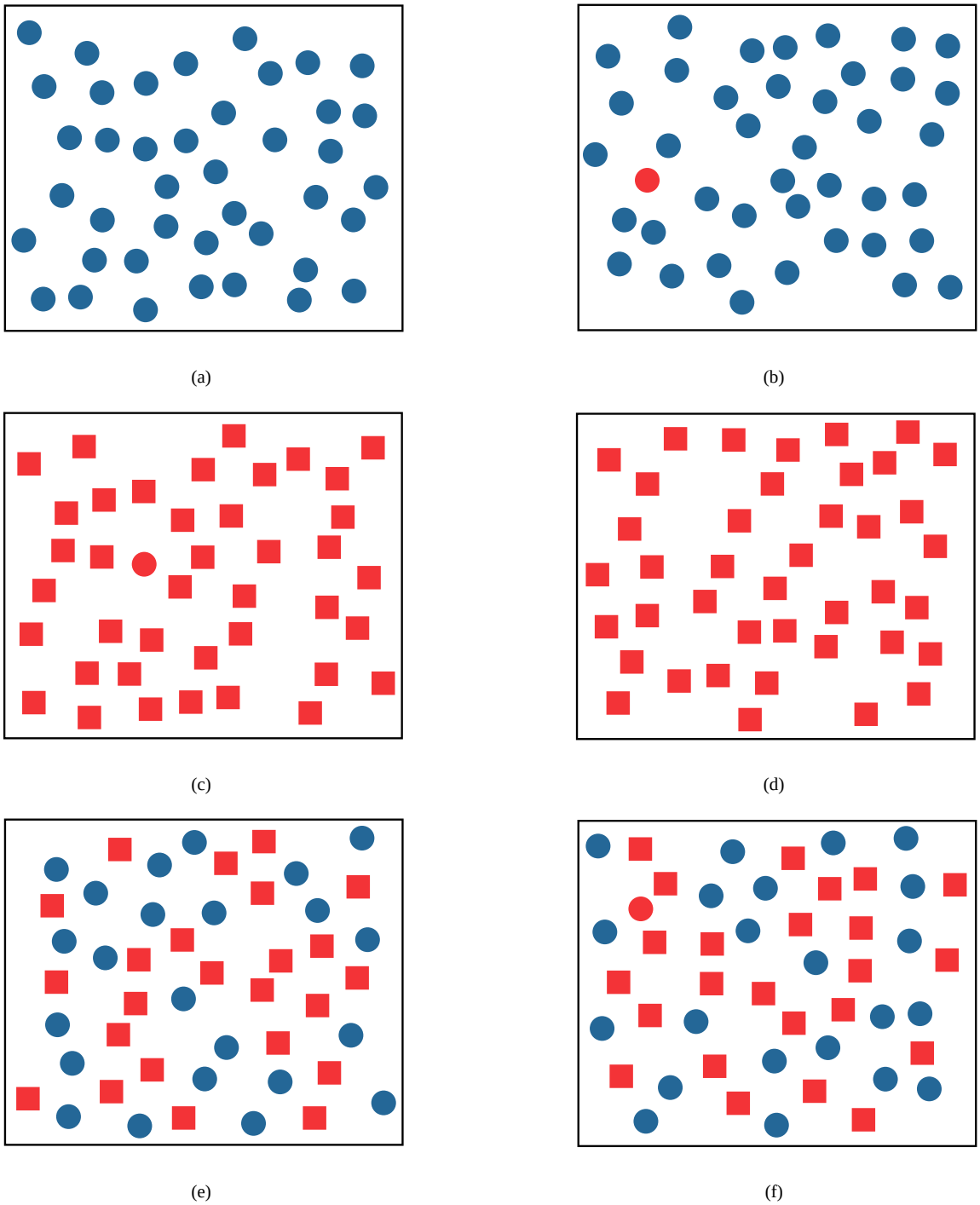


Figure 1: Examples of target search: (a, b) identifying a red target in a sea of blue distracters is rapid and accurate, target absent in (a), present in (b); (c, d) identifying a red circular target in a sea of red square distracters is rapid and accurate, target present in (c), absent in (d); (e, f) identifying the same red circle target in a combined sea of blue circular distracters and red square distracters is significantly more difficult, target absent in (e), present in (f)

that viewers are searching small subgroups of elements (or even individual elements themselves) to identify the target. In this example the low-level visual system has no unique feature to search for, since circular elements (blue circles) and red elements (red squares) are also present in the display. The visual system cannot integrate preattentively the presence of multiple visual features (circular and red) at the same spatial location. This is a very simple example of a situation where knowledge of preattentive vision would have allowed us to avoid displays that actively interfere with our analysis task.

In spite of the perceptual salience of the target in Fig. 1a-1d, bottom-up influences cannot be assumed to operate independently of the current goals and attentional state of the observer. Recent studies have demonstrated that many of the bottom-up factors only influence perception when the observer is engaged in a task in which they are expected or task-relevant (see the review by [14]). For example, a target defined as a color singleton will "pop out" of a display only when the observer is looking for targets defined by color. The same color singleton will not influence perception when observers are searching exclusively for luminance defined targets. Sometimes observers will fail completely to see otherwise salient targets in their visual field, either because they are absorbed in the performance of a cognitively-demanding task [41], there are a multitude of other simultaneous salient visual events [51], or because the salient event occurs during an eye movement or other change in viewpoint [62]. Therefore, the control of attention must always be understood as an interaction between bottom-up and top-down mechanisms.

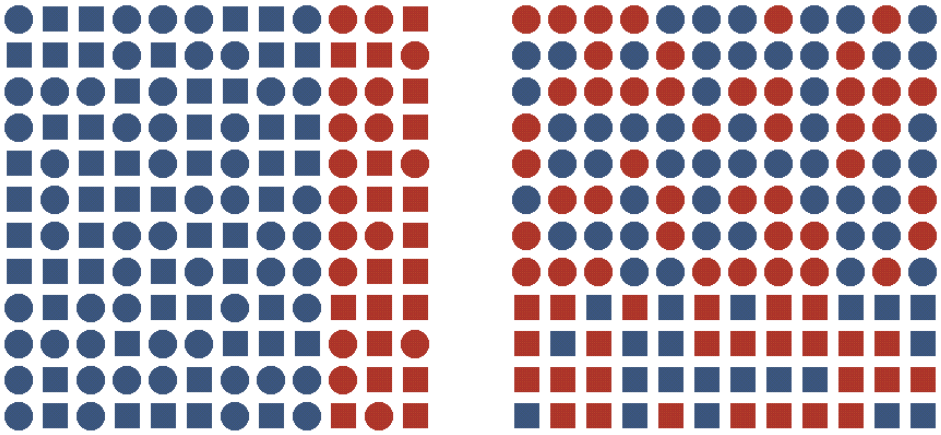
1. Visual analysis is rapid, accurate, and relatively effortless since preattentive tasks can be completed in 200 ms or less. We have shown that tasks performed on static displays extend to a dynamic environment where data frames are shown one after another in a movie-like fashion [23] (*i.e.,* tasks that can be performed on an individual display in 200 ms can also be performed on a sequence of displays shown at five frames a second).
2. The time required for task completion is independent of display size (to the resolution limits of the display). This means we can increase the number of data elements in a display with little or no increase in the time required to analyze the display.
3. Certain combinations of visual features cause interference patterns that mask information in the low-level visual system. Our experiments are designed to identify these situations. This means our visualization tools can be built to avoid data-feature mappings that might interfere with the analysis task.

Properties that are processed preattentively can be used to highlight important image characteristics. Experiments in both the cognitive psychology and scientific visualization domains have used various features to assist in performing the following visual tasks:

- *target detection*, where users attempt to rapidly and accurately detect the presence or absence of a "target" element that uses a unique visual feature within a field of distracter elements (Fig. 1),
- *boundary detection*, where users attempt to rapidly and accurately detect a texture boundary between two groups of elements, where all the elements in each group have a common visual feature (Fig. 2), and
- *counting and estimation*, where users attempt to count or estimate the number or percentage of elements in a display that have a unique visual feature.

Callaghan [7, 8] first reported the interference effects shown in Fig. 2. The visual system seems to prioritize features in order of importance. This means that the presence of visually "important" features can interfere with tasks that use lower priority features. In Fig. 2a, the vertical boundary defined by hue is detected preattentively, even though the shape of each element is random. In Fig. 2b, however, it is difficult to detect the horizontal boundary defined by form due to random hue variations. If hue were fixed to a constant value for each element, the form boundary could be detected preattentively. Callaghan explains this phenomena by suggesting that the visual system assigns a higher importance to hue than to form during boundary detection. Thus, a random hue interferes with form boundary detection, but a random form has no effect on hue boundary detection. A similar asymmetry exists between hue and intensity. Random hue has no effect on detecting boundaries defined by intensity. However, random intensity interferes with hue boundary detection. Callaghan concluded that intensity is more important than hue to the low-level visual system during boundary identification [6].

Researchers continue to expand preattentive processing in a number of exciting directions. To date, most of the features used in preattentive tasks have been relatively simply properties (*e.g.,* hue, orientation, line length, and size). Enns and Rensink,



(a) (b)

Figure 2: Region segregation by form and hue: (a) hue boundary is identified preattentively, even though form varies in the two regions; (b) random hue variations interfere with the identification of a region boundary based on form

however, have identified a class of three-dimensional elements that can be detected preattentively [15, 16]. They have shown that three-dimensional orientation, shape, and direction of lighting can be used to make elements "pop-out" of a visual scene (Fig. 3b and 3c). This is important, because it suggests that complex high-level concepts may be processed preattentively by the low-level visual system.

New tasks that can be performed preattentively are also being investigated. Research has recently been conducted on counting and estimation in preattentive processing. Varey describes experiments in which subjects were asked to estimate the relative frequency of white or black dots [76]. Her results showed that subjects could estimate in four different ways: "percentage" of white dots, "percentage" of black dots, "ratio" of black dots to white dots, and "difference" between the number of black and white dots. She also found that subjects consistently overestimated small proportions and underestimated large proportions. Estimation of relative frequency using hue and orientation was shown to be preattentive in experiments conducted in our laboratory [22, 24]. Moreover, our results showed that there was no feature interaction. Random orientation did not interfere with estimation of targets with a unique hue, and random hue did not interfere with estimation of targets with a unique orientation. This is important because it suggests that hue and orientation can be used to encode two independent data values in a single display without causing visual interference.

A number of scientists have proposed competing theories to explain how preattentive processing occurs, in particular Triesman's feature integration theory [70], Julész' texton theory [34], Quinlan and Humphreys' similarity theory [48], and Wolfe's guided search theory [83]. Our interest is in the use of visual features that have already been shown to be preattentive. Results from psychology are extended, modified, tested, and then integrated into our visualization environment.

Since preattentive tasks are rapid and insensitive to display size, we believe visualization techniques that make use of these properties will support high-speed exploratory analysis of large, multivariate datasets. Care must be taken, however, to ensure that we choose data-feature mappings that maximize the perceptual salience of all the data attributes being displayed.

2.1 Real-Time Preattentive Visualization

Most preattentive techniques are validated by studying a single data frame in isolation. This leads to an interesting question with important relevance to visualization. If a preattentive task can be performed on a single frame in 100 milliseconds, can the same task can be performed on a real-time sequence of frames displayed at ten frames per second? We hypothesized that important aspects of preattentive processing will extend to a real-time environment. A visualization tool that uses preattentive features will allow viewers to perform visual tasks such as grouping of similar data elements (boundary detection), detection of elements with a unique characteristic (target detection), and estimation of the number of elements with a given value or range of

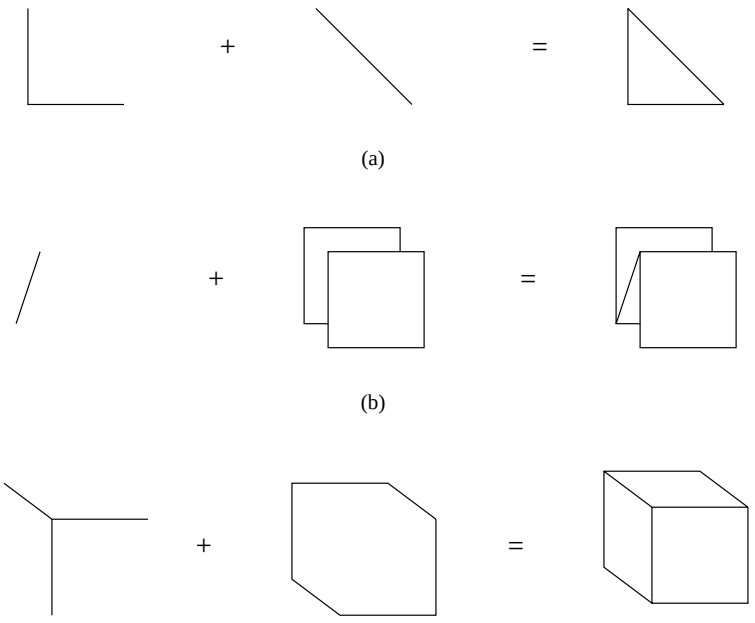


Figure 3: Combination of simple components to form emergent features: (a) closure, a simple closed figure is seen; (b) three-dimensionality, the figure appears to have depth; (c) volume, a solid figure is seen

values, all in real-time on temporally animated data frames. We tested this hypothesis using experiments that simulated the use of our preattentive visualization tools in a real-time environment. Analysis of the experimental results supported our hypothesis for boundary and target detection using hue and shape. Moreover, interference properties previously reported for static frames were found to apply to a dynamic environment.

Our initial experiment addressed two general questions about the preattentive features hue and shape, and their use in our visualization tools:

- *Question 1:* Is it possible for subjects to detect a data frame with a horizontal boundary within a sequence of random frames? If so, what features allow this and under what conditions?
- *Question 2:* Do Callaghan's feature hierarchy effects apply to our real-time visualization environment? Specifically, does random hue interfere with form boundary detection within a sequence of frames? Does random form interfere with hue boundary detection within a sequence of frames?

Experimental results showed accurate boundary detection can be performed using either hue or form on sequences of frames displayed at ten frames per second. Moreover, feature hierarchy effects extended to a dynamic environment, specifically, hue dominated form during boundary detection. A random hue pattern masked form boundaries, while a random form pattern had no effect on hue boundary detection.

A corresponding set of experiments were run to test target detection, with similar results. While both hue and form targets can be detected preattentively in a real-time environment (at frame rates of ten to twenty frames per second), form targets were only visible when the background hue was held constant. Hue variation masked form targets. Form variation had no effect on the detection of hue targets.

We have built a number of visualization tools that allow users to perform exploratory analysis on their datasets in real-time. Experience from using these tools confirmed that our experimental results hold for these datasets and tasks.

3 Color Selection

Color is a powerful and often-used visual feature. Previous work has addressed the issue of choosing colors for certain types of data visualization. For example, Ware and Beatty describe a simple color visualization technique for displaying correlation in a five-dimensional dataset [79]. Robertson, Ware, Rheingans and Tebbs, and Levkowitz and Herman discuss various methods for building effective color gamuts and colormaps [38, 52, 53, 78]. Recent work at the IBM Thomas J. Watson Research Center has focused on a rule-based visualization tool that considers how a user perceives visual features like hue, luminance, height, and so on [4, 54].

If we use color to represent our data, one important question to ask is: "How can we choose effective colors that provide good differentiation between data elements during the visualization task?" We addressed this problem by trying to answer three related questions:

- How can we allow rapid and accurate identification of individual data elements through the use of color?
- What factors determine whether a "target" element's color will make it easy to find, relative to differently colored "non-target" elements?
- How many colors can we display at once, while still allowing for rapid and accurate target identification?

None of the currently-available color selection techniques were specifically designed to investigate the rapid and accurate identification of individual data elements based on color. Also, since the color gamut and colormap work uses continuous color scales to encode information, they have not addressed the question of how many colors we can effectively display at once, while still providing good differentiation between individual data elements.

We began by using the perceptually balanced CIE LUV color model to provide control over color distance and isoluminance. We also exploited two specific results related to color target detection: linear separation [12, 2] and color category [35]. These effects are controlled to allow for the rapid and accurate identification of color targets. Target identification is a necessary first step towards performing other types of exploratory data analysis. If we can rapidly and accurately differentiate elements based on their color, we can apply our results to other important visualization techniques like detection of data boundaries, the tracking of data regions in real-time, and enumeration tasks like counting and estimation [24, 71, 76].

3.1 CIE LUV

The CIE LUV color model was proposed by the Commission Internationale de L'Èclairge (CIE) in 1976 [85]. Colors are specified using the three dimensions L^* (which encodes luminance), u^*, and v^* (which together encode chromaticity). CIE LUV provides two useful properties for controlling perceived color difference. First, colors with the same L^* are isoluminant. Second, Euclidean distance and perceived color difference (specified in ΔE^* units) can be interchanged, since the color difference between two color stimuli x and y (positioned in CIE LUV at (L^*_x, u^*_x, v^*_x) (L^*_y, u^*_y, v^*_y), respectively) is roughly:

$$\Delta E^*_{xy} = \sqrt{(\Delta L^*_{xy})^2 + (\Delta u^*_{xy})^2 + (\Delta v^*_{xy})^2} \tag{1}$$

3.2 Linear Separation

The linear separation effect has been described by both D'Zmura and Bauer et. al [2, 12]. D'Zmura was investigating how the human visual system finds a target color in a sea of background non-target colors. D'Zmura ran experiments that asked observers to determine the presence or absence of an orange target. Two groups of differently colored non-target elements were also present in each display (*e.g.,* in one experiment half the non-targets in each display were colored green and half were colored red). Results showed that when the target could be separated by a straight line from its non-targets in color space (Fig. 4, target T and non-targets A and C), the time required to determine the target's presence or absence was constant, and independent of the total number of elements being displayed. This suggests detection occurs preattentively in the low-level

visual system. When the target was collinear with its non-targets (Fig. 4, target T and non-targets A and B), the time required to identify the target was linearly proportional to the number of elements being displayed. Observers had to search serially through each display to determine whether the target was present or absent. Bauer et. al strengthened D'Zmura's results by showing that perceptually balanced color models cannot be used to overcome the linear separation effect. Bauer et. al also replicated their findings in three additional color regions: green, blue, and green-yellow. This suggests linear separation applies to colors from different parts of the visible color domain.

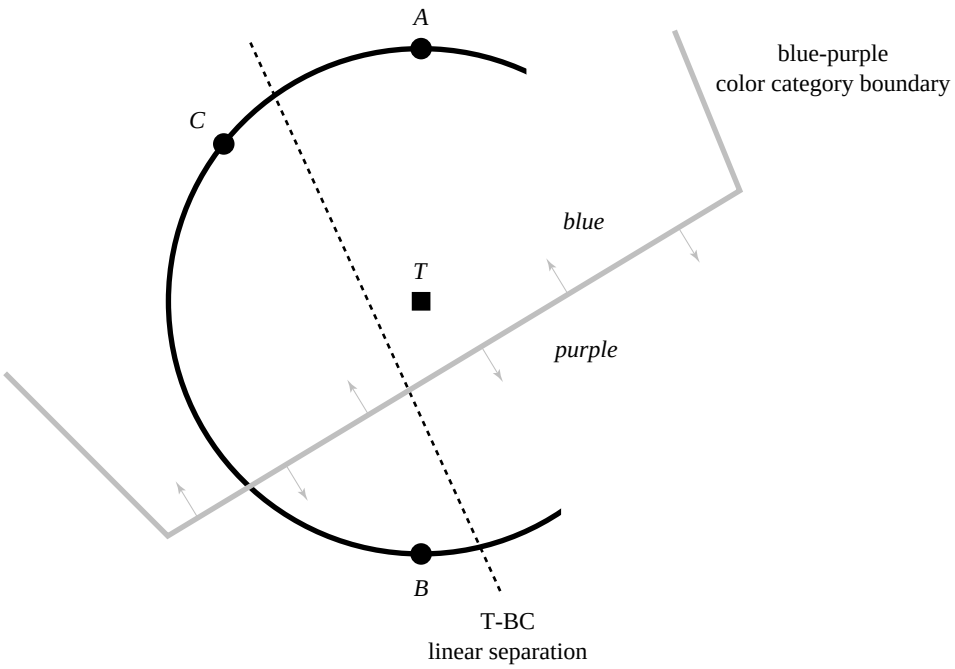


Figure 4: A small, isoluminant patch within the CIE LUV color model, showing a target color *T* and three background distracter colors *A*, *B*, and *C*; note that *T* is collinear with *A* and *B*, but can be separated by a straight line from *B* and *C*; note also that *T*, *A*, and *C* occupy the "blue" color region, while *B* occupies the "purple" color region

3.3 Color Category

Kawai et. al [35] reported results that suggest that the time required to identify a color target depends in part on the named color regions occupied by the target and its non-targets. Kawai et. al tested displays that contained a uniquely color target and a constant number of uniformly colored non-targets. They divided an isoluminant, perceptually balanced color slice into named color regions. Their results showed that search times decreased dramatically whenever the non-target was moved outside the target's color region. For example, finding a target colored T in a set of non-targets colored A was significantly more difficult than than finding T in a set of non-targets colored B (Fig. 4). Since the target–non-target distances $\overline{TA}$ and $\overline{TB}$ are equal, there was an expectation of perceptual balance that should have been provided by the underlying color model. This expectation was not met. Kawai et. al suggests the difference in performance is due to the fact that both T and A are located in the blue color region, but A is not.

3.4 Experiments

We ran a number of experiments to study the effects of color distance, linear separation, and color category. Subjects were shown displays that contained multiple colored squares. Each subject was asked to determine whether a target square with a particular color was present or absent in each display. The experiments were designed to test the following conditions:

- *selection criteria*, which selection criteria (color distance, linear separation, and color category) need to be considered to guarantee equally distinguishable colors,
- *simultaneous colors*, how many colors can we display at the same time, while still allowing users to rapidly and accurately determine the presence or absence of any particular color, and
- *display size*, is performance affected by the number of elements in a display.

We found that up to seven isoluminant colors can be displayed simultaneously, while still allowing for the rapid and accurate identification of any one of the seven. The time required to perform identification was independent of display size, suggesting that target detection is occurring preattentively. Our results also showed that all three selection criteria needed to be considered to guarantee consistent performance. When only some of the selection criteria were used (*e.g.* only distance and separation, or only category) the amount of time required to identify targets depended on the color of the target: some colors were very easy to identify, while other colors were very difficult. This asymmetry suggested that the colors we chose were not *equally* distinguishable, and therefore that the selection criteria being used were not sufficient to properly control perceived difference between the colors during target detection.

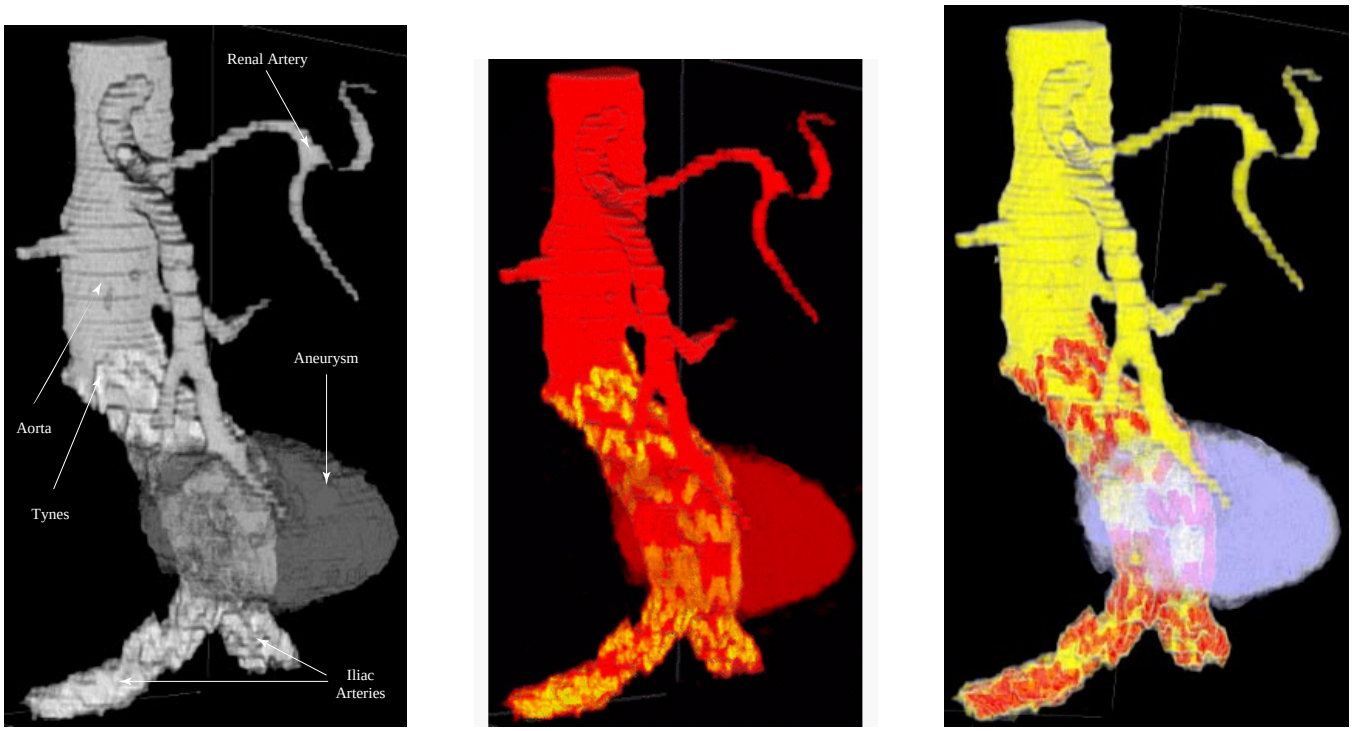
4 Visualizing CT Medical Scans

One practical application of our color selection technique is the use of color to highlight regions of interest during volume visualization [66]. Radiologists from the University Hospital at the University of British Columbia are studying methods for visualizing abdominal aneurisms. Traditional repair of an abdominal aneurism entails a major operation with an incision into the aneurism, evacuation of the clot contained within, placement of a synthetic graft, and wrapping of the graft with the remnants of the wall of the aneurism. Recently, a new treatment option, endovascular stenting, has been proposed and is currently undergoing clinical trials. This procedure does not require general anesthesia and can be done less invasively by simply placing a self-expanding stent graft via a catheter into the aneurism to stabilize it. Less fit patients are able to withstand the procedure, hospital stay is cut to 1 to 2 days, and post-operative recovery is shortened considerably.

After the operation computed tomography (CT) scans are used to obtain two-dimensional slices of a patient's abdominal region. These slices are reconstructed to produce a three-dimensional volume. The volume is visualized by the radiologists to perform post-operative analysis. A two-pass segmentation step is used to strip out material in each CT slice that does not correspond to one of the regions of interest: the artery running through the abdomen, the aneurism, and the metal hooks (called tynes) used to embed the stent graft within the aneurism. The reconstructed volumes must show clearly each of these three regions.

Normally, greyscale is used to display reconstructed medical volumes. Changes in luminance are most effective for representing the high spatial frequency data contained in these kinds of datasets. For our application, however, one of the most important tasks is identifying the exact position of the tynes (which in turn identify the positions of each of the corresponding stent grafts). In our greyscale volume the location of tynes within the aneurism are obscured by the wall of the aneurism itself (Fig. 5a). Different levels of transparency were used to try to "see through" the aneurism, however, we could not find any appropriate value that showed the tyne locations within the artery, while at the same time providing an effective representation of the three-dimensional shape and extent of the wall of the aneurism. We decided that, for this application, it might be more appropriate to high the three regions of interest using color.

Although the radiologists had already chosen a set of colors based on context and aesthetic considerations, it did a poor job of showing the size and shape of the aneurism (Fig. 5b). We replaced their colors with three new ones using our color selection technique. The radiologists asked us to avoid greens and green-yellows, since these are associated with bile. We decided to use yellow to represent the artery, purple to represent the aneurism, and red to represent the tynes (Fig. 5c). These colors show clearly the location of all three regions of interest within the volume. For example, consider the large patches of yellow within the aneurism. These are areas of "low support"where the grafts in the lower part of the artery were not inserted far enough to mesh with their upstream partner. Although not dangerous, these are exactly the kinds of features the radiologists want to identify during post-operative visualization.



(a) (b) (c)

Figure 5: A reconstructed CT volume showing an abdominal aneurism: (a) greyscale hides the location of the tynes within the aneurism; (b) a color scale chosen by the radiologists obscures the shape and extent of the wall of the aneurism; (c) three colors chosen with our perceptual color selection technique

5 Perceptual Textures

One of the important issues in scientific visualization is designing methods for representing multiple values (or attributes) at a single spatial location. Although it is possible to assign different visual features to each attribute, simply "stacking" multiple features together will most likely lead to displays that are unintelligible.

Rather than choosing multiple individual features (*i.e.*, color, shape, size, orientation, line length), we decided to try using a single visual feature that could be decomposed into a set of fundamental parts or dimensions. We chose to investigate texture for this purpose.

Texture has been studied extensively in the computer vision, computer graphics, and cognitive psychology communities. Although each group focuses on separate tasks (texture identification and texture segmentation in computer vision, displaying information with texture patterns in computer graphics, and modeling the low-level human visual system in cognitive psychology) they each need ways to describe precisely the textures being identified, classified, or displayed.

Researchers have used different methods to study the perceptual features inherent in a texture pattern. Bela Julész [32] conducted numerous experiments that investigated how a texture's first, second, and third-order statistics affect discrimination in the low-level human visual system. This led to the texton theory [33], which suggests that early vision detects three types of features (or textons, as Julész called them): elongated blobs with specific visual properties (*e.g.*, hue, orientation, and width), ends of line segments, and crossings of line segments. Tamura et al. [67] and Rao and Lohse [49, 50] identified texture dimensions by conducting experiments that asked subjects to divide pictures depicting different types of textures (Brodatz images) into groups. Tamura et al. used their results to propose methods for measuring coarseness, contrast, directionality, line-likeness, regularity, and roughness. Rao and Lohse used multidimensional scaling to identify the primary texture dimensions used by

their subjects to group images: regularity, directionality, and complexity. Haralick et al. [19] built greyscale spatial dependency matrices to identify features like homogeneity, contrast, and linear dependency. These features were used to classify satellite images into categories like forest, woodlands, grasslands, and water. Liu and Picard [40] used Wold features to synthesize texture patterns. A Wold decomposition divides a 2D homogeneous pattern (*e.g.,* a texture pattern) into three mutually orthogonal components with perceptual properties that roughly correspond to periodicity, directionality, and randomness. Malik and Perona [42] designed computer algorithms that use orientation filtering, nonlinear inhibition, and computation of the resulting texture gradient to mimic the discrimination ability of the low-level human visual system. We used these results to choose the perceptual texture dimensions we wanted to investigate during our experiments.

Work in computer graphics has studied methods for using texture patterns to display information during visualization. Schweitzer [59] used rotated discs to highlight the orientation of a three-dimensional surface. Pickett and Grinstein [17] built "stick-men" icons to produce texture patterns that show spatial coherence in a multidimensional dataset. Ware and Knight [80, 81] used Gabor filters to construct texture patterns; attributes in an underlying dataset are used to modify the orientation, size, and contrast of the Gabor elements during visualization. Turk and Banks [75] described an iterated method for placing streamlines to visualize two-dimensional vector fields. Interrante [29] displayed texture strokes to help show three-dimensional shape and depth on layered transparent surfaces; principal directions and curvatures are used to orient and advect the strokes across the surface. Finally, Salisbury et al. [57] and Wikenbach and Salesin [82] used texturing techniques to build computer-generated pen-and-ink drawings that convey a realistic sense of shape, depth, and orientation. We built upon these results to try to develop an effective method for displaying multidimensional data through the use of texture.

5.1 Pexels

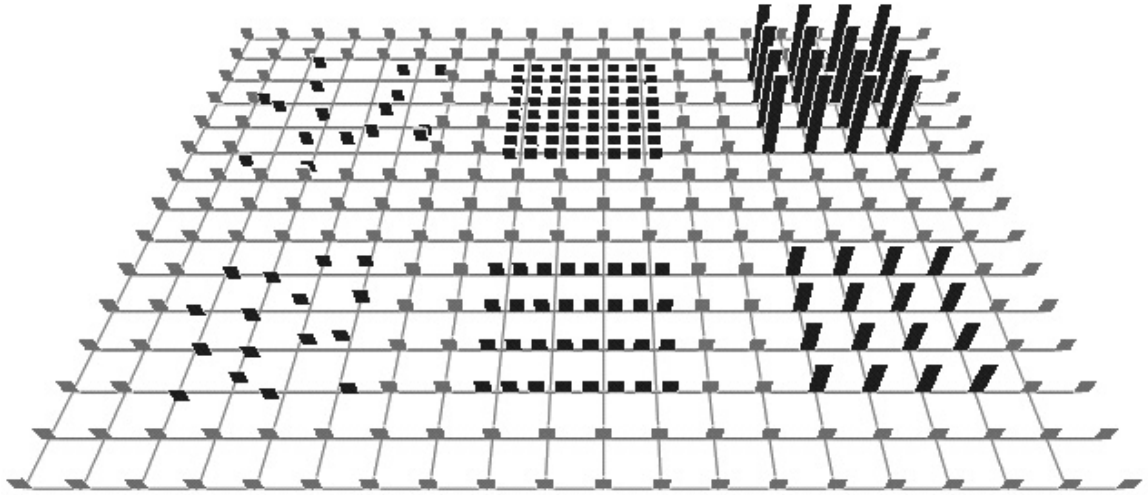
We wanted to design a technique that will allow users to visualize multidimensional datasets with perceptual textures. To this end, we used a method similar to Ware and Knight to build our displays. Each data element is represented with a single perceptual texture element, or pexel. Our visualization environment consists of a large number of elements arrayed across a three-dimensional surface (*e.g.,* a topographical map or the surface of a three-dimensional object). Each element contains one or more attributes to be displayed. Attribute values are used to control the visual appearance of a pexel by modifying its texture dimensions. Texture patterns formed by groups of spatially neighboring pexels can be used to visually analyze the dataset.

We chose to study three perceptual dimensions: density, regularity, and height. Density and regularity have been identified in the literature as primary texture dimensions [49, 50, 67]. Although height might not be considered an "intrinsic textural cue", we note that height is one aspect of element size, and that element size is an important property of a texture pattern. Moreover, results from cognitive vision have shown that differences in height are detected preattentively by the low-level visual system [1, 70]. We wanted to build three-dimensional pexels that "sit up" on the underlying surface. This allows the possibility of applying various orientations (another important perceptual dimension) to a pexel. Because of this, we chose height as our third texture dimension.

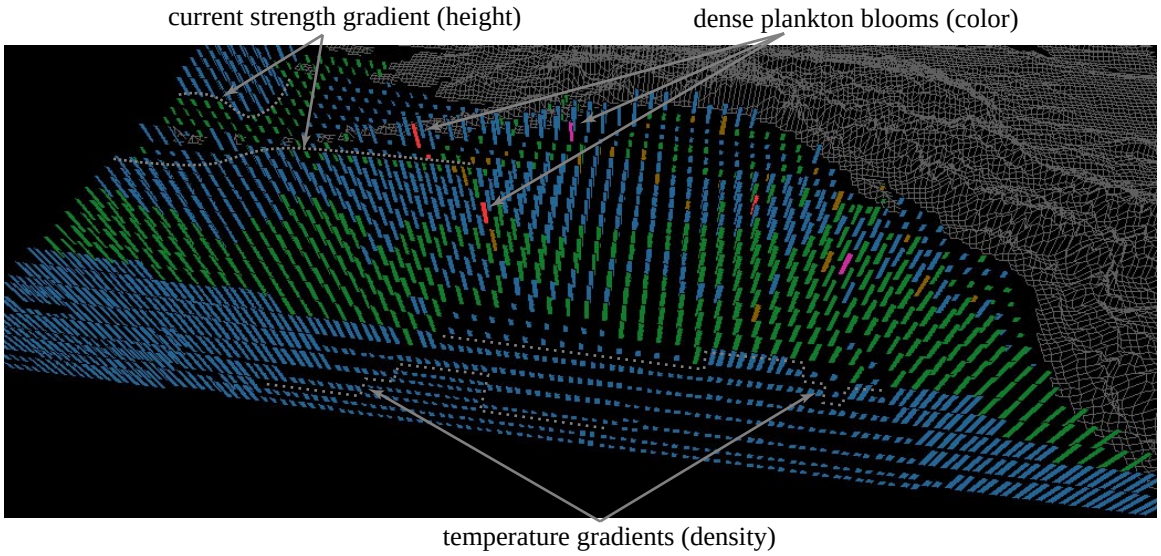
In order to support variation of height, density, and regularity, we built pexels that look like a collection of paper strips. The user maps attributes in the dataset to the density (which controls the number of strips in a pexel), height, and regularity of each pexel. Examples of each of these perceptual dimensions are shown in Fig. 6a. Fig. 6b shows an example of our technique applied to the oceanographic dataset: environmental conditions in the northern Pacific Ocean are visualized using multicolored pexels. In this display, color represents open-ocean plankton density, height represents ocean current strength (taller for stronger), and density represents sea surface temperature (denser for warmer). Fig. 6b is only one frame from a much larger time-series of historical ocean conditions. Our choice of visual features was guided by experimental results that show how different color and texture properties can be used in combination to represent multivariate data elements.

5.2 Experiments

In order to test our perceptual dimensions and the interactions that occur between them during visualization, we ran a set of psychophysical experiments. Our experiments were designed to investigate a user's ability to rapidly and accurately identify target pexels defined by a particular height, density, or regularity. Users were asked to determine whether a small group of pexels with a particular type of texture (*e.g.,* a group of taller pexels, as in Fig. 7a) was present or absent in a 20×15 array.



(a)



(b)

Figure 6: Pexel examples: (a) a background array of short, sparse, regular pexels; the lower and upper groups on the left contain irregular and random pexels, respectively; the lower and upper groups in the center contain dense and very dense pexels; the lower and upper groups to the right contain medium and tall pexels; (b) Color, height, and density used to visualize open-ocean plankton density, ocean current strength, and sea surface temperature, respectively; low to high plankton densities represented with blue, green, brown, red, and purple, stronger currents represented with taller pexels, and warmer temperatures represented with denser pexels

(a) (b)

Figure 7: Two display types from the taller and regular pexel experiments: (a) a target of medium pexels in a sea of short pexels with a background density pattern (2×2 target group located left of center); (b) a target of regular pexels in a sea of irregular pexels with no background texture pattern (2×2 target group located 8 grids step right and 2 grid steps up from the lower-left corner of the array)

Conditions like target pexel type, exposure duration, target group size, and background texture dimensions differed for each display. This allowed us to test for preattentive task performance, visual interference, and a user preference for a particular target type. In all cases, user accuracy was used to measure performance.

Each experimental display contained a regularly-spaced 20×15 array of pexels rotated 45° about the X-axis (Fig. 7). All displays were monochromatic (*i.e.*, grey and white), to avoid variations in color or intensity that might mask the underlying texture pattern. Grid lines were drawn at each row and column, to ensure users perceived the pexels as lying on a tilted 3D plane. After a display was shown, users were asked whether a group of pexels with a particular target value was present or absent. In order to avoid confusion, each user searched for only one type of target pexel: taller, shorter, sparser, denser, more regular, or more irregular. The appearance of the pexels in each display was varied to test for preattentive performance, visual interference, and feature preference. For example, the following experimental conditions were used to investigate a user's ability to identify taller pexels (similar conditions were used for the shorter, denser, sparse, regular, and irregular experiments):

- two target-background pairings: a target of medium pexels in a sea of short pexels, and a target of tall pexels in a sea of medium pexels; different target-background pairings allowed us to test for a subject preference for a particular target type,
- three display durations: 50 msec, 150 msec, and 450 msec; we varied exposure duration to test for preattentive performance, specifically, does the task become more difficult during shorter exposures,
- three secondary texture dimensions: none (every pexel is sparse and regular), density (half the pexels are randomly chosen to be sparse, half to be dense), and regularity (half the pexels are regular, half are random); we added a "background" texture feature to test for visual interference, that is, does the task become more difficult when a secondary texture dimension appears at random spatial locations in the display, and
- two target group sizes: 2×2 pexels and 4×4 pexels; we used different target group sizes to see how large a group of pexels was needed before the target could be detected by a viewer.

Our results suggest that pexels can be used to represent multidimensional data elements, but only if specific data-feature mappings are chosen. Some dimensions were more salient than others, and interference occurred when certain types of pexels were displayed. Specifically, we found that:

- taller pexels can be identified at preattentive exposure durations (*i.e.*, 150 msec or less) with very high accuracy (approximately 93%); background density and regularity patterns produce no significant interference,
- shorter, denser, and sparser pexels are more difficult to identify than taller pexels, although good results are possible at both 150 and 450 msec; height, regularity, and density background texture patterns cause interference for all three target types,

- irregular pexels are difficult to identify, although reasonable accuracy (approximately 76%) is possible at 150 and 450 msec with no background texture pattern, and
- regular pexels cannot be accurately identified; the percentage of correct results approached chance (*i.e.*, 50%) for every condition.

These results show that height and density can be used to to form texture patterns that can be identified preattentively. Regularity, however, can only be used as a secondary dimension. While differences in regularity cannot be detected consistently by the low-level visual system, in many cases users will be able to see changes in regularity when areas of interest in a dataset are identified and analyzed in a focused or attentive fashion.

6 Orientation

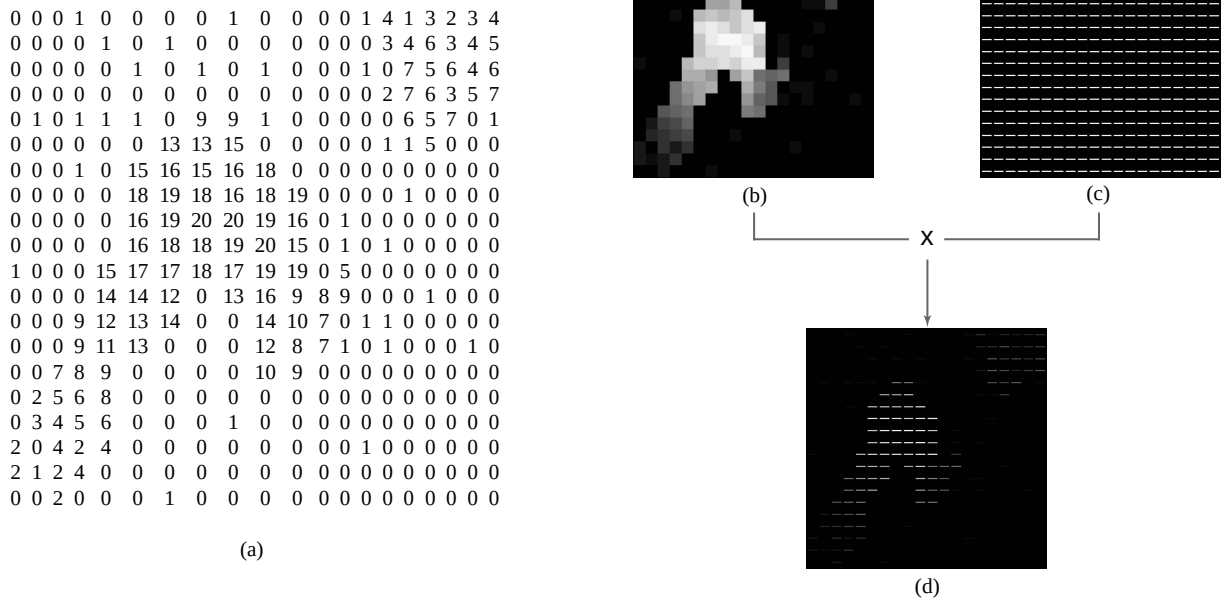


Figure 8: (a) A 20×20 patch of values from a scalar field; (b) the patch represented by greyscale swatches; (c) a collection of slivers oriented $0°$ at each data value location; (d) the greyscale map and slivers and combined to produce the final sliver layer

A follow-up study was recently conducted to test another perceptual texture dimension: orientation. This work was motivated in part by the desire to construct an alternative method for visualizing multiple overlapping data fields. The well-known method of selecting m visual features to represent each of the m data attributes has a number of inherent limitations:

- *dimensionality:* as the number of attributes n in the dataset grows, it becomes more and more difficult to find additional visual features to represent them.
- *interference:* different visual features will often interact with one another, producing visual interference; these interference effects must be controlled or eliminated to guarantee effective exploration and analysis.
- *attribute-feature matching:* different visual features are best suited to a particular type of attribute and analysis task; an effective visualization technique needs to respect these preferences.

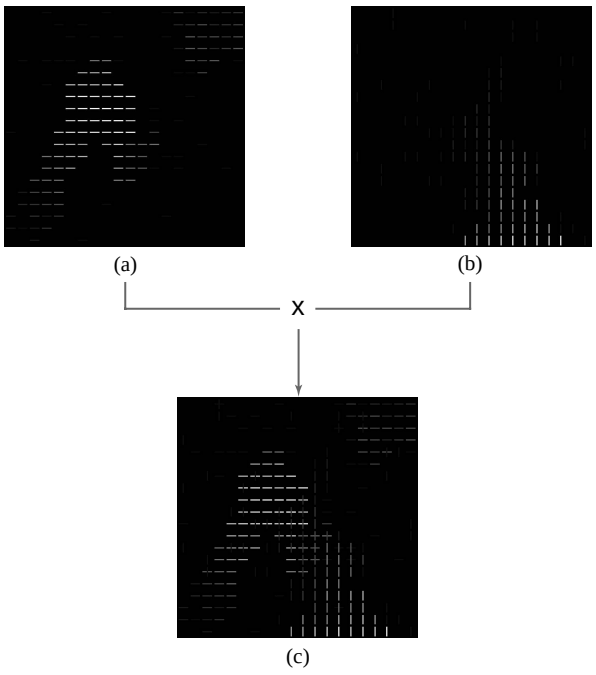


Figure 9: (a,b) two scalar fields represented with 0° and 90°, respectively; (c) both fields displayed in a single image, overlapping values show as elements that look like plus signs

Multidimensional datasets can often be viewed as a collection of m scalar fields that overlap spatially with one another. Rather than using m visual features to represent these fields, we can use only two: orientation and luminance. For each scalar field (representing attribute A_j) we select a constant orientation o_j; at various spatial locations where a value $a_{i,,j} \in A_j$ exists, we place a corresponding sliver texture oriented at o_j. The luminance of the sliver texture depends on $a_{i,j}$: the maximum $a_{max\_j} \in A_j$ produces a white (full luminance) sliver, while the minimum $a_{min\_j} \in A_j$ produces a black (zero luminance) sliver. A perceptually-balanced luminance scale running from black to white is used to select a luminance for an intermediate values $a_{i,j},\ a_{min\_j} < a_{i,j} < a_{max\_j}$.

Fig. 8a shows a uniformly-sampled patch from a hypothetical scalar field. Values in the field are represented as greyscale swatches in Fig. 8b. A constant orientation of 0° is used to represent values in the field (slivers rotated 0° a replaced at the spatial locations for each reading in the field, shown in Fig. 8c). Blending these two representations together produces the final image (Fig. 8d), a layer of variable-luminance slivers showing the positions and values of all the data in the original field.

Multiple scalar fields are displayed by compositing their sliver layers together. Fig. 9a-b shows two separate sliver layers representing two scalar fields. The first field uses slivers oriented 0°; the second uses slivers oriented 90°. When a viewer visualizes both fields simultaneously, the sliver layers are overlayed to produce the single image shown in Fig. 9c. This image allows the viewer to locate values in each individual field, while at the same time identifying important interactions between the fields. The use of thin, well separated slivers is key to allowing values from multiple fields to show through in a common spatial location. A viewer can use these images to:

- determine which fields are prominent in a region,
- determine how strongly a given field is present,
- estimate the relative weights of the field values in the region, and
- locate regions where all the fields have low, medium, or high values.

6.1 Experiments

Our initial experiments investigated a viewer's ability to distinguish sliver textures with different 2D orientations (*i.e.,* slivers with different rotations embedded in the XY-plane). Each trial contained a 20×20 grid of rectangles rotated bg. A randomly selected 2×2 patch of target rectangles was then rotated tg. In half the trials $bg = tg$ (*i.e.*, the target was absent, Fig. 10b). In the other half, $bg \neq tg$ (*i.e.*, a target was present, Fig. 10a and 10c). We tested background orientations ranging from 0-45° and 45-90° in 5° steps. For each background, every orientation was tested as a target (*i.e.*, ten target rotations 0, 5, . . ., 45° were tested for each background in the range 0-45°; ten target rotations 45, 50, . . ., 90° were tested for each background in the range 45-90°).

In summary, our results showed:

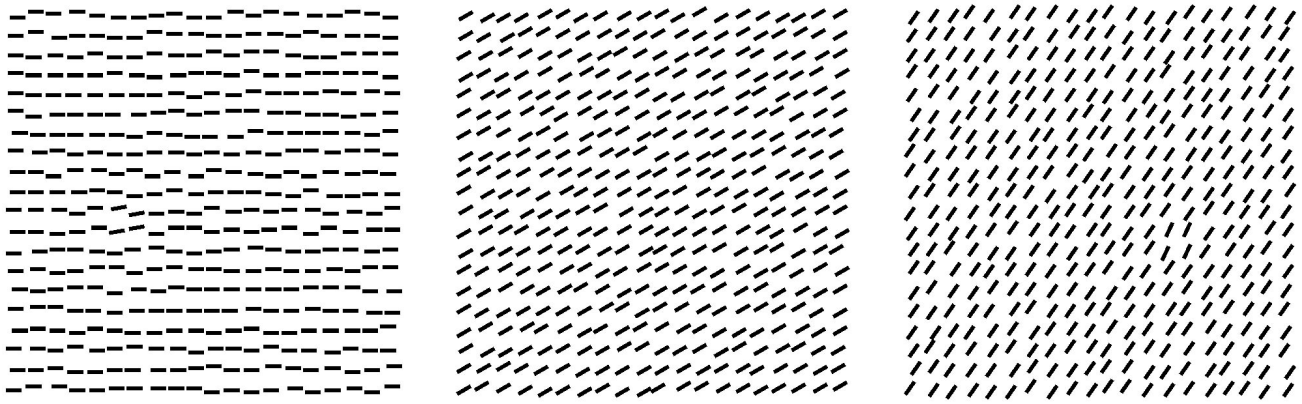


Figure 10: An example of three experiment displays: (a) a 10° target in a 0° background (target is five steps right and eight steps up from the lower left corner of the array); (b) a 30° background with no target; (c) a 65° target in a 55° background (target is six steps left and seven steps up from the lower right corner of the array)

1. A target oriented $\pm 15^\circ$ or more from its background elements is rapidly and accurately distinguishable.
2. Errors for backgrounds oriented 0° or 90° were significantly lower than for other backgrounds (*e.g.*, tilted targets are easier to see in backgrounds of 0° or 90°).
3. Errors for targets oriented 0° or 90° were significantly higher than for other targets, suggesting an asymmetry (good as a background, bad as a target) for these orientations.

7 Scanning Electron Microscope Images

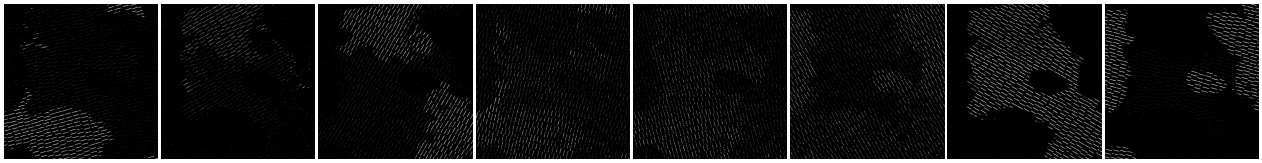
The application for which this technique was originally developed is the display of multiple data fields from a scanning electron microscope (SEM). Each field represents the concentration of a particular element (oxygen, silicon, carbon, and so on) across a surface. Physicists studying mineral samples need to determine what elements make up each part of the surface and how those elements mix. By allowing the viewer to see the relative concentrations of the elements in a given area, our technique enables recognition of composites more easily than side-by-side comparison, especially for situations where there are complex amalgams of materials.

Fig. 11a shows sliver layers representing eight separate elements: calcium (15°), copper (30°), iron (60°), magnesium (75°), manganese (105°), oxygen (120°), sulphur (150°), and silicon (165°), The orientations for each layer were chosen to ensure no two layers have an orientation difference of less than 15° Fig. 11b shows the eight layers blended together to form a single image. Fig. 11c changes the orientations of silicon and oxygen to 90° and 180°, respectively, to investigate a potential interaction between the two (the presence of silicon oxide in the upper right, upper left, and lower left where regions of "plus sign" textures appear).

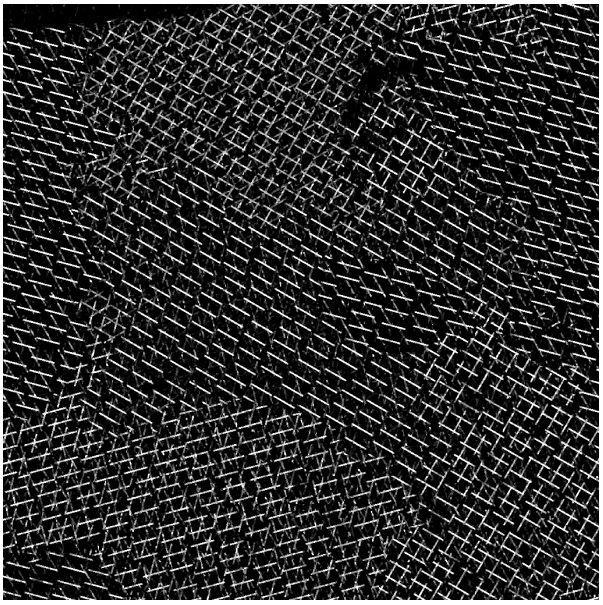
Our results suggest up to 15 orientations in the 0-180° range can be rapidly and accurately differentiated. The greyscale ramp used to assign a luminance to each sliver is also constructed to be perceptually linear. The result is an image that shows data values in each individual field, while at the same time highlighting important interactions between the fields.

8 Combining Color and Texture

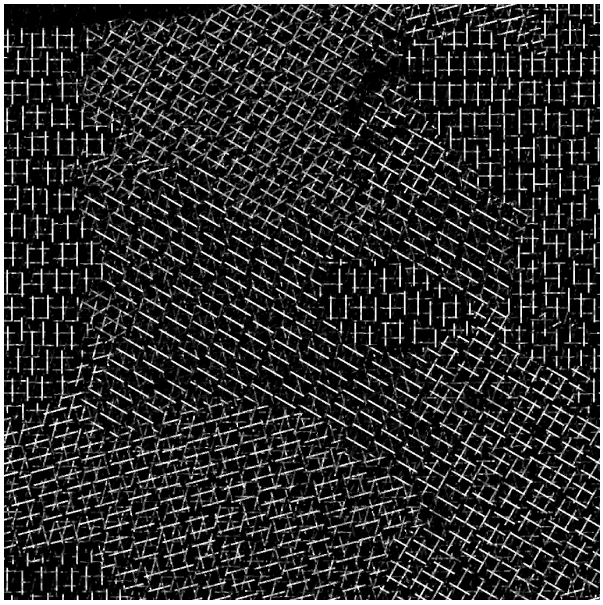
Although texture and color have been studied extensively in isolation, much less work has focused on their combined use for information representation. An effective method of displaying color and texture patterns simultaneously would increase the



(a)



(b)



(c)

Figure 11: (a) eight sliver layers representing calcium (15°), copper (30°), iron (60°), magnesium (75°), manganese (105°), oxygen (120°), sulphur (150°), and silicon (165°), (b) all eight layers blended into a single image; (c) silicon and oxygen re-oriented at 90° and 180°, respectively, to highlight the presence of silicon oxide (as a "plus sign" texture) in the upper right, upper left, and lower left corners of the image

number of attributes we can represent at one time. The first step towards supporting this goal is the determination of the amount of visual interference that occurs between these features during visualization.

Experiments in psychophysics have produced interesting but contradictory answers to this question. Callaghan reported asymmetric interference of color on form during texture segmentation: a random color pattern interfered with the identification of a boundary between two groups of different forms, but a random form pattern had no effect on identifying color boundaries [7, 8]. Triesman, however, showed that random variation of color had no effect on detecting the presence or absence of a target element defined by a difference in orientation (recall that directionality has been identified as a fundamental perceptual texture dimension) [73]. Recent work by Snowden [64] recreated the differing results of both Callaghan and Triesman. Snowden ran a number of additional experiments to test the effects of random color and stereo depth variation on the detection of a target line element with a unique orientation. As with Callaghan and Triesman, results differed depending on the target type. Search for a single line element was rapid and accurate, even with random color or depth variation. Search for a spatial collection of targets was easy only when color and depth were fixed to a constant value. Random variation of color or depth produced a significant reduction in detection accuracy. Snowden suggests that the visual system wants to try to group spatially neighboring elements with common visual features, even if this grouping is not helpful for the task being performed. Any random variation of color or depth interferes with this grouping process, thereby forcing a reduction in performance.

These results left unanswered the question of whether color variation would interfere with texture identification during visu-

alization. Moreover, work in psychophysics studied two-dimensional texture segmentation. Our pexels are arrayed over an underlying height field, then displayed in 3D using a perspective projection. Most of the research to date has focused on color on texture interference. Less work has been conducted to study how changes in texture dimensions like height, density, or regularity will affect the identification of data elements with a particular target color. The question of interference in this kind of height-field environment needs to be addressed before we can recommend methods for the combined use of color and texture.

8.1 Experiments

In order to investigate the combined use of color and texture, we designed a new set of psychophysical experiments. Our two specific questions were:

1. Does random variation in pexel color influence the detection of a region of target pexels defined by height or density?
2. Does random variation in pexel height or density influence the detection of a region of target pexels defined by color?

We chose to ignore regularity, since it performed poorly as a target defining property during all phases of our original texture experiments [25, 26]. We chose three different colors using our perceptual color selection technique [20, 25]. Colors were initially selected in the CIE LUV color space, then converted to our monitor's RGB gamut. The three colors corresponded approximately to red (monitor RGB = 246, 73, 50), green (monitor RGB = 49, 144, 21) and blue (monitor RGB = 82, 109, 168). Our new experiments were constructed around a set of conditions similar to those used during the original texture experiments. We varied target-background pairings, exposure duration, the presence or absence of a secondary background feature, and target patch size from trial to trial to test user performance in a number of different circumstances.

Mean percentage target detection accuracy was the measure of performance. Observer responses were collected, averaged, and analyzed using multi-factor ANOVA. In summary, we found:

1. Color targets were detected rapidly (*i.e.,* at 150 ms) with very high accuracy (96%). Background variation in height and density produced no interference effects in this detection task.
2. Detection accuracy for targets defined by density or height were very similar to results from our original texture experiments [25, 26]. When there was no background variation in color, excellent detection accuracy was obtained for density defined targets (*i.e.,* denser and sparser targets) at 150 ms (94%). Height defined targets (*i.e.,* taller and shorter) were detected somewhat less accurately at 150 ms (88%) but were highly detectable at 450 ms (93%). As we had also found previously, taller targets were generally easier to detect than shorter targets, and denser targets were easier than sparser targets.
3. In all four texture experiments, background variation in color produced a small but significant interference effect, averaging 6% in overall accuracy reduction.
4. The absolute reduction in accuracy due to color interference depended on the difficulty of the detection task. Specifically, color interfered more with the less visible target values (shorter and sparser targets yielded a mean accuracy reduction of 8%) than with the more visible targets (taller and denser targets yield a mean accuracy reduction of 4%).

Our results showed an asymmetric interference effect, similar to those reported in the psychophysical literature. As described by [64], we found that color produces a small but statistically reliable interference effect during texture segmentation. Moreover, we found color and texture form a "feature hierarchy" that produces asymmetric interference: color variation interferes with an observer's ability to see texture regions based on height or density, but variation in texture has no effect on region detection based on color. This is similar to reports by [7, 8], who reported asymmetric color on shape interference in a boundary detection task involving two-dimensional textures. The amount of color on texture interference depended on the difficulty of the segmentation task. Targets that were harder to identify in isolation (*e.g.,* shorter and sparser targets) showed a much higher sensitivity to color interference, compared to targets that were easy to identify (*e.g.,* taller and denser targets). This suggests that color and texture can be combined in a single display, but only in cases where the texture targets have a strong perceptual salience.

9 Nonphotorealistic Visualization

For many years the areas of modeling and rendering in computer graphics have studied the problem of producing photorealistic images, images of graphical models that are indistinguishable from photographs of an equivalent real-world scene. Advances in areas like the simulation of global light transport, modeling of natural phenomena, and image-based rendering have made dramatic strides towards achieving this goal. At the same time, researchers have approached the issue of image generation from a completely different direction. Although photographs are common, there are many other compelling methods of visual discourse, for example, oil and watercolor paintings, pen and ink sketches, cel animation, and line art. In the proper situations, these types of pictures are often considered more effective, more appropriate, or even more expressive than an equivalent photograph. The study of methods that construct images of these types is known as *nonphotorealistic rendering*.

Our current interests lie specifically in nonphotorealistic rendering methods that use simulated brush strokes to produce images that look like paintings. Strassmann [65] constructed a "hairy brush", a collection of bristles placed along a line segment; Japanese-style brush strokes are produced by applying ink to the brush, then moving it along a path over a simulated paper surface. Haberli [18] built a system that paints with a brush that a user strokes across an underlying image; the size, shape, color, location, and direction of brush strokes can all be varied. Hsu et al. [28] used vector-based skeletal strokes with variable thickness drawn along a parametric curve to produce interesting line-art images. Meier [44] attached particles to surfaces in a 3D scene, then rendered a brush stroke (with variable color, size, and direction) at each particle position to paint the scene. Litwinowicz [39] clipped strokes to object boundaries, then rendered them as lines or texture maps with variable length, thickness, direction, and color. Curtis et al. [10] built a fluid-flow simulation to model the interactions of brush, watercolor, and paper during the painting of watercolor images. Shiraishi and Yamaguchi [60] computed image moments on a target image to control the color, location, orientation, and size of the brush strokes in their painterly rendering. Hertzmann [27] used a multilayer painting technique, where each new layer contains finer details painted with smaller brush strokes; brush paths are modeled with splines, while the brush itself allows variation of length, size, opacity, placement, and color jitter. Interrante [30] discussed applying natural textures to visualize multidimensional datasets. Laidlaw [36, 37] extended the layered approach of Meier to visualize multidimensional data in a painterly fashion. Finally, Ebert and Rheingans [13] used nonphotorealistic techniques like silhouettes, sketch lines, and halos to highlight important features in a volumetric dataset.

9.1 Painterly Styles

We believe that fundamental properties of a nonphotorealistic image can be identified in part by studying the styles used by artists to construct their paintings. Our investigation of painterly styles is directed by two separate criteria. First, we are restricting our search to a particular movement in art known as Impressionism. Second, we attempt to pair each style with a corresponding visual feature that has been shown to be effective in a perceptual visualization environment. There are no technical reasons for why Impressionism was chosen over any other movement. In fact, we expect the basic theories behind our technique will extend easily to other types of artistic presentation. For our initial work, however, we felt it was important to narrow our focus to a set of fundamental goals in the the context of a single type of painting style.

The term "Impressionism" was attached to a small group of French artists (initially including Monet, Degas, Manet, Renoir, and Pissaro, and later Cézanne, Sisley, and Van Gogh, among others) who broke from the traditional schools of the time to approach painting from a new perspective. The Impressionist technique was based on a number of underlying principles [5, 58, 77], for example:

1. *Object and environment interpenetrate.* Outlines of objects are softened or obscured (*e.g.*, Monet's water lilies); objects are bathed and interact with light; shadows are colored and movement is represented as unfinished outlines.
2. *Color acquires independence.* There is no constant hue for an object, atmospheric conditions and light moderate color across its surface; objects may reduce to swatches of color.
3. *Show a small section of nature.* The artist is not placed in a privileged position relative to nature; the world is shown as a series of close-up details.
4. *Minimize perspective.* Perspective is shortened and distance reduced to turn 3D space into a 2D image.

5. *Solicit a viewer's optics.* Study the retinal system; divide tones as separate brush strokes to vitalize color rather than graying with overlapping strokes; harness simultaneous contrast; use models from color scientists like Chevruel [9] or Rood [55].

Although these general characteristics are perhaps less precise than we might prefer, we can still draw a number of important conclusions. Properties of hue, luminance, and lighting were explicitly controlled and even studied in a scientific fashion by some of the Impressionists. Rather than using an "object-based" representation, the artists appear to be more concerned with subdividing a painting based on the interactions of light with color and other surface properties. Additional painterly styles can be identified by studying the paintings themselves. These styles often varied dramatically between individual artists, acting to define their unique painting techniques. Examples include:

- *path:* the direction a brush stroke follows; Van Gogh made extensive use of curved paths to define boundaries and shape in his paintings; other artists favored simpler, straighter strokes,
- *length:* the length of individual strokes on the canvas, often used to differentiate between contextually different parts of a painting,
- *density:* the number and size of strokes laid down in a fixed area of canvas,
- *coarseness:* the coarseness of the brush used to apply a stroke; a coarser brush causes visible bristle lines and surface roughness, and
- *weight:* the amount of paint applied during each stroke; heavy strokes highlight coarseness and produce ridges of paint that cause underhanging shadows when lit from the proper direction.

Although this collection of painterly styles provides a good starting point, it is by no means exhaustive. All of the styles we use are evaluated for effectiveness by identifying their perceptual characteristics, and by validating their ability to support visualization, discovery, analysis, and presentation in a real-world application environment.

A comparison of perceptual color and texture properties with painterly styles from Impressionist art reveals a strong correspondence between the two. Reduced to perceptual elements, color and texture are the precise properties that an artist varies in the application of colored pigments of paint to a canvas with a brush. From this perspective, color and lighting in Impressionism has a direct relationship to the use of hue and luminance in perceptual vision. Other painterly styles (*e.g.*, path, density, and length) have similar partners in perception (*e.g.*, orientation, contrast, and size). This close correspondence between perceptual features and many of the painterly styles we hope to apply is particularly advantageous. Since numerous controlled experiments on the use of perception have already been conducted, we have a large body of evidence to use to predict how we expect painterly styles to react in a multidimensional visualization environment.

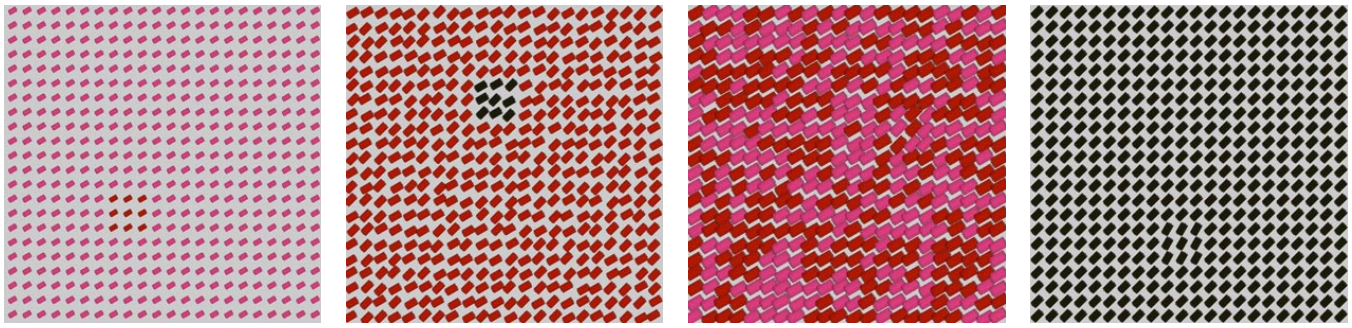
9.2 Experiments

We conducted a set of psychophysical experiments to test our hypothesis that guidelines from human perception will extend to a painterly environment. Our experiments were designed to investigate a viewer's ability to rapidly and accurately identify target brush strokes defined by a particular color or orientation. Background orientation, color, regularity, and density varied between displays. This allowed us to test for preattentive task performance, and for visual interference effects. The experimental results were then used to identify similarities and differences between painterly images and existing perceptual visualization techniques.

Each experimental display contained a 22×22 array of simulated brush strokes (Fig. 12). Viewers were asked to determine whether a small, 3×3 group of strokes with a particular target type was present or absent in each display. Displays were shown for 200 milliseconds, after which the screen was cleared; the system then waited for viewers to enter their answer (either "target present" or "target absent").

The displays were equally divided into two groups: one studied a viewer's ability to identify target strokes based on color, the other studied identification based on orientation. The appearance of the strokes in each display was varied to test for preattentive performance and visual interference. For example, Fig. 12a shows an orange target in a sea of pink strokes; all the strokes have a constant orientation of 30^{circ}; they are sparsely packed, and are located in a completely regular, grid-like pattern.

Fig. 12b shows a green target in a sea of orange strokes; these strokes, however, have a random orientation, a dense packing, and an irregular placement. Fig. 12c shows a 45° orientation target in a sea of strokes rotated 30°; the strokes have a random background color, very dense packing, and irregular placement. Finally, Fig. 12d shows a 60° target in a 45° background; these strokes have a constant color, dense packing, and regular placement.



(a) (b) (c) (d)

Figure 12: Example experiment displays: (a) orange target in pink strokes, constant 30° background orientation, sparse packing, regular placement; (b) green target in orange strokes, random background orientation, dense packing, irregular placement; (c) 45° target in 30° strokes, random background color, very dense packing, irregular placement; (d) 60° target in 45° strokes, constant background color, dense packing, regular placement

Viewer accuracy and response time were combined and tested for significance using a multi-factor analysis of variance (ANOVA). In summary, we found:

- color targets were easy to detect at a 200 millisecond preattentive exposure duration; a random orientation pattern had no interfering effect on performance,
- orientation targets were easy to detect when a constant color was displayed in the background; a random background color pattern caused a significant reduction in performance,
- background density had a significant effect on both color and orientation targets; denser displays produced an improvement in performance, and
- background regularity had a significant effect on both color and orientation targets; irregular displays caused a reduction in performance.

Our results match previous findings in both the psychophysical and the visualization literature, specifically: (1) color produces better performance than orientation during target identification, and (2) an asymmetric color on texture interference effect exists (random color patterns interfere with orientation identification, but not vice-versa). Both results have been shown to exist in experimental [8, 64] and real-world visualization settings [25, 26]. These results extend our previous work [26], which found a general color on texture interference pattern, but no corresponding texture on color effect. Overall, our results provide positive evidence to support the belief that perceptual findings will carry to a painterly visualization environment.

10 Visualizing Environmental Weather Data

Based on the results from our experiments, we decided to build a nonphotorealistic visualization system that varied brush stroke color, spatial density, direction (*i.e.*, orientation), and stroke placement (*i.e.*, regularity) to encode multiple data attributes (in addition to the two spatial values used to position each stroke). The presence of feature hierarchies suggest color should be used to represent the most important attribute, followed by texture properties. Our results further refine this to mapping color, direction, density, and placement in order of attribute importance (from most important to least important).

The brush strokes in our current prototype are similar to the ones shown during our experiments. They are constructed using a simple texture mapping scheme. This technique is common in nonphotorealistic rendering (*e.g.*, in [18, 27, 39, 44]). Real painted strokes are captured and converted into texture maps. These textures are applied to an underlying polygon to produce an approximation of a collection of generic brush strokes. We currently use a small library of five representative brush stroke textures; one of the textures is randomly selected and applied when a stroke is rendered. A stroke's color, direction, density, and placement are controlled by modifying its texture and by transforming the polygon it maps to.

One of the application testbeds for our nonphotorealistic visualization technique is a collection of monthly environmental and weather conditions collected and recorded by the Intergovernmental Panel on Climate Change. This dataset contains mean monthly surface climate readings in $\frac{1}{2}^\circ$ latitude and longitude steps for the years 1961 to 1990 (*e.g.*, readings for January averaged over the years 1961-1990, readings for February averaged over 1961-1990, and so on). We chose to visualize values for mean temperature, windspeed, pressure, precipitation, and frost frequency (or *temp*, *wind pressure*, *precip*, and *frost)*. Based on this order of importance, we built a data-feature mapping M that varies brush stroke color, density, orientation, and regularity. We divided density into two separate parts: *energy*, the number of strokes used to represent a data element e_i, and *coverage*, the percentage of e_i's screen space covered by its strokes. Both properties represent painterly styles. Energy describes the number and vitality of strokes in a fixed region of a painting (*e.g.*, a few long, broad, lazy strokes or many small, short, energetic strokes). Coverage describes the amount of the underlying canvas, if any, that shows through the strokes. This produced the following attribute to visual feature pairings:

- *temp* $\rightarrow$ *color:* dark blue for low *temp* to bright pink for high *temp*,
- *wind* $\rightarrow$ *coverage:* low coverage for weak *wind* to full coverage for strong *wind*,
- *pressure* $\rightarrow$ *energy:* a single stroke, a 1×2 array of strokes, or a 2×2 array of strokes for low to high *pressure*,
- *precip* $\rightarrow$ *orientation:* upright (90° rotation) for light *precip* to flat (0° rotation) for heavy *precip*, and
- *frost* $\rightarrow$ *regularity:* regular for low *frost* frequency to irregular for high *frost* frequency.

Fig. 13 shows an example of applying M to data for February along the east coast of the continental United States. The top five images use a perceptual color ramp (running from dark blue and green for small values to bright red and pink for large values) to show the individual variation in *temp*, *pressure*, *wind*, *precip*, and *frost*. The result of applying M to construct a nonphotorealistic visualization of all five attributes is shown in the bottom image. Various color and texture patterns representing different weather phenomena are noted on this image. Changes in temperature are visible as a smooth blue-green to red-pink color variation running north to south over the map. Pressure gradients produce energy boundaries, shown as regions with different numbers of strokes packed into a unit area of screen space (*e.g.*, higher energy strokes in Florida represent higher *pressure* readings). Windspeed modifies stroke coverage: weak *wind* values produce small strokes with a large amount of background showing through (*e.g.*, north of the Great Lakes), while strong *wind* values produce larger strokes that completely fill their corresponding screen space (*e.g.*, in central Texas and Kansas). Increases in rainfall are shown as a increasing stroke tilt running from upright (light *precip)* to flat (heavy *precip)*. Finally, a higher frost frequency produces more irregularity (*e.g.*, strokes in Florida and southern Texas are completely regular, while strokes in the northern states and Canada are highly irregular).

Fig. 14 uses the same mapping M to visualize weather conditions over the western United States for January and August. These visualizations provide a number of interesting insights into historical weather conditions for this part of the continent. In January (Fig. 14a) weak *wind* values (shown as small, low coverage strokes) highlight the locations of the Rocky Mountains, the Cascades, and the Sierra Nevada range. Typically heavy rainfall in the Pacific Northwest is represented by nearly flat strokes. Regions of severe cold east of the Rocky Mountains near Denver and in the northern plains and Canadian prairies appear as patches of dark green and blue strokes. Low pressure (*i.e.*, a single low energy stroke for each data element) and high frost frequency (shown as irregular stroke placement) covers most of the map. Conditions in August (Fig. 14b) are markedly different. Most of the western United States is warm (a completely regular placement of strokes denoting little or no frost during this month). An area of intense heat, shown as bright pink strokes, is visible in southern California and Arizona. High pressure regions cover the coast, the south, and much of the central and northern plains. Little or no rainfall is evident. Finally, an area of weak *wind* values is visible as small, low coverage strokes in northern Washington, Idaho, and Montana.

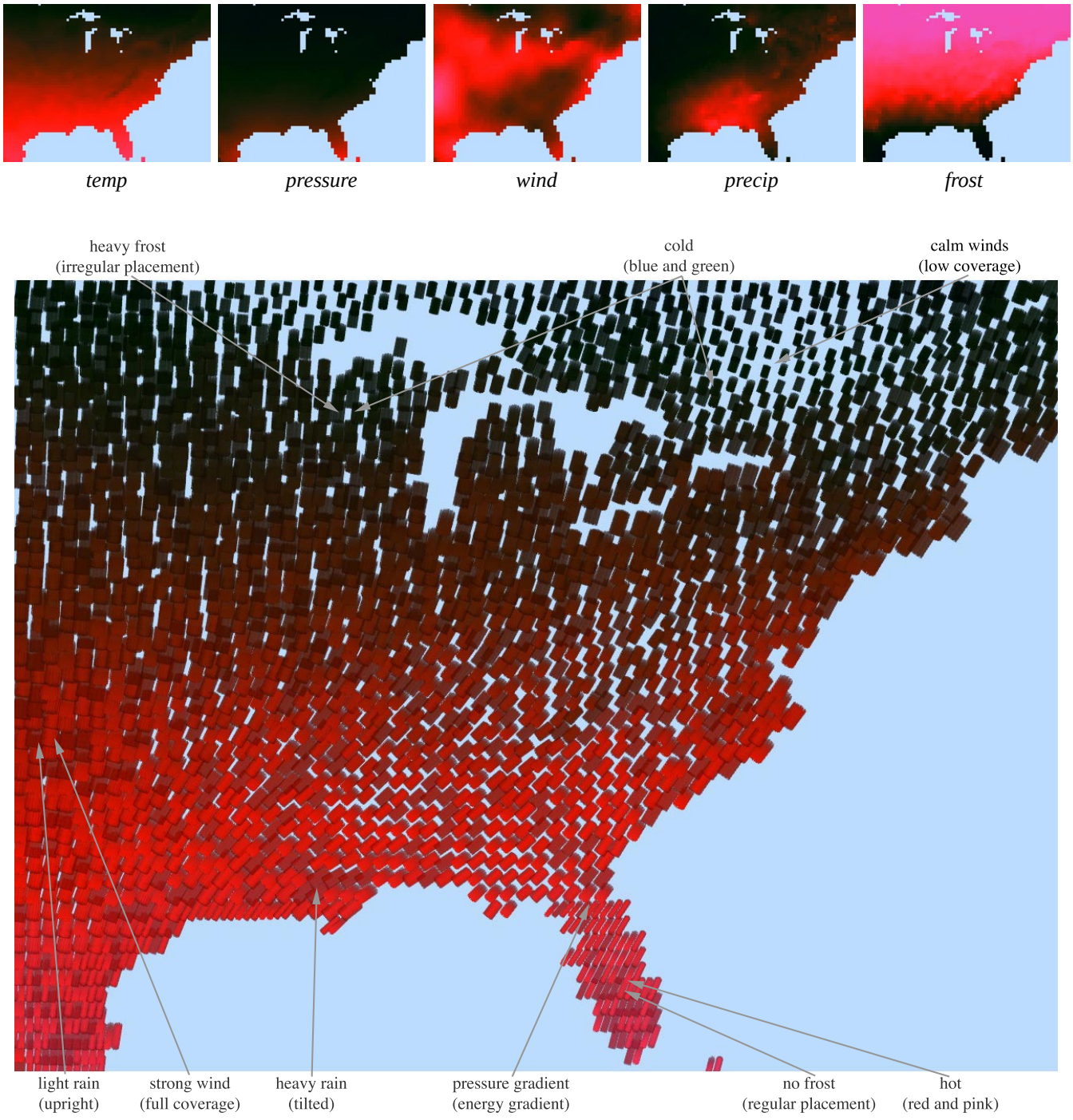
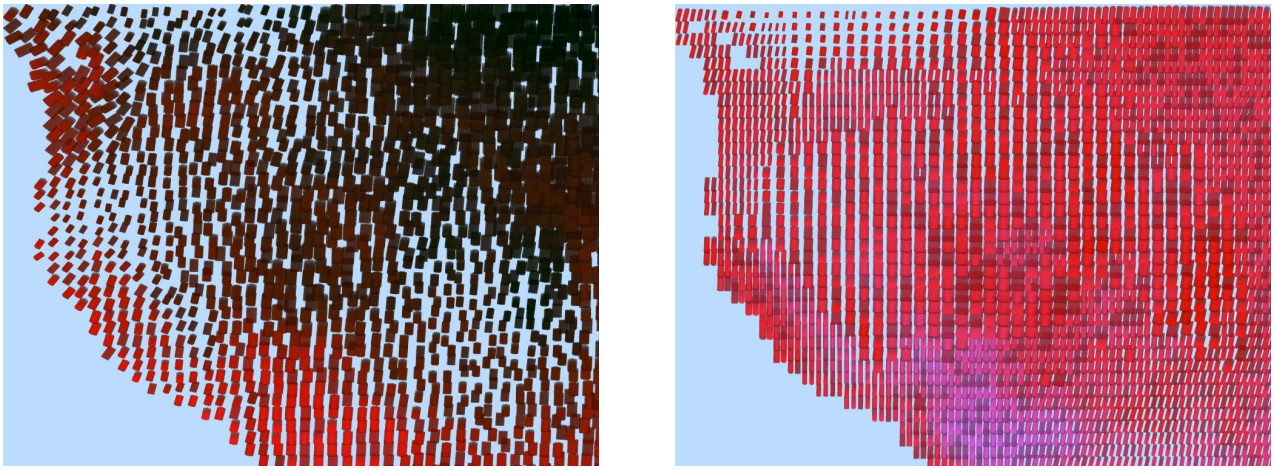


Figure 13: Nonphotorealistic visualization of weather conditions over the eastern United States: (top row) perceptual color ramps (dark blue for low to bright pink for high) of mean temperature, pressure, windspeed, precipitation, and frost frequency in isolation; (bottom row) combined visualization of temperature (dark blue to bright pink for cold to hot), pressure (low to to high energy for low to high), windspeed (low to high coverage for weak to strong), precipitation (upright to flat for light to heavy), and frost frequency (regular to irregular for low to high)



(a) (b)

Figure 14: Weather conditions over the western United States: (a) mean *temp*, *pressure*, *wind*, *precip* and *frost* for January; (b) mean conditions for August

11 Visualizing Typhoon Data

One of our current testbeds for using pexels to visualize multidimensional data is the analysis of environmental conditions related to typhoons. We used pexels to visualize typhoon activity in the Northwest Pacific Ocean during the summer and fall of 1997. The names "typhoon" and "hurricane" are region-specific, and refer to the same type of weather phenomena: an atmospheric disturbance characterized by low pressure, thunderstorm activity, and a cyclic wind pattern. Storms of this type with windspeeds below 17m/s are called "tropical depressions". When windspeeds exceed 17m/s, they become "tropical storms". This is also when storms are assigned a specific name. When windspeeds reach 33m/s, a storm becomes a typhoon (in the Northwest Pacific) or a hurricane (in the Northeast Pacific and North Atlantic).

We combined information from a number of different sources to collect the data we needed. A U.S. Navy elevation dataset[2] was used to obtain land elevations at ten minute latitude and longitude intervals. Land-based weather station readings collected from around the world and archived by the National Climatic Data Center[3] provided daily measurements for eighteen separate environmental conditions. Finally, satellite archives made available by the Global Hydrology and Climate Center[4] contained daily open-ocean windspeed measurements at thirty minute latitude and longitude intervals. The National Climatic Data Center defined the 1997 typhoon season to run from August 1 to October 31; each of our datasets contained measurements for this time period.

We chose to visualize three environmental conditions related to typhoons: windspeed, pressure, and precipitation. All three values were measured on a daily basis at each land-based weather station, but only daily windspeeds were available for open-ocean positions. In spite of the missing open-ocean pressure and precipitation, we were able to track storms as they moved across the Northwest Pacific Ocean. When the storms made landfall the associated windspeed, sea-level pressure, and precipitation were provided by weather stations along their path.

Based on our experimental results, we chose to represent windspeed, pressure, and precipitation with height, density, and color, respectively. Localized areas of high windspeed are obvious indicators of storm activity. We chose to map increasing windspeed to an increased pexel height. Although our experimental results showed statistically significant interference from background color variation, the absolute effect was very small. Taller and denser pexels were easily identified in all other cases, suggesting there should be no changes in color interference due to an increase in task difficulty. Windspeed has two important boundaries:

[2] http://grid2.cr.usgs.gov/dem/
[3] http://www.ncdc.noaa.gov/ol/climate/online/gsod.html
[4] http://ghrc.msfc.nasa.gov/ghrc/list.html

17m/s (where tropical depressions become tropical storms) and 33m/s (where storms become typhoons). We mirrored these boundaries with height discontinuities. Pexel height increases linearly from 0-17m/s. At 17m/s, height approximately doubles, then continues linearly from 17-33m/s. At 33m/s another height discontinuity is introduced, followed by a linear increase for any windspeeds over 33m/s.

Pressure is represented with pexel density. Since our results showed it was easier to find dense pexels in a sea of sparse pexels (as opposed to sparse in dense), an increase in pressure is mapped to a decrease in pexel density (*i.e.*, very dense pexels represent the low pressure regions associated with typhoons). Three different texture densities were used to represent three pressure ranges. Pressure readings less than 996 millibars, between 996 and 1014 millibars, and greater than 1014 millibars produce very dense, dense, and sparse pexels, respectively.

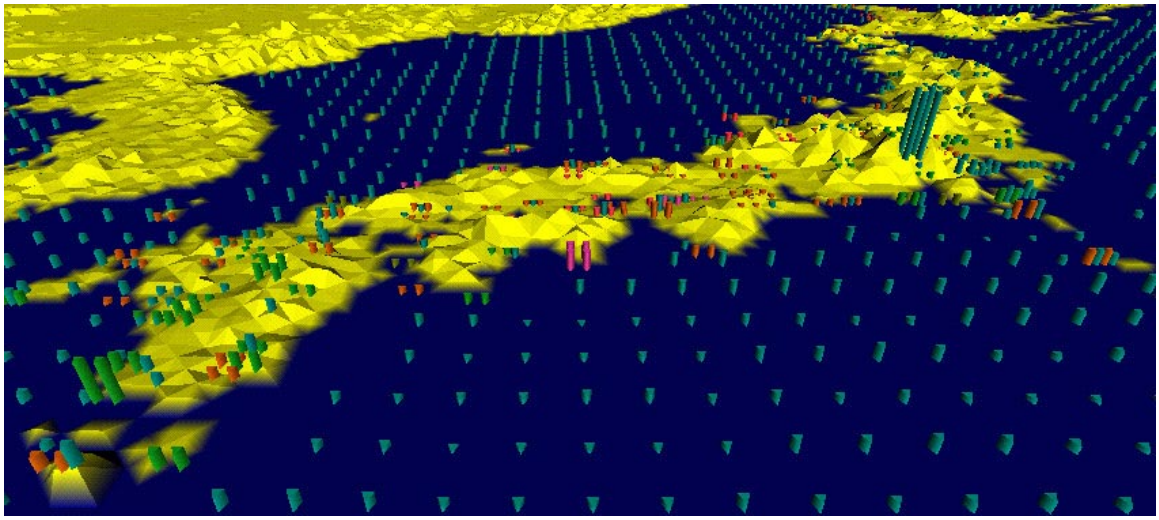
Precipitation is represented with color. We used our perceptual color selection technique to choose five perceptually uniform colors. Daily precipitation readings of zero, 0–0.03 inches, 0.03–0.4 inches, 0.4–1.0 inches, and 1.0–10.71 inches were colored green, yellow, orange, red, and purple, respectively (each precipitation range had an equal number of entries in our typhoon dataset). Pexels on the open ocean or at weather stations where no precipitation values were reported were colored blue-green. Our experimental results showed no texture-on-color interference. Moreover, our color selection technique is designed to produce colors that are equally distinguishable from one another. Our mapping uses red and purple to highlight the high-precipitation areas associated with typhoon activity.

We built a simple visualization tool that maps windspeed, pressure, and precipitation to their corresponding height, density, and color. Our visualization tool allows a user to move forwards and backwards through the dataset day-by-day. One interesting result was immediately evident when we began our analysis: typhoon activity was not represented by high windspeed values in our open-ocean dataset. Typhoons normally contain severe rain and thunderstorms. The high levels of cloud-based water vapor produced by these storms block the satellites that are used to measure open-ocean windspeeds. The result is an absence of any windspeed values within a typhoon's spatial extent. Rather than appearing as a local region of high windspeeds, typhoons on the open-ocean are displayed as a "hole", an ocean region without any windspeed readings (see Fig. 15b and 15d). This absence of a visual feature (*i.e.*, a hole in the texture field) is large enough to be salient in our displays, and can be preattentively identified and tracked over time. Therefore, users have little difficulty finding storms and watching them as they move across the open ocean. When a storm makes landfall, the weather stations along the storm's path provide the proper windspeed, as well as pressure and precipitation. Weather stations measure windspeed directly, rather than using satellite images, so high levels of cloud-based water vapor cause no loss of information.

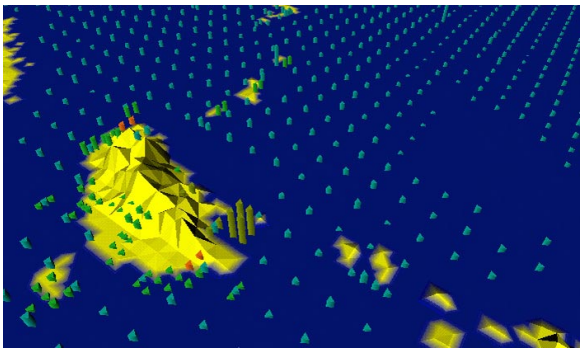
Fig. 15 shows windspeed, pressure, and precipitation around Japan, Korea, and Taiwan during August 1997. Fig. 15b, looking northeast, tracks typhoon Amber (one of the region's major typhoons) approaching along an east to west path across the Northwest Pacific Ocean on August 27, 1997. Fig. 15c shows typhoon Amber one day later as it moves through Taiwan. Weather stations within the typhoon show the expected strong winds, low pressure, and high levels of rainfall. These results are easily identified as tall, dense, red and purple pexels. Compare these images to Fig. 15d and 15e, where windspeed was mapped to regularity, pressure to height, and precipitation to density (a mapping without color that our original texture experiments predict will perform poorly). Although viewers can identify areas of lower and higher windspeed (*e.g.*, on the open ocean and over Taiwan), it is difficult to identify *a change* in lower or higher windspeeds (e.g., the change in windspeed as typhoon Amber moves onshore over Taiwan). In fact, viewers often searched for an increase in density that represents an increase in precipitation, rather than an increase in irregularity; pexels over Taiwan become noticeably denser between Fig. 15d and 15e.

12 Oceanography Simulations

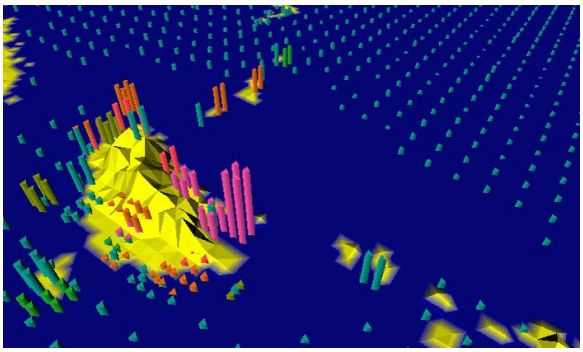
Our final example describes a set of oceanography simulations being jointly conducted at North Carolina State University and the University of British Columbia [21]. Researchers in oceanography are studying the growth and movement patterns of different species of salmon in the northern Pacific Ocean. Underlying environmental conditions like plankton density, sea surface temperature (SST), current direction, and current strength affect where the salmon live and how they move and grow [68]. For example, salmon like cooler water and tend to avoid ocean locations above a certain temperature. Since the salmon feed on plankton blooms, they will try to move to areas where plankton density is highest. Currents will "push" the salmon as they swim. Finally, SST, current direction, and current strength affect the size and location of plankton blooms as they form.



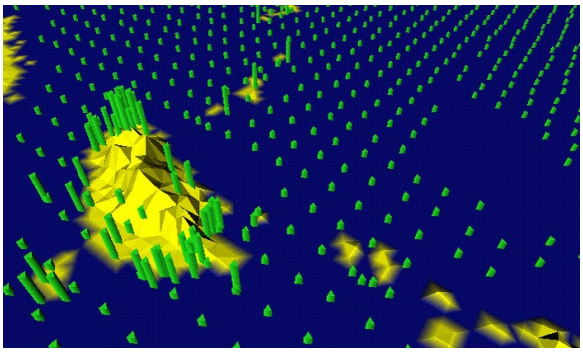
(a)



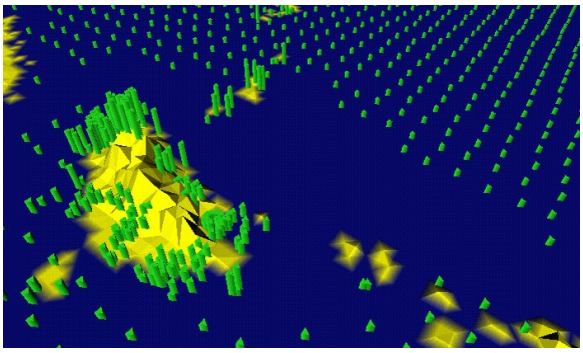
(b)



(c)



(d)



(e)

Figure 15: Typhoon conditions across Southeast Asia during the summer of 1997: (a) August 7, normal weather conditions over Japan; (b) August 27, typhoon Amber approaches the island of Taiwan from the southeast; (c) August 28, typhoon Amber strikes Taiwan, producing tall, dense pexels colored orange, red, and purple (representing high precipitation); (d,e) the same data as in (b) and (c) but with windspeed represented by regularity, pressure by height, and precipitation by density.

The oceanographers are designing models of how they believe salmon feed and move in the open ocean. These simulated salmon will be placed in a set of known environmental conditions, then tracked to see if their behavior mirrors that of the real fish. For example, salmon that migrate back to the Fraser River to spawn chose one of two routes. When the Gulf of Alaska is warm, salmon make landfall at the north end of Vancouver Island and approach the Fraser River primarily via a northern route through the Johnstone Strait (the upper arrow in Fig. 16). When the Gulf of Alaska is cold, salmon are distributed further south, make landfall on the west coast of Vancouver Island, and approach the Fraser River primarily via a southern route through the Juan de Fuca Strait (the lower arrow in Fig. 16). The ability to predict salmon distributions from prevailing environmental conditions would allow the commercial fishing fleet to estimate how many fish will pass through the Johnstone and Juan de Fuca straits. It would also allow more accurate predictions of the size of the salmon run, helping to ensure that an adequate number of salmon arrive at the spawning grounds.

In order to test their hypotheses, the oceanographers have created a database of SSTs and ocean currents for the region 35° north latitude, 180° west longitude to 62° north latitude, 120° west longitude (Fig. 16). Measurements within this region are available at $1^\circ \times 1^\circ$ grid spacings. This array of values exists for each month for the years 1956 to 1964, and 1980 to 1989.

Partial plankton densities have also been collected and tabulated; these are obtained by ships that take readings at various positions in the ocean. We estimated missing plankton values using a set of knowledge discovery (KD) algorithms that we have specifically modified for use during visualization. Our KD algorithms identified month, SST, and current magnitude as the attributes used to estimate missing plankton values. Because of this, we restricted our initial visualization to a monthly time-series of plankton density, SST, and current magnitude.

Figure 16: Map of the North Pacific; arrows represent possible salmon migration paths as they pass through the either Johnstone Strait (upper arrow) or the Strait of Juan de Fuca (lower arrow)

Displaying the three attributes together allows the oceanographers to search for relationships between plankton density, current strength, and SST. Plankton is displayed using color; SST and current strength are displayed using texture. Colors for the five plankton ranges were chosen using our color selection technique [20]. Although other color scales were available (for example, by Ware [78]), our colors are specifically designed to highlight outliers, and to show clearly the boundaries between groups of elements with a common plankton density. We display the five plankton density ranges from low to high using blue (monitor RGB=36, 103, 151), green (monitor RGB=18, 127, 45), brown (monitor RGB=134, 96, 1), red (monitor RGB=243, 51, 55), and purple (monitor RGB=206, 45, 162),

For the underlying texture, we mapped current strength to height and SST to density. Our choices were guided by results we observed from our texture experiments:

- differences in height (specifically, taller elements) may be easier to detect, compared to differences in density or randomness,
- variation in height may mask differences in density or randomness; this appears to be due to the occlusion that occurs when tall pexels in the foreground hide short pexels in the background; this will be less important when users can control their viewpoint into the dataset (our visualization tool allows the user to interactively manipulate the viewpoint), and
- tightly spaced grids can support up to three easily distinguishable density patterns; placing more strips in a single pexel (*e.g.,* arrays of 3×3 or 4×4 strips) will either cause the strips to overlap with their neighbors, or make each strip too

thin to easily identify.

Because there may be a feature preference for height over density, and because current strength was deemed "more important" than SST during knowledge discovery, we used height to represent currents and density to represent SSTs. The five ranges of current strength are mapped to five different heights. We do not use a linear mapping, rather the lower two ranges (corresponding to the weakest currents) are displayed using two types of short pexels, and the upper three ranges (corresponding to the strongest currents) are displayed using three types of tall pexels. This allows a user to rapidly locate boundaries between weak and strong currents, while still being able to identify each of the five ranges. For SSTs, the lower three ranges (corresponding to the coldest SSTs) are displayed with a pexel containing a single strip, while the upper two ranges (corresponding to the warmest SSTs) are displayed with pexels containing arrays of 2×1 and 2×2 strips, respectively. The densities we chose allow a user to see clearly the boundaries between cold and warm temperature regions. If necessary, users can change the range boundaries to focus on different SST gradients.

The oceanographers want to traverse their datasets in monthly and yearly steps. Experiments run in our laboratory have shown that preattentive tasks performed on static frames can be extended to a dynamic environment, where displays are shown one after another in a movie-like fashion [23]. Our visualization tool was designed to allow users to scan rapidly forwards and backwards through the dataset. This makes it easy to compare changes in the value and location of any of the environmental variables being displayed. The oceanographers can track seasonal changes in current strength, SST, and plankton density as they move month by month through a particular year. They can also see how interannual variability affects the environmental conditions and corresponding plankton densities for a particular month across a range of years.

Fig. 17 shows three frames from the oceanography dataset: February 1956, June 1956, and October 1956. Color shows the seasonal variation in plankton densities. Height and density allow the oceanographers to track current strengths and SSTs. In February (Fig. 17a), most plankton densities are less than 28 g/m^3 (*i.e.*, blue and green strips). Currents are low in the north-central Pacific; a region of weak currents also sits off the south coast of Alaska. Most of the ocean is cold (sparse pexels), although a region of higher temperatures can easily be seen as dense pexels in the south. In June (Fig. 17b) dense plankton blooms (red and purple strips) are present across most of the northern Pacific. The positions of the strong currents have shifted (viewing the entire dataset shows this current pattern is relatively stable for the months March to August). Warmer SSTs have pushed north, although the ocean around Alaska and northern British Columbia is still relatively cold. By October the plankton densities have started to decrease (green, brown, and red strips); few high or low density patches are visible. Current strengths have also decreased in the eastern regions. Overall a much larger percentage of the ocean is warm (*i.e.*, dense pexels). This is common, since summer temperatures will sometimes last in parts of the ocean until October or November.

References

[1] AKS, D. J., AND ENNS, J. T. Visual search for size is influenced by a background texture gradient. *Journal of Experimental Psychology: Perception and Performance 22*, 6 (1996), 1467–1481.

[2] BAUER, B., JOLICOEUR, P., AND COWAN, W. B. Visual search for colour targets that are or are not linearly-separable from distractors. *Vision Research 36* (1996), 1439–1446.

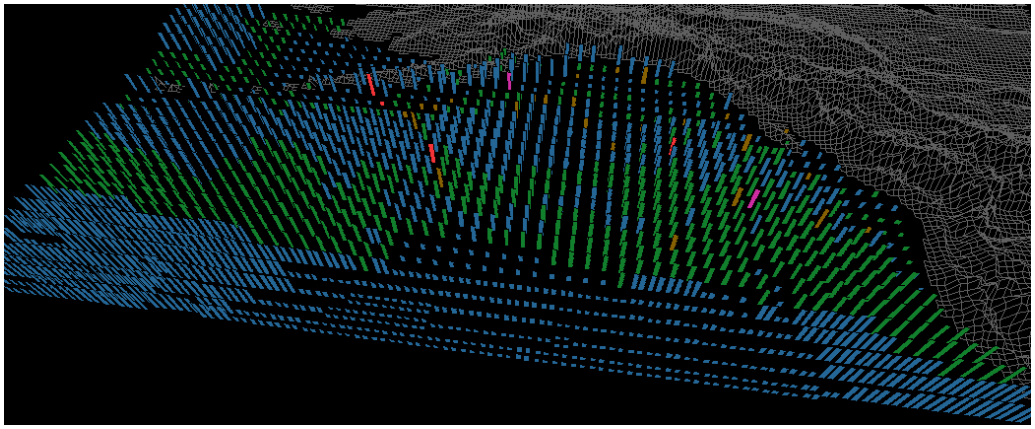
[3] BECK, J., PRAZDNY, K., AND ROSENFELD, A. A theory of textural segmentation. In *Human and Machine Vision*, J. Beck, K. Prazdny, and A. Rosenfeld, Eds. Academic Press, New York, New York, 1983, pp. 1–39.

[4] BERGMAN, L. D., ROGOWITZ, B. E., AND TREINISH, L. A. A rule-based tool for assisting colormap selection. In *Proceedings Visualization '95* (Atlanta, Georgia, 1995), pp. 118–125.

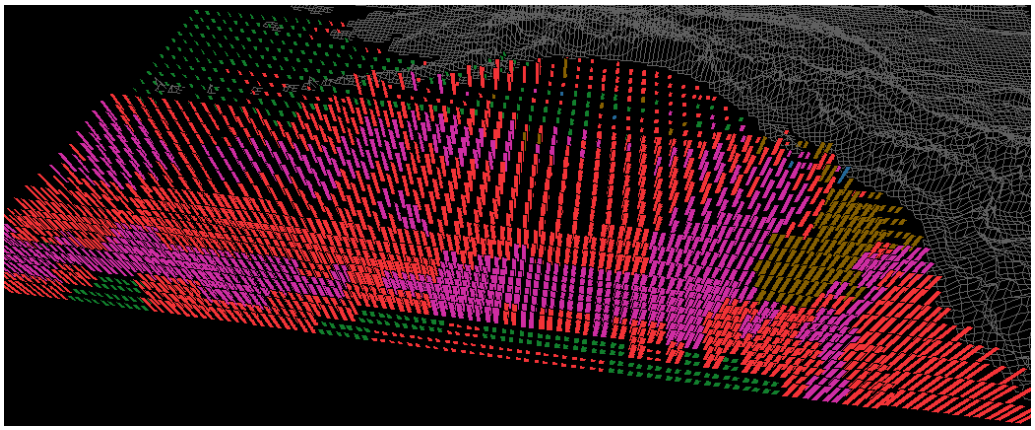
[5] BROWN, R. Impressionist technique: Pissarro's optical mixture. In *Impressionism in Perspective*, B. E. White, Ed. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1978, pp. 114–121.

[6] CALLAGHAN, T. C. Dimensional interaction of hue and brightness in preattentive field segregation. *Perception & Psychophysics 36*, 1 (1984), 25–34.

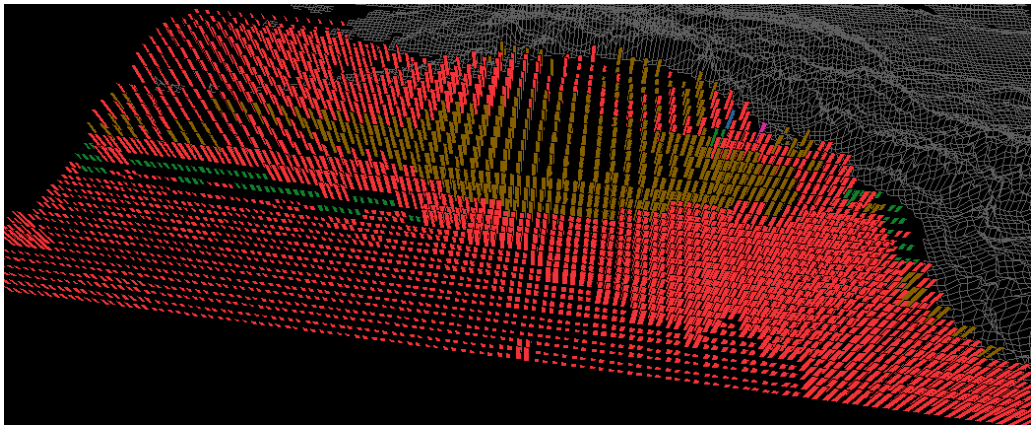
[7] CALLAGHAN, T. C. Interference and domination in texture segregation: Hue, geometric form, and line orientation. *Perception & Psychophysics 46*, 4 (1989), 299–311.



(a)



(b)



(c)

Figure 17: Visualization of the oceanography datasets, color used to represent plankton density (blue, green, brown, red, and purple represent lowest to highest densities), height used to represent current strength, texture density used to represent SST: (a) February, 1956; (b) June, 1956; (c) October, 1956

[8] CALLAGHAN, T. C. Interference and dominance in texture segregation. In *Visual Search*, D. Brogan, Ed. Taylor & Francis, New York, New York, 1990, pp. 81–87.

[9] CHEVREUL, M. E. *The Principles of Harmony and Contrast of Colors and Their Applications to the Arts*. Reinhold Publishing Corporation, New York, New York, 1967.

[10] CURTIS, C. J., ANDERSON, S. E., SEIMS, J. E., FLEISCHER, K. W., AND SALESIN, D. H. Computer-generated watercolor. In *SIGGRAPH 97 Conference Proceedings* (Los Angeles, California, 1997), T. Whitted, Ed., pp. 421–430.

[11] DRIVER, J., MCLEOD, P., AND DIENES, Z. Motion coherence and conjunction search: Implications for guided search theory. *Perception & Psychophysics 51*, 1 (1992), 79–85.

[12] D'ZMURA, M. Color in visual search. *Vision Research 31*, 6 (1991), 951–966.

[13] EBERT, D., AND RHEINGANS, P. Volume illustration: Non-photorealistic rendering of volume models. In *Proceedings Visualization 2000* (San Francisco, California, 2000), pp. 195–202.

[14] EGETH, H. E., AND YANTIS, S. Visual attention: Control, representation, and time course. *Annual Review of Psychology 48* (1997), 269–297.

[15] ENNS, J. T. The promise of finding effective geometric codes. In *Proceedings Visualization '90* (San Francisco, California, 1990), pp. 389–390.

[16] ENNS, J. T., AND RENSINK, R. A. Sensitivity to three-dimensional orientation in visual search. *Psychology Science 1*, 5 (1990), 323–326.

[17] GRINSTEIN, G., PICKETT, R., AND WILLIAMS, M. EXVIS: An exploratory data visualization environment. In *Proceedings Graphics Interface '89* (London, Canada, 1989), pp. 254–261.

[18] HABERLI, P. Paint by numbers: Abstract image representations. *Computer Graphics (SIGGRAPH 90 Conference Proceedings) 24*, 4 (1990), 207–214.

[19] HARALICK, R. M., SHANMUGAM, K., AND DINSTEIN, I. Textural features for image classification. *IEEE Transactions on System, Man, and Cybernetics SMC-3*, 6 (1973), 610–621.

[20] HEALEY, C. G. Choosing effective colours for data visualization. In *Proceedings Visualization '96* (San Francisco, California, 1996), pp. 263–270.

[21] HEALEY, C. G. One the use of perceptual cues and data mining for effective visualization of scientific datasets. In *Proceedings Graphics Interface '98* (Vancouver, Canada, 1998), pp. 177–184.

[22] HEALEY, C. G., BOOTH, K. S., AND ENNS, J. T. Harnessing preattentive processes for multivariate data visualization. In *Proceedings Graphics Interface '93* (Toronto, Canada, 1993), pp. 107–117.

[23] HEALEY, C. G., BOOTH, K. S., AND ENNS, J. T. Real-time multivariate data visualization using preattentive processing. *ACM Transactions on Modeling and Computer Simulation 5*, 3 (1995), 190–221.

[24] HEALEY, C. G., BOOTH, K. S., AND ENNS, J. T. High-speed visual estimation using preattentive processing. *ACM Transactions on Computer-Human Interaction 3*, 2 (1996), 107–135.

[25] HEALEY, C. G., AND ENNS, J. T. Building perceptual textures to visualize multidimensional datasets. In *Proceedings Visualization '98* (Research Triangle Park, North Carolina, 1998), pp. 111–118.

[26] HEALEY, C. G., AND ENNS, J. T. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE Transactions on Visualization and Computer Graphics 5*, 2 (1999), 145–167.

[27] HERTZMANN, A. Painterly rendering with curved brush strokes of multiple sizes. In *SIGGRAPH 98 Conference Proceedings* (Orlando, Florida, 1998), M. Cohen, Ed., pp. 453–460.

[28] HSU, S. C., AND LEE, I. H. H. Drawing and animation using skeltal strokes. In *SIGGRAPH 94 Conference Proceedings* (Orlando, Florida, 1994), A. Glassner, Ed., pp. 109–118.

[29] INTERRANTE, V. Illustrating surface shape in volume data via principle direction-driven 3D line integral convolution. In *SIGGRAPH 97 Conference Proceedings* (Los Angeles, California, 1997), T. Whitted, Ed., pp. 109–116.

[30] INTERRANTE, V. Harnessing natural textures for multivariate visualization. *IEEE Computer Graphics & Applications 20*, 6 (2000), 6–11.

[31] JULÉSZ, B. *Foundations of Cyclopean Perception*. University of Chicago Press, Chicago, Illinois, 1971.

[32] JULÉSZ, B. A theory of preattentive texture discrimination based on first-order statistics of textons. *Biological Cybernetics 41* (1981), 131–138.

[33] JULÉSZ, B. A brief outline of the texton theory of human vision. *Trends in Neuroscience 7*, 2 (1984), 41–45.

[34] JULÉSZ, B., AND BERGEN, J. R. Textons, the fundamental elements in preattentive vision and perception of textures. *The Bell System Technical Journal 62*, 6 (1983), 1619–1645.

[35] KAWAI, M., UCHIKAWA, K., AND UJIKE, H. Influence of color category on visual search. In *Annual Meeting of the Association for Research in Vision and Ophthalmology* (Fort Lauderdale, Florida, 1995), p. #2991.

[36] LAIDLAW, D. H. Loose, artistic "textures" for visualization. *IEEE Computer Graphics & Applications 21*, 2 (2001), 6–9.

[37] LAIDLAW, D. H., AHRENS, E. T., KREMERS, D., AVALOS, M. J., JACOBS, R. E., AND READHEAD, C. Visualizing diffusion tensor images of the mouse spinal cord. In *Proceedings Visualization '98* (Research Triangle Park, North Carolina, 1998), pp. 127–134.

[38] LEVKOWITZ, H., AND HERMAN, G. T. Color scales for image data. *IEEE Computer Graphics & Applications 12*, 1 (1992), 72–80.

[39] LITWINOWICZ, P. Processing images and video for an impressionist effect. In *SIGGRAPH 97 Conference Proceedings* (Los Angeles, California, 1997), T. Whitted, Ed., pp. 407–414.

[40] LIU, F., AND PICARD, R. W. Periodicity, directionality, and randomness: Wold features for perceptual pattern recognition. In *Proceedings 12th International Conference on Pattern Recognition* (Jerusalem, Israel, 1994), pp. 1–5.

[41] MACK, A., AND ROCK, I. *Inattentional Blindness*. MIT Press, Menlo Park, California, 1998.

[42] MALIK, J., AND PERONA, P. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A 7*, 5 (1990), 923–932.

[43] MCCORMICK, B. H., DEFANTI, T. A., AND BROWN, M. D. Visualization in scientific computing—a synopsis. *IEEE Computer Graphics & Applications 7*, 7 (1987), 61–70.

[44] MEIER, B. J. Painterly rendering for animation. In *SIGGRAPH 96 Conference Proceedings* (New Orleans, Louisiana, 1996), H. Rushmeier, Ed., pp. 477–484.

[45] NAGY, A. L., AND SANCHEZ, R. R. Critical color differences determined with a visual search task. *Journal of the Optical Society of America A 7*, 7 (1990), 1209–1217.

[46] NAGY, A. L., SANCHEZ, R. R., AND HUGHES, T. C. Visual search for color differences with foveal and peripheral vision. *Journal of the Optical Society of America A 7* (1990), 1995–2001.

[47] NAKAYAMA, K., AND SILVERMAN, G. H. Serial and parallel processing of visual feature conjunctions. *Nature 320* (1986), 264–265.

[48] QUINLAN, P. T., AND HUMPHREYS, G. W. Visual search for targets defined by combinations of color, shape, and size: An examination of task constraints on feature and conjunction searches. *Perception & Psychophysics 41*, 5 (1987), 455–472.

[49] RAO, A. R., AND LOHSE, G. L. Identifying high level features of texture perception. *CVGIP: Graphical Models and Image Processing 55*, 3 (1993), 218–233.

[50] RAO, A. R., AND LOHSE, G. L. Towards a texture naming system: Identifying relevant dimensions of texture. In *Proceedings Visualization '93* (San Jose, California, 1993), pp. 220–227.

[51] RENSINK, R. A., O'REGAN, J. K., AND CLARK, J. J. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science 8* (1997), 368–373.

[52] RHEINGANS, P., AND TEBBS, B. A tool for dynamic explorations of color mappings. *Computer Graphics 24*, 2 (1990), 145–146.

[53] ROBERTSON, P. K. Visualizing color gamuts: A user interface for the effective use of perceptual color spaces in data displays. *IEEE Computer Graphics & Applications 8*, 5 (1988), 50–64.

[54] ROGOWITZ, B. E., AND TREINISH, L. A. An architecture for rule-based visualization. In *Proceedings Visualization '93* (San Jose, California, 1993), pp. 236–243.

[55] ROOD, O. N. *Modern Chromatics, with Applications to Art and Industry*. Appleton, New York, New York, 1879.

[56] ROSENBLUM, L. J. Research issues in scientific visualization. *IEEE Computer Graphics & Applications 14*, 2 (1994), 61–85.

[57] SALISBURY, M., WONG, M. T., HUGHES, J. F., AND SALESIN, D. H. Orientable textures for image-based pen-and-ink illustration. In *SIGGRAPH 97 Conference Proceedings* (Los Angeles, California, 1997), T. Whitted, Ed., pp. 401–406.

[58] SCHAPIRO, M. *Impressionism: Reflections and Perceptions.* George Brazillier, Inc., New York, New York, 1997.

[59] SCHWEITZER, D. Artificial texturing: An aid to surface visualization. *Computer Graphics (SIGGRAPH 83 Conference Proceedings) 17*, 3 (1983), 23–29.

[60] SHIRAISHI, M., AND YAMAGUCHI, Y. Image moment-based stroke placement. In *SIGGRAPH 99 Sketches & Applications* (Los Angeles, California, 1999), R. Kidd, Ed., p. 247.

[61] SILBERSHATZ, A., STONEBRAKER, M., AND ULLMAN, J. D. The "Lagunita" report of the NSF invitational workshop on the future of database systems research. Tech. Rep. TR-90-22, Department of Computer Science, University of Texas at Austin, 1990.

[62] SIMON, D. J., AND LEVIN, D. T. Change blindness. *Trends in Cognitive Science 1* (1997), 261–267.

[63] SMITH, P. H., AND VAN ROSENDALE, J. Data and visualization corridors report on the 1998 CVD workshop series (sponsored by DOE and NSF). Tech. Rep. CACR-164, Center for Advanced Computing Research, California Institute of Technology, 1998.

[64] SNOWDEN, R. J. Texture segregation and visual search: A comparison of the effects of random variations along irrelevant dimensions. *Journal of Experimental Psychology: Human Perception and Performance 24*, 5 (1998), 1354–1367.

[65] STRASSMANN, S. Hairy brushes. *Computer Graphics (SIGGRAPH 86 Proceedings) 20*, 4 (1986), 185–194.

[66] TAM, R., HEALEY, C. G., AND FLAK, B. Volume visualization of abdominal aortic aneurysms. In *Proceedings Visualization '97* (Phoenix, Arizona, 1997), pp. 43–50.

[67] TAMURA, H., MORI, S., AND YAMAWAKI, T. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics SMC-8*, 6 (1978), 460–473.

[68] THOMSON, K. A., INGRAHAM, W. J., HEALEY, M. C., LEBLOND, P. H., GROOT, C., AND HEALEY, C. G. Computer simulations of the influence of ocean currents on Fraser River sockeye salmon (*oncorhynchus nerka*) return times. *Canadian Journal of Fisheries and Aquatic Sciences 51*, 2 (1994), 441–449.

[69] TRICK, L., AND PYLYSHYN, Z. Why are small and large numbers enumerated differently? A limited capacity preattentive stage in vision. *Psychology Review 101* (1994), 80–102.

[70] TRIESMAN, A. Preattentive processing in vision. *Computer Vision, Graphics and Image Processing 31* (1985), 156–177.

[71] TRIESMAN, A. Search, similarity, and integration of features between and within dimensions. *Journal of Experimental Psychology: Human Perception & Performance 17*, 3 (1991), 652–676.

[72] TRIESMAN, A., AND GELADE, G. A feature-integration theory of attention. *Cognitive Psychology 12* (1980), 97–136.

[73] TRIESMAN, A., AND GORMICAN, S. Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review 95*, 1 (1988), 15–48.

[74] TRIESMAN, A., AND SOUTHER, J. Illusory words: The roles of attention and top-down constraints in conjoining letters to form words. *Journal of Experimental Psychology: Human Perception & Performance 14* (1986), 107–141.

[75] TURK, G., AND BANKS, D. Image-guided streamline placement. In *SIGGRAPH 96 Conference Proceedings* (New Orleans, Louisiana, 1996), H. Rushmeier, Ed., pp. 453–460.

[76] VAREY, C. A., MELLERS, B. A., AND BIRNBAUM, M. H. Judgments of proportions. *Journal of Experimental Psychology: Human Perception & Performance 16*, 3 (1990), 613–625.

[77] VENTURI, L. Impressionist style. In *Impressionism in Perspective*, B. E. White, Ed. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1978, pp. 105–113.

[78] WARE, C. Color sequences for univariate maps: Theory, experiments, and principles. *IEEE Computer Graphics & Applications 8*, 5 (1988), 41–49.

[79] WARE, C., AND BEATTY, J. C. Using colour dimensions to display data dimensions. *Human Factors 30*, 2 (1988), 127–142.

[80] WARE, C., AND KNIGHT, W. Orderable dimensions of visual texture for data display: Orientation, size, and contrast. In *Proceedings SIGCHI '92* (Monterey, California, 1992), pp. 203–209.

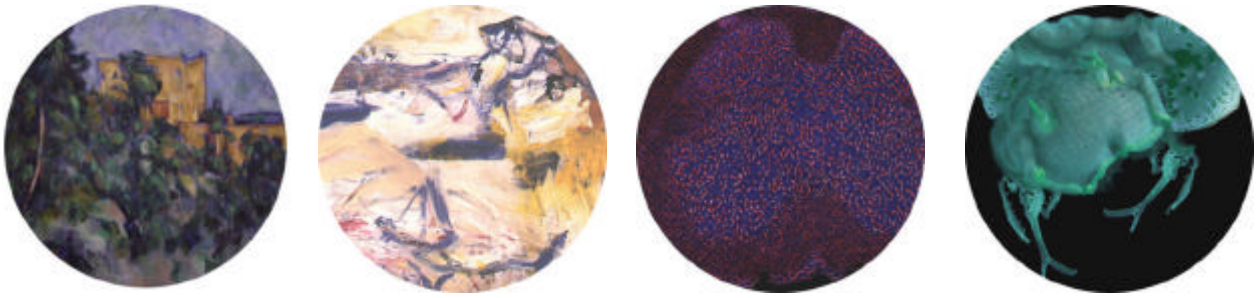
[81] WARE, C., AND KNIGHT, W. Using visual texture for information display. *ACM Transactions on Graphics 14*, 1 (1995), 3–20.

[82] WINKENBACH, G., AND SALESIN, D. H. Rendering free-form surfaces in pen-and-ink. In *SIGGRAPH 96 Conference Proceedings* (New Orleans, Louisiana, 1996), H. Rushmeier, Ed., pp. 469–476.

[83] WOLFE, J. M. Guided Search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review 1*, 2 (1994), 202–238.

[84] WOLFE, J. M., FRIEDMAN-HILL, S. R., STEWART, M. I., AND O'CONNELL, K. M. The role of categorization in visual search for orientation. *Journal of Experimental Psychology: Human Perception & Performance 18*, 1 (1992), 34–49.

[85] WYSZECKI, G., AND STILES, W. S. *Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd Edition*. John Wiley & Sons, Inc., New York, New York, 1982.



some thoughts on pictures in more dimensions

davidkremers, caltech distinguished conceptual artist in biology

...cezanne said that every brushstroke has its own perspective. he didn't mean it in the sense of renaissance perspective, but every brushstroke has its own point of view. de kooning

traditional modeling of 3d objects in space also contains an invisible 4th, or time, dimension as a feedback circuit completed by the reaction of the viewer. each time an object is looked at, even by the same viewer, the viewed response is changed due to the viewer's intervening experience/position. thus the contemplation of any object is both refined and expanded at the same time.

the line drawn where painting crosses over into sculpture is called bas-relief. at this point the density of the picture plane attains a weight which the picture plane cannot visually support, and in a process converse to a rocket reaching escape velocity, the object is launched into orbit only to find itself a captive of gravity. if the visual density continues to climb, the gravity of the image solidifies into sculpture. sculpture, the tradition of representing three dimensional objects in spacetime, has relied on either freezing time, or abstracting form as a means of representing fluidity.

our task then, is to explore the surface fields suggested by cezanne in such a fashion that we can learn to defy gravity. in order to depict multiple dimensions of data we must retain the fluidity of the visual imagery. this will require the discovery of a picture plane that can support density minus visual weight. we must learn to serve water without a glass.

Wind over Wheatfield, van Gogh

Chateau Noir, Cezanne

paraxial mesoderm, davidkremers

DeKooning details

Exhibits and Bibliography

1. ’microMacro : theexpandinglandscape’ de verbeelding art landscape nature, zeewolde 2001
2. ’paradise now’ exit art, new york 2000
3. ’art meets science’ royal academy of fine arts, charlottenborg palace, copenhagen 1996
4. davidkremers, ’the delbruck paradox.’ genetische kunst - kunstlisches leben [exh. cat] pvs verleger, wien, 1993.
5. warn, dana. ’unconventional collaborations.’ abcnews.com, august 21, 2000.
6. robbins, david. ’the rise of systems man.’ art issues, no. 38 [summer 1995] : 28-31.

Visualization Viewpoints

Editors: Theresa-Marie Rhyne and Lloyd Treinish

Loose, Artistic "Textures" for Visualization

David H. Laidlaw

Brown University

In the November/December 2000 issue of *IEEE Computer Graphics and Applications*,[1] Vicki Interrante posed a visualization problem she and I have been interested in for several years. The problem is that of visually representing a 2D field of data that has multiple data values at each point. For example, 2D fluid flow has a vector value at each location and derived values are often available at each location. Interrante suggests using natural textures to attack this problem, because the textures can potentially encode lots of information. She provides some intriguing examples and proposes a psychology-based approach for developing an understanding of how we perceive natural textures, like those Brodach photographed.[2] Understanding this helps us build better visualizations.

Based on Interrante's suggestions, I would like to posit and explore what is, perhaps, a less well-defined approach. Through evolution, the human visual system has developed the ability to process natural textures. However, in addition to natural textures, humans also visually process man-made textures—some of the richest and most compelling of which are in works of art. Art goes beyond what perceptual psychologists understand about visual perception and there remain fundamental lessons that we can learn from art and art history and apply to our visualization problems.

The rest of this article describes and illustrates some of the visualization lessons we have learned studying art. I believe that these examples also illustrate some of the potential benefits of further study. While this approach is more open-ended than a perceptual psychology approach, both approaches are worthy of pursuit, and the potential benefits of using the less structured approach outweigh any risk of failure.

How humans see and understand

Scientific visualization, a term coined only a little over 10 years ago, is the process of using the human visual system to increase our understanding of phenomena studied in various scientific disciplines. While the term is young, the process (modulo the computer) has been used since the beginning of science. Many scientists have created drawings or built 3D models to understand and communicate their science. The history of science and art can provide us with lessons for using computers effectively. Over time, artists have developed techniques to create visual representations in particular communication goals. Art history provides a language for understanding and communicating that knowledge.

Historically, two disciplines approach the human visual system from different perspectives. Art history provides a phenomenological view of art—painting X evokes response Y. Art history, however, doesn't deconstruct the perceptual and cognitive processes underlying responses. Perceptual psychology, on the other hand, strives to explain how humans understand those visual representations. There's a gap between art and perceptual psychology—we don't know how humans combine visual inputs to arrive at the responses art evokes. Shape, shading, edges, color, texture, motion, and interaction are all components of an interactive visualization. But how do these components interact and how can they most effectively be deployed for a particular scientific task? Answers to these questions are likely to fill some of the gap between art and perceptual psychology. As an example, the human-computer interaction (HCI) community is using and extending knowledge about perception to test and develop better user interfaces. We can find analogous inspiration for improved methods for scientific visualization in the gap between art and perceptual psychology. Many of these lessons will impact the visual representation of multivalued data.

Looking up from our monitors

A number of times over the last few years I've shepherded my students to art museums for guided tours by my artist collaborator davidkremers, the Caltech Distinguished Conceptual Artist in Biology. After initially searching for scientific visualization inspiration in art, these visits let us formulate a plan for finding and applying the concepts. Our initial focus was on oil painting, particularly from the Impressionist period, because these paintings are so visually rich. The multiple layers of brush strokes in these paintings provide a natural metaphor for constructing visualizations from layers of synthetic "brush strokes." Some of my colleagues look at me askance when I describe our research field trips, as if to say, "This is research?" But stepping out of the lab helps students build a new picture of what they can accomplish when they come back to the computer. It trains their eyes and minds to see differently.

During these field trips, we studied, in particular, the works of three painters:

- Van Gogh, whose large, expressive, discrete strokes carry meaning both individually and collectively.

(a) (b)

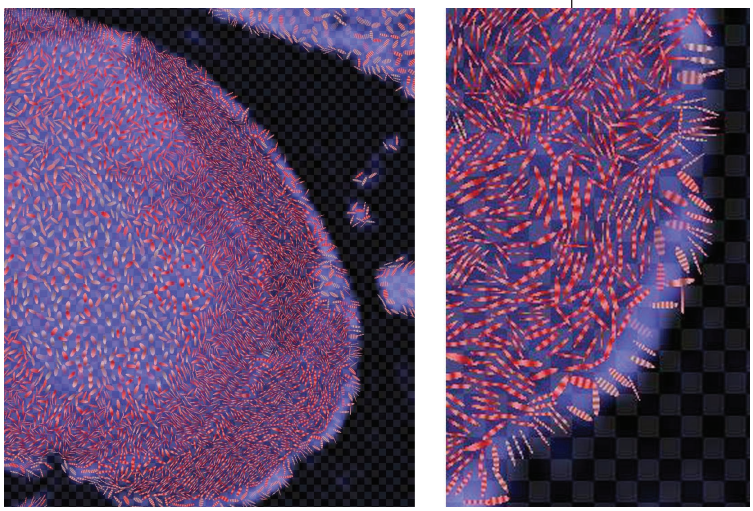
1 (a) Van Gogh's *The Mulberry Tree* (1889, oil on canvas) illustrates the visual shorthand that van Gogh used with his expressive strokes. Multiple layers of strokes combine to define regions of different ground cover, aspects of the hillside, and features of the tree. (b) An underpainting shows the "anatomy" or composition of the scene in broad strokes. (Image of *The Mulberry Tree* granted by the Norton Simon Foundation, Pasadena, California. Gift of Norton Simon, 1976.)

- Monet, whose smaller strokes are often meaningless in isolation—the relationships among the strokes give them meaning, far more than in van Gogh.
- Cezanne, who combined strokes into cubist facets, playing with 3D perspective and time within his paintings more then either van Gogh or Monat. His layering also incorporates more atmospheric effects. In a sense, his work shifts from surface rendering toward volume rendering.

The three artists' work in this sequence builds in complexity and subtlety. In our field trips, we studied all three, but most of our experiments thus far are limited to ideas we learned from van Gogh's work.

Van Gogh introduced us to the concept of underpainting, or laying down a rough value sketch of the entire painting. The underpainting shows through the overlying detailed brush strokes to define the anatomy of the painting. Figure 1b shows underpainting for Figure 1a. It divides the canvas into two parts—a primed lower region of hillside, rocks, and ground cover and a darker upper region of tree, sky, and distant hills. Underpainting helped us present some overall parts of our data. We found that an analogous underlying form in our visualizations anchors and literally gives shape to disparate data components. Outlines around regions provide separation and emphasis, lending definition to our sea of data.

In van Gogh's *The Mulberry Tree* (1889, oil on canvas), brush strokes represent the solid trunk of the tree, bending branches, leaves blowing in the wind, and tufts of grass (Figure 1a). We learned many shorthand ways of depicting complexity using icons, geometric shapes, or textures that evoke a characteristic of the subject, or the data—and with that comes the responsibility of choosing brush strokes that don't create opposing or unwanted secondary impressions. Beyond this direct representation, they also invite the viewer to experience the scene, not just view it passively. Similarly, brush-stroke size and proximity depict density, weight, and velocity. In our visualizations, we want to capture this marriage between direct representation of independent data and the overall intuitive feeling of the data as a whole.

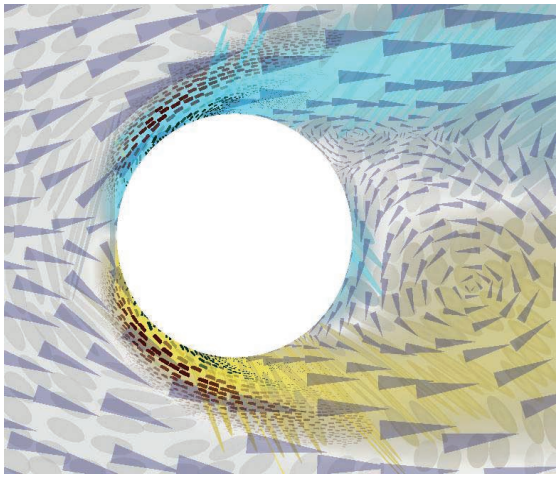


2 Visualization of half of a section through a mouse spinal cord. The data is a symmetric 3D second-order tensor field, with the equivalent of six independent scalar values at each point. The detail on the right shows the lower right part of the section.

Back in the lab

Returning to our computer lab, we tried to use some of the ideas we had gleaned, once again drawing mostly from van Gogh's work. We experimented with brush stroke shapes and ways of layering them. Our initial attempts were free–form and produced interesting results. Our next attempts were more directly applied to scientific problems. We show two of the images we generated in Figures 2 and 3 (next page). The problem-directed approach led us to iconic-looking strokes.

In Figure 2, we show one 2D slice of a 3D second-



3 Visualization of 2D fluid velocity together with several derived data values. Approximately nine values are represented visually at each point in the image.

4 Variances in viewing van Gogh's *Mulberry Tree*. Viewed in this article from about 18 inches, Figure 1a shows what you would see 15 feet from the painting. Comparatively, Figure 4 shows the following: (a) a detail of what you would see 5 feet from the painting, and (b) a detail at actual size (what you would see from 18 inches). Look at (b) more closely for viewing distances less than 18 inches.

order tensor field, which has about six different data values at each point in the image. The image shows the right half of a section through a mouse spinal cord. To create the visualization, we used a layer resembling underpainting with ellipse-shaped strokes on top of it. On each of the strokes, a texture represents more of the data. For more details on the scientific interpretation and the visualization, see Laidlaw et al.[3]

In Figure 3, we show 2D fluid velocity together with a number of derived quantities. About nine values are represented at each spatial location in this visualization. We again used a layer resembling underpainting with layers of ellipse, wedge, and box strokes on top. The ellipse strokes have a subtle texture superimposed. More details on the visualization appear in Kirby et al.[4]

Space

We learned that paintings (and, in some cases, visualizations) are multiscale. They can be viewed from different distances and seen and understood differently. This raises interesting issues about the definition of texture. Let's consider van Gogh's *Mulberry Tree* (Figure 1a). From a few inches away (look closely at Figure 4b), you can see shapes from the bristles of the brush as well as colors mixed within a stroke. At this distance, the shape and color features might be considered texture, but they could also be interpreted individually. At a distance of 18 inches (Figure 4b at a normal reading distance of 18 inches), these features appear smaller and resemble a texture on each stroke. The strokes themselves are still individual. At a distance of five feet (Figure 4a), the strokes merge together to appear more like a texture. Finally, at 15 feet (Figure 1a), the strokes blend together and become almost invisible individually.

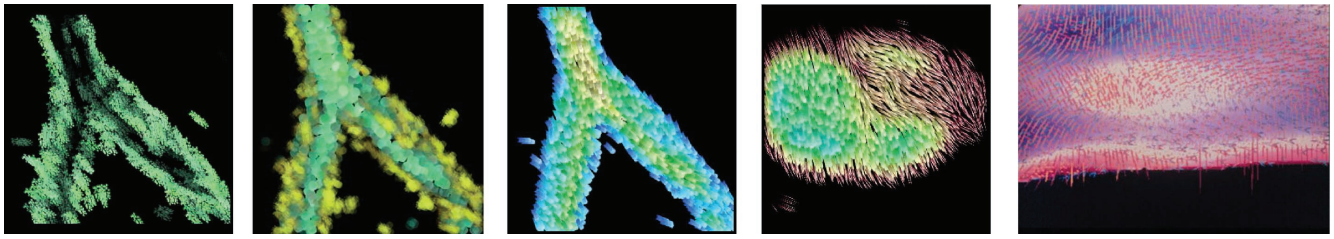
We can use this lesson by encoding different information at different scales. Iconic information at one scale can turn into texture information at another scale. With care, we can design features at different scales into the same images. In the scientific visualizations of Figures 2 and 3, we design visual features at different scales. The texture on strokes is at a much finer scale than the strokes themselves, and the dark box strokes of Figure 3 are at a different scale than the other strokes.

To take full advantage of the multiscale nature of paintings and visualizations we have to have ways of interacting with them—that is, ways of changing our viewing distance. We rarely change the distance from which we view our monitors—only a bit more frequently do we do so with paper publications. That's why the same image is shown at different scales in Figure 4 and in some of the other detailed figures. However, we do view images hung on a wall from different distances. And some images on paper—often artistic—inspire that sort of study. Projection systems like PowerWalls and CAVEs may be good options for encouraging this sort of exploration, as may other hangable large-format output media.

Time

We also learned that paintings (and, in some cases, visualizations) have a temporal component. For instance, we see different aspects of an image at different viewing times. Some parts stand out quickly, like the overall composition or palette of a painting, and some take more time to become apparent, like the texture or shape of individual strokes. The scale and speed of recognition correlate, as do contrast and speed of recognition—but these are not the only factors.

To use this lesson, we can design our visualizations so that important data features are mapped to quickly seen visual features. For example, features we want to measure directly from an image are present for detailed study but don't intrude on the visualization's initial impression. The multiscale examples from the "Space" section illustrate this temporal concept. Figure 3 gives another example: we can read the wedges more quickly than the



5 Loose texture examples.

ellipses because of a difference in contrast.

Studies of preattentive vision and knowledge about low-level vision are useful for designing quickly seen visualization parts. It's more difficult to test the more slowly seen parts, which makes it more difficult to design them. Task-oriented experimental tests seem logical, but the tasks are often so complex that the performance variance is relatively high, making methods difficult to compare.

Our initial experiments

Our initial experiments were much looser than the examples shown in Figures 2 and 3. Some examples in Figure 5 show 2D or 3D fluid flow. Since I want to emphasize the overall texture and visual qualities, I won't go into detail about the mappings for each. To many, the images are visually compelling, yet it has been difficult to extract concrete visualization lessons from them beyond those I previously described. What people see in these images includes not only the mappings that were used for the data value, but also other visual characteristics. Despite being 2D, some images give an overall sense of depth. Some of the strokes appear to layer, like feathers or scales. One of our challenges with these looser images is in understanding what works, what doesn't, and (we hope) why.

Closing thoughts

I've tried to illustrate some examples of looking toward art for inspiration in creating visualizations. Here we feature van Gogh and mention Monet and Cezanne for context. In your artistic searches, choose the artists in whom you have a passionate interest. I believe that any artist has lessons to offer to visualization.

Working on scientific visualization problems, we already interact with scientists and adopt their problems. As toolsmiths, we do better computer science through addressing scientists' problems on scientists' terms.[5] Similarly, we benefit from critical feedback from artists, despite the difficulty of creating and maintaining these relationships. I try to look at and understand art—early and often—and emulate it in scientific visualizations and get critical feedback from artists. I explain what I'm trying to do visually and have artists critique it. Then I iterate, iterate, and iterate.

Of course, scientists must be involved in this iterative process. Artists can help with inspiration and feedback on the visual and communicative aspects of visualization, but scientists define the tasks performed and therefore must ultimately evaluate the success of the methods. For instance, the fluid flow example in Figure 3 may be aesthetically pleasing, but without explanation—perhaps via a legend or key—it's not scientifically useful.

Figure 3 displays as many as nine values at each point of the image. With some research indicating that texture has roughly three independent dimensions, the ability to represent nine values is somewhat surprising—perhaps it's due to combining color with texture or layering textures at different scales.

Texture is hard to define. Understanding black and white natural textures like the photographs in Brodatz[2] is a good start, but we also need to look broadly. Task-oriented user testing may help, and perhaps we can use the critiques that are part of artists' training. This might combine perceptual psychology and art to fill in part of the gap in our understanding of how humans see. By having artists cognitively analyze what is shown by more complex textures, we might come to a consensus on what works, what doesn't work, and why it does or doesn't work in the context of art and art history. ■

Acknowledgments

I'd like to thank my collaborators—in particular, artist davidkremers—as well as Clifford Elgin, Mike Kirby, and Matthew Avalos. I also want to thank Victoria Interrante and Barbara Meier for their encouragement and ideas. The NSF (CCR-9619649, CCR-0086065) and the Human Brain Project (NIMH and NIDA) supported some of the work described.

References

1. V. Interrante, "Harnessing Natural Textures for Multivariate Visualization," *IEEE Computer Graphics and Applications*, vol. 20, no. 6, Nov./Dec. 2000, pp. 6-10.
2. P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover, New York, 1966.
3. D.H. Laidlaw et al., "Visualizing Diffusion Tensor Images of the Mouse Spinal Cord," *Proc. Visualization 98*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1998, pp. 127-134.
4. R.M. Kirby, H. Marmanis, and D.H. Laidlaw, "Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting," *Proc. Visualization 99*, IEEE Computer Soc. Press, Los Alamitos, Calif., 1999, pp. 333-340.
5. F.P. Brooks, "The Computer Scientist as Toolsmith II," *Comm. ACM*, vol. 39, no. 3, Mar. 1996, pp. 61-68.

Readers may contact Laidlaw at the Department of Computer Science, P.O. Box 1910, Brown University, Providence, RI 02912, email dhl@cs.brown.edu.

Readers may contact department editors Rhyne and Treinish by email at rhyne@siggraph.org and lloydt@us.ibm.com.

Immersive VR for Scientific Visualization: A Progress Report

Andries van Dam, Andrew S. Forsberg, David H. Laidlaw, Joseph J. LaViola, Jr., and Rosemary M. Simpson
Brown University

Immersive virtual reality (IVR) has the potential to be a powerful tool for the visualization of burgeoning scientific data sets and models. In this article we sketch a research agenda for the hardware and software technology underlying IVR for scientific visualization. In contrast to Brooks' excellent survey last year,[1] which reported on the state of IVR and provided concrete examples of its production use, this article is somewhat speculative. We don't present solutions but rather a progress report, a hope, and a call to action, to help scientists cope with a major crisis that threatens to impede their progress.

Immersive virtual reality can provide powerful techniques for scientific visualization. The research agenda for the technology sketched here offers a progress report, a hope, and a call to action.

Brooks' examples show that the technology has only recently started to mature—in his words, it "barely works." IVR is used for walkthroughs of buildings and other structures, virtual prototyping (vehicles such as cars, tractors, and airplanes), medical applications (surgical visualization, planning, and training), "experiences" applied as clinical therapy (reliving Vietnam experiences to treat post-traumatic stress disorder, treating agoraphobia), and entertainment. Building on Brooks' work, here we concentrate on why scientific visualization is also a good application area for IVR.

First we'll briefly review scientific visualization as a means of understanding models and data, then discuss the problem of exploding data set size, both from sensors and from simulation runs, and the consequent demand for new approaches. We see IVR as part of the solution: as a richer visualization and interaction environment, it can potentially enhance the scientist's ability to manipulate the levels of abstraction necessary for multi-terabyte and petabyte data sets and to formulate hypotheses to guide very long simulation runs. In short, IVR has the potential to facilitate a more balanced human-computer partnership that maximizes bandwidth to the brain by more fully engaging the human sensorium.

We argue that IVR remains in a primitive state of development and is, in the case of CAVEs and tiled projection displays, very expensive and therefore not in routine use. (We use the term cave to denote both the original CAVE developed at the University of Illinois' Electronic Visualization Laboratory[2] and CAVE-style derivatives.) Evolving hardware and software technology may, however, enable IVR to become as ubiquitous as 3D graphics workstations—once exotic and very expensive—are today.

Finally, we describe a research agenda, first for the technologies that enable IVR and then for the use of IVR for scientific visualization. Punctuating the discussion are sidebars giving examples of scientific IVR work currently under way at Brown University that addresses some of the research challenges, as well as other sidebars on data set size growth and IVR interaction metaphors.

What is IVR?

By immersive virtual reality we mean technology that gives the user the psychophysical experience of being surrounded by a virtual, that is, computer-generated, environment. This experience is elicited with a combination of hardware, software, and interaction devices. Immersion is typically produced by a stereo 3D visual display, which uses head tracking to create a human-centric rather than a computer-determined point of view. Two common forms of IVR use head-mounted displays (HMDs), which have small display screens in front of the user's eyes, and caves, which are specially constructed rooms with projections on multiple walls and possibly floor and/or ceiling. Forms of IVR differ along a number of dimensions, such as user mobility and field of view, which we discuss briefly when talking about the tradeoffs that exist in IVR technology for scientific visualization.

Closely related to the sensation of immersion is the sensation of presence—usually loosely described as the feeling of "being there"—which gives a sense of the reality of objects in a scene and the user's presence with those objects. Immersion and presence are enhanced by

0272-1716/00/$10.00 © 2000 IEEE

a wider field of view than is available on a desktop display, and leverage peripheral vision when working with 3D information. This helps provide situational awareness and context, aids spatial judgments, and enhances navigation and locomotion. The presentation may be further enhanced by aural rendering of spatial 3D sound and by haptic (touch, force) rendering to create representations of geometries and surface material properties.

Interaction is provided through a variety of spatial input devices, most providing at least six degrees of freedom (DOF) based on tracker technology. Such devices include 3D mice, various kinds of wands with buttons for pointing and selecting, data gloves that sense joint angles, and pinch gloves that sense contacts. Both types of gloves provide position and gesture recognition. Additional sensory modalities may be engaged with speech recognizers and haptic input and feedback.

IVR aims to create a rich, highly responsive environment, one that engages as many of our senses as possible. Realism—mimicking the physical world as faithfully as possible—is often a goal, but for experiencing many environments, it need not be. We can view IVR as the technology that currently lies at an extreme on a spectrum of display technologies and corresponding interaction technologies. This spectrum starts with keyboard-driven, text-only displays and proceeds through 2D graphics with keyboard and mouse to 3D desktop graphics with 3D interaction devices to IVR. Thus, IVR can be seen as a natural extension of existing computing environments. As we argue later, however, it's more appropriately seen as a substantially new medium that differs more from conventional desktop 3D environments than those environments differ from 2D desktop environments. Conventional desktop 3D displays give one the sensation of looking through a window into a miniature world on the other side of the screen, with all the separation that sensation implies, whereas IVR makes it possible to become immersed in and interact with life-sized scenes.

Once mastered, post-WIMP (that is, post-windows, -icons, -menus, -pointing) multimodal interaction,[3] such as simultaneous speech and hand input, provides a far richer, potentially more natural way of interacting with a synthetic environment than do mouse and keyboard. Fish Tank VR on a monitor,[4] workbenches,[5] and single-wall projection displays,[6] all with head-tracked stereo, provide semi-immersive VR environments—between desktop 3D and fully immersive VR.

IVR for scientific visualization

We believe that IVR is a rich way of interacting with virtual environments (VEs). It holds great promise for scientists, mathematicians, and engineers who rely on scientific visualization to grapple with increasingly complex problems that produce correspondingly larger and more complex models and data sets. These data sets often describe complicated 3D structures or can be visualized with derived 3D abstractions (such as isosurfaces) possessing complicated geometry. We contend that people can more readily explore and understand these complex structures with the kinesthetic feedback gained by peering around at them from within, walking around them to see them from different aspects, or handling them.

Scientific visualization isn't an end in itself, but a component of many scientific tasks that typically involve some combination of interpretation and manipulation of scientific data and/or models. To aid understanding, the scientist visualizes the data to look for patterns, features, relationships, anomalies, and the like. Visualization should be thought of as task driven rather than data driven.

Indeed, it's useful to think of simulation and visualization as an extension of the centuries-old scientific method of formulating a hypothesis, then performing experiments to validate or reject it. Scientists now use simulation and visualization as an alternate means of observation, creating hypotheses and testing the results of simulation runs against data from physical experiments. Large simulation runs may use visualization as a completely separate postprocess or may interlace visualization and parameter setting with re-running the simulation, in a mode called interactive steering,[7] in which the user monitors and influences the computation's progress.

Unfortunately, our ability to simulate or use increasingly numerous and refined sensors to produce ever larger data sets outstrips our ability to understand the data, and there's compelling evidence that the gap is widening. Hence, we look for ways to make human-in-the-loop visualization more powerful. IVR has begun to serve as one such power tool.

We use the term scientific visualization and chose examples primarily from science and technology, but much of the discussion would apply equally well to closely related areas of information visualization, used for commercial and organizational data, and to concept visualization. Our discussions of archaeology (see the sidebar "Archave") and color theory (see the sidebar "Color Museum") IVR systems give some sample domains and approaches.

Why scientific visualization?

Visualization is essential in interpreting data for many scientific problems. Other tools such as statistical analysis may present only a global or an extremely localized partial result. Statistical techniques and monitoring of individual points or regions in data sets expected to be of interest prove useful for learning the effect of a simulation, but these techniques generally cannot explain the effect.

Visualization is such a powerful technique because it exploits the dominance of the human visual channel (more than 50 percent of our neurons are devoted to vision). While computers excel at simulations, data filtering, and data reduction, humans are experts at using their highly developed pattern-recognition skills to look through the results for regions of interest, features, and anomalies. Compared to programs, humans are especially good in seeing unexpected and unanticipated emergent properties.

Much scientific computing uses parallel computers, benefiting from the productive synergy of using parallel hardware and software to do computation and par-

Archave

Daniel Acevedo Feliz and Eileen Vote

Members of the Computer Science Department at Brown University have been collaborating with archaeologists from the Anthropology and Old World Art and Archaeology departments to develop Archave, a system that uses the virtual environment of a cave as an interface for archaeological research and analysis.

VR in archaeology

Archaeologists and historians develop applications to reconstruct and document archaeological sites. The resulting models are displayed in virtual environments in museums, on the Internet, or in video kiosks at excavation sites. Recently, a number of projects have been tested in IVR environments such as caves and VR theaters.[1] Although archaeologists have used VR and IVR primarily for visualization since Paul Reilly introduced the concept of virtual archaeology in 1990, interest is increasing in using VR to improve techniques for discovering new knowledge and helping archaeologists perform analysis rather than simply presenting existing knowledge.[2] One proposed area for application of IVR is in the presentation and analysis of three-dimensionally referenced excavation data.

Archaeological method

The database for the Great Temple excavation in Petra, Jordan (see Figure A), contains more than 200,000 entries recorded since 1993. Following standard archaeological practice, artifacts recovered from the excavation site are recorded with precise 3D characteristics. All artifacts are also recorded in the site database in their relative positions by loci or excavation layer and excavation trench, with a number of feature attributes such as object type (bone, pottery, coin, metal, sculpture), use, color, size, key features, and date. This method ensures that all site data is precisely recorded for an accurate record of the disturbance caused by the excavation and for the analysis that occurs after the excavation is complete. Unfortunately, the full potential of spatially defined archaeological data is rarely realized, in part because archaeologists find it difficult to analyze the geometric characteristics of the artifacts and spatial relationships with other elements of the site.[3]

Photo courtesy of A.A.W. Joukowsky

A Aerial of the Great Temple site in Petra, Jordan.

Current problems in analysis

As the excavation proceeds, there's a strong need to correlate all the objects in order to observe patterns within the data set and perform standard analysis. Methods for this type of analysis vary widely depending on excavation site features, dig strategy, and data obtained. A quantitative analysis of all materials grouped and sorted in various ways presented in The Great Temple five-year report (1998) showed statistics about the percentages of different artifacts and their find locations, such as pottery by phase, pottery by area, and frequency of occurrence of pottery by area.[4] This type of analysis can help in a variety of statistical analyses using fairly comprehensive information from the database. It can also let the archaeologist quantify obvious patterns within the data set.

Unfortunately, many factors cannot be represented well with a traditional database approach and in reports generated from it. Specifically, these methods cannot integrate important graphical information from the maps and plans, and specific attribute data, location, and relational data among artifacts and site features.

Besides obvious conclusions that can be drawn when objects are correlated spatially, combinations of artifacts when viewed by a trained eye in their original spatial configurations can yield important and unlikely discoveries. Lock and Harris suggested that "vast quantities of locational and thematic information can be stored in a single map and yet, because the eye is a very effective image processor, visual analysis and information retrieval can be rapid."[3]

Although processing information visually would seem a more intuitive and thus effective way of processing 3D data, the idea hasn't yet been proven. More graphical methods of analysis have been explored in geographic information systems (GIS) systems that overlay multiple types of 2D graphic representations of data such as maps, plans, and raster images with associated attribute data in an attempt to present relationships among spatial data. However, many feel that it's not clear that GIS systems are sophisticated enough to provide a thorough description of height relationships. As Clarke observed, "The spatial relationships between the artifacts, other artifacts, site features, other sites, landscape elements and environmental aspects present a formidable matrix of alternative individual categorizations and cross-combinations to be searched for information."[5]

A new method

The Archave system displays all the components of the excavation with recorded artifacts and features in a cave environment. Like the excavation site, the virtual site is divided into the grid of excavation trenches excavated over the past seven years (see Figure B). Each trench is modeled so that the user can look at the relative layers or loci the excavator established during the removal of debris in that

B Color-coded excavation trenches from the past seven years of the dig at the Great Temple in Petra, Jordan.

trench. As the user dictates, information about artifacts can be viewed throughout the site (see Figures C1 and C2).

We believe that the system makes it easier to associate objects in all three dimensions, so it can accommodate objects that cannot be related to each other in 2D or even 3D map-based GIS. In addition, multiple data types such as pottery concentrations, coin finds, bone, sculpture, architecture, etc. can be visualized together to test for patterns and latent relationships between variables.

Users have commented that they feel more comfortable using the system in a cave because it allows them to access the data at a natural, life-size scale. The immersion provided by the cave gives the users improved accessibility to the objects they need to identify and study, and a very flexible interface for exploring the data at different levels of detail, going smoothly from close-range analysis to site-wide overviews. More importantly, the wide field of view provided by the cave's three walls and floor let the user assimilate and compare a larger section of data at once. For example, a user working at close range in a trench has full visual access to neighboring trenches or other parts of the site. Therefore, it's possible to assimilate and compare more information at one time. This becomes crucial when users look for patterns or try to match elements throughout the site or between trenches.

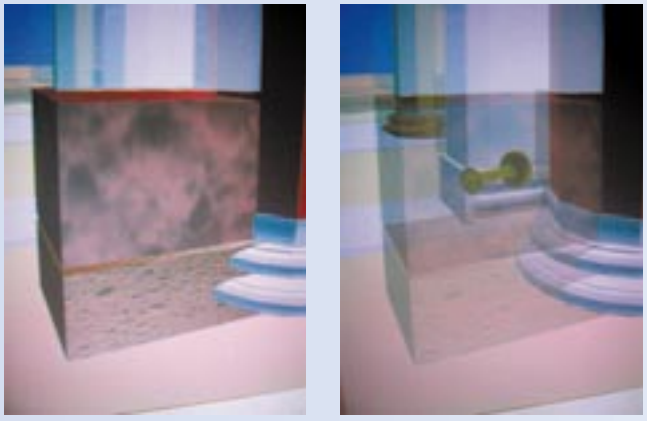
Conclusion

The Archave system lets the user establish new hypotheses and conclusions about the archaeological record because data can be processed comprehensively in its natural 3D format. However, along with the ability to visually process a coherent, multidimensional data sample comes the need for an intuitive and flexible environment. IVR provides the user with the ability to access the system in an environment similar to the conditions that a working archaeologist encounters on site.

As stated in the beginning, current tendencies to implement VR for reconstruction and visualization need not be the only use for this technology. The standard archaeological method provides a rich record in which high-level analysis can occur, and IVR can provide a significant test-bed for advanced forms of analysis not heretofore available.

References

1. B. Frischer et al., "Virtual Reality and Ancient Rome: The UCLA Cultural VR Lab's Santa Maria Magggiore Project," *Virtual Reality in Archaeology*, J.A. Barcelo, M. Forte, and D. Sanders, eds., BAR International Series 843, Oxford, 2000, pp. 155-162.
2. P. Miller and J. Richards, "The Good, the Bad, and the Downright Misleading: Archaeological Adoption of Computer Visualization," *Computer Applications in Archaeology*, J. Huggett and N. Ryan, eds., British Archaeological Reports (Int. Series, 600), Oxford, 1994, pp. 19-22.
3. G. Lock and T. Harris. "Visualizing Spatial Data: The Importance of Geographic Information Systems," *Archaeology in the Information Age: A Global Perspective,* P. Reilly and S. Rahtz, eds., Routledge, London, 1992, pp. 81-96.
4. M.S. Joukowsky, *Petra Great Temple: Volume I*, Brown University Excavations 1993-1997, E.A. Johnson Company, USA, 1998, p. 245.
5. D.L. Clarke, "A Provisional Model of an Iron Age Society and its Settlement System," *Models in Archaeology*, D.L. Clarke, ed., Methuen, London, 1972, pp. 801-869.



C Visualization of excavation data in one of the trenches inside the Great Temple. (1) The texture maps show two parameters: The color saturation indicates the concentration of pottery, and the density of the texture indicates the concentration of bone—here only a significant difference in bone concentration exists. (2) Making the trench semitransparent reveals special finds in the exact location in which they were found inside the volume.

allel human wetware to interpret the results. Computational steering—inappropriate for massive, lengthy production runs—often proves useful for smaller-scale problems with coarser spatiotemporal resolution or for test cases to help set up parameters for production runs.

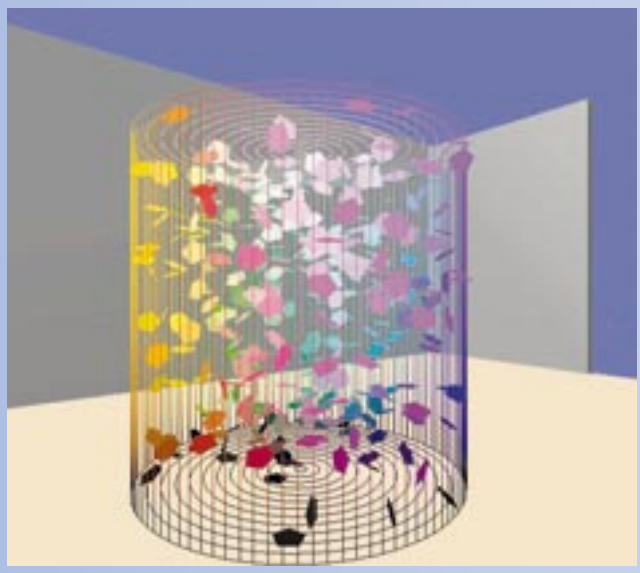
Computer-based scientific visualization exploiting human pattern recognition is scarcely a new idea. It started with real-time oscilloscope data plotting and offline paper plotting in the 1950s. Science and engineering applications were the first big customers of the

Color Museum

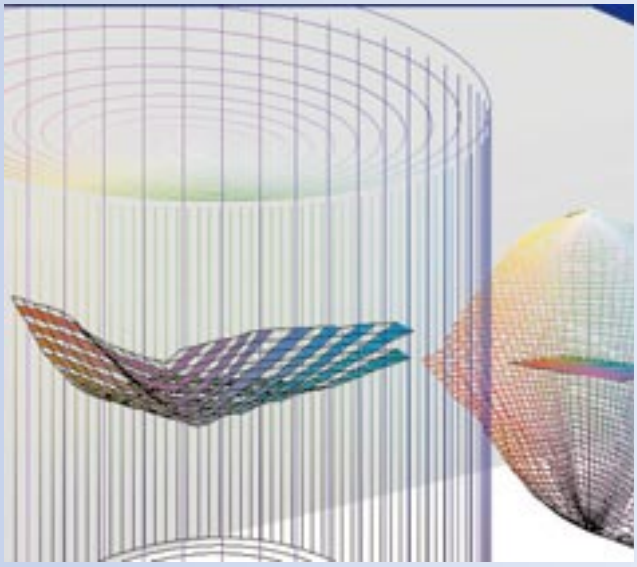
Anne Morgan Spalter

IVR can enable students to interact with ideas in new ways. In an IVR environment, students can engage in simulations of real-world environments that are inaccessible due to financial or time constraints (high-end chemistry laboratories, for instance) or that cannot be experienced, such as the inside of a volcano or the inside of an atom. IVRs also enable students to interact with visualizations of abstract ideas, such as mathematical equations or elements of color theory. The hands-on, investigative learning most natural in IVR offers an excellent way to train new scientists and engineers. In addition, because the environment is computer-generated, it's an ideal future platform for individual and collaborative distance-learning efforts.

We've used our cave to teach elements of color theory.[1] Color theory is often highly abstract and multidimensional, making it difficult to explain well with static diagrams or even 3D real-world models. The desktop environment provides valuable flexibility in the study of color (for example, making it easy to modify colors rapidly), but doesn't address the difficulty of understanding the 3D nature of color spaces and the complex interactions between lights and colored objects in a real-world setting. In the cave, users of our Museum of Color can view fully 3D color spaces from any point of view and use a variety of interaction and visualization techniques (for example, a rain of disks colored by their location in a space and an interactive cutting plane) to explore individual spaces and compare spaces with one another (see Figure D). In addition, viewers can enter the spaces and become fully immersed in them, seeing them from the inside out.



D Falling disks are colored according to their changing positions within the color space.



E A plane of constant perceived value is flat in Munsell space (right) but warped in HSV space (left).

We believe that the experience of entering a color space in an IVR differs fundamentally from examining 3D desktop models and that the experience will help users develop a better understanding of color structures and the relative merits of different spaces. For example, our color space comparison exhibit lets the user move a cutting plane in Munsell space and see it mapped into both RGB and HSV spaces. The plane is defined by gradients of constant hue, saturation, and value, and thus is flat in the perceptually based Munsell space. In RGB, and especially HSV, however, the plane deforms, at times quite radically, demonstrating the nonlinearities of those color spaces. Although we could show a single example of such a comparison in a picture (see Figure E), actual use of this technique in the cave lets users actively explore different areas of the spaces and experience their changing degrees of nonlinearity.

In other interactive exhibits in the museum, users can experiment with the effects of additive and subtractive mixing by shining colored lights on paintings and 3D objects. This provides a hands-on approach impossible with desktop software while offering a more easily controlled and more varied environment than practical in a real-life laboratory. Future plans include more exhibits (such as one on color scale that shows the user why choosing a color from a little swatch for one's walls is often misleading), as well as user testing of the pedagogy and interaction techniques.

References

1. A.M. Spalter et al., "Interaction in an IVR Museum of Color," *Proc. of ACM Siggraph 2000 Education Program*, ACM Press, New York, 2000, pp. 41-44.

expensive interactive 2D vector displays commercialized in the 1960s. Graphing packages of various kinds were designed for both offline and interactive viewing then as well. In the mid-1980s considerably higher-level interactive visualization packages such as Mathematica and AVS leveraged the power of raster graphics and modern user interfaces.

The landmark 1987 National Science Foundation

report "Visualization in Scientific Computing"[7] stressed the importance of interactive scientific visualization, especially for large-scale problems, and reminded us of Hamming's famous dictum: "The purpose of computing is insight, not numbers." The authors' observation that "Today's data sources [simulations and sensors] are such fire hoses of information that all we can do is gather and warehouse the numbers they generate" is unfortunately as true today as it was then.

Why use IVR for scientific visualization?

Several factors prompt the use of IVR for scientific visualization. IVR also shows potential to surpass other forms of desktop-based visualization.

Exponential growth of data set size

Moore's Law for computer processing power and similar improvements in storage, network, and sensing device performance continue to give us ever-greater capacity to collect and compute raw data. Unfortunately, computational requirements and data set size in science research are growing faster than Moore's Law. Thus, the gap between what we can gather or create and what we can properly analyze and interpret is widening (see the sidebar "Examples of Data Set Size"). In the limit, the real issue is nature's overwhelming complexity. Galaxy or plasma simulations, for example, are seven-dimensional problems, so doubling resolution can increase computation by a factor of 128. It's extremely difficult to make headway against problems this hard, and there are hundreds of comparable complexity.

Problems and proposed solutions

The 1998 DOE report "Data and Visualization Corridors"[8] proposed three technology roadmaps to address the crisis: data manipulation; visualization and graphics; and interaction, shared spaces, and collaboration. This document and subsequent reports show that systems are unbalanced today and that our ability to produce and collect data far outweighs our ability to visualize or work with it. The main bottleneck continues to be the ability to visualize the output and gain insight.

While the raw polygon performance of graphics cards may be on a faster track than Moore's law, visualization environments aren't improving commensurately. In graphics and visualization, the key barriers to achieving really effective visualizations are underpowered hardware, underdeveloped software, inadequate visual idioms/encoding representations and interaction metaphors not based on a deep understanding of human capabilities, and disproportionately little funding for visualization. We address some of these problems as research issues in the sections below.

The accelerating data crisis demands new approaches short term and long term, new forms of abstraction, and new tools. Short term, Moore's Law, visualization clusters (parallel approaches), tiled displays (increased image fidelity), and IVR should help the most. Long term, artificial-intelligence-based techniques will cull, organize, and summarize raw data prior to IVR viewing, while ensuring that the links to the original data remain. These techniques will support adjustable detail-and-con-

Examples of Data Set Size

Andrew Forsberg

At the Department of Energy's Accelerated Strategic Computing Initiative (ASCI, http://www.llnl.gov/asci/), a wide range of applications that generate huge amounts of data are run to gain understanding through simulations. Table A gives an overview of the size of current and anticipated ASCI simulations. These estimates are based on an actual run of a "multi-physics code." The very largest ASCI simulation runs, known as hero calculations, produce even larger data sets that generate an order of magnitude more data than typical runs. Developing techniques to manage and visualize these data sets is a formidable problem being actively researched at both DOE laboratories and universities.

The National Center for Atmospheric Research (NCAR, http://www.ncar.ucar.edu/ncar/index.html) studies data volumes associated with earth system and related simulation. Climate, weather, and general turbulence are of particular interest. Climate simulations generally produce about 1 TBytes of raw data per 100-year run. Running ensemble runs—several simulations with small differences—is important and multiplies the amount of data that must be analyzed.

Monthly time dumps are currently used; in the future, these time dumps may be hourly for certain studies, increasing the data size by many orders of magnitude. Hurricane simulations at 1 km resolution and sampled every 15 minutes may produce as much as 3 TBytes of raw data. Unlike some simulations, all this data (in terms of both time and space) must be visualized for many variables. In addition, geophysical and astrophysical turbulence has been a particularly active and fruitful research area at NCAR, but researchers are limited by both computational and analytical resources.

One current effort in astrophysical turbulence runs at 2.5 km resolution, and even with what is considered a crude and insufficient time sampling, produces net data volumes of about .25 TBytes. Given more resources, researchers could use a finer time sampling, add more variables, and conduct several runs for comparison with one another. This would result in a final data set size of about 6 TBytes. As soon as it's practical to do so, researchers will double the resolution of the simulation, yielding a 50-TByte data set. And if it were possible, researchers would benefit from running at four times the resolution.

Sensors that collect data produce data sets on the order of petabytes today. For example, the compact muon solenoid detector experiment on CERN's Large Hadron Collider (http://cern.web.cern.ch/CERN/LHC/pgs/general/detectors.html) will collect about a petabyte of physics data per year. A more visualizable data set is CACR's collection of all-sky surveys known as the Digital Sky (http://www.cacr.caltech.edu/SDA/DigiSky/whatis.html); this is starting out at tens of terabytes and will grow.

Another example comes from developmental biology. Using multispectral, time-lapse video microscopy, it's now possible to acquire volume images showing where and when a gene is expressed in developing avian embryos. To accomplish this, a given gene is modified so that when expressed it produces not only the protein it represents, but also an additional marker protein. The marker can be imaged in a live avian embryo as it develops, producing a time-varying volume image.

continued on p. 32

continued from p. 31

The acquired data sets are large. Changes recorded every 90 minutes—a moderate timestep in the scale of development, corresponding to the formation of one somite in a developing embryo—over the first four days of development roughly correspond to the first trimester of human development. A single acquisition can measure expression of three genes in a volume of $512 \times 768 \times 150$ spatial points for 64 time steps, producing 22 gigabytes of data. As many as 10 images are necessary to cover an entire embryo. Images of the 100,000 genes expressed in avians would exceed a petabyte. The real complexity of the problem, and its true potential, lies in correlating hundreds or thousands of these images in order to understand the many different proteins working in concert. IVR has the potential to help solve this problem by showing the multivalued data simultaneously so that the human visual system, arguably the best pattern-finding system known, can search for these correlations.

Table A. Data output from one ASCI code.*

Sizing Requirements	FY00 4 TFLOP	FY02 30 TFLOP	FY04 100 TFLOP
Number of zones (locations where material properties such as pressure, temperature, chemical species, stress tensor, and so on are tracked)	25 million	200 million	1 billion
Number of material properties per zone	10-50	10-50	10-50
Small visualization file (such as description of mesh, one zonal, one nodal)	2 GBytes	12 GBytes	50 GBytes
Large plot/restart file size (with all physics variables saved)	60 GBytes	450 GBytes	1,500 GBbytes
Average length of run	20.5 days	20-40 days	20-40 days
Number of visualization files per run	100 – small	200 – small	200 – small
	180 – large	180 – large	180 – large
Visualization data set size per major run	6.4 TBytes	84 TBytes	280 TBytes

*Data has been scaled linearly to FY02 and FY04 using data from a recent run. Data courtesy of Terri Quinn, Lawrence Livermore National Laboratory.

text views to let researchers zoom in on specific areas while maintaining the context of the larger data set.

IVR versus 3D desktop-based visualization

Visualization that leverages our human pattern-recognition ability can be a powerful tool for understanding, and any technique that lets the user "see more" enhances the experience. Complex 3D or higher-dimensional and possibly time-varying data especially benefit from interactive exploration to see more. One way of seeing more is to use greater levels of abstraction/encoding in the data. However, for a given data representation, the more the eye can rapidly take in, the better.

IVR allows much more use of peripheral vision to provide global context. It permits more natural and thus quicker exploration of three- and higher-dimensional data.[9] Additionally, body-centric judgments about 3D spatial relations come more easily,[10] as can recognition and understanding of 3D structures.[11] It's easier to do such tasks when 3D depth perception is enhanced by stereo and motion parallax (via head tracking).

In a sense, using IVR's kinesthetic depth perception to visualize phenomena represents the life-size interactive generalization of stereo pairs in textbooks. Bryson made the case that real-time exploration is a desirable capability for scientific visualization and that IVR greatly facilitates exploration capabilities.[11] In particular, a proper IVR environment provides a rich set of spatial and depth cues, and rapid interaction cycles that enable probing volumes of data. Such rapid interaction cycles rely on multimodal interaction techniques and the high performance of typical IVR systems. While any of IVR's input devices and interaction techniques could, in principle, work in a desktop system, IVR seems to encourage more adventurous use of interaction devices and techniques. (See the sidebar "Interaction in Virtual Reality: Categories and Metaphors.")

Other interesting differences separate IVR and conventional desktop environments. Current research at Carnegie Mellon University and the University of Virginia shows that users make the same kinds of mistakes in spatial judgments in the virtual world that they do in the real world (such as overestimating height-width differences of objects), which isn't the case in 3D desktop graphics. Also, certain kinds of hand-eye coordination tasks, such as rotating small objects, are easier in IVR. Typical times for rotating objects to match a target orientation using a virtual trackball or arcball at the desktop[12] fall in the range of 17 to 27 seconds, but having the user's hand and the virtual object collocated in 3D space for optimal hand-eye coordination can reduce this time by an order of magnitude to around two seconds.[13]

All these phenomena exemplify how the IVR experience comes closer to our real-world experience than does 3D desktop graphics. Indeed, IVR produces an undeniable qualitative difference. Looking at a picture of the Grand Canyon, however large, differs fundamentally from being there. Again, IVR is more like the real world than any photograph, or any conventional "through the window" graphics system, could be.

IVR is used in scientific visualization in two sorts of problems: human-scale and non-human-scale problems. The case for using IVR is more obvious for the former, as Brooks described,[1] citing vehicle operation, vehicle design, and architectural design. For example, an architectural walkthrough will, in general, be more effective in an IVR environment than in a desktop environment because humans have a lifetime of experience in navigating through, making spatial judgments in, and manipulating 3D physical environments. Ergonomic validation tasks, like checking viewable and reachable cockpit instrumentation and control placement, can be performed more quickly and efficiently in a virtual prototyping environment than with labor-intensive physical prototyping. Bryson's pioneering work on the virtual wind tunnel lets researchers "experience" fluid flow over

Interaction in Virtual Reality: Categories and Metaphors

Joseph J. LaViola, Jr.

When discussing 3D user interfaces for IVR, it's important to break down different interaction tasks into categories so as to provide a framework for the design of new interaction metaphors. In contrast to 2D WIMP (windows, icons, menus, pointing) interfaces, which have a commonality in their structure and appearance, only a few standard sets of sophisticated interface guidelines, classifications, and metaphors for 3D user interfaces[1] have emerged due, in part, to the complex nature of the interaction space and the additional degrees of freedom. However, 3D user interaction can be roughly classified into navigation, selection and manipulation, and application control.

Navigation

Navigation can be classified into three distinct categories. Exploration is navigation without any explicit target (that is, the user simply explores the environment). Search tasks involve moving through the environment to a particular location. Finally, maneuvering tasks are characterized by short, high-precision movements usually done to position users better for performing other tasks. The navigation task itself is broken up into a motor component called travel, the movement of the viewpoint from place to place, and a cognitive component called wayfinding, the process of developing spatial knowledge and awareness of the surrounding space.[2] For example, with large scientific data sets, users must be able to move from place to place and make spatial correlations between different parts of the data visualization.

Most travel interaction metaphors fall into one of five categories:

- *Physical movement*: using the motion of the user's body to travel through the environment. Examples include walking, riding a stationary bicycle, or walking in place on a virtual conveyer belt.
- *Manual viewpoint manipulation*: the user's hand motions effect travel. For example, in Multigen's SmartScene navigation, users grab points in space and pull themselves along as if holding a virtual rope.
- *Steering*: the continuous specification of the direction of motion. Examples include gaze-directed steering, in which the user's head orientation determines the direction of travel, or two-handed flying, in which the direction of flight is determined by the vector between the user's two hands and the speed is proportional to the user's hand separation.[3]
- *Target-based travel*: the user specifies the destination and the application handles the actual movement. An example of this type of travel is teleportation, in which the user immediately jumps to the new location.
- *Route planning*: the user specifies a path and the application moves the user along that path. An example is drawing a path on a map of the space or actual environment to plan a route.

Selection and manipulation

A number of metaphors have been developed for selecting, positioning, and rotating objects. The classical approach provides the user with a virtual hand or 3D cursor whose movements correspond to the physical movement of the hand tracker. This metaphor simulates real-world interaction but is problematic because users can pick up and manipulate objects only within the area of reach.

One way around this problem is to use ray-casting or hand-extension metaphors such as the Go-Go technique[4] for object selection and manipulation. The Go-Go technique interactively "grows" the user's arm using a nonlinear mapping. Thus the user can reach and manipulate both near and distant objects. Ray-casting metaphors are based on shooting a ray from the virtual hand into the scene. When the ray intersects an object, the user can select and manipulate it.

With simple ray casting the user may find it difficult to select very small or distant objects. Variants of ray casting developed to handle this problem include the spotlight technique[5] and aperture-based selection.[6] Another approach within the metaphor of ray casting is the image-plane family of interaction techniques,[7] which bring 2D image-plane object selection and manipulation, commonly found in 3D desktop applications, to virtual environments.

Instead of having users reach out into the virtual world to select and manipulate objects, another metaphor for this interaction task is to bring the virtual world closer to the user. One of the first examples of this approach, the 3DM immersive modeler,[8] lets users grow and shrink themselves to manipulate objects at different scales. In another approach, the World-In-Miniature (WIM) technique[9] provides users with a handheld copy of the virtual world. The user can indirectly manipulate the virtual objects in the world by interacting directly with their representations in the WIM.

In addition to direct manipulation of objects, users can control objects indirectly with 3D widgets.[10] They extend familiar 2D widgets of WIMP interaction and are combinations of geometry and behavior. Some are general, such as transformer widgets with handles to constrain the translation, rotation, and scale of an object. Others are task-specific, such as the rake emitting colored streamlines used in computational fluid dynamics (shown to the left of the user's left hand in Figure F). They're used to modify parameters

continued on p. 34

F A user interacting with a scientific data set. He uses a multimodal interface combining hand gesture and speech to change modes and application state, such as creating and controlling visualization widgets, and starting and stopping recorded animations.

continued from p. 33

associated with an object and to invoke particular operations.

A third way of selecting and manipulating virtual objects is to create physical props, or phicons,[11] that act as proxies for virtual objects, giving users haptic feedback and a cognitive link between the virtual object and its intended use.

A number of other techniques and metaphors have been developed for selection and manipulation in virtual environments. For references see Poupyrev and Kruijff's annotated bibliography of 3D user interfaces of the 20th century, available on the Web at http://www.mic.atr.co.jp/~poup/3dui/3duibib.htm.

One of the areas we must continue to explore is how to combine the various interaction styles to enrich user interaction. For example, physical props can help to increase the functionality of virtual widgets and vice versa. The Virtual Tricorder[12] (see Figure F), which uses a physical prop and has a corresponding virtual widget, is a good example of this combination. These types of hybrid interface approaches help reduce the user's cognitive load by providing familiar transitions between tools and their functionality.

Application control

Application control tasks change the state of the application, the mode of interaction, or parameter adjustment, and usually include the selection of an element from a set. There are four different categories of application control techniques: graphical menus, voice commands, gestural interaction, and tools (virtual objects with an implicit function or mode). Example application control tools include the Virtual Tricorder and rake widgets mentioned above.

These application control techniques can be combined in a number of different ways to create hybrid interfaces. For example (see Figure G), combining gestural and voice input provides users with multimodal interaction, which has the potential of being a more natural and intuitive interface. Although application control is a part of most VR applications, it hasn't been studied in a structured way, and evaluations of the different techniques are sparse.

G The display geometry of the virtual tricorder closely reflects the geometry of the 6DOF Logitech FlyMouse, enhanced with transparent menus.

The future

Within the last couple of years, we've seen a significant slowdown in the number of novel IVR interaction techniques appearing in the literature, largely because it's becoming more and more difficult to develop them. Among the new directions to pursue in 3D user interface research should be more evaluations of already existing techniques to see which ones work best for which applications and IVR environments. Despite our many different 3D interaction metaphors, we lack an application-based structure to tell us where these metaphors are best used.

Another direction to consider is extending 3D interaction techniques with artificial intelligence. For example, with increasing data set size, AI techniques will become essential in feature extraction and detection so that users can visualize these massive data sets. Another example is using machine-learning algorithms so that the application can detect various user patterns that could aid users in interaction tasks. Incorporating AI into 3D interaction has the potential to spawn new sets of 3D interface techniques for VR applications that would otherwise not be possible.

References

1. D. Bowman et al., *3D User Interface Design: Fundamental Techniques, Theory, and Practice*, Course #36, Siggraph 2000, ACM, New York, July, 2000.
2. K. Hinckley et al., "A Survey of Design Issues in Spatial Input," in *Proc. of ACM UIST '*94, 1994, pp. 213-222.
3. M. Mine, "Moving Objects in Space: Exploiting Proprioception in Virtual Environment Interaction," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, 1997, pp. 19-26.
4. I. Poupyrev et al., "The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR," *Proc. of the ACM User Interface Software and Technology (UIST) 96*, ACM Press, New York, 1996, pp. 79-80.
5. J. Liang and M. Green, "JDCAD: A Highly Interactive 3D Modeling System," *Computer and Graphics*, Vol. 4, No. 18, 1994, pp. 499-506.
6. A.S. Forsberg, K.P. Herndon, and R.C. Zeleznik, "Aperture-Based Selection for Immersive Virtual Environments," *Proc. of User Interface Software and Technology (UIST) 96*, ACM Press, New York, 1996, pp. 95-96.
7. J.S. Pierce et al., "Image Plane Interaction Techniques in 3D Immersive Environments," *Proc. 1997 ACM I3D (Interactive 3D Graphics)*, ACM Press, New York, 1997, pp. 39-43.
8. J. Butterworth et al., "3DM: A Three Dimensional Modeler Using a Head-Mounted Display," *Proc. ACM Symp. on Interactive 3D Graphics (I3D)*, ACM Press, New York, 1992, pp. 135-138.
9. R. Stoakley, M. Conway, and R. Pausch, "Virtual Reality on a WIM: Interactive Worlds in Miniature," *Proc. ACM Computer-Human Interaction (CHI) 95*, ACM Press, New York, 1995, pp. 265-272.
10. K.P. Herndon and T. Meyer, "3D Widgets for Exploratory Scientific Visualization," *Proc. of ACM User Interface Software and Technology (UIST) 94*, ACM Press, New York, 1994, pp. 69-70.
11. H. Ishii and B. Ullmer, "Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms," *Proc. of ACM Computer-Human Interaction (CHI) 97*, ACM Press, New York, 1997, pp. 234-241.
12. M. Wloka and E. Greenfield, "The Virtual Tricorder: A Uniform Interface for Virtual Reality," *Proc. of ACM User Interface Software and Technology (UIST) 95*, ACM Press, New York, 1995, pp. 39-40.

a life-sized replica of the space shuttle.[14] Finally, in complex 3D environments such as oil refineries, orientation and navigation seem easier with IVR—the simulated environment stays fixed while your body moves—than with desktop environments, where the mouse or joystick makes the VE rotate around you.

While some problems (such as visualizing numerical simulation of arterial blood flow) aren't naturally human-scale, they can be cast into a human-scale setting. There they can arouse the normal human reactions to 3D environments. For example, in arterial flow (see the "Artery" sidebar), when our users enter the artery, they think of it as a pipe—a familiar object at their own macro scale. By entering the artery and viewing the 3D vorticity geometry in 3D, they can make better decisions about which viewpoints and 2D projections are most meaningful. A similar example, which Brooks described,[1] is the University of North Carolina nanomanipulator project, in which the humans and their interactions are scaled down to the nanoscale.

The key question for non-human-scale problems is whether the added richness of life-size immersive display allows faster, easier, or more accurate perceptions and judgment. So far, not enough controlled studies have been done to answer this question definitively. Anecdotal evidence indicates that it's easier, for example, to do molecular docking[15] for drug design in IVR than on the desktop.

In addition to varying perspectives of scale in data sets with inherent physical geometries, we often face data that have no inherent geometry (such as flow field data) and perhaps even no physical scale (such as data describing statistical phenomena). The 3D abstractions through which we visualize these data sets often present very irregular structure with complicated geometries and topologies. Just as IVR allows better navigation through complicated architectural-scale structures, we believe that IVR will be a better environment for conception, navigation, and exploration in any visualization of complex 3D structure. Indeed, what's often far more difficult in nonimmersive interactive visualizations is to gain rapid insight through simple head movement and natural multimodal interaction. As Haase et al. pointed out in 1994, "[IVR can provide] easy-to-understand presentations, and more intuitive interaction with data" and "rather than relying almost exclusively on human cognitive capabilities, [IVR applications for analysis of complex data] engage the powerful human perceptual mechanisms directly."[16]

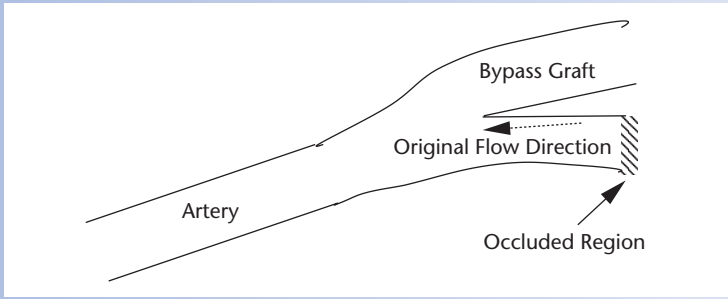
Despite the lack of conclusive evidence that today's IVR "is better" for scientific visualization, we remain optimistic that in due time it will improve sufficiently in cost, performance, and comfort to become the medium of choice for many visualization tasks.

Artery

Andrew Forsberg

In collaboration with Spencer Sherwin of Imperial College, researchers at Brown University are studying blood flow in arterial branches.[1] Understanding flow features and transport properties within arterial branches is integral to modeling the onset and development of diseases such as arteriosclerosis.

We're currently examining the geometries of arterial bypass grafts, which are surgically constructed bypasses around a blockage in an artery. They have a downstream (proximal) junction where the bypass vessel attaches to the original (host) artery and an upstream (distal) junction where the bypass vessel is reattached after the blockage. The disease occurs most frequently at this downstream junction, therefore this geometry is of greatest interest (see Figure H).



H **Diagram of an arterial graft.**

These flows are typically unsteady and have no clearly defined symmetries. The fields we're interested in can be expressed in terms of a vector, velocity, and scalar field. Interpreting this type of data requires understanding the forces on a fluid element due to local pressure and viscous stresses. In general, these forces aren't collinear or aligned with any preferred Cartesian direction. The use of traditional 2D visualization can therefore be limiting, especially when considering a geometrically complicated situation with no planes of symmetry.

It's useful to consider the coherent structure identification as a whole to get a general picture. The scale of this structure is typically of the order of the artery's diameter. The physical scales of the problem are bounded from above by the geometry diameter and from below by the viscosity length scale (the length at which structures disappear). These coherent structures typically occur along the local primary axis of the vessels. Therefore, at the junction of the vessels we have two sets of flow structures in different planes. At higher flow speeds smaller flow features can also occur due to the nonlinear nature of the flow as the transition to turbulence takes place.

Output data is often so large it cannot be visualized. In their work on suppressing turbulence, Du and Karniadakis[2] processed only a small percentage of this data, typically in terms of statistics such as

continued on p. 36

Research challenges

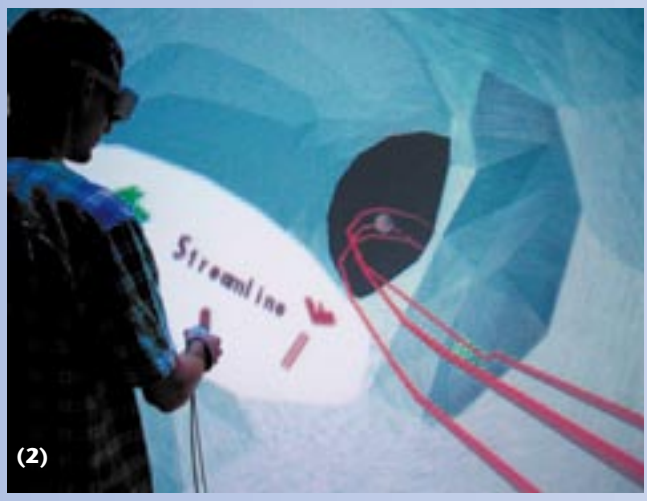
First we summarize the field's current state. Next we explore some IVR research challenges: display hardware, rendering (including parallelism in rendering), haptics, interaction, telecollaboration, software standards and interoperability, and user studies. Finally, we cover the challenges for scientific and information visualization.

Where is IVR today?

Thanks to Moore's Law and clever rethinking of graphics architecture at both the hardware and algorithm levels, 3D graphics hardware has seen much progress recently. Commodity chips and cards, such as those in NVidia's GeForce2 Ultra and Sony's PlayStation-2, provide an astonishing improvement over previous genera-

continued from p. 35

mean value at one point. However, it's difficult to relate these point-wise quantities to the flow structures to construct theories: the statistics can show the effect, but not the cause. Examining the detailed small-scale flow structures can lead to discovery of the cause.



I (1) A view within the artery looking downstream from the bifurcation. Shear-stress values on the artery wall are shown using a standard color mapping technique in which blue represents lower values and red represents higher values. Regions of low shear stress tend to correlate with locations of future lesions. (2) A view within the artery looking upstream at the bifurcation. The user holds a wand that controls a virtual menu and has created three streamline widgets and a particle advection widget. A texture map on the artery wall helps the user perceive the 3D geometry.

The challenge is how to interpret the dynamics at the junction. One problem is that, when viewed from the outside, some of the structures block each other; thus viewing from inside the junction can let us understand how each flow feature interacts with others. Note that viewing a time-varying isosurface alone will not be sufficient to understand the whole flow pattern.

Figure I shows snapshots of the user visualizing the arterial flow data. In Figure I, part 2, the user is positioned just downstream from the bifurcation, facing the bypass graft from which three streamlines emanate. The bifurcation partially occludes the graft passage, and to the right of the bifurcation is the occluded region (see the corresponding features in Figure H). The user holds a wand to control a virtual menu of options ("Streamline" is the current selection) and to create and adjust the parameters of visualization widgets.

IVR may help in understanding the artery data in several ways. Most immediately, viewing the 3D data from the inside is easier in IVR than in desktop environments. IVR's fundamental attributes (such as a wide field of view, a larger number of pixels, and head-tracked stereo) contribute to enhanced 3D viewing. In an immersive environment you can stand at a point such as the intersection of the bypass graft and consider how flow features such as rotation and straining occurring in different physical planes can interact and exchange information. Multimodal user interfaces enable users to control the visualization process rapidly and more naturally than desktop WIMP interfaces. Group interaction is also a benefit of IVR.

Of course, problems remain to be solved. For example, we desperately need to increase the fidelity (both in terms of spatio-temporal resolution and overall aesthetic quality) of the visualization while maintaining interactive frame rates. When more than one person views the data, they can have difficulties communicating because each person has a different point of view. Consequently, very natural and common tasks like pointing at features are deceptively difficult to implement (despite some research to enhance multi-user interaction[3]).

References

1. A.S. Forsberg et al., "Immersive Virtual Reality for Visualizing Flow through an Artery," *Proc. of IEEE Visualization 2000,* in publication.
2. Y. Du and G.E. Karniadakis, "Suppressing Wall Turbulence by Means of a Transverse Traveling Wave," *Science*, Vol. 288, No. 5469, 19 May 2000, pp. 1230-1234.
3. M. Agrawala et al., "The Two-User Responsive Workbench: Support for Collaboration through Individual Views of a Shared Space," *Proc. of ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, 1997, pp. 327-332.

tions of even high-end workstations. These advances are made possible by graphics processor designs larger than those of microprocessors and produced on a timetable even more aggressive. One could construct a very decent personal IVR system from a PC with such a high-end card, a high-quality lightweight HMD, and robust, high-resolution, large-volume trackers for head-tracking and interaction devices. Of these three components, tracker technology is actually the most problematic, given today's primitive state of the art.

Meanwhile, the number of IVR installations, including semi-immersive workbenches plus fully immersive caves and HMD-based environments, is steadily increasing. Wall-sized single or tiled displays offer an

increasingly popular alternative to IVR, particularly for group viewing. IVR environments augment traditional design studios and "wet labs" in such areas as biochemistry and drug design. Lin et al.[17] reported increasing use in the geosciences. Supercomputer centers, such as those at the National Center for Supercomputing Applications and the Cornell Theory Center, provide production use of their IVR facilities. Thus it's fair to say that, slowly but surely, scientists are becoming acquainted with IVR in general and multimodal post-WIMP interaction in particular.

On the downside, while money does go into immersive environments, scientists and their management continue to treat investments in all visualization technologies as secondary to investments in computation and data handling.

Challenges in IVR

The following list of research issues in IVR is a partial one; space and time constraints prevent a more detailed listing. As often with systems-level research areas, there's no way to partition the issues neatly into categories such as hardware and software. Also, latency—a key concern for IVR—affects essentially all issues.

Furthermore, many of the issues outlined apply equally to 3D desktop graphics. Indeed, IVR can be considered as merely the most extreme point on the spectrum for all these research issues; solutions will come from researchers and commercial vendors not focused uniquely on IVR. A telling example is the Sony GScube, shown at Siggraph 2000, which demonstrated very high end performance using a special-purpose scalable graphics solution based on 16 PlayStation-2 cards. The carefully hand-tuned demo showed 140 ants (from the PDI movie *AntZ*), made up of around 7,000 polygons each, running in real time at 60 Hz at HDTV resolution, effectively over 1M polygons per frame at 1920×1080 resolution. A Sony representative quoted about 60M triangles per second and peak rates of roughly 300M triangles. Load-balancing algorithms helped improve the performance of the 16 PlayStation-2 cards and additional graphics hardware.

While 2D graphics is mature, with the most progress occurring in novel applications, we've reached no such maturity in 3D desktop graphics, let alone in IVR. This immaturity is manifested at all levels, from hardware through software, interaction technology, and applications. Progress will have to be dramatic rather than incremental to make IVR a generally available productive environment. This is especially true if our hope that IVR will become a standard work environment is to be realized.

Improve display technologies. Hardware display systems for IVR applications have two important and interrelated components. The first is the technology underlying how the light we see gets produced; the second is the type and geometrical form of surface on which this light gets displayed.

Invent new light production technologies. A number of different methods exist for producing the light displayed on a surface. While CRTs—and increasingly LCD panels and projectors—are the workhorses for graphics today, newer light-producing techniques are still being invented. Texas Instrument's Digital Micromirror Device (DMD) technology is available in digital light projectors. Even more exotic technology from Silicon Light, which uses Grating Light Valve technology, will soon handle theater projection of digital films. There's hope that this kind of technology may be commoditized for personal displays.

Jenmar Visual System's BlackScreen technology (used in the ActiveSpaces telecollaboration project of Argonne Laboratories) captures image light into a matrix of optical beads, which focus it and pass it through a black layer into a clear substrate. From there it passes relatively uniformly into the viewing area. This screen material presents a black level undegraded by ambient light, making it ideal for use with high-luminosity projection sources and nonplanar tiled displays such as caves.

A very different approach to light production, the Virtual Retinal Display (VRD), projects light directly onto the retina.[18] The VRD was developed at the Human Interface Technology (HIT) Lab in 1991 and is now being commercially offered by Microvision in a VGA form factor with 640×480 resolution. Because the laser must shine directly onto the retina, visual registration is lost if the eye wanders. Autostereoscopic displays such as that described by Perlin et al.[19] are less obtrusive than stereo displays, which require shutter glasses. Light-emitting polymers hold the promise of display surfaces of arbitrary size and even curved shape. The large, static, digital holograms of cars displayed at Siggraph 2000 demonstrated an impressive milestone toward the real-time digital holography we can expect in the far future.

Create and evaluate new display surfaces. Unfortunately, no "one size fits all" display surface exists for IVR applications. Rather, many different kinds offer advantages and disadvantages. Choosing the appropriate display surface depends on the application, tasks required, target audience, financial and human resources available, and so on. In addition to Fish Tank VR, workbenches, caves, HMDs and PowerWalls, new ways of displaying light continue to emerge.

Tiled display surfaces, which combine many display surfaces and light-producing devices, are very popular for visualization applications. Tiled displays offer greater image fidelity than other immersive and desktop displays today due to an increased number of pixels displayed (for example, $6.4K \times 3K$ in Argonne's 15-projector configuration, in contrast to typical display resolution of 1280×1024) over an area that fills most of a user's, or often a group of users', field of view.[20] For a thorough discussion of tiled wall displays, see *CG&A*'s special issue on large displays.[6]

Not surprisingly, practical drawbacks of IVR display systems are their cost and space requirements. This problem plays a significant role in inhibiting the production use of IVR by scientists. However, semi-immersive personal IVR displays such as CMU's CUBE (Computer-driven Upper Body Environment) are emerging. In addition, the VisionStation from Elumens and Fakespace Systems' conCAVE are hemispheric per-

sonal displays that use image warping to compensate for the nonplanar display topology.

Understanding which IVR display surfaces best suit which application areas occupies researchers today. For example, head-tracked stereo displays typically provide a one-person immersive experience, since current projection hardware can only generate accurate stereo images for one person. (Some proposed strategies time-multiplex the output of a single projector,[21] but exhibit problems such as decreased frame rate and darker images.) Non-head-tracked displays, such as a PowerWall, which takes some advantage of peripheral vision, proves much better for group viewing. Given the still primitive state of IVR, scientists—not surprisingly—generally choose a higher-resolution, nonimmersive, single-wall display over a much lower-resolution immersive display. Our optimism about the use of IVR for scientific visualization is bolstered by the belief that the same high resolution will eventually be available for IVR.

Improve immersion in multiprojector environments. Although large-scale display systems, such as multiprojector tiled and dome-based displays, show promise in providing more pixels to the user's visual field, a number of technological and research challenges remain to be addressed. For example, even though cave technology is more than eight years old, the seams between display walls still have visual discontinuities that can break the illusion of immersion. Large-scale tiled wall and dome displays also have problems with seams. Making images seamless across display surfaces and multiple projectors requires sophisticated image blending algorithms.[22]

We must also continue to explore methods for maintaining projector calibration and alignment, and color and luminosity matching. In addition, with front-projected displays (typical for domes), the user may occlude the projected images when performing various spatial 3D interaction tasks.

Finally, large-scale displays also require higher resolution than is currently possible. To match human acuity, we need to display at least 200 pixels per inch in the circular region with a 0.16 to 0.31 inch radius roughly 18 inches distant in the gaze direction (the region of foveal vision); lower resolution could be displayed outside this region (for example, on portions of the display more distant from the viewer position, and outside the cone of foveal gaze).

A simple calculation assuming a limiting discernable resolution of an arc-minute (typical for daylight grating discrimination[23]) yields a requirement for a conventional desktop display of 2400×1920 pixels—achieved, for example, by IBM's Roentgen active-matrix LCD display. However, for a 10-foot-diameter cave environment, in which today's typical projection systems evenly distribute pixels on a display surface and users can in principle come as close to the walls as they do to a normal monitor, each wall must have $23{,}000 \times 23{,}000$ pixels to achieve the same resolution. Variable-resolution display technology could use many fewer pixels because the pixels could be positioned to best accommodate the human eye.

Human visual acuity also varies with the task. For example, our visual acuity increases dramatically with stereo vision (so-called hyperacuity). For discriminating the relative depth of two lines in stereo, our acuity is 10 to 20 times finer than the value quoted for daylight grating discrimination,[23] resulting in a 10-to-20-fold increase in the numbers quoted here or a 100-to-400-fold increase in the total number of pixels needed (though careful and sophisticated antialiasing could permit coarser resolutions).

Develop office IVR. The history of computing has shown that only a few early adopters will flock to a new technology as long as it remains expensive and fragile, and requires using a lab away from one's normal working environment. IVR will not become a normal component of the scientist's work environment until it literally becomes indistinguishable from that environment.

UNC's ambitious Office of the Future project[24] and its various near-term and longer-term variations are designed to bring IVR to the office in a powerful and yet affordable way. The system is based on a unified application of computer vision and computer graphics. It combines and builds on the notions of the cave, tiled display systems, and image-based modeling. The basic idea is to use real-time computer vision techniques to dynamically extract per-pixel depth and reflectance information for the visible surfaces in the office, including walls, furniture, objects, and people, and then to project images on the surfaces or interpret changes in the surfaces.

To accomplish the simultaneous scene capture and display, computer controlled cameras and projectors replace ceiling lights. By simultaneously projecting images and monitoring geometry and reflectivity of the designated display surfaces, we can dynamically adjust for geometric, intensity, and resolution variations resulting from irregular and dynamic display geometries, and from overlapping images.

The projectors work in two modes: scene extraction (in coordination with the cameras) and normal display. In the scene extraction mode, 3D objects within each camera's view are extracted using imperceptible structured light techniques, which hide projected patterns used for scene capture through a combination of time-division multiplexing and light cancellation techniques. In display mode, the projectors display high-resolution images on designated display surfaces.[25] Scene capture can also be achieved passively and in real time[26] using a cluster of cameras and view-independent acquisition algorithms based on stereo matching.

Ultimately, such an office system will lead to more compelling and useful systems for shared telepresence and telecollaboration between distant individuals. It will enable rich experiences and interaction not possible with the through-the-window paradigm.

Improve rendering performance and flexibility. Although display processors have become fast, we still need additional orders of magnitude in performance. This problem increases in the context of tiled displays on a single wall or multiple walls—we have megapolygons-per-second rates, while we need gigapolygons-per-second, not to mention texels, voxels, and other display primitives.

A question pertaining to both hardware and software is how rendering can take advantage of the full capabilities of the human visual and cognitive systems. Examples of this concept include rendering with greater precision where the eye is focusing and with less detail in the peripheral vision, and rendering so as to emphasize the cues most important for perception and discrimination.[27]

The field of rendering remains in flux as new techniques such as perceptually based rendering, volumetric rendering, image-based rendering, and nonphotorealistic rendering[28] join our lexicon. (Volume-rendering hardware—particularly important in scientific visualization—is specialized now in dedicated chips like Mitsubishi's VolumePro, whose output goes into a conventional polygon graphics pipeline as texture maps.) Current systems don't integrate all these rendering techniques completely. (For example, the Visualization ToolKit can integrate 2D, 3D polygonal, volumetric, and texture-based approaches, but not interactively for all situations, such as when translucent geometric data is used.[29]) Integrating all these techniques into a common framework is difficult, both at the API level, so that the application programmer can mix and match as suits the occasion, and at the system level, where the various types of data must be efficiently rendered and merged. Especially at the hardware level, this will require considerable redesign of the conventional polygon-centric graphics pipeline. Furthermore, visual rendering needs to be coordinated with audio rendering and haptic rendering.

Use parallelism to speed up rendering. Parallel rendering aims to speed up the rendering process by decomposing the problem into multiple pieces that can execute in parallel. We need to learn how to make scalable graphics systems by ganging together commodity processors and graphics components in the same way as high-performance parallel computers are built by ganging together commodity processors. Projects at Stanford, Princeton, Argonne National Lab, and other labs use this strategy to create tiled displays.[6] Some groups are also experimenting with special-purpose hardware such as compositors. For example, Stanford has built the experimental, distributed, frame-buffer architecture Lightning2.

Parallel rendering is a less well studied problem than parallel numerical computing, except for the embarrassingly parallel problem of ray tracing, which has been well studied.[30] While multiple parallel rendering approaches are known (object or image space partitioning,[31] for example), vendors haven't committed to any standard scalable parallel approach. Furthermore, the typical goal of parallel computation is to increase batch throughput, while the goal for IVR is to maximize interactivity. For IVR this is accomplished by minimizing latency and maximizing frame rate (discussed later).

Another way of looking at it is that scientific computing is most interested in asymptotic performance, whereas IVR is most interested in worst-case performance on a subsecond time-scale. Parallel rendering has much in common with parallel databases (for example, efficient distribution and transaction mechanisms are needed), but is focused at the hardware level (for example, the memory subsystem and the rendering pipeline).

Whitman[31] proposed the following criteria for evaluating parallel graphics display algorithms:

- granularity of task sizes
- nature of algorithm decomposition into parallel tasks
- use of parallelism in the display algorithm without significant loss of coherence
- load balancing of tasks
- distribution and access of data through the communication network
- scalability of the algorithm on larger machines

An important requirement is making a parallel rendering system easy to use—that is, isolating the application programmers from the complexities of driving a parallel rendering system. Hereld et al.[32] stated that "the ultimate usability of tiled display systems will depend on the ease with which applications can be developed that achieve adequate performance." They suggested that existing applications should run without modification, that the details of rendering should be hidden from the user, and that new applications should have simple mechanisms to exploit advanced properties of tiled displays.

WireGL[33] is an example of a system that makes it easy for the user and application programmer to leverage scalable rendering systems. Specifically, an OpenGL program needs no modification to run on a tiled display. The distributed OpenGL (DOGL) system at Brown University is a similar, although less optimized, library—it requires minor modification to an OpenGL program, but can drive a tiled, head-tracked, stereo cave display. It uses MPI, a message-passing interface common in parallel scientific computing applications. These and other systems work by intercepting normal (nonparallel) OpenGL calls and multicasting them, or channeling them to specific graphics processors.

WireGL includes optimizations to minimize network traffic for a tiled display and, for most applications, provides scalable output resolution with minimal performance impact. The rendering subsystem relies not only on parallelism, but also on traditional techniques for geometry simplification, such as view-dependent culling. (We discuss geometry simplification techniques later.)

To run a sequential OpenGL program written for a conventional 3D display device without modification requires that the interceptor deal with the complexities of managing head-tracking and stereo. In particular, this requires transformations that conflict with those in the original OpenGL code. Furthermore, the application programmer must provide any additional interaction devices needed. Retaining the simplicity of the WireGL approach (running an OpenGL application in an IVR environment without modification) while extending its capabilities to a wider range of display types is an open problem.

Make haptics useful for interaction and scientific visualization. Haptic output devices need significant improvements before they can become generally useful. Commodity haptic devices, including the Phantom, the Wingman Force Feedback Mouse, and

a number of gaming devices such as the Microsoft Sidewinder Force Feedback joystick and wheel, deliver six degrees of freedom at best; handling objects requires many more degrees of freedom to distribute forces more widely, such as over a whole hand.

Probably the most compelling use of force feedback today (besides in games and certain kinds of surgical training) comes from UNC's nanomanipulator project, where a natural mapping takes place between the atomic forces on the tip of the probe and what the Phantom can provide.

In addition to force displays that emphasize the force itself, the other major kind of haptic display is tactile.[34] Tactile feedback has been simulated with vibrating components such as Virtual Technologies' CyberTouch Glove, pin arrays, and robot-controlled systems that present physical surfaces at the appropriate locations, like those in Tachi's lab at the University of Tokyo (http://www.star.t.u-tokyo.ac.jp/), but these techniques are either experimental or unable to span a sufficient range of tactile sensations.

Earlier molecular docking experiments showed that a robot-arm force-feedback device could significantly speed up the exploration process.[35] Feeling forces, as in electromagnetic or gravitational fields, can also aid visualization.[35] Haptics used in a problem-solving environment for scientific computing called SCIRun lets the user feel vector fields.[36]

Because of their small working volume, today's commodity haptic rendering devices better suit Fish Tank VR than walk-around environments like the cave, let alone larger spaces instrumented with wide-area trackers. There are three basic approaches to haptics in such larger spaces:

- make a larger ground- or ceiling-based exoskeleton with a larger working area,
- put a desktop system such as a Phantom on a pedestal and move it around, or
- ground the forces in a backpack worn by the user,[34] for example to apply forces by pulling on wires attached to the fingers.[37]

None of these options is ideal. The first can be expensive and involve lesser fidelity and greater possibility of hitting bystanders in the device's working volume. The second is clumsy and may cause visual occlusion problems, and the third makes the user carry around significant extra weight.

Besides distributed and mobile haptics, we list two additional research problems:

- Developing real-time haptic rendering algorithms for complex geometry. Forces must be computed and created at high rates (on the order of 1,000 Hz) to maintain a realistic sensation of even the simplest solid objects.
- Actually using haptics for interaction, not just doing haptic rendering. Miller and Zeleznik described a recent example of adding haptics to the 2D user interface,[38] but only a much smaller effort has begun on the general problem of what should be displayed haptically in general 3D environments, in addition to the surfaces of objects.[39]

Make interaction comfortable, fast, and effective. Several options tackle the problem of making interaction better: improving the devices, minimizing system latency, maximizing frame rate, and scaling interaction techniques. We consider each in turn.

Improve interaction device technology. Any input device provides a way for humans to communicate with computer applications. A major distinction can be made, however, between traditional desktop input devices, such as the keyboard and mouse, and post-WIMP input devices for IVR applications. In general, traditional desktop input devices have both a level of predictability that users trust and good spatiotemporal resolution, accuracy, and repeatability. For example, when a user manipulates a mouse, hand motions typically correspond directly to cursor movement, and the control-to-display ratio (the ratio of physical mouse movement to mouse pointer movement) is set in a useful mapping from mouse to display.

In contrast, many IVR input devices, such as 3D trackers, often exhibit chaotic behavior. They lack good spatiotemporal resolution, range, accuracy, and repeatability, not to mention their problems with noise, ergonomic comfort, and even safety (in the case of haptics). For example, a 2D mouse has good accuracy and repeatability regardless of whether the mouse pointer is at the center or at the edges of the screen, but a 3D mouse has adequate accuracy and repeatability only in the center of its tracking range, deteriorating as it moves towards the boundaries of this range. IVR input devices thus frequently have a level of unpredictability that makes them frustrating and difficult to use. Although the HiBall Tracker,[40] originally developed at UNC and now available commercially from 3rdTech, has extremely low latency, and high accuracy and update rates, it's too big for anything other than head tracking. Indeed, miniaturization of tracking devices is another important technological challenge.

Position and orientation tracking is a vital input technology for IVR applications because it lets users get the correct viewing perspective in response to head motion. In most IVR applications, the user's head and either one or both hands are tracked. However, to go beyond traditional IVR interaction techniques, we need to track very precisely other parts of the body such as the fingers (not just fingertips), the feet, pressure on the floor that changes as a user's weight shifts, gaze direction, and the user's center of mass. With the standard tracking solutions, the user would potentially have not just two or three tethers, but perhaps 10 to 15, which is completely unacceptable.

Another example of this tethering problem is the current IVR configuration at the Brown University Virtual Environment Navigation Lab. An HMD and wide-area tracking system (40×40 feet) enables scientists to perform IVR-based navigation, action, and perception experiments. An assistant must accompany the human participant to manage the tethered cables for HMD video

and tracking. Although wireless tracking solutions are commercially available, for instance the Polhemus Star Trak, they have a limited working volume and require the user to somehow carry signal transmitting electronics.

An alternate method of tracking employs computer vision-based techniques so that users can interact without wearing cumbersome devices. This type of tracking is commonly found in perceptual user interfaces (PUIs), which work towards the most natural and convenient interaction possible by making the interface aware of user identity, position, gaze, gesture, and, in the future, even affect and intent.[41] Vision-based tracking has a number of drawbacks, however. Most notably, even with multiple cameras, occlusion is a major obstacle. For example, it's difficult to track finger positions when the hands are at certain orientations relative to the user's body. The form factor of an IVR environment can also play a role in vision-based tracking—consider using camera-based tracking in a six-sided cave without obstructing the visual projection. Vision-based tracking systems are commercially available (for example, the MotionAnalysis motion capture system), but they're mostly used in motion-capture applications for animation and video games. Like the wireless tracking technologies from Polhemus and Ascension, these vision-based systems are expensive, and they also require users to wear a body suit.

In addition to researching robust, accurate, and unobtrusive tracking that scientists can use easily, we must continue to develop other new and innovative interaction devices. Many input devices are designed as general-purpose devices, but, as with displays, one size doesn't fit all. We need to develop specific devices for specific tasks that leverage a user's learned skills. For example, flight simulators don't use instrumented gloves that sense joint angles or finger contact with a virtual cockpit, but instead recreate the exact physical interface of a real cockpit, which is manipulated with the hands. Physical hand-held props are also useful devices for some task- and application-specific interactions (see the sidebar "Interaction in Virtual Reality" for specific examples).

Another task-specific device is the Cubic Mouse,[42] an input device designed for viewing and interacting with volumetric data sets. Intended to let users specify 3D coordinates intuitively, the device consists of a box with three perpendicular rods passing through the center and buttons for additional input. The rods represent the *x*-, *y*-, and *z*-axes of a given coordinate system. Pushing and pulling the rods specifies constrained motion along the corresponding axes. Embedded within the device is a 6DOF tracking sensor, which permits the rods to be continually aligned with a coordinate system located in a virtual world.

Other forms of interaction device technology still need improvement as well. For example, speech recognizers remain underdeveloped and suffer from accuracy and speed problems across a range of users. Motion platforms such as treadmills and terrain simulators are expensive, clumsy, and not yet satisfactory.

A main interface goal in IVR applications is to develop natural, human-like interaction. Multimodal interfaces, which combine different modalities such as speech and gesture, are one possible route to human-to-human style interaction. To create robust and powerful multimodal systems, we need more research on how multiple sensory channels can augment each other. In particular, we need better sensor fusion "unification" algorithms[43] that can improve probabilistic recognition algorithms through mutual reinforcement.

Minimize system latency. According to Brooks,[1] "end-to-end system latency is still the most serious technical shortcoming of today's VR systems." IVR imposes far more stringent demands on end-to-end latency (including both command specification and completion) than a desktop environment. In the latter, the major concern is task performance, and the general rule of thumb is that between 0.1- and 1.0-second latency is acceptable.[44] In IVR we're concerned not only with task performance but also with retaining the illusion of IVR—and more importantly, not causing fatigue, let alone cybersickness. It's useful to think of a hierarchy of latency bounds:

- Some human subjects can notice latencies as small as 16 to 33 ms.[45]
- Latency doesn't typically result in cybersickness until it exceeds a task- and user-dependent threshold. According to Kennedy et al., any delay over 35 ms can cause cue conflicts (such as visual/vestibular mismatch).[46] Note that motion sickness and other discomforts related to cybersickness can occur even in zero-latency environments (like the real world, which effectively has no visual latency) because cue conflicts can result from factors other than latency.
- For hand-eye motor control in navigation and object selection and manipulation, latency must remain less than 100 to 125 ms.[46] For visual display, latency requirements are more stringent for head tracking than for tracking other body parts such as hands and feet. The HiBall tracking system[40] uses Kalman filter-based prediction-correction algorithms to increase accuracy and to reduce latency.
- Permissible latency until the result of an operation appears (as opposed to feedback while performing it) is a matter of the user's expectations and patience. For example, a response to a data query ranging from subseconds to a few seconds may be adequate. Conversely, if time-series data produced by sampling either simulation or observational data plays back too rapidly, the user may become confused. The problem can worsen if the original sampling rate is too low and temporal aliasing occurs. (For the purpose of discussion, we assume the time to render a query's result doesn't affect frame rate—not always the case in practice.)
- Latency affects user performance nonlinearly; when latencies exceed 200 ms, user interaction strategies tend to change to "move and wait" from more continuous and fluid control.[47]

In IVR the dominant criterion is that the overall system latency must fall below the cybersickness threshold. Latency per se doesn't cause cybersickness. However, in the context of IVR, where the coupling between body tracking (especially of the head) and dis-

play is so tight, latency can cause cybersickness. In addition, minimizing latency is important to make it easier for users to transition from performing tasks at the cognitive level to the perceptual level (that is, using muscle memory to perform tasks). Worse, in the context of variable latency, it's even more difficult to make this transition. It's still a significant research problem to identify and especially to minimize all the possible causes of latency at the tracking device, operating-system, application, and rendering levels[48] for a local system, let alone for one whose components are networked, as discussed next.

Unfortunately, latency thus poses a system-wide problem in which no single cause is fatal in itself but the combination produces the equivalent of "the death of a thousand cuts." Singhal and Zyda[49] pointed out that latency is one of the biggest problems with today's Internet, yet has received relatively little attention. Networks introduce uncontrollable, indeed potentially unpredictable, time delays. The causes are numerous, including travel time through routers to and from the network, through the network hardware, operating system, and into the application.

Research in modern network protocols deals with minimizing and guaranteeing maximum network latency, and other aspects of quality-of-service management, but clearly a lower bound exists on network latency due to the speed of light. Since latency in network transmission is inevitable, we need strategies for making the user experience acceptable. Some strategies for masking network latency include using separate time frames and filters for each participant with periodic resynchronization when more accurate information becomes available,[50] and visual effects allowing immediate interaction with proxies at each location and subsequent resynchronization.[51]

Unfortunately, the networking research community isn't well aware of the needs of the real-time graphics community, let alone the far more demanding IVR community. Conversely, the graphics community is insufficiently aware of ongoing developments in networking. An urgent need exists to bring these communities together to come up with acceptable algorithms. However, no matter how good these networking algorithms turn out to be, some IVR tasks are clearly inappropriate for today's networks. Head tracking, for example, isn't a candidate for distributed processing.

Maximize frame rate to meet application needs. Closely related to latency is frame rate—the number of distinct frames generated per second. While most sources quote a rate of 10 Hz as the minimum for IVR, the game community has long known that for fluid interactivity and minimal fatigue 60 Hz is a more acceptable minimum. Worse, given the complexity of scenes that VR users want to generate and the fact that stereo automatically doubles rendering requirements, attaining an acceptable frame rate for real smoothness in interactivity will be an ongoing battle, even with faster hardware. While developers often switch to using lower resolutions in stereo to compensate for the extra computation time required to render left and right eye images, this is hardly a solution, since it compromises the already-inadequate resolution to gain frame rate.

The comparison with the requirements of video games also doesn't take into account the inordinate investment of time by game designers in constructing and tuning their geometry and behavior models to achieve high frame rates, an investment typically impossible for IVR applications. Nonetheless, some of the geometry simplification techniques used in the game community can transfer to IVR, such as level-of-detail management and view-dependent culling. These techniques were, in fact, first developed for IVR walkthroughs. We review them briefly below.

A distinction separates simulation frame update rate and visualization frame update rate. Many scientific simulations will probably never be fast enough to overwhelm the human visual system. However, animated sequences derived from a sequence of simulation time steps may need to be slowed down so that viewers don't miss important details. (Think of watching a slow-motion replay of a close play in a baseball game because the normal—or fastest—playing speed is inappropriate.)

Vogler and Metaxas[52] showed that applications with a lot of motion (such as decoding American Sign Language) require more than a 60-Hz update frame rate to capture all the information. It's unlikely that many scientific visualization situations will require such a stringent update rate, since typically the scientist can control the dynamic visualization's speed (slow motion playback or fast-forward). Of course, actually providing these capabilities in real time may be a formidable problem, given the large size of some data sets.

Finally, the frame rate and system latency often change over time in IVR applications, which can play a significant role in user performance. Watson et al.[53] showed that for an average frame rate of 10 fps, 40 percent fluctuations in the frame rate about the mean can degrade user performance. However, the same experiment showed that for an average frame rate of 20 fps, no significant degradation occurred.

Scale up interaction techniques with data and model size. To combat the accelerating data crisis, we need not only scalable graphics and massive data storage and management systems, but also scalable interaction techniques so that users can view and manipulate the data. Existing interaction techniques don't scale well when users visualize massive data sets. For example, the state-of-the-art interaction techniques for selection, manipulation, and application control described in the sidebar "Interaction in Virtual Reality" are mostly proof-of-concept techniques designed for and tested only with small sets of objects. Techniques that work well with 100 objects in the scene won't necessarily work with 10,000, let alone 100,000. Consider, for example, selecting one out of 100 objects versus selecting one out of 100,000 objects. If culling techniques are used for object selection, some objects of interest may not even be visible, and some kind of global context map may be needed. Unfortunately, IVR remains in an early stage of development, and performance limitations are one of the main reasons that our interaction techniques are based

on small problems. We must either modify existing interaction techniques or develop novel techniques for handling massive data sets.

In considering how we interact with data during an exploration, we must consider all the various steps in a typical process: sensing or computing the raw data, culling a subset to visualize, computing the presentation (the visualization itself), and the interaction techniques themselves. Below we discuss only the last three steps. A more in-depth discussion on managing gigabyte data sets in real time appears elsewhere.[54] Here we discuss various techniques for cutting down the size of the data set, automatic feature extraction techniques, a class of algorithms known as time-critical computing that compute on a given time budget to guarantee a specified frame rate, and the high-performance demands for computing the visualization itself.

A standard approach to interacting with massive data sets is first to subset, cull, and summarize, then to use small-scale interaction techniques with the more manageable extracts from the original data. Probably our best examples of this type of approach are vehicle and building walkthroughs. These require a huge amount of effort in real-time culling and geometry simplification, but the interaction is fairly constrained. While the underlying geometry of, say, a large airplane or submarine may be defined in terms of multiple billions of polygons, only 100,000 to a million polygons may be visible from a particular point of view.

After semantically filtering the data (for example, displaying only HVAC or hydraulic subsystems), we must use a number of approximation techniques to best present the remaining geometry. Nearby geometry may be rendered using geometric simplification, typified by Hoppe's progressive meshes,[55] while distant geometry may be approximated using a variety of image-based techniques, such as texture mapping and image warping. Objects or scenes with a high degree of fine detail approaching one polygon per pixel may benefit from point cloud or other volumetric rendering techniques. UNC's Massive Model Rendering System offers an excellent example of a system combining many of these techniques.[56] Rendering may be further accelerated through hardware support for compression of the texture or geometric data[57] to reduce both memory and bandwidth needs.

Similarly, a scientific data set size can be reduced to allow visualization algorithms to operate on a relevant subset of the full data set if additional information can be added to the data structure. Two techniques for reducing the data set on which visualization algorithms operate are

- precomputation of data ranges via indices so that pieces can quickly be identified at runtime, and
- spatial partitioning for view-dependent culling.

Data compression and multiresolution techniques are also commonly used for reducing the size of data sets. However, many compression techniques are lossy in nature; that is, the original data cannot be reconstructed from the compressed format. Although lossy techniques have been applied to scientific data, many scientists, especially in the medical field, remain skeptical—they fear losing potentially important information.

Bryson et al. pointed out that there's little point in interactively exploring a data set when a precise description of a feature can be used to extract it algorithmically.[54] Feature extraction tends to reduce data size because lower-level data is transformed into higher-level information. Some examples of higher-level features that can be automatically detected in computational fluid dynamics data are vortex cores,[58] shocks,[59] flow separation and attachment,[60,61] and recirculation.[62] We need many more techniques like these to help shift the burden of pattern recognition from human to machine and move toward a more productive human-machine partnership.

Time-critical computing (TCC) algorithms also prove useful for interacting with large-scale data sets.[63] TCC techniques[64,65,54] guarantee some result within a time budget while meeting user-specified constraints. A scheduling algorithm balances the cost and benefit of alternatives from the parameter space (such as time budget per object, algorithm sophistication, level-of-detail, and so forth) to provide the best result in a given time. TCC suits many situations; here we discuss how visualization techniques can benefit from TCC.

Two important challenges face TCC: determining the time budgets for the various visualizations—a type of scheduling problem—and deciding which visualization algorithms with which parameter settings best meet that budget. In traditional 3D graphics, time budgets often depend on the position and size of an object on the screen.[63] In scientific visualization applications, the traditional approach doesn't work, since it can take substantial time to compute the visualization object's position and size on the screen, thereby defeating the purpose of the time-critical approach. A way around this problem is to have equal time budgets for all visualization objects, possibly augmented by additional information such as explicit object priorities provided by the user.

Accuracy versus speed tradeoffs can also help in computing visualization objects within the time budget. Of course, sacrificing accuracy to meet the time budget introduces errors. Unfortunately, we have an insufficient understanding of quality error metrics that would allow us to demonstrate that no significant scientific information has been lost. Metrics such as image quality used for traditional graphics don't transfer to scientific data.[54]

Transforming data into a visualization takes visualization algorithms that depend on multiple resources. In many cases, the time required to create visualizations creates a significant bottleneck. For example, an analysis of the resources used by visualization algorithms reveals that many techniques (such as streamlines, isosurfaces, and more sophisticated algorithms) can result in severe demands on computation and data access that can't be handled in real time. These resources include many components—raw computation, memory, communication, data queries, and display capabilities, among others.

To help illustrate the situation, consider part of Bryson's 1996 analysis[11] based on just the floating-point operation capabilities of a typical 1996 high-performance workstation. He showed that you could expect

25 streamlines to be computed in one tenth of a second. Bryson's assumptions for this result were instantaneous data set access, second-order Runge-Kutta integration, a single integration of a streamline required about 200 floating-point operations, a 20 megaflop machine, and most systems perform about half their rated performance. Today, the same calculation scaled up by Moore's Law would yield about 130 streamlines. However, when you consider the cost of data access and today's very large data sets, these performance figures drop substantially to perhaps tens of streamlines, or even fractions of a single streamline, depending on the data complexity and algorithms used. We need a balanced, high-performance computational system in conjunction with efficient algorithms, with both designed to support interactive exploration.

Many approaches to large data management for visualizing large data sets operate on the output of a simulation. An alternative approach would investigate how ideas used to produce the data (that is, ideas in the simulation community) might be leveraged by the visualization community. At Brown University we're exploring the use of spectral element methods[66] as a data structure for both simulation and visualization. Their natural hierarchical and continuous representation are indeed desirable traits for visualization of large data sets. Nonetheless, we see significant research issues in discovering how best to use them in visualization, minimizing the loss of important detail and accuracy, and visualizing high-order representations.

Spectral elements combine finite elements, which provide discretization flexibility on complex geometries, and spectral methods using higher-order basis functions, which provide excellent convergence properties. A key benefit of spectral elements is that they support two forms of refinement: allowing the size and number of elements to be changed, and allowing the order of polynomial expansion to be changed in a region. Both can occur while the simulation is running. This attribute can be leveraged to perform simulation steering and hierarchical visualization. As in the use of wavelets, you could choose the necessary level of summarization or zoom into a region of interest. Thus you could change the refinement levels to increase resolution only where needed while running a simulation.

Using scalable interfaces means not just interacting with more objects, but also interacting with them over a longer period of time. Small enough problems or a reduced-complexity version of a large problem (with a coarser spatio-temporal resolution) may run in real time, but larger simulations won't be able to engage the user continuously. One interesting aspect of the system proposed by de St. Germain et al.[67] is the "detachable" user interface, which lets the user connect to a long-running simulation to monitor or steer it. This type of interface allows the user to dynamically connect to and disconnect from computations for massive batch jobs. This system also lets multiple scientists steer disjoint parts of the simulation simultaneously.

Facilitate telecollaboration. The problems of interaction multiply when multiple participants interact with a shared data set and with one another over a network. Such collaboration—increasingly a hallmark of modern science and engineering—is poorly supported by existing software. The considerable literature in computer-supported collaborative work primarily concerns various forms of teleconferencing, shared 2D whiteboards, and shared productivity applications; very little literature deals with the special problems of 3D, let alone of teleimmersion, which remains embryonic. Singhal and Zyda[49] detailed many of the network-related challenges involved in this area.

An interesting question is how participants share an immersive 3D space, especially one that may incorporate sections of their physical offices acquired and reconstructed via real-time vision algorithms.[26] A special issue in representing remote scenes is the handling of participant avatars—an active area of research, particularly for the subcase of facial capture. Participants want to see not just their collaborators, but especially their manipulation of the data objects. This may involve avatars for hands, virtual manipulation widgets, explanatory voice and audio, and so on. As in 2D, issues of "floor control" arise and are exacerbated by network-induced latency. Collaborative telehaptics present extremely difficult problems, especially because of latency considerations, and have scarcely been broached.

Other interesting questions involve how to handle annotation and metadata so that they're available and yet don't clutter up the scene, and how to record visualization sessions for later playback and study. In addition to the common interaction space, participants must also have their own private spaces where they can take notes, sketch, and so on, shielded from other participants. In some instances, small subgroups of participants will want the ability to share their private spaces.

An excellent start on many of these questions appears in Haase's work,[16] and in the Cave6D and TIDE (TeleImmersive Data Explorer)[68] projects at EVL. In particular, TIDE has demonstrated distributed interactive exploration and visualization of large scientific data sets. It combines a centralized collaboration and data-storage model with multiple processes designed to allow researchers all over the world to collaborate in an interactive and immersive environment. In Sawant et al.[68] the researchers identified annotation of data sets, persistence (to allow asynchronous sessions), time-dependent data visualization, multiple simultaneous sessions, and visualization comparison as immediate research targets.

In a telecollaboration setting with geographically distributed participants, one strategy Bryson and others adopt is that architectures must not move raw data, but instead move extracts. Extracts are the transmitted visual representation of a structure computed from a separate computation "close to the raw data." In other words, raw data doesn't move from the source (such as a running simulation, processor memory, or disk); rather, the result of a visualization algorithm is transmitted to one or more recipients. The extracts strategy also provides a mechanism for decoupling the computation of visualization objects and rendering, which is essentially a requirement for interactive visualization of very large data sets.

The visualization server the US Department of Energy (DOE) expects to site at Lawrence Livermore, attached to the 14 teraop "ASCI White" machine, offers an example of the use of extracts. This server will compute extracts for all users, but do rendering only for local users. One option being explored for remote visualization is to do some form of partial rendering, such as producing RGBZ images or multilayered images that can be warped in a final rendering step at the remote site. Such image-based rendering techniques can effectively mask the latency of the wide-area network and provide smooth real-time rotation of objects, even if the server or network can handle only a few frames per second.

Cope with lack of standards and interoperability. A Tower of Babel problem afflicts not just graphics but especially the virtual environments community—a situation about which Jaron Lanier, a pioneer of modern IVR, is vocal. Lanier laments that the job of building virtual environments, already hard enough, is made even harder by the lack of interoperability—no mechanism yet exists by which virtual worlds built in different environments can interoperate.

Dozens of IVR development systems exist or are currently under development at the OpenGL and scene-graph level. Unlike the early 1990s, when the most high-performance and robust software was available only commercially and ran on expensive workstations, currently multiple open-source efforts are available for many platforms. Nevertheless, the existence of so many options addressing similar problems hinders progress. In particular, code reuse and interoperability prove very difficult.

This is a general graphics problem, not just an IVR problem. The field has been plagued since its inception by the lack of a single, standard, cross-platform graphics library that would permit interoperability of graphics application programs. Among the proto-standards for interactive 3D graphics are the oldest and most established SGI-led OGL, as well as newer packages such as Microsoft's DirectX, Sun's Java3D, Sense 8's World ToolKit, W3C's X3D (previously called VRML), and UVA/CMU's Alice. Only a few of these support IVR, and many only the bare essentials, such as projection matrices for a single-pipe display engine. Multiple efforts may be the only path in the short term to find a best of breed, but ultimately a concerted effort is needed.

Interoperability is especially important in immersive telecollaboration, in which collaborators share a virtual space populated potentially by their own application objects and avatars. The problem isn't just one of sharing geometry; it also concerns interoperability of behaviors and especially interaction techniques, intrinsically very difficult problems.

One simple but partial solution conceived by Lanier and implemented under sponsorship of Advanced Network & Services, whose teleimmersion project he directs, is Scene Graph as Bus (SGAB).[69] This framework permits writing applications to different scene graphs to interoperate over a network. This approach also captures behavior and interaction because object attribute modifications such as transformation changes are observed and distributed on the bus.

Other systems have been built to allow interoperability of 3D applications, but most assume homogeneous client software[70] or translate only between two scene graphs (without the generality of SGAB). The Distributed Interactive Simulation (DIS)[71] and High-Level Architecture (HLA)[72] standards, for example, make possible cooperation between heterogeneous clients, but only as long as they follow a set of network protocols. The SGAB approach instead attempts to bridge the informational gap between independently designed standalone systems, with minimal or no modification to those systems. More research needs to be done to create interoperability mechanisms as long as no single networked, component-based development and runtime environment standard is in sight.

For scientific visualization in IVR, rendering and interaction libraries are only the beginning. A whole layer of more discipline-specific software must be built on top. A number of such scientific visualization toolkits exist or are under development (such as VTK,[73] SCIRun, EnSight Gold, AVS, Open DX, Open RM Scene Graph). As with lower-level libraries, no interoperability exists between these separate software development and delivery platforms, and users must choose one or another. In some cases, conversion routines may be used to link individual packages, but this solution becomes inefficient as problem size increases.

One group beginning to address the problem, the Common Component Architecture Forum (http://www.acl.lanl.gov/cca-forum), plans to define a common-component software architecture approach to allow more interchanging of modules. While not yet focused on visualization, the group has explored how the same approaches can apply to visualization software architectures.

Develop a new design discipline and validation methodology. IVR differs sufficiently from conventional 3D desktop graphics in terms both of output and interaction that it must be considered not just an extension of what has come before but a medium in its own right. Furthermore, as a new medium, new metrics and evaluation methodologies are required to determine which techniques are effective and why.

Create a new design discipline for a fundamentally new medium. In IVR, all the conventional tasks (navigation—travel for the motor component and wayfinding for the cognitive component—object identification, selection, manipulation, and so on) typically aren't simple extensions of their counterparts for the 2D desktop—they require their own idioms. Programming for the new medium is correspondingly far more complex and requires a grounding in computer, display, audio, and possibly haptics, hardware, and interaction device technology, as well as component-based software engineering and domain-specific application programming.

In addition, while IVR may provide an experience remarkably similar to the real world, it differs from the real world in significant ways:

- Cybersickness is similar to but not the same as motion sickness, and its causes and cures are different.
- Avatars don't look or behave like human beings (extending the message of the classic cartoon "On the Internet, no one knows you're a dog").
- The laws of Newtonian physics may be augmented or even overturned by the laws of cartoon physics.
- Interaction isn't the same. It's both less (for example, haptic feedback is wholly inadequate) and more, through the suitable use of magic (for example, navigation and object manipulation can mimic, extend, or replace comparable real-world actions).
- Application designers may juxtapose real and fictional entities.

In short, IVR creates a computer-generated world with all the power and limitations that implies.

Furthermore, because so much of effective IVR deals with impedance-matching the human sensorium, designers must have a far better understanding of perceptual, cognitive, and even social science (such as small group interaction for telecollaboration) than the traditionally trained computer scientist or engineer. It's especially important to know what humans are and aren't good at in executing motor, perceptual, and cognitive tasks. For example, evidence indicates that humans have one visual system for pattern recognition and a separate one for the hand-eye coordination involved in motor skills. Interactive visualization techniques must be designed to take into account, if not take advantage of, these separate processing channels.

Finally, since IVR focuses on providing a rich, multisensory experience with its own idioms, we can learn much from other design and communication disciplines that aim to create such experiences. These include print media (books and magazines), entertainment (theater, film and video, and computer games), and design (architectural, user interface, and Web page).

Prove immersion's effectiveness for scientific visualization applications and tasks. One of the most important research challenges is proving, for certain types of scientific visualization applications and tasks, that IVR provides a better medium for scientific visualization than traditional nonimmersive approaches. Unfortunately, this is a daunting task, and very little formal experimentation has tested whether IVR is effective for scientific visualization.

Application-level experimentation is extremely difficult to do for a number of reasons. Experimental subjects must be both appropriately familiar with the domain and the tasks to be performed, and willing to devote sufficient time to the experiment. Simple, single-parameter studies not related to application tasks aren't necessarily meaningful. It's also difficult to define meaningful tasks useful to multiple application areas. From a technological standpoint, the hardware and software may still be too primitive and flaky to establish proper controls for the experiments. Finally, the cost in manpower and money of doing controlled experiments is large, and funding for such studies is lacking.

Although some anecdotal evidence from informal user evaluations and demos supports IVR's effectiveness for certain tasks and applications in scientific visualization, the literature contains only a few examples of quantified user studies in these areas, with mixed results. For example, Arns et al.[74] showed that users perform better on statistical data visualization tasks, such as cluster analysis, in an IVR environment than on a traditional desktop. However, users had a more difficult time performing interaction tasks in an IVR environment.[1] Salzman et al.[75] showed that students were better able to define electrostatics concepts after their lessons in an IVR application than in a traditional 2D desktop learning environment. However, retention tests for these concepts after a five-month period showed no significant difference between the desktop and IVR environments. Lin et al.[17] showed that users preferred visualizations of geoscientific data in an IVR environment but also found the IVR interaction techniques less effective than desktop techniques. Although this study presented statistical results based on questionnaires, it collected no speed or accuracy data, and only a handful of the human participants actually used the system during the experiment.

Hix et al.[76] showed how to use heuristic, formative, and summative evaluations to iteratively design a battlefield visualization VE. Although they didn't try to show that the VE was better than traditional approaches to battlefield visualization, their work was significant in being among the first on developing 3D user interfaces using a structured, user-centered design approach. Salzman et al.[75] used similar user-centered design approaches for their IVR application development.

Challenges in scientific (and information) visualization

To address the challenges facing scientific and information visualization, we can pull from the knowledge in other fields, specifically art, perceptual psychology, and artificial intelligence.

Develop art-motivated, multidimensional, visual representations. The size and complexity of scientific data sets continues to increase, making more critical the need for visual representations that can encode more information simultaneously. In particular, inherently multivalued data, like the examples in the sidebar "Art-Motivated Textures for Displaying Multivalued Data," can in many cases be viewed only one value at time. Often, correlations among the many values in such data are best discovered through human exploration because of our visual system's expertise in finding visual patterns and anomalies.

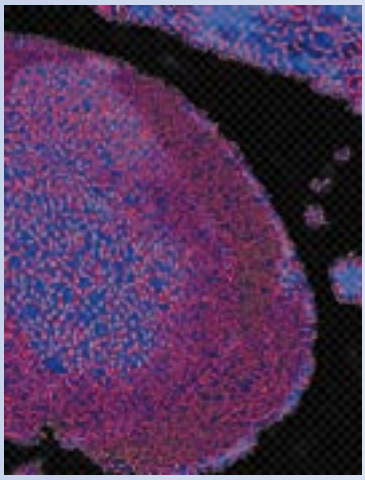
Experience from art and perceptual psychology has the potential to inspire new, more effective, visual representations. Over several centuries, artists have evolved a tradition of techniques to create visual representations for particular communication goals. Inspiration from painting, sculpture, drawing, and graphic design all show potential applicability. The 2D painting-motivated example in the "Art-Motivated Textures" sidebar offers one example. That work is being extended to show multivalued data on surfaces in 3D. These methods use

Art-Motivated Textures for Displaying Multivalued Data

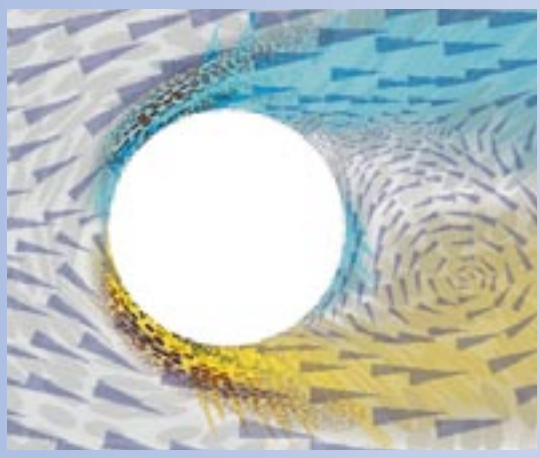
David H. Laidlaw

Concepts from oil painting and other arts can be applied to enhance information representation in scientific visualizations. This sidebar describes some examples developed in 2D. I'm currently extending this work to curved surfaces in 3D immersive environments and facing many of the challenges described in the article.

Examples of methods developed for displaying multivalued 2D fluid flow images[1] appear in Figure J and 2D slices of tensor-valued biological images[2] in Figure K. While the images don't look like oil paintings, concepts motivated by the study of painting, art, and art history were directly applied in creating them. Further ideas gathered from the centuries of artists' experience have the potential to revolutionize visualization.



K 3D tensor-valued data from a slice of a mouse spinal cord. The visualization methods show six interrelated data values at each point of the slice.



J Using art-motivated methods to display multivalued 2D fluid flow around a cylinder at Reynolds number 100. Shown at each point are velocity, vorticity, the rate-of-strain tensor, turbulent change, and turbulent current.

The images are composited from layers of small icons analogous to brush strokes. The many potential visual attributes of such strokes—size, shape, orientation, colors, texture, density, and so on—can all represent components of the data. The use of multiple partially transparent layers further increases the information capacity of the medium. In a sense, an image can become several images when viewed from different distances. The approach can also introduce a temporal aspect into still images, using visual cues that become visible more or less quickly to direct a viewer through the temporal cognitive process of understanding the relationships among the data components.

Figure J, created through a collaboration with R. Michael Kirby and H. Marmanis at Brown, shows simulated 2D flow around a cylinder at Reynolds number 100. The quantities displayed include two newly derived hydrodynamic quantities—turbulent current and turbulent charge—as well as three traditional flow quantities—velocity, vorticity, and rate of strain. Visualizing all values simultaneously gives a context for relating the different flow quantities to one another in a search for new physical insights.

Figure K, created through a collaboration with Eric Ahrens, Russ Jacobs, and David Kremers at Caltech, shows a section through a mouse spinal cord. At each point a measurement of the rate of diffusion of water gives clues about the microstructure of the underlying tissues. The image simultaneously displays the six interrelated values that make up the tensor at each point.

Extending these ideas to display information on surfaces in 3D has the potential to increase the visual content of images in IVR.

References

1. R.M. Kirby, H. Marmanis, and D.H. Laidlaw, "Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting," *Proc. of IEEE Visualization 99*, ACM Press, New York, Oct. 1999, pp. 333-340.
2. D.H. Laidlaw, K.W. Fleischer, and A.H. Barr, "Partial-Volume Bayesian Classification of Material Mixtures in MR Volume Data using Voxel Histograms," *IEEE Trans. on Medical Imaging*, Vol.17, No. 1, Feb. 1998, pp. 74-86.

relatively small, discrete, iconic "brush strokes" layered over one another to represent up to nine values at each point in an image of a 2D data set.[77,78]

Applying these ideas to map multivalued data onto surfaces in 3D comes up against some barriers. First, parts of surfaces may face away from a viewpoint or be

obscured by other surfaces. In an interactive environment, moving an object around can alleviate this problem, but doesn't eliminate it. Second, and perhaps more fundamental, the human visual system can misinterpret the visual properties that represent data values.

Many visual cues can be used to map data. Some of the most obvious are color and texture. Within texture, density, opacity, and contrast can often be distinguished independently. At a finer level of detail, texture can consist of more detailed shapes that can convey information. What makes the problem complex is that the human visual system takes cues about the shape and motion of objects from changes in the texture and color of surfaces. For example, the shading of an object gives cues about its shape. Therefore, data values mapped onto the brightness of a surface may be misinterpreted as an indication of its shape.

Just as brightness cues from shading are wrongly interpreted as shape information, the visual system uses the appearance and motion of texture to infer shape and motion properties of an object. Consider a surface covered with a uniform texture. The more oblique the view of the surface, the more compressed the texture appears. The human visual system is tuned to interpret that change in the density of the texture as an indication of the angle of the surface. Thus any data value mapped onto the density of a texture may be misinterpreted as an indication of its orientation. Texture is also analyzed visually to infer the motion of an object. These shape and motion cues are important both for understanding objects and for navigating through a virtual world, so confounding their interpretation by mapping data values to them carries a risk.

The visual system already "knows" how to interpret the visual attributes that we "know" how to map our data onto. Unfortunately, the interpretation doesn't match our intent, and the results are ambiguous. Avoiding these ambiguities requires an understanding of perception and perceptual cues as well as how the cues combine and when they can be discounted. Because stereo and motion are the primary cues for shape, perhaps shading can be overloaded with a different interpretation. Only on a task-by-task basis can hypotheses like this be evaluated.

A third barrier to representing multivalued data with textures is the difficulty of defining and rendering meaningful textures on arbitrary surfaces. Interrante[79] and others have made some excellent progress here, but many unsolved problems in representing details of surface appearance efficiently remain.

Learn from perceptual psychology. Perceptual psychology has the potential to yield important knowledge for scientific visualization problems. Historically, the two disciplines of art history and perceptual psychology have approached the human visual system from different perspectives. Art history provides a phenomenological view of art—painting *X* evokes response *Y*—but doesn't consider the perceptual and cognitive processes underlying the responses. Perceptual psychology investigates how humans understand those visual representations. A gap separates art and perceptual psychology: no one knows how humans combine the visual inputs they receive to arrive at their responses to art.

Shape, shading, edges, color, texture, motion, and interaction are all components of an interactive visualization. How do they interact? And how can they most effectively be deployed for a particular scientific task? We need to look to perceptual psychology for lessons on the effectiveness of our visualization methods. Evaluating this is difficult, because not only are the goals difficult to define and codify, but tests that evaluate them meaningfully are difficult to design and execute. These evaluations are akin to evaluating how the human perceptual system works. Visual psychophysics can help to understand how an observer interprets and acts upon visual information,[80] as well as how an observer combines different sources of information into coherent percepts.[81]

The experience of perceptual psychologists in designing experiments has much to offer, and the study of perception provides several clear directions for data visualization. First, a better understanding of how visual information is perceived will allow the creation of more effective displays. Second, modeling how different types of information combine to create structures will allow the presentation of complex multivalued data. Third, psychophysical methods can be used to assess objectively the effectiveness of different visualization techniques.

Perceptual psychology can also help us to understand limitations of IVR environments and the impacts of those limits. For example, cybersickness is believed to arise from conflicting perceptual cues—our visual system perceives us as moving in one way, but the vestibular system in our inner ears senses no motion. IVR systems today include many causes of discrepancies like this. One is the time lag in visual feedback due to input, rendering, and output latencies. A second is the disagreement between where our eyes focus (on the IVR projection surface) and where a virtual object is located (usually not on the surface).

Visualized data almost always includes some amount of error. IVR's stringent update and latency constraints force the issue of accuracy tradeoffs. A thorough understanding of error is critical to a theory of techniques for scientific visualization. Pang et al.[82] reported that the common underlying problem is visually mapping data and uncertainty together into a holistic view. There is an inherent difficulty in defining, characterizing, and controlling the uncertainty in the visualization process, coupled with a lack of methods that effectively present uncertainty and data.

Transcend human limitations with scalable AI techniques. As technology improves, displays will approach the limits of human visual bandwidth, and data sets will become so large that even with the best resolution and tools we can never look at more than a tiny fraction of them. Even after we finish leveraging and impedance-matching the human visual system optimally, we're still going to hit a brick wall. We need intelligent software to perform human-like pattern

recognition tasks on massive data sets to winnow the potential areas of interest for further human study. We lump such intelligent software under the title of AI. While many researchers remain skeptical of AI's track record, other enthusiasts such as Ray Kurzweil[83] feel that the expected exponential increase in computational power and algorithms together will make significant advances possible—a bet that's clearly fueling data mining projects, for example.

As Don Middleton of the National Center for Atmospheric Research (NCAR) wrote in an e-mail,

> It's not clear the commercial marketplace or the community efforts currently under way will address the visualization and analysis requirements of the largest problems—the terascale problems. It's my own belief that by mid-decade we'll need to be looking very seriously at quasi-intelligent and autonomous analysis agents that can sift through the data volumes on behalf of the researcher.

These kinds of techniques are precisely the focus of the Intelligent Data Understanding component of NASA's new Intelligent Systems Program (http://ic.arc.nasa.gov/ic/nra/).

Scaling, which has always been a problem in adapting AI techniques to real-world needs, is a particularly crucial issue here, especially if we consider the consequences of latency. Thus, we must not only develop the techniques, but also ways of evaluating their scalability to terabyte and petabyte data sets.

Conclusion

It's generally accepted that visualization is key to insight and understanding of complex data and models because it leverages the highest-bandwidth channel to the brain. What's less generally accepted, because there has been so much less experience with it, is that IVR can significantly improve our visualization abilities over what can be done with ordinary desktop computing. IVR isn't "just better" 3D graphics, any more than 3D is just better 2D graphics. Rather, IVR can let us "see" (that is, form a conception of and understand) things we could not see with desktop 3D graphics.

We need to push any means available for making visualization more powerful, because the gap between the size and complexity of data sets we can compute or sense and those we can effectively visualize is increasing at an alarming rate. IVR's potential to display larger and more complex data, to interact more naturally with that data, and possibly to reveal new patterns in the data through the use of our intrinsic 3D perception, navigation, and manipulating skills, is tantalizing. We anticipate increasing interest in both the research agenda and in production uses of IVR for visualization.

However, many barriers block rapid progress. Technology for IVR remains primitive and expensive, and investment for visualization—let alone IVR visualization—both for R&D and for deployment, lags far behind investment in computation and data gathering. Scientific computing facilities typically spend 10 percent or less of their hardware budget on visualization systems. Nonetheless, we see considerable cause for optimism, given the success stories and partial success stories available and the inexorable improvements in hardware, software, and interaction technology. Those of us in the field, sensing the potential, wonder whether we are roughly at the 1903 Kitty Hawk stage of powered flight, with the equivalent of modern jet-airplane travel and same-day package delivery inevitably to come, or whether we are deluded by the late-1930s popular science prediction of a helicopter in every garage.

We do believe that we're seeing the slow and somewhat painful birth of a new medium, although we're far from being at "Stage 3" (see *The Three Stages of a New Medium*, http://www.alice.org/stage3/whystage3.html) of using the new medium to its fullest potential, that is, with idioms unique to it rather than imitating the old ones. Much research is needed to develop visualization and interaction techniques that take proper advantage of the IVR medium. The learning curve for the new medium is far steeper than for 3D, let alone 2D graphics, at least in part because the technology is so much more complex and our lack of knowledge of human perception, cognition and manipulation skills is so much greater a limitation.

The research agenda for progress in using IVR for scientific visualization is long and provides interesting challenges for researchers in many fields, especially in interdisciplinary problems. The agenda includes the traditional research areas of dramatically improving device technology, producing scalable high-performance graphics architectures from commodity parts, and developing ways of significantly reducing end-to-end latency and of coping strategies for unavoidable latency. Among the interesting newer research problems are

- how to do computational steering of exploratory computations and monitoring of production runs, especially in IVR
- how to use art-inspired visualization techniques for showing large multivalued data sets
- how to use our growing knowledge of perceptual, cognitive, and social science to construct effective and comfortable IVR environments and visualizations
- how to do application-oriented user studies that show under what circumstances and for what reasons IVR is or isn't effective

We hope that this article will stimulate serious interest in the research agenda we have highlighted here. Also, we hope that a review of this kind done in a decade will start by listing important scientific discoveries or designs that would not have happened without the production use of IVR-based scientific visualization as an integral part of the discovery or design process. ■

Acknowledgments

This article was truly a group effort, with many colleagues helping in substantive ways. Some contributed sidebars, others submitted detailed comments and suggested fixes that we have shamelessly but gratefully incorporated. Yet others answered time-critical techni-

cal questions with very helpful responses. Any sins of omission or commission are our responsibility.

First, we wish to thank John van Rosendale, with whom Andy van Dam collaborated to produce the keynote addresses that John gave at IEEE Visualization 99 and that Andy gave at IEEE VR 2000. These talks formed the basis for this article. Other major contributers whose text we used literally or paraphrased included Steve Bryson, Sam Fulcomer, Chris Johnson, Mike Kirby, Tim Miller, and Dave Mizell.

Next, we thank those who provided useful feedback or technical information: Steve Ellis, Steve Feiner, Jim Foley, David Johnson, George Karniadakis, Jaron Lanier, Don Middleton, Jeff Pierce, Terri Quinn, Spencer Sherwin, and Colin Ware. Katrina Avery, Nancy Hays, and Debbie van Dam greatly improved the readability of the article.

Finally, we thank our sponsors: NSF, NIH, the ANS National Tele-Immersion Initiative, DOE's ASCI program, IBM, Microsoft Research, and Sun Microsystems.

References

1. F.P. Brooks, Jr. "What's Real About Virtual Reality," *IEEE Computer Graphics and Applications*, Vol. 19, No. 6, Nov./Dec. 1999, pp. 16-27.
2. C. Cruz-Neira, D.J. Sandin, and T.A. DeFanti, "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE," *Proc. ACM Siggraph 93*, Ann. Conf. Series, ACM Press, New York, 1993, pp. 135-142.
3. A. van Dam, "Post-Wimp User Interfaces: The Human Connection," *Comm. ACM*, Vol. 40, No. 2, 1997, pp. 63-67.
4. C. Ware, K. Arthur, and K.S. Booth, "Fish Tank Virtual Reality," *Proc. InterCHI 93*, ACM and IFIP, ACM Press, New York, 1993, pp. 37-42.
5. W. Krüger and B. Fröhlich, "The Responsive Workbench," *IEEE Computer Graphics and Applications*, Vol. 14, No. 3, May 1994, pp. 12-15.
6. "Special Issue on Large Wall Displays," *IEEE Computer Graphics and Applications*, Vol. 20, No. 4, July/Aug. 2000.
7. B.H. McCormick, T.A. DeFanti, and M.D. Brown, *Visualization in Scientific Computing*, Report of the NSF Advisory Panel on Graphics, Image Processing and Workstations, 1987.
8. *Data and Visualization Corridors: Report on the 1998 DVC Workshop Series,* CACR-164, P. Smith and J. Van Rosendale, eds., Center for Advanced Computing Research, California Institute of Technology, Pasadena, Calif., 1998.
9. J. Leigh, C.A. Vasilakis, T.A. DeFanti, "Virtual Reality in Computational Neuroscience," *Proc. Conf. on Applications of Virtual Reality*, R. Earnshaw, ed., British Computer Society, Wiltshire, UK, June 1994.
10. R. Pausch, D.R. Proffitt, and G. Williams, "Quantifying Immersion in Virtual Reality," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, Aug. 1997, pp. 13-18.
11. S. Bryson, "Virtual Reality in Scientific Visualization," *Comm. ACM*, Vol. 39, No. 5, May 1996, pp. 62-71.
12. M. Chen et al., "A Study in Interactive 3D Rotation Using 2D Control Devices," *Computer Graphics*, Vol. 22, No. 4, Aug. 1988, pp. 121-129.
13. C. Ware and J. Rose, "Rotating Virtual Objects with Real Handles," *ACM Trans. CHI*, Vol. 6, No. 2, 1999, pp. 162-180.
14. S. Bryson and C. Levit, "The Virtual Windtunnel," *IEEE Computer Graphics and Applications*, Vol. 12, No. 4, July/Aug. 1992, pp. 25-34.
15. H. Haase, J. Strassner, and F. Dai, "VR Techniques for the Investigation of Molecule Data," *J. Computers and Graphics*, Vol. 20, No. 2, Elsevier Science, New York, 1996, pp. 207-217.
16. H. Haase et al., "How Scientific Visualization Can Benefit from Virtual Environments," *CWI Quarterly*, Vol. 7, No. 2, 1994, pp. 159-174.
17. C.-R. Lin, H.R. Nelson, Jr., and R.B. Loftin. "Interaction with Geoscience Data in an Immersive Environment," *Proc. IEEE Virtual Reality 2000*, IEEE Computer Society Press, Los Alamitos, Calif., 2000, pp. 55-62.
18. M. Tidwell et al., "The Virtual Retinal Display—A Retinal Scanning Imaging System," *Proc. Virtual Reality World 95*, Mecklermedia, Westport, Conn., 1995, pp. 325-333.
19. K. Perlin, S. Paxia, and J.S. Kollin, "An Autostereoscopic Display," *Proc. Siggraph 2000*, Ann. Conf. Series, ACM Press, New York, 2000, pp. 319-326.
20. M. Hereld et al., "Developing Tiled Projection Display Systems," *Proc. IPT 2000 (Immersive Projection Technology Workshop)*, University of Iowa, Ames, Iowa, 2000.
21. M. Agrawala et al., "The Two-User Responsive Workbench: Support for Collaboration through Individual Views of a Shared Space," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, 1997, pp. 327-332.
22. R. Raskar et al., "Multi-Projector Displays using Camera-Based Registration," *Proc. IEEE Visualization 99*, ACM Press, New York, Oct. 1999.
23. *Virtual Reality: Scientific and Technological Challenges*, N.I. Durlach and A.S. Mavor, eds., National Academy Press, Washington, D.C., 1995.
24. R. Raskar et al., "The Office of the Future: A Unified Approach to Image-based Modeling and Spatially Immersive Displays," *Proc. ACM Siggraph 98*, Ann. Conf. Series, ACM Press, New York, 1998, pp. 179-188.
25. G. Bishop and G. Welch, "Working in the Office of the "Real Soon Now," *IEEE Computer Graphics and Applications*, Vol. 4, No. 4, July/Aug. 2000, pp. 88-95.
26. K. Daniilidis et al., "Real-Time 3D Tele-immersion," in *The Confluence of Vision and Graphics*, A. Leonardis et al., eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000, pp. 215-314.
27. J.A. Ferwerda et al., "A Model of Visual Masking for Computer Graphics," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, 1997, pp. 143-152.
28. L. Markosian et al., "Real-Time Nonphotorealistic Rendering," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, Aug. 1997, pp. 415-420.
29. W.J. Schroeder, L.S. Avila, and W. Hoffman, "Visualizing with VTK: A Tutorial," *IEEE Computer Graphics and Applications*, Vol. 20, No. 5, Sep./Oct. 2000, pp. 20-28.
30. S. Parker et al., "Interactive Ray Tracing for Volume Visualization," *IEEE Trans. Visualization and Computer Graphics*, July-Sept. 1999, pp. 238-250.
31. S. Whitman, *Multiprocessor Methods for Computer Graphics Rendering*, Jones and Bartlett Publishers, Boston, 1992.
32. M. Hereld, I.R. Judson, and R.L. Stevens, "Tutorial: Introduction to Building Projection-based Tiled Display Systems," *IEEE Computer Graphics and Applications*, Vol. 20, No. 4, July/Aug. 2000, pp. 22-28.

33. G. Humphreys et al., "Distributed Rendering for Scalable Displays," to appear in *Proc. IEEE Supercomputing 2000*, IEEE Computer Society Press, Los Alamitos, Calif., 2000.
34. G.C. Burdea, *Force and Touch Feedback for Virtual Reality*, John Wiley & Sons, New York, 1996.
35. F.P. Brooks, Jr. et al., "Project Grope—Haptic Displays for Scientific Visualization," *Proc. ACM Siggraph 90*, ACM Press, New York, Aug. 1990, pp. 177-185.
36. L.J.K. Durbeck et al., "SCIRun Haptic Display for Scientific Visualization," *Proc. Third Phantom User's Group Workshop*, MIT RLE Report TR624, Massachusetts Institute of Technology, Cambridge, Mass., 1998.
37. M. Hirose et al., "Development of Wearable Force Display (HapticGear) for Immersive Projection Displays," *Proc. IEEE Virtual Reality 99 Conf.*, IEEE Computer Society Press, Los Alamitos, Calif., 1999, p. 79.
38. T. Miller and R. Zeleznik, "An Insidious Haptic Invasion: Adding Force Feedback to the X Desktop," *Proc. ACM User Interface Software and Technologies (UIST) 98*, ACM Press, New York, 1998, pp. 59-64.
39. T. Anderson, "Flight—A 3D Human-Computer Interface and Application Development Environment," *Proc. Second Phantom User's Group Workshop*, MIT RLE TR618, Massachusetts Institute of Technology, Cambridge, Mass., Oct. 1997.
40. G. Welch et al., "The HiBall Tracker: High-Performance Wide-area Tracking for Virtual and Augmented Environments," *Proc. ACM Symp. Virtual Reality Software and Technology (VRST)*, ACM Press, New York, Dec. 1999.
41. "Perceptual User Interfaces," M. Turk and G. Robertson, eds., special issue of *Comm. ACM*, Vol. 43, No. 3, Mar. 2000.
42. B. Fröhlich et al., "Cubic-Mouse-Based Interaction in Virtual Environments," *IEEE Computer Graphics and Applications*, Vol. 20, No. 4, July/Aug. 2000, pp. 12-15.
43. P.R. Cohen et al., "QuickSet: Multimodal interaction for Distributed Applications," *Proc. ACM Multimedia 97*, ACM Press, New York, 1997, pp. 31-40.
44. J. Nielsen, *Usability Engineering*, Academic Press, San Diego, 1993.
45. S.R. Ellis et al., "Discrimination of Changes of Latency during Voluntary Hand Movement of Virtual Objects," *Proc. of Human Factors and Ergonomics Society (HFES) 99*, Human Factors and Ergonomics Society, Santa Monica, Calif., 1999, pp. 1182-1186.
46. R.S. Kennedy et al., "A Simulator Sickness Questionnaire (SSQ): An Enhanced Method for Quantifying Simulator Sickness," *Int'l J. Aviation Psychology*, Vol. 3, 1993, pp. 203-20.
47. T.B. Sheridan and W.R. Ferrell, "Remote Manipulative Control with Transmission Delay," *IEEE Trans. Human Factors in Electronics (HFE)*, Vol. 4, No. 1, 1963, pp. 25-29.
48. M. Wloka, "Lag in Multiprocessor VR," *Presence: Teleoperators and Virtual Environments*, Vol. 4, No. 1, Winter 1995, pp. 50-63.
49. S. Singhal and M. Zyda, *Networked Virtual Environments: Design and Implementation*, Addison Wesley, Reading, Mass., 1999.
50. P.M. Sharkey, M.D. Ryan, and D.J. Roberts, "A Local Perception Filter for Distributed Virtual Environments," *Proc. IEEE Virtual Reality Ann. Int'l Symp. (VRIAS) 98*, IEEE Computer Society Press, Los Alamitos, Calif., Mar. 1998, pp. 242-249.
51. D.B. Conner and L.S. Holden, "Providing a Low-Latency User Experience in a High-Latency Application," *Proc. ACM Symp. on Interactive 3D Graphics 97*, ACM Press, New York, Apr. 1997, pp. 45-48.
52. C. Vogler and D. Metaxas, "Toward Scalability in ASL Recognition: Breaking Down Signs into Phonemes," *Proc. Gesture Workshop 99*, Springer, Berlin, Mar. 1999.
53. B. Watson et al., *Evaluation of Frame Time Variation on VR Task Performance*, Tech. Report 96-17, College of Computing, Georgia Institute of Technology, Atlanta, Ga., 1996.
54. S. Bryson et al., "Visually Exploring Gigabyte Data Sets in Real Time," *Comm. ACM*, Vol. 42, No. 8, Aug. 1999, pp. 83-90.
55. H. Hoppe, "View-Dependent Refinement of Progressive Meshes," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, Aug. 1997, pp. 189-198.
56. D. Aliaga et al., "MMR: An Integrated Massive Model Rendering System Using Geometric and Image-Based Acceleration," *Proc. ACM Symp. on Interactive 3D Graphics (I3D)*, ACM Press, New York, Apr. 1999, pp. 199-206.
57. M. Deering, "Geometry Compression," *Proc. ACM Siggraph 95*, Ann. Conf. Series, ACM Press, New York, 1995, pp. 13-20.
58. D. Sujudi and R. Haimes, "Identification of Swirling Flow in 3D Vector Fields," *Proc. AIAA (Am. Inst. of Aeronautics and Astronautics) Computational Fluid Dynamics Conf.*, American Institute of Aeronautics and Astronautics, Reston, Va., June 1995, pp. 151-158.
59. P. Walatka et al, *Plot3D User's Manual*, NASA Tech. Memo 101067, Moffett Field, Calif., 1992.
60. J. Helman and L. Hesselink, "Visualizaing Vector Field Topology in Fluid Flows," *IEEE Computer Graphics and Applications*, Vol. 11, No. 3, May/June 1991, pp. 36-40.
61. D. Kenwright, "Automatic Detection of Open and Closed Separation and Attachment Lines," *Proc. IEEE Visualization 98*, ACM Press, New York, Oct. 1998, pp. 151-158.
62. R. Haimes, "Using Residence Time for the Extraction of Recirculation Regions," *Proc. 14th AIAA (Am. Inst. of Aeronautics and Astronautics) Computational Fluid Dynamics Conf.*, Paper No. 97-3291, American Institute of Aeronautics and Astronautics, Reston, Va., June 1999.
63. S. Bryson and S. Johan, "Time Management, Simultaneity, and Time-critical Computation in Interactive Unsteady Visualization Environments," *Proc. IEEE Visualization 96*, ACM Press, New York, 1996, pp. 255-261.
64. T. Funkhouser and C. Séquin, "Adaptive Display Algorithm for Interactive Frame Rates during Visualization of Complex Virtual Environments," *Proc. ACM Siggraph 93*, Ann. Conf. Series, ACM Press, New York, 1993, pp. 247-254.
65. T. Meyer and Al Globus, *Direct Manipulation of Isosurfaces and Cutting Planes in Virtual Environments*, Tech. Report 93-54, Computer Science Dept., Brown University, Providence, R.I., Dec. 1993.
66. G.E. Karniadakis and S.J. Sherwin, *Spectral Element Methods for CFD*, Oxford University Press, Oxford, UK, 1999.
67. J.D. de St. Germain et al., "Uintah: A Massively Parallel Problem Solving Environment," *Proc. of HPDC 00, Ninth IEEE Int'l Symp. on High Performance and Distributed Computing*, IEEE Computer Society Press, Los Alamitos, Calif., Aug. 2000, pp. 33-41.
68. N. Sawant et al., " The Tele-Immersive Data Explorer: A Distributed Architecture for Collaborative Interactive Visualization of Large Data Sets," *Proc. IPT 2000 (Immersive*

Projection Technology Workshop), University of Iowa, Ames, Iowa, June 2000.
69. B. Zeleznik et al., "Scene-Graph-As-Bus: Collaboration Between Heterogeneous Stand-alone 3D Graphical Applications," *Proc. of Eurographics 2000*, Blackwell, Cambridge, U.K., pp. 91-98.
70. G. Hesina et al., "Distributed Open Inventor: A Practical Approach to Distributed 3D Graphics," *Proc. ACM Virtual Reality Software and Technology (VRST) 99*, ACM Press, New York, Dec. 1999.
71. Institute of Electrical and Electronics Engineers, International Standard, ANSI/IEEE Standard 1278-1993, "Standard for Information Technology, Protocols for Distributed Interactive Simulation," IEEE Press, Piscataway, N.J., Mar. 1993.
72. F. Kuhl, R. Weatherly, and J. Dahmann, *Creating Computer Simulation Systems*, Prentice Hall, Upper Saddle River, N.J., 1999.
73. *The Visualization Toolkit, An Object-Oriented Approach to 3D Graphics*, Prentice Hall, Upper Saddle River, N.J., 1997.
74. L. Arns, D. Cook, and C. Cruz-Neira, "The Benefits of Statistical Visualization in an Immersive Environment," *Proc. IEEE Virtual Reality 99*, IEEE Computer Society Press, Los Alamitos, Calif., Mar. 1999, pp. 88-95.
75. M.C. Salzman et al., "ScienceSpace: Lessons for Designing Immersive Virtual Realities," *Proc. ACM Computer-Human Interaction (CHI) 96*, ACM Press, New York, 1996, pp. 89-90.
76. D. Hix et al., "User-Centered Design and Evaluation of a Real-Time Battlefield Visualization Virtual Environment," *Proc. IEEE Virtual Reality 99*, IEEE Computer Society Press, Los Alamitos, Calif., 1999, pp. 96-103.
77. R.M. Kirby, H. Marmanis, and D.H. Laidlaw, "Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting," *Proc. IEEE Visualization 99*, ACM Press, New York, Oct. 1999, pp. 333-340.
78. D.H. Laidlaw, K.W. Fleischer, and A.H. Barr, "Partial-Volume Bayesian Classification of Material Mixtures in MR Volume Data using Voxel Histograms," *IEEE Trans. on Medical Imaging*, Vol. 17, No. 1, Feb. 1998, pp. 74-86.
79. V. Interrante, "Illustrating Surface Shape in Volume Data via Principle Direction-Driven 3D Line Integral Convolution," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, 1997, pp. 109-116.
80. W.H. Warren and D.J. Hannon, "Direction of Self-Motion Is Perceived from Optical Flow," *Nature*, Vol. 335, No. 10, 1988, pp. 162-163.
81. M.S. Landy and J.A. Movshon, *Computational Models of Visual Processing*, MIT Press, Cambridge, Mass., 1991.
82. A. Pang, C.M. Wittenbrink, and S.K. Lodha, *Approaches to Uncertainty Visualization*, Tech. Report UCSC-CRL-96-21, Computer Science Dept., University of California, Santa Cruz, 1996.
83. R. Kurzweil, *The Age of Spiritual Machines: When Computers Exceed Human Intelligence*, Viking, New York, 1999.

Andries van Dam *is Thomas J. Watson, Jr., University Professor of Technology and Education, and professor of computer science at Brown University, where he co-founded the Computer Science Department and served as its first chairman. His research has concerned computer graphics, text processing, and hypermedia systems.*

van Dam co-founded ACM Siggraph and sits on the technical advisory boards of several startups and Microsoft Research. He became an IEEE Fellow and an ACM Fellow in 1994. He received honorary PhDs from Darmstadt Technical University, Germany, in 1995 and from Swarthmore College in 1996. He was inducted into the National Academy of Engineering in 1996 and became an American Academy of Arts and Sciences Fellow in 2000. See his curriculum vitae for a complete list of his awards and publications: http://www.cs.brown.edu/people/avd/long_cv.html.

Contact van Dam at Brown University, Dept. of Computer Science, Box 1910, Providence, RI 02912, e-mail avd@cs.brown.edu.

Andrew Forsberg *is a research staff member in the Brown University Graphics Group. He works primarily on developing 2D and 3D user interfaces and on applications of virtual reality. He received his Masters degree from Brown in 1996.*

David Laidlaw *is the Robert Stephen Assistant Professor in the Computer Science Department at Brown University. His research centers on applications of visualization, modeling, computer graphics, and computer science to other scientific disciplines. He received his PhD in computer science from the California Institute of Technology, where he also did postdoctoral work in the Division of Biology.*

Joseph J. LaViola, Jr.*, is a PhD student in computer science in the Computer Graphics Group and a Masters student in applied mathematics at Brown University. He also runs JJL Interface Consultants, which specializes in interfaces for virtual reality applications. His primary research interests are multimodal interaction in virtual environments and user interface evaluation. He received his ScM in computer science from Brown in 1999.*

Rosemary Michelle Simpson *is a resources coordinator for the Exploratories project in the Computer Science Department at Brown University, and webmaster for several graphics and hypertext Web sites. She received a BA in history from the University of Massachusetts.*

Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting

R.M. Kirby, H. Marmanis
Division of Applied Mathematics
D.H. Laidlaw
Department of Computer Science
Brown University

Abstract

We present a new visualization method for 2d flows which allows us to combine multiple data values in an image for simultaneous viewing. We utilize concepts from oil painting, art, and design as introduced in [1] to examine problems within fluid mechanics. We use a combination of discrete and continuous visual elements arranged in multiple layers to visually represent the data. The representations are inspired by the brush strokes artists apply in layers to create an oil painting. We display commonly visualized quantities such as velocity and vorticity together with three additional mathematically derived quantities: the rate of strain tensor (defined in section 4), and the turbulent charge and turbulent current (defined in section 5). We describe the motivation for simultaneously examining these quantities and use the motivation to guide our choice of visual representation for each particular quantity. We present visualizations of three flow examples and observations concerning some of the physical relationships made apparent by the simultaneous display technique that we employed.

1 Introduction

Within the study of fluid mechanics, many mathematical constructs are used to enhance our understanding of physical phenomena. The use of visualization techniques as tools for developing physical intuition of mathematically defined quantities is common. Scientific visualization not only expands our understanding of physical phenomena, by allowing us to examine the evolution of quantities like momentum, but also it provides a catalyst for the development of mathematical models which describe the time evolution of complex flows. In addition to the examination of the primitive variables, i.e. the velocity and the pressure, the examination of derived quantities such as the vorticity has provided a better understanding of the underlying processes of fluid flow.

Vorticity is a classic example of a mathematical construct which provides information not immediately assimilated by merely viewing the velocity field. In Figure 1, we illustrate this idea. When examining only the velocity field, it is difficult to see that there is a rotational component of the flow in the far wake region of the cylinder. But, when vorticity is combined with the velocity field, the underlying dynamics of vortex generation and advection is more apparent.

Though vorticity cannot be measured directly, its relevance to fluid flow was recognized as early as 1858 with Helmholtz's pioneering work. Vorticity as a physical concept is not necessarily intuitive to all, yet visualizations of experiments demonstrate its usefulness, and hence account for its popularity. Vorticity is derived from velocity, and *vice versa* under certain constraints [2]. Hence, vorticity does not give any new information that was not already available from the velocity field, but it does emphasize the rotational component of the flow. The latter is clearly demonstrated in Figure 1, where the rotational component is not apparent when one merely views the velocity.

In the same way that vorticity as a derived quantity provides us with additional information about the flow characteristics, other derived quantities such as the rate of strain tensor, the turbulent charge and the turbulent current could be of equal use. Because the examination of the rate of strain tensor, the turbulent charge and the turbulent current within the fluids community is relatively new, few people have ever seen visualizations of these quantities in well known fluid mechanics problems. Simultaneous display of both the velocity and quantities derived from it is done both to allow the fluids' researcher to examine these new quantities against the canvas of previously examined and understood quantities, and also to allow the fluids' researcher to accelerate the understanding of these new quantities by visually correlating them with well known fluid phenomena.

To demonstrate the application of these concepts, we present visualizations of a geometry that, although simple in form, demonstrate many of the major concepts which motivate our work. By examining the well studied problem of flow past a cylinder we demonstrate the usefulness of the visualizations in a context familiar to most fluids' researchers. We examined two-dimensional direct numerical simulation of flow past a cylinder for Reynolds number 100 and 500 [3]. This range of Reynolds numbers provides sufficient phenomenological variation to allow us to discuss the impact of visualization of the newly visualized quantities. In addition to the simulation results presented, we examine data obtained experimentally for a different geometry. This comparison demonstrates one use of the visualization method for experimentalists: data verification.

We extend the visualization methods presented in [1] to problems in fluid mechanics. As in [1], we seek representations that are inspired by the brush strokes artists apply in layers to create an oil painting. We copied the idea of using a primed canvas or underpainting that shows through the layers of strokes. Rules borrowed from art guided our choice of colors, texture, visual elements, composition, and focus to represent data components. Our new methods simultaneously display 6-9 data values, qualitatively representing the underlying phenomena, emphasizing different data values to different degrees, and displaying different portions of the data from different viewing distances. These qualities lead a viewer through the temporal cognitive process of understanding interrelationships in the data much as a painting can lead a viewer through a process designed by the painter.

In the remainder of the paper we first discuss the related work in visualizing multivalued data. We then describe the painting-motivated method we employed, with specific details concerning the combination of scaler, vector, and tensor data into one visualization. In Sections 4 and 5 we present fluid flow examples where multivalued data visualization was used. We summarize and

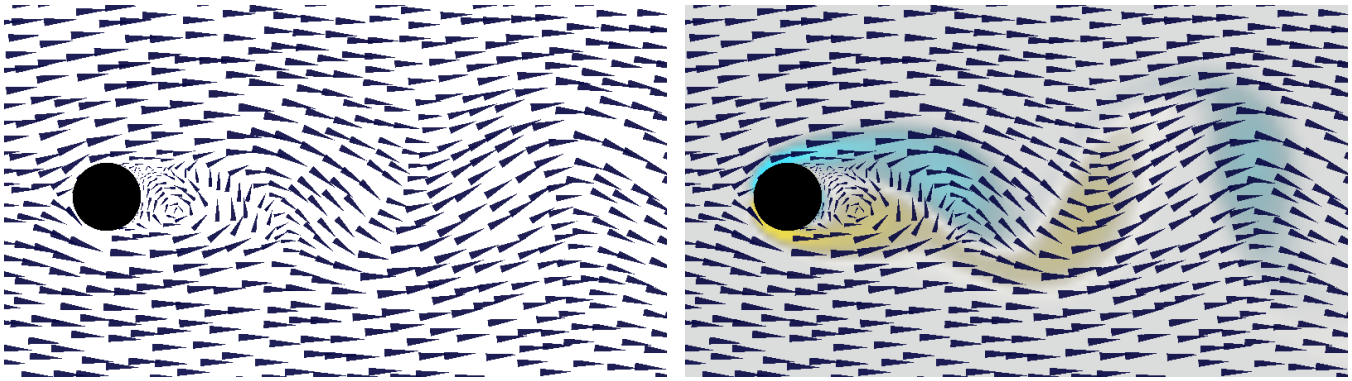


Figure 1: Typical visualization methods for 2D flow past a cylinder at Reynolds number 100. On the left, we show only the velocity field. On the right, we simultaneously show velocity and vorticity. Vorticity represents the rotational component of the flow. Clockwise vorticity is blue, counterclockwise yellow.

present conclusions in Section 6.

2 Related work

2.1 Multivalued data visualization

Hesselink et al. [4] give an overview of research issues in visualization of vector and tensor fields. While they describe several methods that apply to specific problems, primarily for vector fields, the underlying data are still difficult to comprehend; this is particularly true for tensor fields. The authors suggest that "feature-based" methods, i.e., those that visually represent only important data values, are the most promising research areas, and our approach embraces this idea.

Statistical methods such as principal component analysis (PCA) [5] and eigenimage filtering [6] can be used to reduce the number of relevant values in multivalued data. In reducing the dimensionality, these methods inevitably lose information from the data. Our approach complements these data-reduction methods by increasing the number of data values that can be visually represented.

Different visual attributes of icons can be used to represent each value of a multivalued dataset. In [7], temperature, pressure, and velocity of injected plastic are mapped to geometric prisms that sparsely cover the volume of a mold. Similarly, in [8] data values were mapped to icons of faces: features like the curve of the mouth or size of the eyes encoded different values. In both cases, the icons capture many values simultaneously but can obscure the continuous nature of fields. A more continuous representation using small line segment-based icons shows multiple values more continuously [9]. Our work builds upon these earlier types of iconic visual representation.

Layering has been used in scientific visualization to show multiple items: in [10, 11], transparent stroked textures show surfaces without completely obscuring what is behind them. These results are related to ours, but our application is 2D, and so our layering is not as spatial as in the 3D case. Our layering is more in the spirit of oil painting where layers are used more broadly, often as an organizing principle.

2.2 Flow visualization

A number of flow-visualization methods display multivalued data. The examples in [12, 13] combine surface geometries representing cloudiness with volume rendering of arrows representing wind velocity. In some cases, renderings are also placed on top of an image of the ground. Unlike our 2D examples, however, the phenomena are 3D and the layering represents this third spatial dimension. Similarly, in [14], surface particles, or small facets, are used to visualize 3D flow: the particles are spatially isolated and are again rendered as 3D objects.

A "probe" or parameterized icon can display detailed information for one location within a 3D flow [15]; it faithfully captures velocity and its derivatives at that location, but does not display them globally. Our data contain fewer values at each location, because we are working with 2D flow, but our visualization methods display results globally instead of at isolated points.

Spot noise [16] and line integral convolution [17] methods generate texture with structure derived from 2D flow data; the textures show the velocity data but do not directly represent any additional information, e.g., divergence or shear. The authors of [16] mention that spot noise can be described as a weighted superposition of many "brush strokes," but they do not explore the concept. Our method takes the placement of the strokes to a more carefully structured level. Of course, placement can be optimized in a more sophisticated manner, as demonstrated in [18]; we would like to explore combining these concepts with ours. Currently our stroke placement is simple and quick to implement while providing adequate results.

2.3 Computer graphics painting

Reference [19] was the first to experiment with painterly effects in computer graphics. Reference [20] extended the approach for animation and further refined the use of layers and brush strokes characteristic for creating effective imagery. Both of these efforts were aimed toward creating art, however, and not toward scientific visualization. Along similar lines, references [21, 22, 23] used software to create pen and ink illustrations for artistic purposes. The pen and ink approach has successfully been applied to 2D tensor visualization in [24].

In reference [1], painterly concepts as used in our work were presented for visualizing diffusion tensor images of the mouse spinal cord. In that work both a motivation and a methodology for the techniques used here were presented. The goal of our work is to visualize simultaneously both new and commonly used scientific quantities within the field of fluid mechanics by building on those concepts.

3 Visualization methodology

We use the methodology and system of [1] to develop our visualizations. We review the methodology here. Developing a visualization method involves breaking the data into components, exploring the relationships among them, and visually expressing both the components and their relationships. For each example we explored different ways of breaking down the data so that we could gain understanding as to how the components were related. Once we achieved an initial understanding, we proceeded to the next step: designing a visual representation.

In the design, we used artistic considerations to guide how we mapped data components to visual cues of strokes and layers. Our brush strokes are affinely transformed images with a superimposed texture. In choosing mappings we looked for geometric components and mapped them to geometric cues like the length or direction of a stroke. We considered the relative significance of different components and mapped them to cues that emphasized them appropriately. For example, two related parameters could map to the length and width of a stroke, giving a clear indication of their relative values. We also considered the order in which components would best be understood and mapped earlier ones to cues that would be seen more quickly. The set of mappings we selected defined a series of stroke images and a scheme for how to layer them.

An iterative process of analysis and refinement followed. Sometimes our refinements involved choosing a mapping we found effective in one visualization and incorporating it into another. Sometimes we needed to change the emphasis among data components by adjusting transparency, size, or color or by representing a component with a different or additional mapping. Sometimes we needed to go further back in the process and choose a new way of breaking down the data.

4 Example 1: Rate of strain tensor

The rate of strain tensor (sometimes called the deformation-rate tensor [25]), is a commonly used derived quantity within fluid mechanics. Though commonly used and reasonably well comprehended, few have visualized this tensor due to the added complexity necessary to view multivalued data. Our motivation for combining visualization of the rate of strain tensor with velocity and vorticity is that despite many years of intense scrutiny, scientific understanding of fluid behavior is still not complete, and qualitative descriptions can still be helpful. Researchers often examine images of individual velocity-related quantities. We thought that good intuition might come from a visual representation that related these values to one another in a single image.

4.1 Data breakdown

We began by choosing a breakdown of data values into components that can be mapped onto stroke attributes. Both the velocity and its first spatial derivatives have meaningful physical interpretations [25], and hence we treat them independently. The velocity is a 2-vector with a direction and a magnitude in the plane, and can be visually mapped directly. The spatial derivatives of velocity form a second-order tensor known at the velocity gradient tensor. This tensor can be written as the sum of symmetric and antisymmetric parts,

$$\frac{\partial u_i}{\partial x_j} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right), \qquad (1)$$

$$= e_{ij} + \Omega_{ij}. \qquad (2)$$

The antisymmetric part Ω_{ij} reduces to the vector quantity vorticity ($\omega_k = \frac{1}{2}\epsilon_{ijk}\Omega_{ij}$), and the symmetric part e_{ij} is known as the rate of strain tensor [26]. The vorticity field determines the axis and the magnitude of rotation for all fluid elements. The rate of strain tensor determines the rate at which a fluid element changes its shape under the particular flow conditions. In incompressible flows, the instantaneous rate of strain consists always of a uniform elongation process in one direction and a uniform foreshortening process in a direction perpendicular to the first. That is, a small circle will change its shape into an ellipse, whose major and minor axes represent the rate of elongation and the rate of squeezing, respectively. In compressible flows, the latter statement is not necessarily true since expansion and compression is allowed. Nevertheless, the visualization of the rate of strain remains valuable and instructive in these flows as well.

4.2 Visualization design

We wanted the viewer to first read velocity from the visualization, then vorticity and its relationship to velocity. Because of the complexity of the second-order rate of strain tensor we want it to be read last. We describe the layers here from bottom up, beginning with a primed canvas, adding an underpainting, representing the tensor values transparently over that, and finishing with a very dark, high-contrast representation of the velocity vectors.

- **Primer** The bottom layer of the visualization is light gray, selected because it would show through the transparent layers to be placed on top.
- **Underpainting** The next layer encodes the scalar vorticity value in semi-transparent color. Since the vorticity is an important part of fluid behavior, we emphasized it by mapping it onto three visual cues: color, ellipse opacity, and ellipse texture contrast (see below). Clockwise vorticity is blue and counter-clockwise vorticity yellow. The layer is almost transparent where the vorticity is zero, but reaches 75% opacity for the largest magnitudes, emphasizing regions where the vorticity is non-zero.
- **Ellipse layer** This layer shows the rate of strain tensor and also gives additional emphasis to the vorticity. The logarithms of the rates of strain in each direction scale the radii of a circular brush shape to match the shape that a small circular region would have after being deformed. The principal deformation direction was mapped to the direction of the stroke to orient the ellipse. The strokes are placed to cover the image densely, but with minimal overlap. The color and transparency of the ellipses are taken from the underpainting, so they blend well and are visible primarily where the vorticity magnitude is large. Finally, a texture whose contrast is weighted by the vorticity magnitude gives the ellipses a visual impression of spinning where the vorticity is larger.
- **Arrow layer** The arrow layer represents the velocity field measurements: the direction of the arrows is the direction of the velocity, and the brush area is proportional to the speed. We chose a dark blue to contrast with the light underpainting and ellipses, to make the velocities be read first. The arrows are spaced so that strokes overlap end-to-end but are well separated side-to-side. This draws the eye along the flow.
- **Mask layer** The final layer is a black mask covering the image where the cylinder was located.

These painting concepts help create a visual representation for the data that encodes all of the data in a manner that allows us to explore the data for a more holistic understanding.

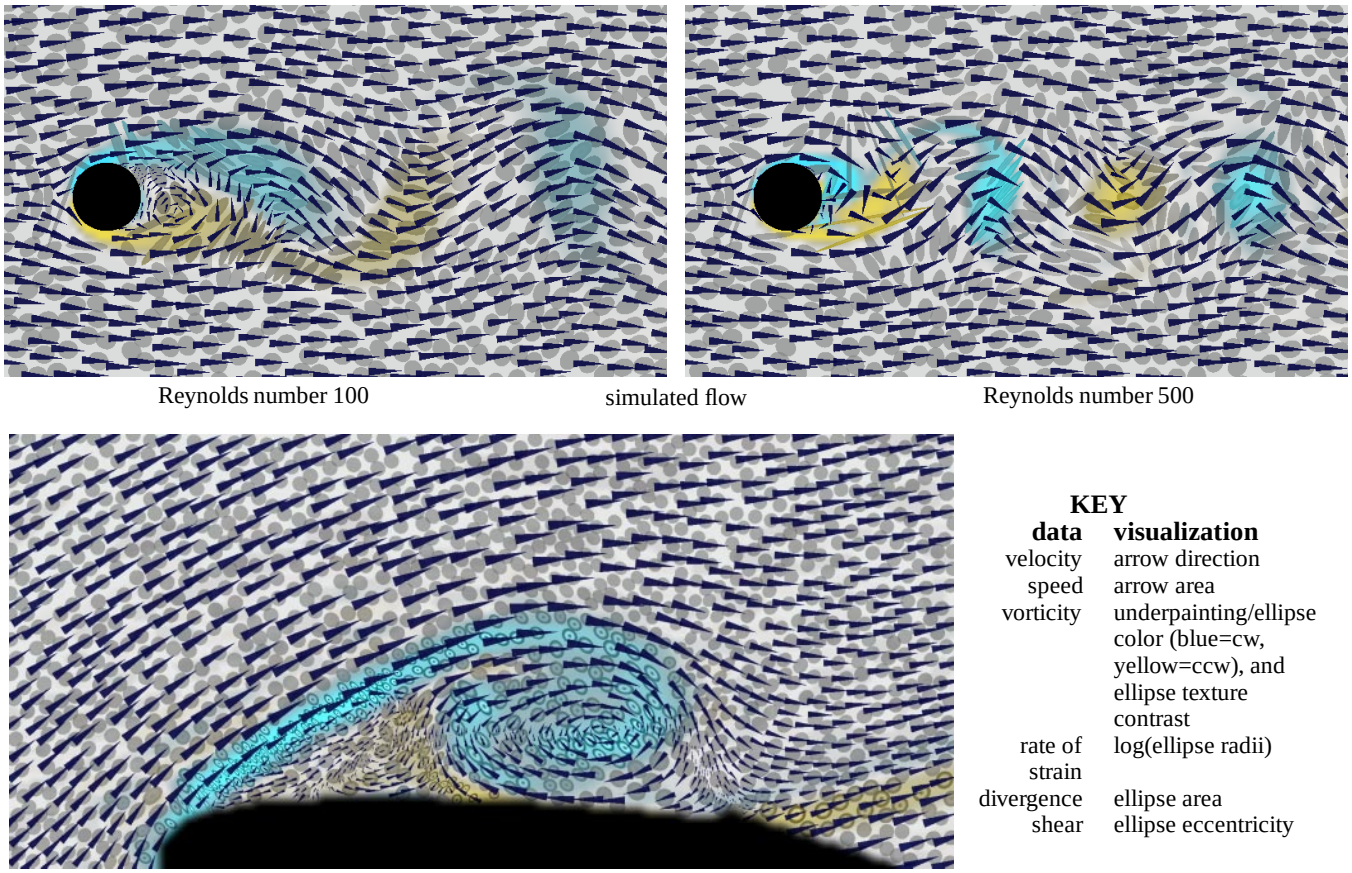


Figure 2: Visualization of simulated 2D flow past a cylinder at Reynolds number = 100 and 500 (top left and top right), and experimental 2D flow past an airfoil (bottom). Velocity, vorticity, and rate of strain (including divergence and shear) are all encoded in the layers of this image. With all six values at each point visible, the image shows relationships among the values that can verify known properties of a particular flow or suggest new relationships between derived quantities.

4.3 Observations

Figure 2 (top left and top right) shows visualizations of 2d flow simulation results obtained using Hybrid $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha r$ [27], a spectral element code for solving the incompressible Navier-Stokes equations. These results were obtained from the work presented in [3]. Figure 2 (bottom) shows data measured experimentally as an airfoil is drawn through a tank of initially stationary water. An image is taken perpendicular to the axis of the airfoil using laser induced fluorescence (LIF) imaging [28]. Velocity data calculated from the LIF images lies on a rectangular grid, with some portions missing, as the figure shows in the black region.

The visualization of single quantities is useful by itself. For instance, if we contrast the simulation results (Figure 2 top right and left) with the experimental data of the airfoil (Figure 2 bottom), we observe that all the ellipses have the same area in the former case whereas they do not have the same area in the latter case. In incompressible flows, the continuity equation implies that the velocity field is divergence-free, which in turn implies that the trace of the rate of strain tensor (i.e. the sum of its diagonal elements) is always zero. This simply means that the area of the fluid elements remains constant in time, regardless of their instantaneous shape. Hence, we can infer that the simulation has reproduced properly the incompressible character of the fluid flow whereas the airfoil data show either compressibility effects or out of plane motion, neither of which can easily be detected by other means.

The multivalued data visualization, however, has additional merits. For instance, we observe that the simultaneous viewing of the vorticity field and the rate of strain tensor provides us with a physical understanding about the deformation of the fluid elements. It clearly shows that at the centers of the vortices the deformation can be rather small, dependent on the eccentricity of the fluid element with respect to the center of the vortex, whereas at the edges of vortices the fluid elements suffer a huge shearing effect. Thus the mathematical decomposition of the velocity gradient tensor (i.e. $\partial u_i/\partial x_j$) into its symmetric (i.e. the rate of strain) and antisymmetric (i.e. the vorticity) parts acquires a visual representation.

Until now the deformation of the fluid elements was represented with qualitative sketches [29] whose direct connection to the rest of the flow field was not obvious. Through our visualization technique, we obtain not only the qualitative character of the fluid element deformation but also its quantitative properties. Moreover, all the information about the deformation can now be visually correlated to the velocity and the vorticity fields.

5 Example 2: Turbulent charge and turbulent current

Turbulent charge and turbulent current are flow quantities that have not been extensively visualized. Our motivation for viewing these quantities, in conjunction with other well-studied quantities (e.g. the vorticity), has its roots in our desire to solve problems that are concerned with *drag reduction*. The importance of fluid mechanics to the problem of reducing the drag on a moving body is unequivocal. All airplane, boat, and car designers, at some stage of their research, have consulted engineers about possible ways of reducing the drag. This is quite reasonable, since drag reduction translates to less fuel consumption.

One method of reducing the drag on a body is the appendage of riblets on the surface of the body. Though experimentally verified, the physical mechanism behind the drag reduction is not well understood. For example, some configurations and shapes of riblets do give drag reduction but some others do not. Thus, the question arises as to why this happens. What are the shapes and which are the configurations that produce drag reduction? The use of riblets everywhere on the surface is costly, thus another question is: what is the location, on the surface of an object, that will provide maximum drag reduction? The answers to the above questions can be found by inspecting visually the turbulent charge on the surface of the body [30].

Suppose we are interested in reducing the drag on a submarine. Our goal from the engineering standpoint is to find geometric modifications to our structure so that we get reasonable drag reduction with a minimum cost (and without inhibiting the purpose of our submarine). Modeling the turbulent charge on the surface of the submarine immediately delineates those regions of the geometry which could most benefit from drag reduction techniques. Unlike all other drag reduction models, the concept of turbulent charge and turbulent current succinctly provide information that is applicable to engineering design.

Unlike the case of simple flows, which can be described easily in terms of vorticity, there are cases in which the visualization of vorticity, and the subsequent description of the flow by it, can be as complex as the one in terms of velocity. For example, in the case of turbulent flows, vortices are shed from the boundaries of the flow domain, they are convected away from it, and subsequently interact with each other in a fashion that has defied a satisfactory solution of practical importance for more than a century. Hence, we can legitimately ask whether we can find other quantities whose visualization in these cases can be as beneficial to our understanding as vorticity is in more simple flows.

5.1 Data breakdown

We began by choosing a breakdown of data values into components that can be mapped onto stroke attributes. It has recently been suggested that two newly introduced quantities, namely, the turbulent charge $n(\mathbf{x},t)$ and the turbulent current $\mathbf{j}(\mathbf{x},t)$, collectively referred to as the turbulent sources, could substitute the role of vorticity in more complicated flows. The nomenclature is not coincidental, it reflects the fact that the derivation of these quantities was based on an analogy between the equations of hydrodynamics and the Maxwell equations [31].

In particular, if we denote by $\mathbf{u}$ the velocity and by p the pressure then the vorticity, $\mathbf{w}$, is given by $\nabla \times \mathbf{u}$, and the Lamb vector is given by $\mathbf{l} \equiv \mathbf{w} \times \mathbf{u}$. The turbulent sources are given by the following expressions:

$$n(\mathbf{x},t) \equiv \nabla \cdot \mathbf{l}(\mathbf{x},t) = -\nabla^2 \left(\frac{p}{\rho} + \frac{u^2}{2} \right) , \qquad (3)$$

and

$$\mathbf{j} = \mathbf{u}n + \nabla\times(\mathbf{u}\cdot\mathbf{w})\mathbf{u} + \mathbf{w}\times\nabla \left(\frac{p}{\rho} + \frac{3u^2}{2} \right) + 2(\mathbf{l}\cdot\nabla)\mathbf{u} . \quad (4)$$

The later two quantities (i.e. $n, \mathbf{j}$) are related to each other through a continuity type of equation where the turbulent current is the flux of the turbulent charge. In the cases where turbulent charge is generated solely at the wall, small turbulent charge implies small turbulent current.

5.2 Visualization design

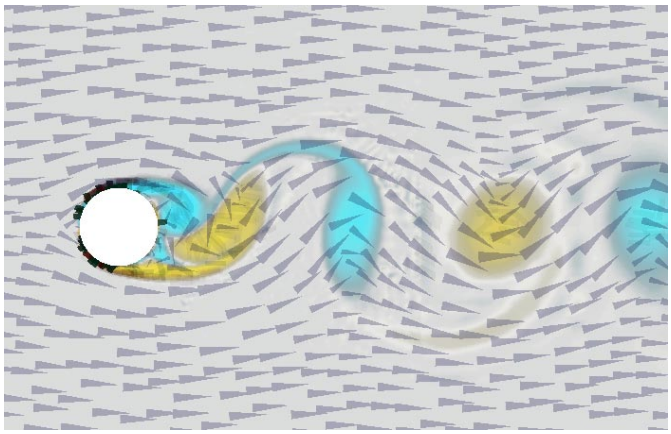
We designed the turbulent source visualizations so that the overall location of the turbulent charge would be visible early. The vorticity was our next priority, since comparison between the two quantities was important. Our third priority was the structure of the flow, as represented by the velocity field. Finally, we wanted fine details about the structure of all the fields, charge, current, velocity, and vorticity, available upon close examination.

We describe the layers here from bottom up, as in the last example. Beginning with the same primed canvas and underpainting, continuing with a low-contrast representation of the velocity vectors, and finishing with a high-contrast representation of the turbulent sources. A final layer represents the geometry of the cylinder.

- **Primer and underpainting** Same as first example.
- **Arrow layer** The arrow layer for this example has the same geometric components – brush area proportional to speed, velocity direction mapped to brush direction, and strokes arranged closer end-to-end to give a sense of flow. This layer differs in that its emphasis is decreased. It has a low contrast with the layers below it. The low contrast is partly achieved through the choice of a light color for the arrows and partly through transparency of the arrows. Without the transparency, the arrows would appear very independent of the underlying layers.
- **Turbulent sources layer** In this layer we encode both the turbulent charge and the turbulent current. The current, a vector, is encoded in the size and orientation of the vector value just as the velocity in the arrow layer. The scalar charge is mapped to the color of the strokes. Green strokes represent negative charge and red strokes positive. The magnitude of the charge is mapped to opacity. Where the charge is large, we get dark, opaque, high-contrast strokes that strongly emphasize their presence. Where the charge is small, the strokes disappear and do not clutter the image. For these quantities, that tend to lie near surfaces, this representation makes very efficient use of visual bandwidth. The strokes in this layer are much smaller than the the strokes in the arrow layer. This allows for finer detail to be represented for the turbulent sources, which tend to be more localized. It also helps the turbulent sources layer to be more easily distinguished from the arrows layer than in the previous visualization, where the stroke sizes were closer and, therefore, harder to disambiguate visually.
- **Mask layer** The final layer is a mask representing the geometry of the cylinder. The mask is white in this example to contrast better with the turbulent sources layer.

5.3 Observations

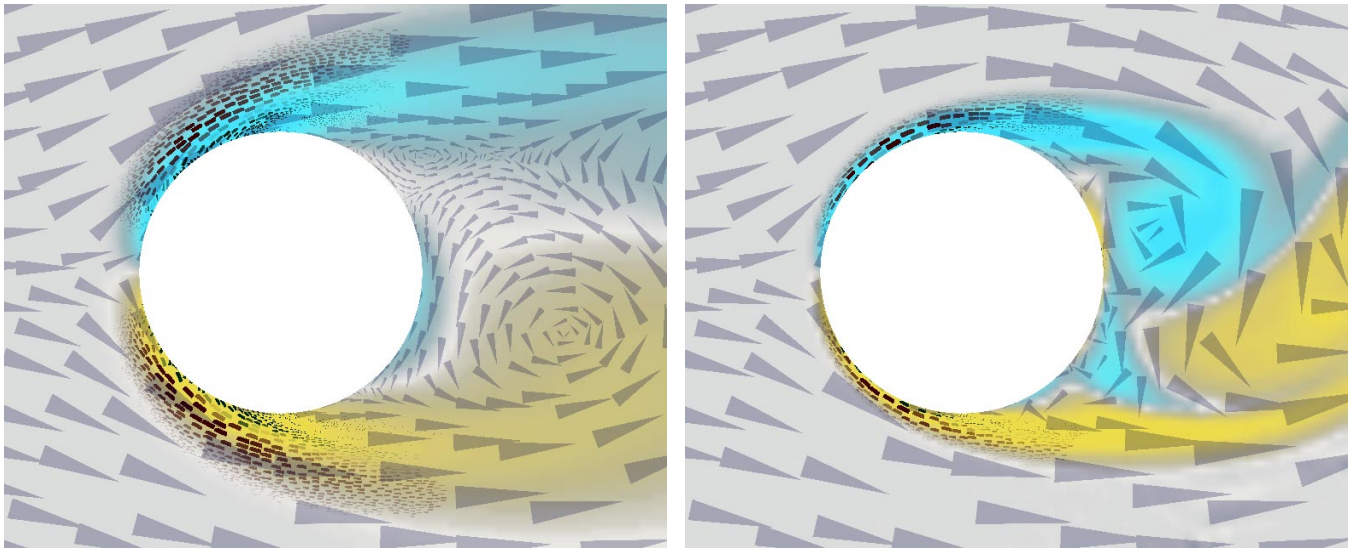
The regions where the turbulent charge achieves its maximum values are the regions where the vorticity field has also very large



KEY

data	**visualization**
velocity	arrow direction
speed	arrow area
vorticity	underpainting: blue=cw, yellow=ccw
turbulent charge	stroke color
turbulent current	stroke direction/area

Figure 3: Visualization of the turbulent charge and the turbulent current for a Reynolds number 500 simulated flow. Observe that charge concentrates near the cylinder and is negligible in other parts the flow. The cylinder geometry is now white to contrast with the visual representation for the turbulent sources



Reynolds number 100 Reynolds number 500

Figure 4: Close up visualization of the turbulent charge and the turbulent current at Reynolds number 100 and 500 (left and right). We are able to see the high concentrations of negative charge at the places where vorticity is being generated.

values. Nevertheless, as we have already mentioned, the advantage in thinking of terms of turbulent charge is related to its permanence close to the boundaries, in contrast to the vorticity field which is conveyed downstream.

The theory proposed in [31] predicts that the turbulent charge, n, and turbulent current, $\mathbf{j}$, are the source terms of the following linear system of equations

$$\begin{aligned} \nabla \cdot \mathbf{W} &= 0\,, \\ \frac{\partial \mathbf{W}}{\partial t} &= -\nabla \times \mathbf{L} - \nu \nabla \times \nabla \times \mathbf{W}\,, \\ \nabla \cdot \mathbf{L} &= N(\mathbf{x}, t)\,, \\ \frac{\partial \mathbf{L}}{\partial t} &= c^2 \nabla \times \mathbf{W} - \mathbf{J}(\mathbf{x}, t) + \nu \nabla \times \nabla \times \mathbf{L}\,, \end{aligned} \tag{5}$$

where $c^2 = \langle u^2 \rangle$, and the use of capital letters denotes that the corresponding quantities have been averaged. From these equations it can be shown that the turbulent current is the dominant forcing term for the velocity. An immediate consequence of this is that the turbulent current and the velocity field should be aligned. In Figure 4 we observe this alignment, especially in the region near the cylinder where we have the most significant change of flow velocity.

Finally, in Figure 5, we add the rate of strain tensor to the turbulent sources visualization, adjusting the blending of the different layers to control their relative emphasis. We observe that the high values of turbulent charge are associated with an extreme deformation of the fluid elements, since it is the shear between adjacent fluid layers that transforms the kinetic energy of the fluid to molecular heat.

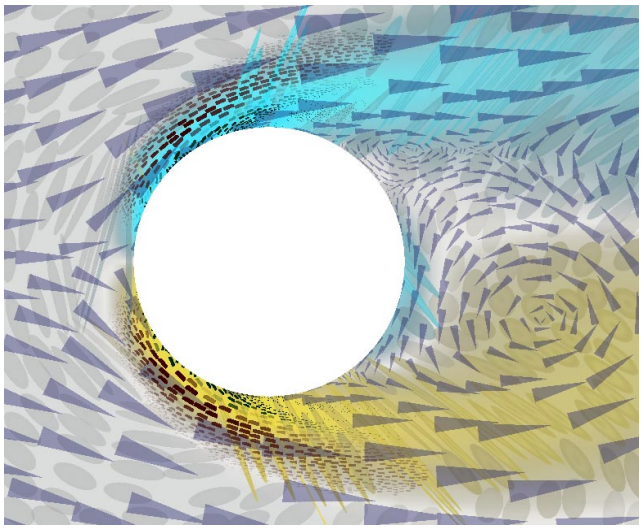


Figure 5: Combination of velocity, vorticity, rate of strain, turbulent charge and turbulent current for Reynolds number 100 flow. A total of nine values are simultaneously displayed.

Visualizing the turbulent sources is very informative for turbulent non-equilibrium flows. In fact, a plot of the turbulent charge distribution immediately allows us to determine whether a particular configuration of the riblets, discussed earlier, is reducing or enhancing the drag. A plot of the turbulent current can immediately reveal which flow directions are dominant (e.g. the streamwise direction in a pipe flow), even if no other information about the flow field is given. The distribution of the turbulent sources reflects succinctly the responses of the flow due to boundary conditions or external fields.

6 Summary and Conclusions

We have presented results of applying the scientific visualization approach outlined in [1] to multivalued incompressible fluid data. The approach borrows concepts from oil painting. Underpaintings showed form. We used brush strokes both individually, to encode specific values, and collectively, to show spatial connections and to generate texture and an impression of motion. We used layering and contrast to create depth. Stroke size, texture, and contrast helped to define a focus within each image and also to influence the order in which different parts of an image were viewed.

The methods we employed produce images that are visually rich and represent many values at each spatial location. From different perspectives, they show the data at different levels of abstraction – more qualitatively at arm's length, more quantitatively up close. Finally, the images emphasize different data values to different degrees, leading a viewer through the temporal cognitive process of understanding the relationships among them.

We visualize quantities that have rarely been viewed before: rate of strain, turbulent charge, and turbulent current. We visualize these new quantities together with more commonly viewed quantities, allowing a scientist to use previously acquired intuition in interpreting the new values and their relationships to one another and to the more traditional quantities.

Our visualization of the rate of strain tensor combined with both the velocity and vorticity fields provides a unique pedagogical tool for explaining the dominant mechanisms responsible for certain fluid flow phenomena. Because an understanding of the deformation tensor (i.e. $\partial u_i / \partial x_j$) is of paramount importance for one's understanding of fluid flow phenomena, visualizing its symmetric and antisymmetric parts separately (i.e. the rate of strain tensor and the vorticity, respectively) clearly accentuates the interplay between rotational and shearing mechanisms within the flow.

The visualization of turbulent charge and turbulent current combined with both velocity and vorticity allows us to use knowledge concerning the latter fields in our effort to understand the usefulness of the newly visualized quantities. It is evident from the visualizations shown that, unlike vorticity, the turbulent charge and the turbulent current are far more localized. This validates the conjectures about the potential usefulness of the model, and also suggests that we focus our attention on viewing the turbulent charge and turbulent current regions close to the surface of the cylinder. By focusing our examination to regions close to the cylinder, we see a high visual correlation between regions where turbulent charge accumulates and regions of vorticity generation.

By visualizing velocity with all the subsequently derived quantities presented here, we can observe through one visualization multiple properties of the flow. The freedom to display multivalued data simultaneously allows us to get a more complete idea of both the dynamics and the kinematics of the flow, and hence provides a catalyst for future understanding of more complex fluid phenomena.

Acknowledgements

We would like to acknowledge Professor George Em. Karniadakis and the CRUNCH group for the manifold support we have received. The Graduate Aeronautics Lab at Caltech (GALCIT), Galen Gornowicz, Haris Catrakis, Jerry Shan, and Paul Dimotakis, provided experimental fluid flow data and discussions of how to understand it.

This work was supported by NSF (CCR-96-19649, CCR-9996209); NSF (ASC-89-20219) as part of the NSF STC for Computer Graphics and Scientific Visualization; and the Human Brain Project with contributions from the National Institute on Drug Abuse and the National Institute of Mental Health.

References

[1] David H. Laidlaw, Eric T. Ahrens, David Kremers, Matthew J. Avalos, Carol Readhead, and Russell E. Jacobs. Visualizing diffusion tensor images of the mouse spinal cord. In *Proceedings Visualization '98*. IEEE Computer Society Press, 1998.

[2] P.G. Saffman. *Vortex Dynamics*. Cambridge University Press, Cambridge, UK, 1992.

[3] C. Evangelinos. *Parallel Simulations of Flexible Cylinders subject to VIV*. PhD thesis, Division of Applied Mathematics, Brown University, 1999.

[4] Lambertus Hesselink, Frits H. Post, and Jarke J. van Wijk. Research issues in vector and tensor field visualization. *IEEE Computer Graphics and Applications*, 14(2):76–79, 1994.

[5] K Jain. *Fundamentals of Digital Image Processing*. Prentice Hall, 1989.

[6] Joe P. Windham, Mahmoud A. Abd-Allah, David A. Reimann, Jerry W. Froelich, and Allan M. Haggar. Eigenimage filtering in MR imaging. *Journal of Computer-Assisted Tomography*, 12(1):1–9, 1988.

[7] Robert B. Haber and David A. McNabb. Visualization idioms: A conceptual model for scientific visualization systems. *Visualization in scientific computing*, pages 74–93, 1990.

[8] Herman Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, 68(342):361–368, 1973.

[9] Robert F. Erbacher, Georges Grinstein, John Peter Lee, Haim Levkowitz, Lisa Masterman, Ron Pickett, and Stuart Smith. Exploratory visualization research at the University of Massachusetts at Lowell. *Computers and Graphics*, 19(1):131–139, 1995.

[10] Victoria Interrante, Henry Fuchs, and Stephen M. Pizer. Conveying the 3D shape of smoothly curving transparent surfaces via texture. *IEEE Transactions on Visualization and Computer Graphics*, 3(2), April–June 1997. ISSN 1077-2626.

[11] Victoria L. Interrante. Illustrating surface shape in volume data via principal direction-driven 3D line integral convolution. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 109–116. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.

[12] Nelson Max, Roger Crawfis, and Dean Williams. Visualization for climate modeling. *IEEE Computer Graphics and Applications*, 13(4):34–40, July 1993.

[13] Roger Crawfis, Nelson Max, and Barry Becker. Vector field visualization. *IEEE Computer Graphics and Applications*, 14(5):50–56, 1994.

[14] Jarke J. van Wijk. Flow visualization with surface particles. *IEEE Computer Graphics and Applications*, 13(4):18–24, July 1993.

[15] Jarke J. van Wijk, Andreaj S. Hin, Willem C. de Deeuw, and Frits H. Post. Three ways to show 3d fluid flow. *IEEE Computer Graphics and Applications*, 14(5):33–39, 1994.

[16] Jarke J. van Wijk. Spot noise-texture synthesis for data visualization. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 309–318, July 1991.

[17] Brian Cabral and Leith (Casey) Leedom. Imaging vector fields using line integral convolution. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 263–272, August 1993.

[18] Greg Turk and David Banks. Image-guided streamline placement. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 453–460. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[19] Paul E. Haeberli. Paint by numbers: Abstract image representations. In Forest Baskett, editor, *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 207–214, August 1990.

[20] Barbara J. Meier. Painterly rendering for animation. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 477–484. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[21] Georges Winkenbach and David H. Salesin. Rendering parametric surfaces in pen and ink. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings*, Annual Conference Series, pages 469–476. ACM SIGGRAPH, Addison Wesley, August 1996. held in New Orleans, Louisiana, 04-09 August 1996.

[22] Georges Winkenbach and David H. Salesin. Computer–generated pen–and–ink illustration. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 91–100. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

[23] Michael P. Salisbury, Sean E. Anderson, Ronen Barzel, and David H. Salesin. Interactive pen–and–ink illustration. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 101–108. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

[24] Michael P. Salisbury, Michael T. Wong, John F. Hughes, and David H. Salesin. Orientable textures for image-based pen-and-ink illustration. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings*, Annual Conference Series, pages 401–406. ACM SIGGRAPH, Addison Wesley, August 1997. ISBN 0-89791-896-7.

[25] Philip A. Thompson. *Compressible-Fluid Mechanics*. Advanced engineering series. Maple Press Co., 1984.

[26] Rutherford Aris. *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*. Dover Publications, Inc, New York, 1989.

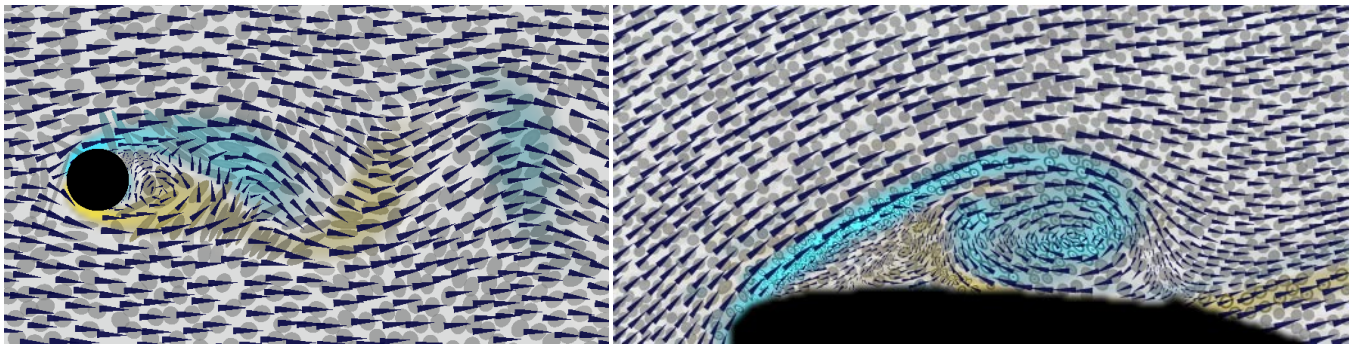
[27] T.C.E. Warburton. *Spectral/hp Methods on Polymorphic Multi-Domains: Algorithms and Applications*. PhD thesis, Division of Applied Mathematics, Brown University, 1999.

[28] Galen G. Gornowicz. *Continuous-Field Image-Correlation Velocimetry and its Application to Unsteady Flow Over an Airfoil*. Engineering thesis, California Institute of Technology, 1997.

[29] M.J. Lighthill. *Laminar Boundary Layers*. (ed. E. Rosenhead), Dover Publications, Inc, New York, 1988.

[30] H. Marmanis, Y. Du, and G.E. Karniadakis. Turbulent control via geometry modifications and electromagnetic fields. *ECCOMAS*. 1998.

[31] H. Marmanis. Analogy between the navier-stokes and maxwell's equations: Application to turbulence. *Phys. Fluids*, 10(6):1428–1437, 1998.



simulated flow, Reynolds number 100 experimental flow

Figure 2: Visualization of 2D flow. Velocity, vorticity, and rate of strain (including divergence and shear) are all encoded in image layers.

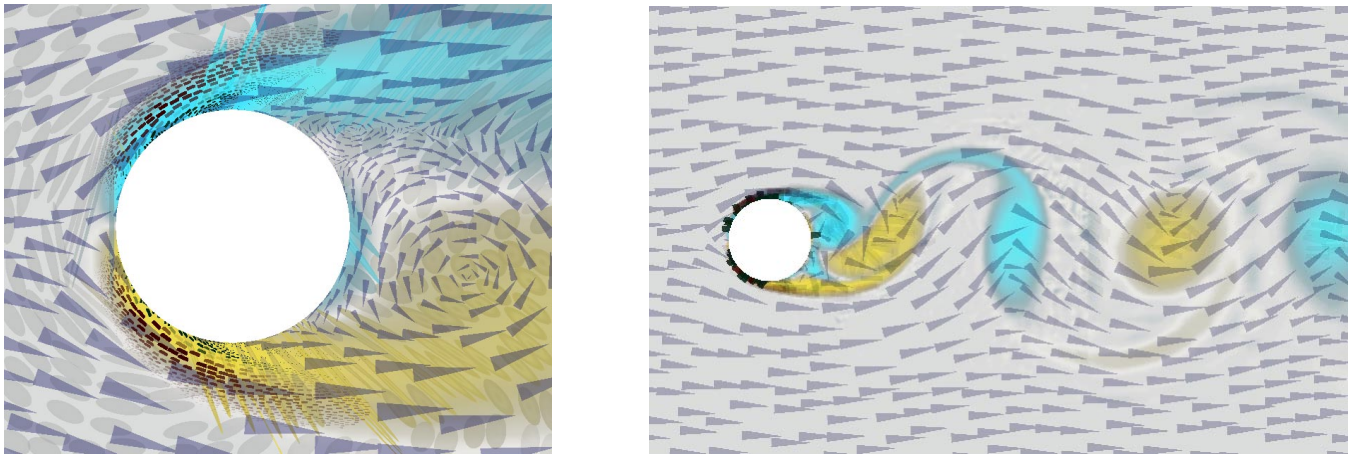
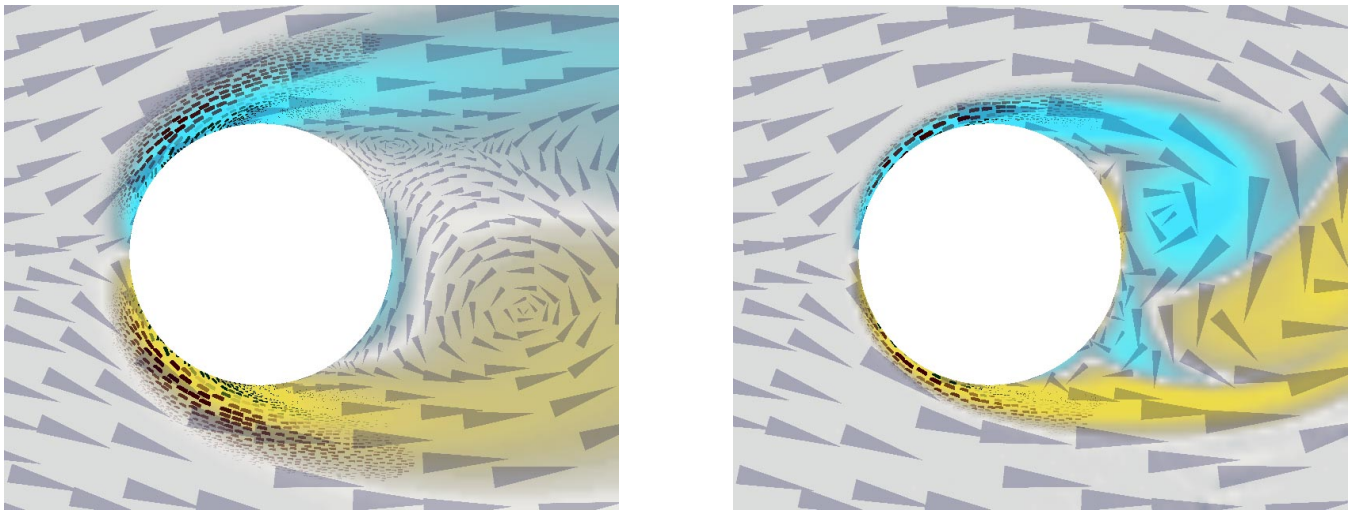


Figure 5: (left) Combination of velocity, vorticity, rate of strain, turbulent charge and turbulent current for Reynolds number 100 flow. A total of nine values are simultaneously displayed. Figure 3: (right) Visualization of the turbulent charge and the turbulent current together for a Reynolds number 500 simulated flow. Observe that charge concentrates near the cylinder and is negligible in other parts the flow.



Reynolds number 100 Reynolds number 500

Figure 4: Close up visualization of the turbulent charge and the turbulent current at Reynolds number 100 and 500 (left and right). We are able to see the high concentrations of negative charge at the places where vorticity is being generated.

Art and Visualization: Oil and Water?

David Laidlaw, Brown University Computer Scientist (organizer)
davidkremers, Caltech Distinguished Conceptual Artist in Biology
Victoria Interrante, University of Minnesota Computer Scientist
Felice Frankel, MIT Artist in Residence and Research Scientist
Thomas Banchoff, Brown University Geometer and Artist

INTRODUCTION

Art and visualization have progressed on parallel paths, often visiting similar points in the space of imagery. This panel session brings together artists who have scientific interests with scientists who have artistic interests. Together, we hope to stimulate excitement about searching the collective experience of centuries of artists to find concepts salient to visualization. Each of the panelists will discuss some of their work, giving concrete examples of joint art/science endeavors. We have organized our statements around the following questions:

1. How can artistic experience benefit visualization? What artistic disciplines have the most to offer?
2. What are the dangers of mixing the two disciplines?
3. How should we proceed? What are the rich research areas to explore?

POSITION STATEMENTS

David Laidlaw

For six centuries artists have developed methods for representing complex scenes in oil paintings. The work that I will show excavates concepts from oil painting and applies them to visualization. We have used multiple layers of brush strokes, motivated by Van Gogh's style, to represent multi-valued data. The resulting images simultaneously display up to eight values at each point. I'll show results of several different types of data displayed with these methods. Surprisingly, these images are richly detailed, and offer different views from different perspectives, much as paintings often do.

Creating visual representations with these methods is a delicate process of balancing the visual bandwidth used for one component of a dataset against the visual bandwidth used for another. Maintaining a relatively continuous representation in one layer without obscuring underlying information also creates tension. And choosing which parts of the data to map to quickly seen visual cues and which to map to cues with a longer latency adds a temporal dimension to the resulting images.

Of course, there are dangers in the process. The potential for misrepresentation is high because the process is subjective. The balancing act can fail, and important features can be obscured or de-emphasized enough that they are missed. More subtle misrepresentations can accidentally map data to cues that have a strong unintended impact. Chernoff used iconic facial features to represent the different values in multi-valued data. Some of the features, such as the upward curve of the mouth, have a very strong emotional impact on western viewers.

There are a number of areas ripe for exploration. Only the surface of painting has been scratched. Van Gogh's brush strokes are wonderfully expressive and discrete in relation to those of many other artists. The work of other painters

is likely to provide many more ideas applicable to visualization. Other artistic disciplines also hold promise. Graphic design, illustration, and sculpture all spring to mind as relevant to visualization, and some exploration has begun in these areas. Consistent "standard" visualization techniques and test cases will make comparisons possible. How to standardize something related to "art" is an interesting problem. And, finally, more visibility for this fledgling yet ancient process of mining the past will help stimulate essential interest and enthusiasm.

davidkremers

the naked human eye can distinguish intervals up to 1/100 of an inch. in order to conceptualize events beyond this limit we developed mathematics and art.

mathematics, along with the tools it helped create, took opinion out of scientific observation. what had once been metaphysics became physics. even better, the church didn't argue with these new views of the universe so long as they remained mathematics, a place invisible to the masses.

in the modern world, our combination of electronics and molecular biology is allowing humans to perceive the invisible at a resolution in sync with the limitations of the human eye. this is putting opinion back into scientific observation.

the fact that we are moving from studying inert samples to working with complex dynamic systems is forcing us to meld art and science into something new. recently i participated in experiments using optical sectioning and 3d reconstruction of stained mouse somites. the first discovery to be made was that our new visualization techniques far outstripped the standards of "artistry" at the bench. our second discovery was that the existing schematic idea of these structures doesn't match up with our organic results.

so we are testing the samples with two forms of high resolution pattern recognition. we are increasing the technical resolution afforded by recent advances in 2-photon microscopy and we are also increasing the observational resolution by taking the unique step of including an artist in the team. artists are very highly trained "eyes" in pattern recognition, and they bring an unbiased eye to biology which can question recurring patterns overlooked by the "practiced" eye of a biologist who is only looking to see what she expects to find.

art is good at qualitative questions, the chief question of art has historically been why? the question facing modern artists is how much objectivity can we afford to let in before we begin to lose the discoveries afforded art by intuition?

science is good at quantitative answers, the chief answer so far has been finding out how things work. the question facing modern science is how much intuition can we afford to let in before we begin to lose the discoveries afforded science by objectivity?

is there a new visual language out there like calculus lying in wait for newton? or are we merely performing a rehabilitation of descriptive biology with high tech pencils? we may be able to make stunning advances in math and simultaneously codify our intuitive complex actions in art to form a new hybrid math/art language. or it may be that the increasing sophistication of our art practice will afford a clearer picture of subatomic phenomena resulting in some new quantum/chemistry language.

Victoria Interrante

Is visualization a science or is it an art? Is there a science behind the art of creating an effective visual representation? How do we know how to begin designing methods for generating pictures that convey the essential information in a dataset in an accurate, efficient, and intuitively meaningful way? How do we know when we have succeeded? When we are on the right track? When we have utterly failed?

Visualization differs from art in that its ultimate goal is not to please the eye or to stir the senses but, far more mundanely, to communicate information – to portray a set of data in a pictorial form that facilitates its understanding. As such, the ultimate success of a visualization can be objectively measured in terms of the extent to which it proves useful in practice. But to take the narrow view that aesthetics don't matter is to overlook the complexity of visual understanding.

Research in perceptual psychology provides a rich source for insight into the fundamental principles underlying the creation of images that can be effectively interpreted by the human visual system. Observation of the practices of artists and illustrators provides a rich source of inspiration for the design of more complex and possibly more intuitively appealing methods for translating data into pictures.

I will present several case study examples, drawn from my research in 3D shape and flow representation, that attempt to demonstrate the potential of looking to art and illustration for insights into design of techniques for more effective visual communication. I will also discuss some of the perceptual issues that underlie the art of representing information in an accessible manner.

Visualization can be viewed as the art of creating a pictorial representation that eloquently conveys the layered complexity of the information in a complicated dataset. But it should also be viewed as the science, behind this art, of defining for others the process through which such pictures can be evolved, providing a theoretical foundation for knowing /em how to create useful images and offering insight into /em why certain representational approaches can be expected to hold more promise than others.

Felice Frankel

The following is excerpted with permission from an original essay [1].

The images I will show are photographs of scientific research, and I state that at the outset because their aesthetic qualities, being immediately apparent, so often seem to dominate initial reactions to them. But I, in fact, created them primarily to serve the scientific community, to record and communicate data, and to further the research. However, I have also recently become aware that the visual impact itself of the photographs I make in the lab can have significant consequences, allowing them to communicate important information about science research not only to other scientists in the lab, or in the field, but to a broader, nonscientific public, as well. So I have come to recognize and to embrace the two worlds my work inhabits, scientific and aesthetic. On the one hand, I bring to science photography my passionately curious, fresh and aesthetic eye. And on the other hand, though I am not an optical or electron microscopist, I use their tools, but I use them with a different point of view: to locate the innate beauty of the research, and to capture it with the kind of technical accuracy that can add information and generate new ways of thinking.

In my work I take the position that we who are privileged to see science's splendor, who image it, diagram it, model it, graph it, and compose its data, can turn the world around, dazzling it with what inspires and nourishes our thinking, if we refine the visual vocabulary we use to communicate our investigations and incorporate - beautifully and above all accurately - the visual component that is already there. Our goal must be to share the visual richness of our world, to make it accessible.

For me, form, shape and composition are integral to a scientific image or representation; I compose data, making it readable and comprehensible and the theorists and experimentalists with whom I work agree that visually clarified information adds another dimension to the exchange of ideas. They tend to be the investigators who are expanding their boundaries, sometimes into scientific disciplines of which they never dreamed. They are learning to use their equipment for visualizing the increasing complexity and dimensions of their work in new ways, with the same rigor in their imaging as in their scientific thinking; when what was once "good enough" is no longer good enough.

Although some of the images I take are displayed in art galleries and museums and are reproduced in books that resemble "art" books, they are not art. I do not view myself as an artist because an artist has a personal agenda and a very particular point of view, that of communicating the part of herself she wants the world to perceive. One may view the images I take as artistic, but their primary purpose is to communicate scientific information. My photographs are spare – compositions of three-dimensional forms and structures recorded on two dimensions. I frame the images in a way that emphasizes the particular point of the investigation, carefully choosing only the components essential for communicating a specific idea; more details do not necessarily add clarity. I find a readable order in the data, a hierarchy of information, guiding the viewer's eye to know where and how to look. If I digitally eliminate a dust particle or scratch, I indicate that I have done so. In sharp contrast, an artist is not necessarily committed to conveying data and may inadvertently subvert the essence of scientific investigation, its intellectual rigor, so to suggest that art

and science are related may dangerously redefine each. Science may be artful, but art is not scientific.

In fact, perpetuating a false connection between science and art cannot provide a permanent basis for greater public interest in science. Science itself in its wonder and beauty can attract enough attention, even if at first it is only a glance. While an amateur, in the true sense of the word, does not deeply understand science, it is a mistake to underestimate the power of enthusiasm from outside the laboratory. For example, my enthusiasm comes with enough understanding of the subject to ask the right questions, to fashion the appropriate visual vocabulary for communication, using images as scientists use equations and formulae. But then there is the more general enthusiasm from the public, whose direct contribution to research is less obvious but whose support is just as important in the long run. That enthusiasm will only expand when science is made more accessible. Accessibility is the first step to convincing the non-scientific community no longer to accept nor be content with ignorance of physical phenomena. It will encourage the confidence to curious; and that curiosity will be reason enough to look at the remarkable world we investigate, to question it, and to attempt to understand it. But first, we must all begin to *see* it.

Thomas Banchoff

Visualizing complicated surfaces in three-dimensional space demands the ability to manipulate and illuminate objects so that their essential features and their interrelationships become more and more apparent. Much more challenging is the process of trying to visualize surfaces in four-dimensional space, requiring even more views and more explorations of shapes from many different perspectives. Communicating the insights gained from visualization activities involves decisions about the best ways of presenting multiple views or animations, especially in circumstances where an object is undergoing deformations.

There is an art to making these decisions, and it is no accident that the choices made by geometers correspond in striking ways with the selections made by professional artists considering the same collections of images.

“Surfaces Beyond the Third Dimension” is the title of a one-person show at the Providence Art Club that first took place in March of 1996 in Providence RI. That exhibit lives on as a virtual art gallery on the Internet [2], and we can learn new ways of interacting with such geometric art by considering the different pieces and their relation to one another.

What have we lost when we no longer have the chance to walk through the actual physical space of the gallery? What have we gained, by allowing each viewer to interact with the various pieces at his or her own level of appreciation of the color, rendering, and shape of the displayed objects, as well as the mathematical background and context that causes these pieces to be chosen for investigation?

Does such a multi-layered gallery enhance the artistic experience of viewers, or can the amount of subsidiary information get in the way of their appreciation? Can the same objects double as art works and illustrations of mathematical relationships? Does the answer to this question depend in essential ways on the kinds of computer renderings or the kinds of mathematical objects under investigation? What lies in the future, as computer graphics opens up new areas for geometric exploration, and new views of geometric objects provide challenges both for communication and for aesthetics?

BIOGRAPHIES

David Laidlaw applies computer graphics and computer science to problems in other scientific disciplines. He is an Assistant Professor of Computer Science at Brown University. He received his Sc.B. and Sc.M. from Brown, where he worked with mathematicians to understand 2- and 3-manifolds. He received his Ph.D. in Computer Science from Caltech in 1995. His thesis presented new methods for extracting geometric models from medical imaging data of biological specimens. He is currently investigating computational methods with applications in developmental neurobiology, diagnostic medical imaging, remote sensing, and fluid mechanics. Research interests include tissue classification, visual representation of data, modeling of imaging data, optimization of data acquisition, geometry,

numerical methods, and statistics.

davidkremers is an artist inspired by the confluence of art and science, and has recently been appointed the Caltech Distinguished Conceptual Artist in Biology. He grew the first paintings from genetically engineered bacteria and his work combining living organisms and digital media has evolved into biospace station concepts and visual information systems for biotechnology research. His art has generated numerous gallery and museum exhibitions in the United States, Germany, Belgium, Denmark, and Austria. His artwork may be viewed in real time at public collections of the Denver Art Museum, the San Francisco Museum of Modern Art, the Eli Broad Family Foundation, and the Panza Collection.

Felice Frankel is artist-in-residence and research scientist at MIT. She has received grants from the Guggenheim Foundation, the National Endowment for the Arts, the Graham Foundation, the Camille and Henry Dreyfus Foundation and has recently received major funding from the National Science Foundation for her work in scientific imaging and visual expression. Her work has appeared on the covers of Nature, Science, and many other journals. Frankel is coauthor of "On the Surface of Things, Images of the Extraordinary in Science" (Chronicle Books, 1997).

Victoria Interrante is an assistant professor of computer science at the University of Minnesota. She received her PhD in 1996 from the University of North Carolina at Chapel Hill, where her dissertation research focused on the design of perceptually inspired artistic techniques for improving the comprehensibility of layered transparent surfaces in radiation therapy treatment planning data. Before coming to the U of M, she spent two years as a staff scientist at ICASE, a center of research in applied mathematics, numerical analysis and computer science operated by the Universities Space Research Association at the NASA Langley Research Center. Her broad research interests are in visualization, visual perception, computer graphics, image processing and human-computer interaction. She is currently working on applying insights from visual perception, art and illustration to the design of methods for more effectively portraying surface shape and depth in computer-generated images.

Thomas Banchoff is a geometer who has been teaching mathematics at Brown for thirty years, and collaborating during all that time with computer scientists and students to produce visualizations of surfaces in four-dimensional space. He studied mathematics and English at the University of Notre Dame and earned his Ph.D. at Berkeley in 1964. He taught two years at Harvard University and one at the University of Amsterdam before coming to Brown. With computer scientist Charles Strauss, he produced the award-winning film "The Hypercube: Projections and Slicing" in 1978, and numerous other films and videotapes, as well as sixty articles and books on geometric topics, including the Scientific American Volume "Beyond the Third Dimension". He received the national award for outstanding teaching from the Mathematical Association of America in 1996 and he is now president-elect of that organization. In 1997, the Carnegie Foundation named him Rhode Island Professor of the Year.

REFERENCES

[1] Felice Frankel, Envisioning Science – A Personal Perspective, *Science*, 280(5370) (1998).

[2] http://www.math.brown.edu/ banchoff/art/PAC-9603/welcome.html

LINE DIRECTION MATTERS: AN ARGUMENT FOR THE USE OF PRINCIPAL DIRECTIONS IN 3D LINE DRAWINGS

Ahna Girshick *Victoria Interrante** *Steven Haker** *Todd Lemoine***

Nissan Cambridge Basic Research *University of Minnesota **LambSoft

ABSTRACT

While many factors contribute to shape perception, psychological research indicates that the direction of lines on the surface may have an important influence. This is especially the case when other techniques (shading, silhouetting) do not present sufficient shape information. The psychology literature suggests that lines in the principal directions of curvature may communicate surface shape better than lines in other directions. Moreover, principal directions have the quality of geometric invariance so line directions are based on the surface geometry and are viewpoint and light source independent, and the lines do not move above over the surface during animation unless desired. In this work we describe principal direction line drawings which show the flow of curvature over the surface. The technique is presented for arbitrary surfaces represented by either 3D volume data or a polygonal surface mesh. The latter format is common in the field of computer graphics yet thus far has not been widely used for principal direction estimation. The methods offered in this paper can be used alone or in conjunction with other NPR techniques to improve artistic 3D renderings of arbitrary surfaces.

Keywords: non-photorealistic rendering, principal direction line drawings, line direction, line drawings, geometrically invariant line drawings.

1 INTRODUCTION

Amongst the varied goals of artistic Non-Photorealistic Rendering (NPR) is the pursuit of *perceptually efficient* images. A perceptually efficient visual representation emphasizes important features and minimizes extraneous detail and is essential for making comprehensible artistic images. Computer-generated line drawings are a particularly effective form of NPR since lines' features (length, width, intensity, density, quality, direction, etc.) can be combined to create shaded, textured, and expressive images which capture the essence of the form of an object. In the field of computer-generated line drawing, 3D representations of curved surfaces generally focus on the silhouette edges, disregarding large amounts of interior curvature information. These depictions often rely on either previous knowledge of the surface or the use of motion (movement of the surface, viewpoint, or light source). In this work we explore a 3D line drawing technique which is independent of the surface's orientation, the viewpoint, or the light source. In particular, we examine line direction and use this paper to raise the question: Does line direction matter?

We argue that line direction does matter, and suggest the use of the principal directions of curvature for directing lines to improve the depiction of surface shape in artistic line drawings. The advantages of principal directions (see Appendix A for a mathematical definition) are that they are geometrically-invariant, highlight the most direct path on a surface between two points, indicate the directions of the curvature extrema at any point, and have been suggested by psychologists as the preferred interpretation for making surface shape judgments.

The importance of geometric invariance should not be underestimated. Geometrically-invariant cues are based on properties of the surface geometry and are by definition viewpoint and light source independent. While shading and silhouetting provide substantial shape information, valuable curvature information can be lost in shadows or the interior of the surface. Furthermore, viewpoint dependent lines may move around in a distracting manner during motion or animation. Geometric invariance does not imply that lines must be rigidly "pasted" onto the surface during animation. If line movement is desired, the geometrically-invariant vector field can help guide more fluid movement over the surface. Combining geometrically-invariant cues with shading or silhouetting can be especially powerful. Geometrically-invariant line attributes such as color and density can be manipulated with respect to viewpoint or light source [7].

Despite the promise for principal directions, their full potential in NPR has yet to be realized. The reasons perhaps may be related to the difficulties in estimating an accurate, smoothly continuous vector field of principal directions. The problem is most challenging for polygonal surface meshes, a particularly common data format for arbitrary 3D surfaces. Additionally, principal direction line drawings must address the complex issues of creating uniformly distributed, non-intersecting, long smooth lines which gracefully traverse umbilics, planar regions, and transitions of directional dominance. Here we examine both 3D volume datasets and polygonal surface meshes, and suggest some techniques for line tracing.

Nissan Cambridge Basic Research, 4 Cambridge Center, Cambridge, MA 02139, girshick@cbr.com

*University of Minnesota, Minneapolis, MN 55455, interran@cs.umn.edu, haker@math.umn.edu

**LambSoft, Minneapolis, MN 55405, tlemoine@lamb.com

The main contribution of this work is to show that for a 3D line drawing, line direction can matter and principal direction line drawings can be used to better convey surface shape. In the next section we motivate the importance of line direction with psychological evidence. We follow with related work in computer-generated 3D line drawing. In section four, we provide a brief overview of principal direction estimation techniques. Section five shows the effects of line direction and section six presents techniques for principal direction line drawings. In the final section we draw some conclusions and discuss areas of future work.

2 PSYCHOLOGICAL EVIDENCE FOR THE IMPORTANCE OF LINE DIRECTION

The psychology literature gives us a sense of how the human visual system perceives images and is an essential reference for making perceptually efficient renderings. Early research asserted that humans can use surface markings, or texture, to perceive surface orientation. Gibson [8] was amongst the first to emphasize the significance of texture cues for shape and depth perception. He was able to show convincingly that observers could reliably interpret the slant of the planar surface by the cues provided by the projection distortion of the texture patterns.

Of relevance to this work is the open question of whether anisotropic (directed) textures are as suitable for conveying shape information as isotropic (undirected) textures. Interrante [12] was unable to show an effect of texture type in shape perception under conditions of stereo and motion for various plausible isotropic and anisotropic textures for transparent surfaces, including grids and principal direction textures. Yet Cumming et al. [3] found an indicative effect of texture type for stereoscopic shape perception between a plausible and unlikely texture. While shape-from-texture research often makes assumptions of isotropy or homogeneity, Knill [16] hypothesized that there are different modes to visually perceive isotropic and anisotropic textures.

While the question of effects of isotropic versus anisotropic texture still remains open, it is evident that when anisotropic surface markings are dependent on surface geometry, surface depth and orientation perception is improved. Knill [16] found that in an anisotropic texture processing mode, the curvature of geodesic surface markings determines perception of local surface orientation. The experiments of Johnston et al. [14] showed that stereoscopic depth perception of curved surfaces with texture which provided a good indication of surface geometry was superior to random dot textures. Stevens [24] was among the first to suggest that humans can make surface shape judgments by assuming that surface contours (lines on the surface) are aligned with the principal directions of curvature. In later work Stevens and Brookes [23] demonstrated that principal direction surface contours are also good indications of relative surface slant. More recently, Mamassian and Landy [17] found that surface shape judgments are biased by the assumption that surface contours are aligned with the principal directions. From the above literature, it is reasonable to believe that surface shape and depth perception may be generally aided by textures, and also by anisotropic textures based on surface geometry, particularly lines aligned with the principal directions

3 RELATED WORK

Computer-generated 3D line drawings borrow from centuries of artists' techniques and have recently received significant attention in the NPR community. Winkenbach and Salesin used stroke textures to create depth and shape in line drawings of parametric surfaces [26]. Markosian et al. emphasized the silhouette edges for viewpoint-dependent images of arbitrary 3D surfaces [18]. Curtis used 3D models to generate loose and artistic sketches and animations [4]. Elber rendered geometrically-invariant line drawings and textures of parametric and implicit surfaces [6].

Principal directions have been suggested [11,26] and approached [2,6] in line drawings. In [26,6], lines were traced along the parametric lines of parametric surfaces, which sometimes coincided with the principal directions. Saito and Takahashi [20] rendered line drawings lines of parametric surfaces along geodesic lines. Interrante et al. [11] used 3D principal direction textures to illustrate surface shape in volume data. However, none of these works addressed the challenge of estimating the principal directions from arbitrary surfaces (particularly polygonal surface mesh formats) nor that of tracing long strokes in one direction (rather than cross-hatching) through umbilics, planar regions, and areas of changing directional dominance. This work is based upon a preliminary sketch by Girshick and Interrante [9].

4 PRINCIPAL DIRECTION ESTIMATION

For data of any format, the first step towards a principal direction line drawing is to estimate the principal direction vector field, comprised of the principal directions at a set of points on the surface. There are a variety of methods for estimating principal directions, each with its various strengths and weaknesses, however a full discussion of the computational details is not in the scope of this paper. Do Carmo outlines analytic calculations of principal directions for parametric surfaces in [5]. For iso-intensity surfaces in 3D volume data, Monga et al. used the Hessian of the 3D data to compute the principal directions [19]. Interrante et al. used a similar technique based on Gaussian-

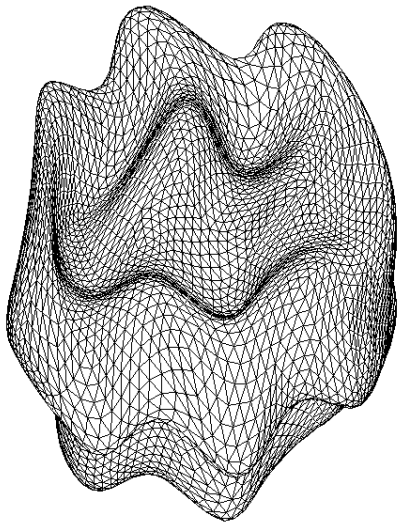


Figure 1 Polygonal surface mesh of arbitrary 3D "blob".

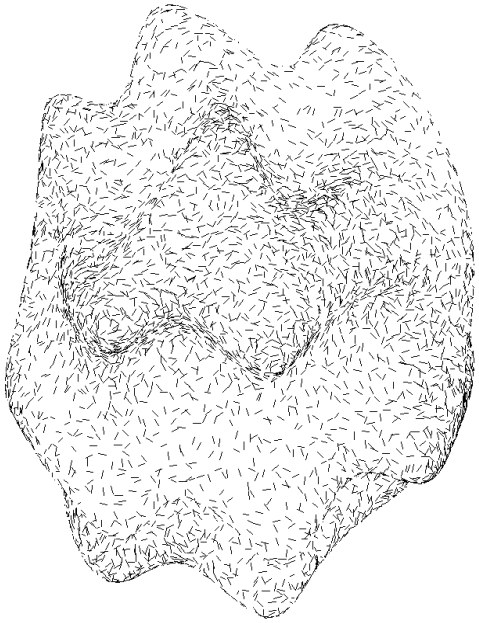


Figure 2 Random vector field of object in figure 1.

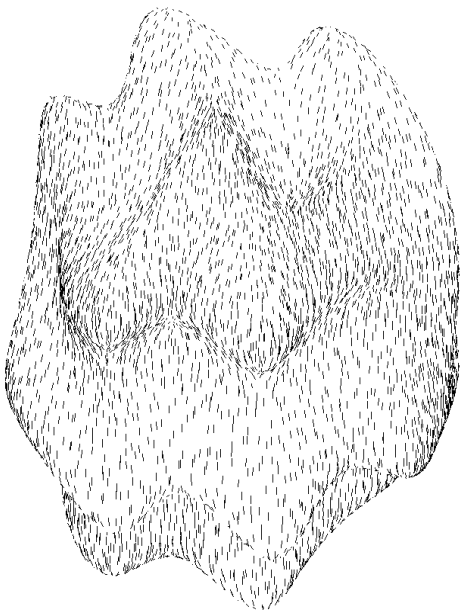


Figure 3 Uniform (vertical) vector field of object in figure 1.

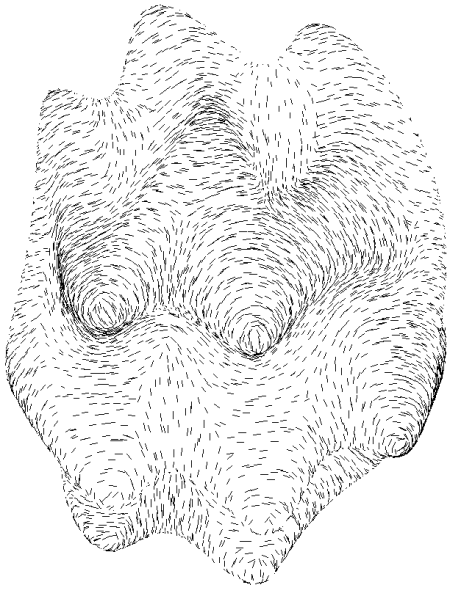


Figure 4 First principal direction vector field of object in figure 1.

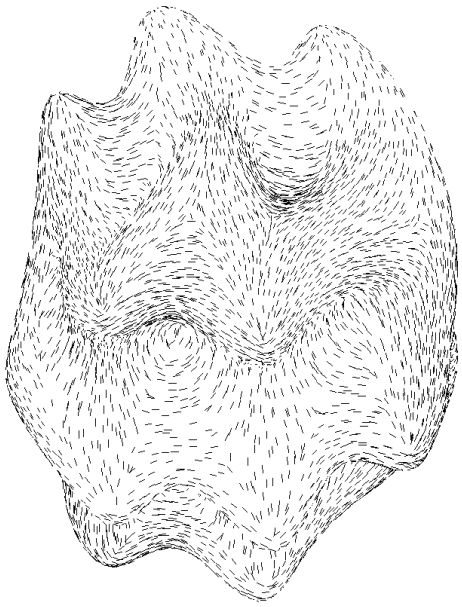
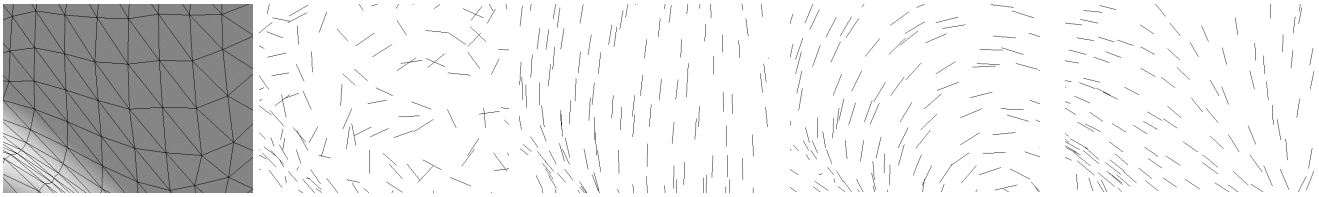


Figure 5 Second principal direction vector field of object in figure 1.



6a Shaded surface mesh **6b** Random vector field **6c** Uniform vector field **6d** First principal direction vector field **6e** Second principle direction vector field

Figure 6 Close-ups of the same region of the object in figure 1.

weighted finite-differencing [12]. We used this approach for the volume datasets in this paper.

As of yet there is no reliable standard technique for locally estimating principal directions from a polygonal surface mesh. Samson and Mallet [21] fit cubic patches to the local neighborhood around a vertex, using the vertex's normal and neighboring normals, and then compute the partial derivatives to obtain principal directions. Hamann [10] employs a similar approach except uses quadratic patches and relies solely on deviation from a vertex's tangent plane without using neighboring vertex normals. Joshi et al. provide good examples of this approach in [15]. Chen and Schmitt [1] and Taubin [25] avoid explicitly describing surface patches but instead construct a quadratic form at each vertex. In [25] the quadratic form represents an orthonormal basis whose eigenvectors are the principal directions. The principal curvatures are the directional curvatures in the principal directions. For the polygonal surface meshes in this work, we use variations of both Hamann's and Taubin's methods, with similar results. The accuracy of both is highly dependent on the symmetry of the local surface geometry and is an area of current work.

5 EFFECTS OF LINE DIRECTION

The significance of line direction for a line drawing is perhaps best illustrated visually with the underlying vector field. As will be explained in the next section, a line drawing can be rendered by tracing strokes which follow the flow of a vector field [22]. Figures 1–5 show various vector fields on the same arbitrary "blob" dataset, shown as a polygonal surface mesh in figure 1. A 3D volume dataset would produce similar results. The vector field is illustrated by projecting the field direction at each vertex of the underlying mesh onto the tangent plane at that point. The random vector field in figure 2 and the uniform vector field in figure 3 convey surface shape only through texture compression, which provides hints of the silhouette edges, but not through the use of line direction. When the silhouette edges are not visible, as in the close-ups in figures 6b and 6c, the surface shape is largely ambiguous.

Figures 4 and 5 show first and second principal direction vector field respectively. Compared to figures 2 and 3, these vector fields appear to better convey local surface orientation, including ridges and valleys, subtle surface undulations, changes in curvature, and interior silhouette edges. Figure 6 shows the close-ups of the vector fields in the absence of silhouette edges. When comparing the four close-ups in figures 6b through 6e,

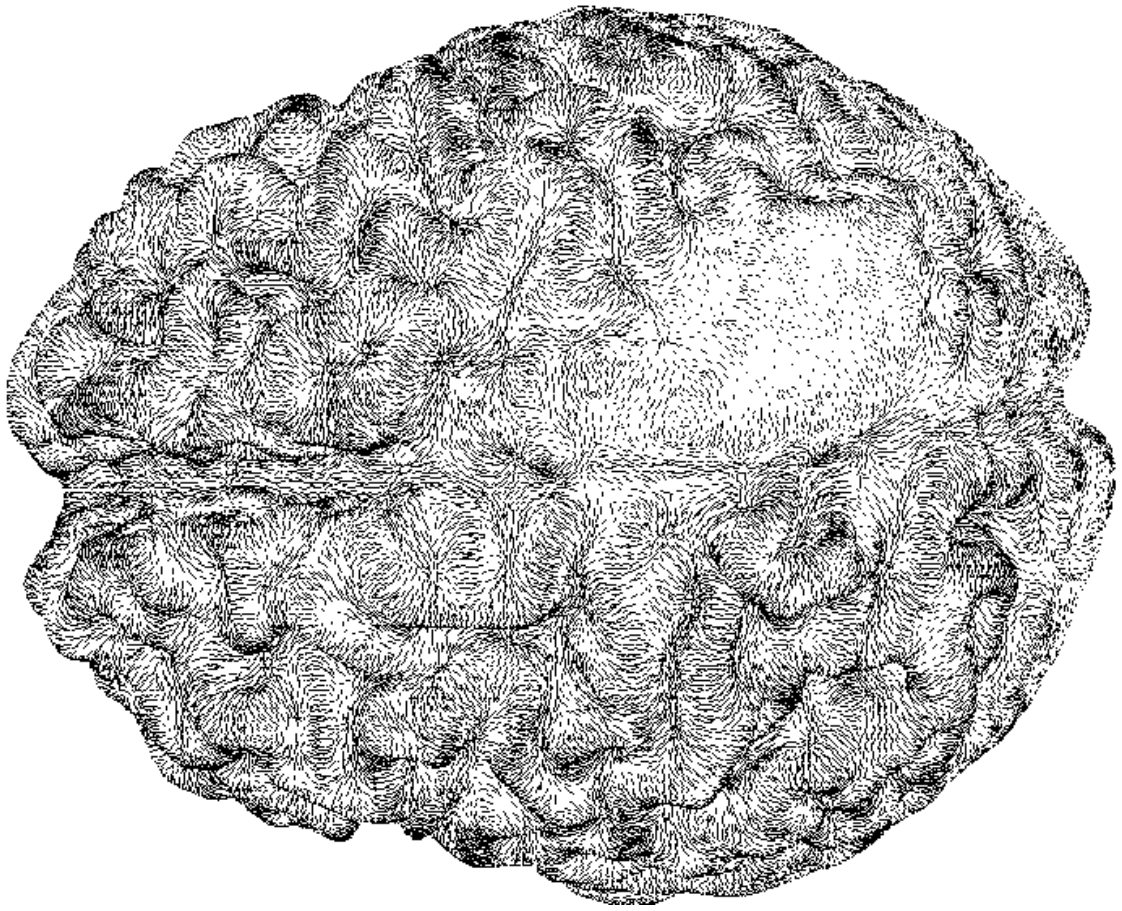


Figure 7 First principal direction vector field of a brain represented by a polygonal surface mesh. Data source: Ron Kikinis, Harvard Medical School.

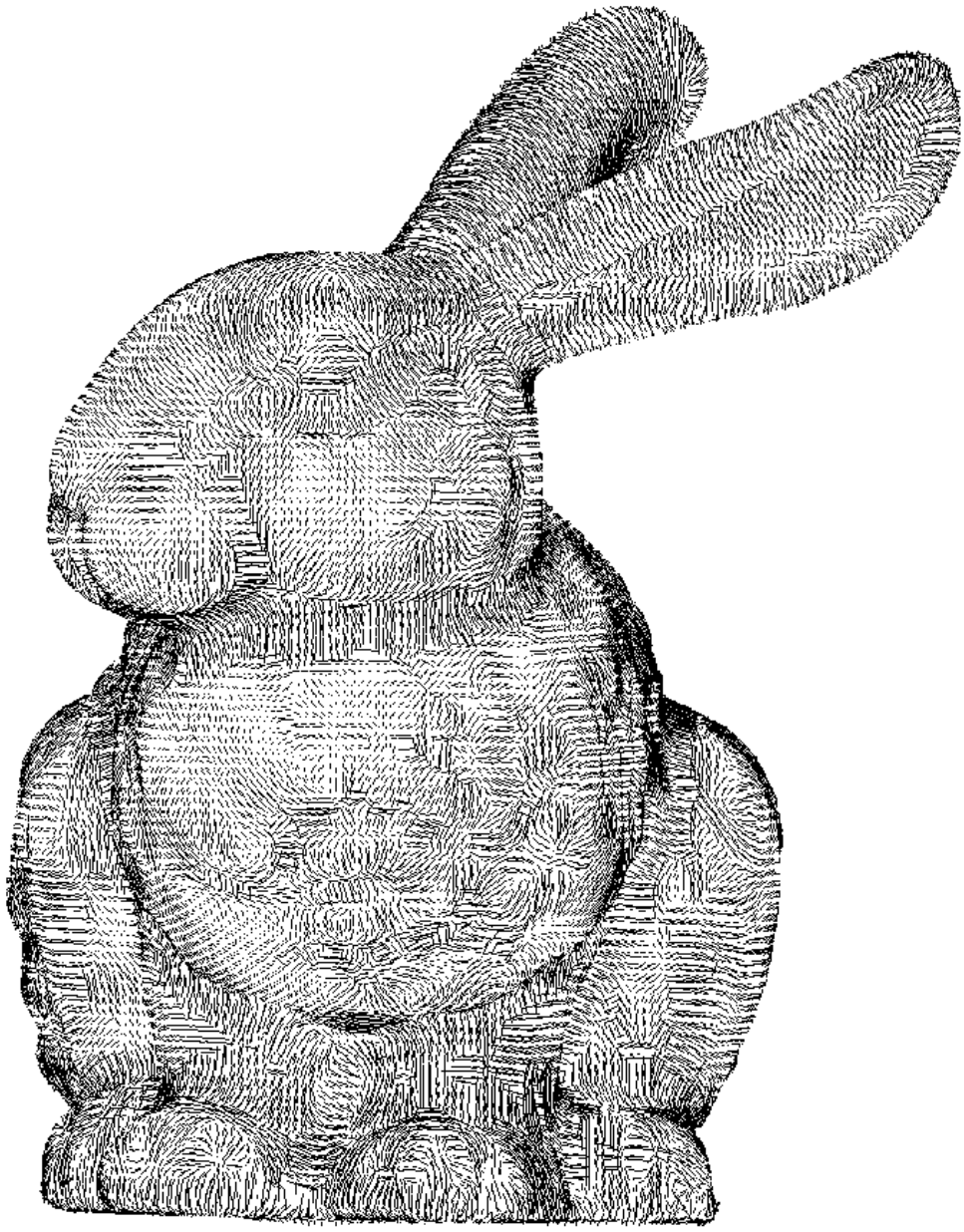


Figure 8 First principal direction vector field of a bunny represented by a polygonal surface mesh. Data source: Stanford University Computer Graphics Lab.

it seems to be easier to judge the surface shape from principal direction vector fields than the random and uniform vector fields. Figures 7 and 8 provide more examples of first principal direction vector fields on more complex surfaces. One can predict the difficulty in perceiving the surface shape if these figure used random or uniform vector fields.

6 PRINCIPAL DIRECTION LINE DRAWINGS

Principal direction line drawings illustrate the flow through the principal direction vector fields described in the previous section. In this section we describe the details for both 3D volume data and polygonal surface meshes. For 3D volume data, the vector field is a 3D volume and the strokes are traced through the volume. For polygonal surface data, the vector field lies on the explicitly defined surface mesh and the strokes must be drawn on the surface.

6.1 Principal Direction Line Drawings of 3D Volume Datasets

Figures 9 and 10 show different styles of principal direction line drawings of the same human pelvis CT volume dataset. Both figures underwent the same preprocessing stage. Initially a first principal direction volume vector field is generated using the technique described in section four. Then a sparse set of strokes

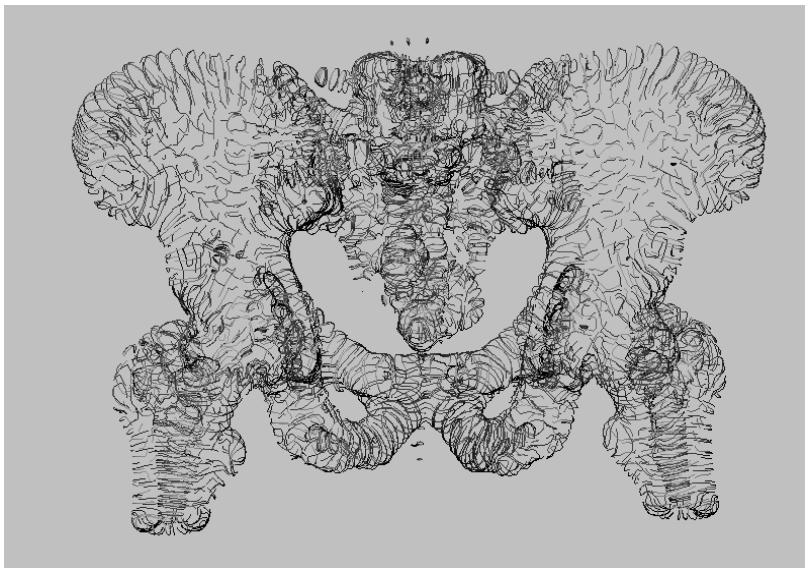


Figure 9 Principal direction line drawing (with shading and without hidden line removal) of a bone/soft tissue boundary iso-intensity surface in a CT 3D volume dataset of a human pelvis.

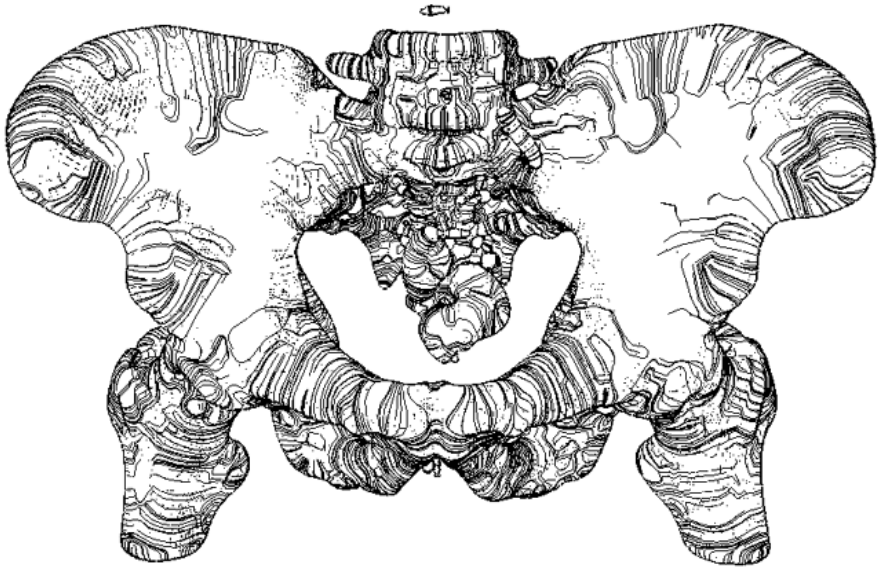


Figure 10 Principal direction line drawing with silhouette edges and hidden line removal of the volume dataset in figure 8.

is traced through the vector field, each stroke originating from a point near the surface which is not too close to neighboring starting points, such that the set has the approximate distribution of a Poisson disk.

In figure 9, the strokes represent individual streamlines [22] through the vector field. The lines are shaded according to the surface normal direction indicated by the gray level gradient in the volume data, but hidden line removal has not been done. The result is especially powerful during animation, when the geometrically-invariant lines "stick" to the surface.

In figure 10 we attempted to create a freer sketch of the volume data set, using hidden line removal, including silhouettes and selecting only a subset of possible strokes. Because it is viewpoint-dependent, by definition it is not geometrically-invariant. However the lines are still directed in the principal directions and defined based on the geometry of the surface, so the static 2D image should provide the same visual cues to the surface shape, at least near the silhouette edges. The subset of lines to render was selected with a preference towards placing lines in areas of higher curvature lines and near silhouette edges. Line length is proportional to the magnitude of the first principal curvature at the start point.

6.2 Principal Direction Line Drawings of Polygonal Surface Meshes

The main steps in creating a principal direction line drawing from a polygonal surface mesh are estimating a smoothly continuous principal direction vector field and tracing evenly spaced strokes which follow the flow of the vector field. The steps are described separately below, but for efficiency they can be done simultaneously.

6.2.1 Creating a continuous principal direction vector field

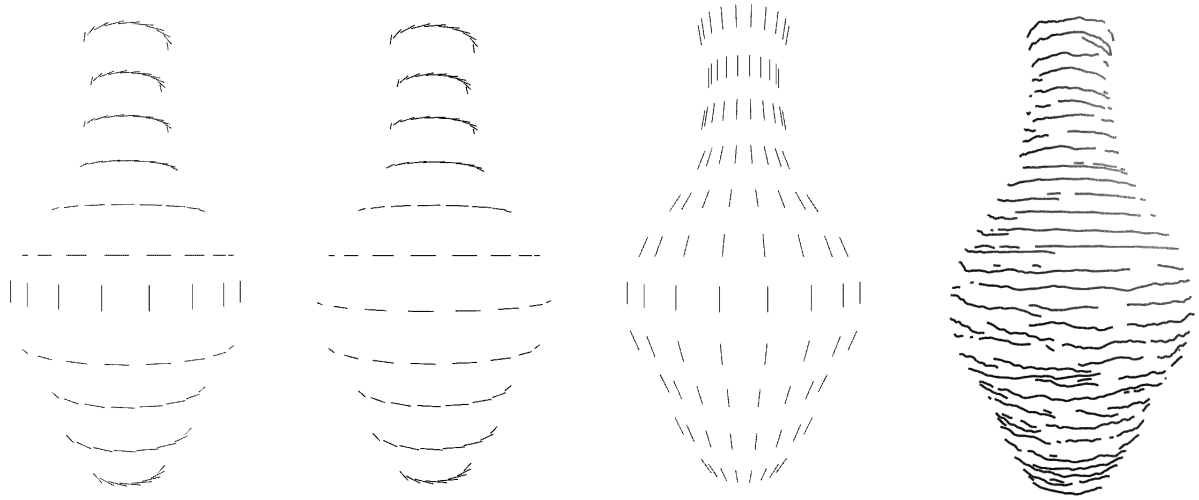
At any point on a 3D surface, each of the orthogonal first and second principal directions have a positive and negative direction. Thus there are four possible directions for the vector field at each point. Ideally we would always choose the first principal direction (either positive or negative). However, in regions close to umbilics and planes, where curvature is almost similar in all directions, the first and second principal directions may suddenly switch places causing a flip of up to 90 degrees, resulting in a sudden disruption of flow. Figure 11a demonstrates this for a simple vase mesh. The first principal direction field is continuous except around the girth of the vase where it is *almost* spherical and the curvature is slightly greater vertically than horizontally. In this case, a *continuous* principal direction line drawing minimizes distracting details and is more aesthetically pleasing than a *first* principal direction line drawing. The continuous vector field is created by first choosing an arbitrary reference vector. In the example of figure 11, the choice of reference vector can lead to only two possible outcomes, but in a more complex dataset it might be advantageous to choose a meaningful starting reference vector. Next, for each vertex, the direction which is closest to the reference vector is chosen. The reference vector is updated to reflect the choice. Figures 11b and 11c show the two possible continuous principal direction vector fields for this dataset. The principal direction line drawing corresponding to 11b is shown in 11d. This approach for creating continuous vector field works well for surface regions with well-defined principal directions. However, at true umbilics, where normal curvature is the same in all directions, and on planes, where normal curvature is zero in all directions, the principal directions are undefined. For these regions, we interpolate between neighboring well-defined regions of the vector field. Even still, for a complex surface such shown in figures 1 and 8, regions may occur where it is necessary to make an abrupt switch in line direction. A possible technique for gracefully transitioning between line directions is to minimally employ cross-hatching using both the first and second principal direction fields combined. However we do not advocate the general use of crosshairs such as in figure 12, as the inelegant crosses can become distracting and muddle the flow of curvature.

6.2.2 Tracing strokes through the vector field on a polygonal surface

The objective of this step is to obtain an approximately uniformly-distributed set of non-intersecting long curved lines, which lie on the surface. The streamline tracing technique of Jobard and Lefer [13] is extended from 2D images to 3D surfaces to generate evenly-spaced non-intersecting lines. The curvature of each line is achieved by continually redirecting it as it traverses the changing vector field.

Each stroke is composed of a set of control points. The criterion for each valid control point is that it lies at a minimum distance threshold from all existing strokes. The first stroke starting point is random, and the remaining stroke starting points are chosen to be as close as possible to existing points without breaking the minimum distance threshold.

The direction of the stroke is updated at frequent distance intervals as well as when a stroke crosses a polygon boundary. The stroke's direction at any given point on a polygon is



11a First principal direction vector field.

11b Continuous vector field of greatest overall curvature.

11c Continuous vector field of less overall curvature.

11d Continuous principal direction line drawing for 11b. Shading and slight randomness added to strokes for artistic effect.

Figure 11 Various principal direction vector fields and principal direction line drawing of a simple vase.

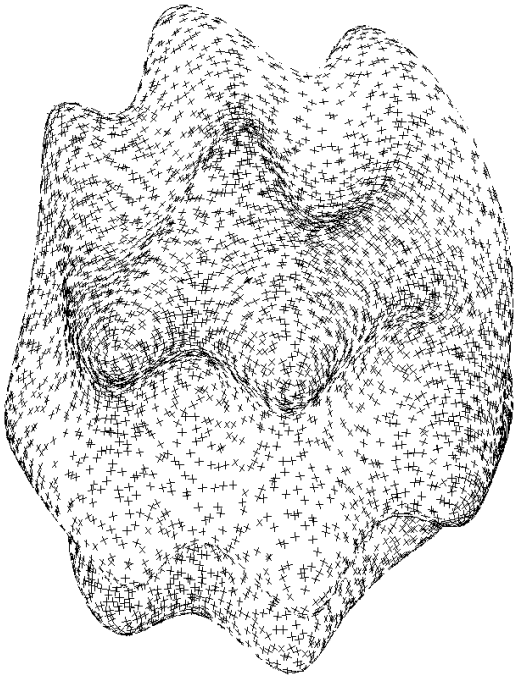


Figure 12 First and second principal direction vector field of the object in figure 1.

determined by trilinearly interpolating the principal directions of the polygon's vertices. Strokes are terminated if they approach the minimum distance threshold. This process is shown in figure 13. To avoid the cost of calculating an implicit surface, each segment of a stroke is projected onto the polygonal surface mesh. Provided a sufficiently fine mesh, this approximation is worth the savings in computation.

Regions of opposing force occur when neighboring principal directions point in opposing directions. These vector field discontinuities crop up near umbilics and planar. The current approach is to terminate strokes when this happens.

The result of this technique is shown in figure 14, with some randomness added for wiggly lines. A more artistic image might be achieved by varying the line density according to the light source, and adding silhouette lines.

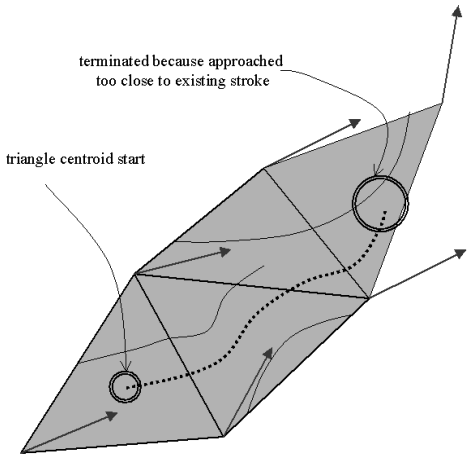


Figure 13 Stroke tracing through a principal direction vector field on a polygonal surface mesh. (image for illustrative purposes only).

Figure 14 Principal direction line drawing of pears, represented by triangular surface meshes. Hidden line removal was used, and slight random noise was added to the stroke tracing process.

6.2.3 Rendering

The rendering of the line drawing is straightforward. A stroke is a set of control points which can be rendered as either a simple polyline or spline. Our approximations were fine enough to use anti-aliased polylines with no perceivable difference over splines.

7 CONCLUSIONS AND FUTURE WORK

The most troublesome areas in obtaining a continuous principal direction line drawing are those where the principal directions are undefined and in regions of opposing force. In the first case, the current interpolation technique works well if the unknown regions are small and bordered by more well-defined principal directions, but fails for larger areas and is a topic of future work. In the latter case, we would like to eventually gracefully merge strokes from neighboring regions of opposing principal directions, possibly with subtle cross-hatching, instead of terminating them.

This work outlined the approach for both 3D volume datasets and polygonal surface. Principal direction line drawings for parametric surfaces can follow a similar approach. For polygonal surface meshes, we found the existing principal direction estimation techniques to be insufficiently accurate for asymmetric local mesh geometries. We are currently working on their improvement which is of great relevance to principal direction line drawings.

We also found that principal direction vector fields work well, in conjunction with silhouette lines or shading, as "short stroke" principal direction line drawings. An example of this is shown in figure 15. Future work includes extending these lines with the line drawing technique described above, using density variations for shading.

This work poses the important question of whether line direction matters for creating a perceptually efficient line drawing. We have provided compelling psychological evidence and visual examples to believe that line direction affects surface shape perception. In particular, the principal directions of curvature appear to be more effective than non-principal directions at conveying surface shape. Principal direction lines on a surface have the advantage that they show the path of greatest curvature and are geometrically-invariant, so they appear the same from all

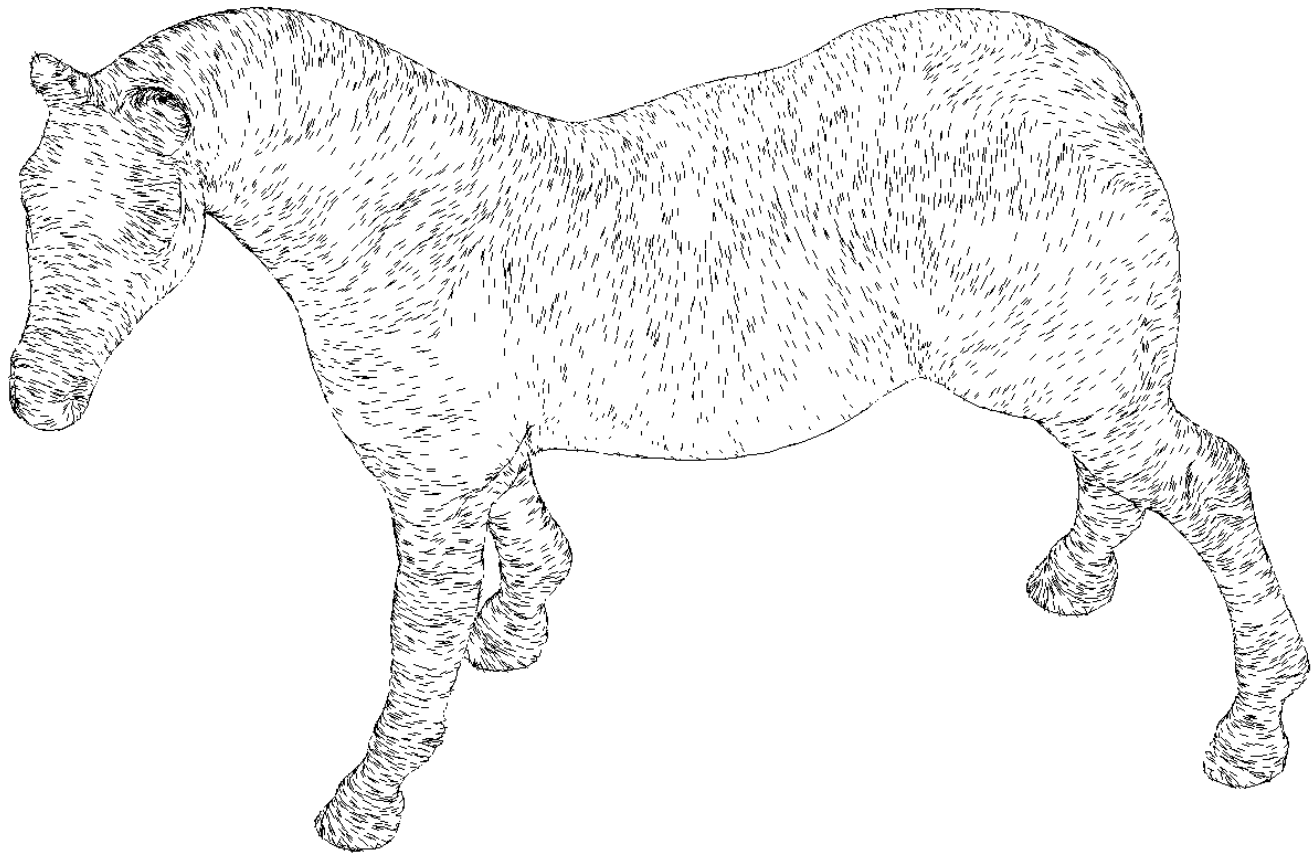


Figure 15 First principal direction vector field and silhouette lines of a horse dataset, courtesy of Cyberware, Inc.

viewpoints and do not shift during animation. Principal direction line drawings are well-suited for showing the subtle undulations of an arbitrary, smoothly curved surface in 3D, especially when silhouette edges are not visible. They can be used alone or in conjunction with other graphics techniques such as shading and drawing silhouette edges. One intention of this work is to serve as a reminder that perceptually efficient images are an important part of artistic NPR. We also wish to inspire more perceptual studies of the effectiveness of principal direction line drawings.

8 ACKNOWLEDGMENTS

This work was supported by a Grant-in-Aid of Research, Artistry, and Scholarship from the University of Minnesota and NSF Grant CCR-9875368. Special thanks to Detlev Stalling, Mireille Boutin, Ron Kikinis, and Erwin Boer.

9 APPENDIX A: DEFINITION OF PRINCIPAL DIRECTIONS OF CURVATURE

The normal curvature at p in a given direction T will be referred to as the directional curvature $\kappa_p(T)$. The *first principal direction*, T_1, is the direction of the maximum magnitude of normal curvature, called the *first principal curvature* (κ_p^1). The *second principal direction*, T_2, is orthogonal to the first, and is the direction of the other curvature extreme, called the *second principal curvature* (κ_p^2). For elliptic surface patches (with positive Gaussian curvature) the second principal direction is the direction which the surface is most nearly flat. For hyperbolic, (saddle-shaped) patches (with negative Gaussian curvature), the second principal direction is the direction of the lesser of the two extrema. The two principal directions T_1 and T_2 are orthogonal and lie in the tangent plane at the point p, creating an orthonormal basis with the normal vector N at p. Figure 14 shows an example of the orthonormal basis on a hyperbolic surface patch. The product of the two principal curvatures equals the Gaussian curvature, $K=\kappa_p^1 \bullet \kappa_p^2$.

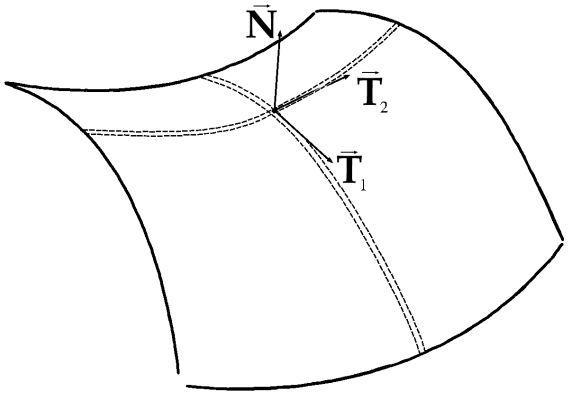


Figure 16 Orthonormal basis formed by normal and two principal directions and curvature strips in the principal directions at a point on a hyperbolic patch.

10 REFERENCES

[1] X.Chen and F.Schmitt. Intrinsic Surface Properties from Surface Triangulation. *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 739-743, 1992.

[2] R.Coutts and D.Greenberg. Rendering with Streamlines. *Visual Proceedings (SIGGRAPH 97)*, p.188, 1997.

[3] B.Cumming, E.Johnston and A.Parker. Effects of Different Texture Cues on Curved Surfaces Viewed Stereoscopically. *Vision Research*, 33(5/6):827-838, 1993.

[4] C. Curtis. Loose and Sketchy Animation. *Visual Proceedings (SIGGRAPH 98).*

[5] M.do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice Hall, 1976.

[6] G.Elber. Line Art Illustrations of Parametric and Implicit Forms. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):77-81, January-March 1998.

[7] G.Elber. Interactive Line Art Rendering of Freeform Surfaces. *Proceedings of Eurographics 99*, July 1999.

[8] J.Gibson. *The Perception of the Visual World*, Houghton Mifflin, 1950.

[9] A.Girshick and V.Interrante. Real-time Principal Direction Line Drawings of Arbitrary 3D Surfaces. *Visual Proceedings (SIGGRAPH 99)*, p. 271.

[10] B.Hamann. Curvature Approximation for Triangulated Surfaces. *Geometric Modelling : Dagstuhl*, pp.139-153, Springer-Verlag, 1993.

[11] V.Interrante. Illustrating Surface Shape in Volume Data via Principal Direction-Driven 3D Line Integral Convolution. *Computer Graphics (Proceedings of SIGGRAPH 97)*, pp.109-116, 1997.

[12] V.Interrante, H.Fuchs and S.Pizer. Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):98-117, April-June 1997.

[13] B.Jobard and W.Lefer. Creating Evenly-Spaced Streamlines of Arbitrary Density. *Proceedings of 8th Eurographics Workshop on Visualization in Scientific Computing*, pp.45-55, 1997.

[14] E.Johnston, B.Cumming and A.Parker. Integration of Depth Modules: Stereopsis and Texture. *Vision Research*, 33(5/6):813-826, 1993.

[15] S. Joshi, J. Wang, M. Miller, D. Van Essen, U. Grenander. On the Differential Geometry of the Cortical Surface. *Vision Geometry IV (Proc. SPIE's 1995 International Symposium on Optical Science, Engineering, and Instrumentation)*, 2573:304-311, 1995.

[16] D.C.Knill. From Contour to Texture: Static Texture Flow is a Strong Cue to Surface Shape. *Perception* 26 *(Supplement: ECVP '97 Abstracts)*, p.111,1997.

[17] P.Mamassian and M.Landy. Observer Biases in the 3D Interpretation of Line Drawings. *Vision Research*, 38:2817-2832, 1998.

[18] L.Markosian, M.Kowalski, S.Trychin, L.Bourdev, D.Goldstein, and J.Hughes. Real-Time Nonphotorealistic Rendering. *Computer Graphics (Proceedings of SIGGRAPH 97)*, pp.415-420, 1997.

[19] O.Monga, S.Benayoun, and O.D.Faugeras. From Partial Derivatives of 3D Density Images to Ridge Lines. *Proceedings of the 1992 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 354-359, 1992.

[20] T.Saito and T.Takahashi, Comprehensible Rendering of 3D Shapes. *Computer Graphics (Proceedings of SIGGRAPH 90)*, pp.197-206, 1990.

[21] P.Samson and J.Mallet. Curvature Analysis of Triangulated Surfaces in Structural Geology. *Mathematical Geology*, 29(3):391-412, 1997.

[22] D.Stalling. Personal communication, 1997.

[23] K.Stevens and A.Brookes. Probing Depth in Monocular Images. *Biological Cybernetics*, 56:355-366, 1987.

[24] K.Stevens. The Visual Interpretation of Surface Contours. *Artificial Intelligence*, 17:47-73, 1981.

[25] G.Taubin. Estimating the Tensor of Curvature of a Surface from a Polyhedral Approximation. *Proceedings of the 5th International Conference on Computer Vision (ICCV)*, pp.902-907, 1995.

[26] G.Winkenbach and D.Salesin. Rendering Parametric Surfaces in Pen and Ink. *Computer Graphics (Proceedings of SIGGRAPH 96)*, pp.496-476, 1996.

Visualization Viewpoints

Editors: Theresa-Marie Rhyne and Lloyd Treinish

Harnessing Natural Textures for Multivariate Visualization

Victoria Interrante

University of Minnesota

In our ongoing quest to convey more information more clearly in a single image, harnessing the full potential of texture for data representation remains an elusive goal. Others have begun excellent work in this area,[1-3] and my efforts are inspired by their example. The grail that I seek is a partially ordered multidimensional palette of richly detailed and varying texture patterns that can be used—in conjunction with lightness and hue—to represent multivariate information. The goal is to facilitate the flexible visual appreciation of the correlations of various quantities across the different dimensions. The approach that I outline here departs a bit from the norm, but is motivated by a desire to proceed more directly from my vision of what I want to achieve, unrestrained by the limitations of the tools I have on hand. In the following discussion, I motivate the adoption of rich, natural textures—resembling those from photographic images[4]—as elemental primitives and sketch some of the approaches that we can take to enhance our understanding of how to effectively harness their properties. My intent here is not to present results, but to expound on the issues and conclude with the questions to which we're still seeking answers.

Why natural textures?

The intricate variety and subtle richness of detail of texture patterns found in nature support possibilities for data representation far more vast and comprehensive than we could ordinarily hope to achieve from standard primitives. Even if we must ultimately rely on synthesized textures for data visualization, by looking to nature for inspiration we have the potential to expand our vision of what to strive for in such a synthesis. The graphic design community has long held that perfectly regular synthetic textures on a flat plane, in particular the infamous hatching patterns that Edward Tufte refers to as "chart junk,"[5] are discomforting to the eye and annoying to look at. Natural textures are not only more aesthetic, but they also put less extraneous stress on the visual system, leaving our eyes freer to observe and attend to the most intrinsically important texture-pattern characteristics.

Understanding human texture perception

To create a perceptually meaningful multidimensional *texture space* that can be indexed in the same fashion as a color space, we must begin by knowing what we're looking for. We need to proceed from a rigorous and experimentally supported understanding of how human observers perceive and interpret texture patterns, under the conditions in which we intend for these patterns to ultimately be viewed. This grounding provides a structure for guiding our search through the complex space of possibilities and formalizes the intuition that a good designer calls upon to create a visualization that works.

A number of researchers[6] have conducted studies to try to elucidate the most significant perceptual dimensions of texture. The results of these experiments will aid us, though some important questions remain. It's beyond the scope of this article to summarize previous findings further than to say that most of the studies used unaltered images from the Brodatz album, subjects were generally asked to cluster the textures into groups, and there appears to be general agreement that a small number (about three) of characteristic dimensions seem sufficient to describe most of the structure underlying this classification. The interpretation of the dimensions varies from study to study, but most often includes aspects of the following:

- periodic (consisting of repeated discrete elements) <-> nonperiodic
- strongly directional <-> rotationally invariant
- coarse <-> fine (spatial frequency of the dominant detail)
- regular (deterministic) <-> random
- high contrast <-> low contrast
- homogeneous (spatially invariant) <-> heterogeneous

Clearly, there is some overlap in these categorizations. Also, it's not evident that we can hope to determine an orthogonal basis that encompasses all members of the texture pattern set. However, the apparent low perceptual dimensionality of the space and the strong agreement between the studies bodes well for our application.

Open questions in visual texture perception

Identifying the features according to which people tend to classify texture patterns gives us important insight into how to structure a perceptually meaningful texture space.

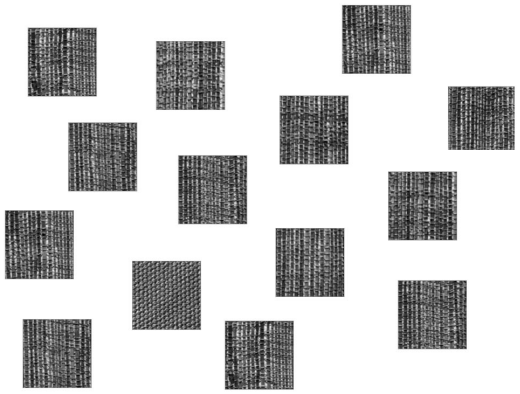
But some important issues have been implicitly overlooked in the studies conducted so far. Foremost are the uncontrolled-for influences of higher level processes.

Payne et al.[7] observed, in studies similar to those referenced above, that material property categories appeared to have a strong influence on a fair number of their observers' clustering decisions. They also noted that their subjects often commented that they felt they were using different criteria to find matches for different texture images. Knowing that people tend to make judgments based on different criteria in different cases, but not knowing who is considering what when or why, weakens the general applicability of experimental results based on these traditional methodologies. I believe that it might be useful to attempt to control for, or at least quantify the impact of, some of these effects by considering alternative paradigms for objectively measuring texture patterns' perceptual similarity.

The question of whether to control for rotation, scale, luminance, and contrast variance among texture samples when seeking insight into the perceptual groupings of texture images is a second important issue that studies using the Brodatz album have frequently overlooked. How do we want to consider apparent texture pattern differences that aren't clearly intrinsic to the pictured material but that could conceivably be attributed to external factors such as viewpoint or lighting?[8]

On the one hand, Ware and Knight[3] have said that orientation, size, and contrast are the primary orderable dimensions of texture. But samples of a texture that are differently oriented, scaled, and lit still intrinsically appear to be the same thing and thus remain good candidates for similarity grouping. As pre-attentive features of individual elements, size, contrast, and orientation differences are undisputedly important in facilitating "pop-out". At the same time, our visual system is remarkably adept at maintaining perceptual constancy across changes in illumination or viewpoint. Many computational methods for classifying texture adopt a rotationally invariant texture recognition approach for this reason. A strong argument exists for equalizing characteristics such as scale, luminance, and contrast before classifying or quantifying texture-pattern differences for the purposes of visualizing multiple distributions across a 2D image. Doing so would let us retain the ability to introduce variations in these feature dimensions universally across all the other texture dimensions and use them to encode additional values. To this end, it might be illuminating to try to separately examine the relative effects of rotation, scale, and contrast differences versus other texture-characterizing differences. Our considerations will also differ in the case of visualizing distributions across surfaces through a 3D domain.

The question of how best to factor out the variations in contrast and luminance—when we choose to do so—must also be carefully considered. Although it's relatively straightforward to equalize the intensity histograms of sample texture images before processing them for similarity, it's not clear that histogram equalization adequately preserves the meaningful qualities of the texture patterns.



1 The salience of differences in various properties of texture patterns might be estimated through psychophysical experiments using briefly presented stimuli such as these.

Quantifying the perceptual similarity of texture patterns

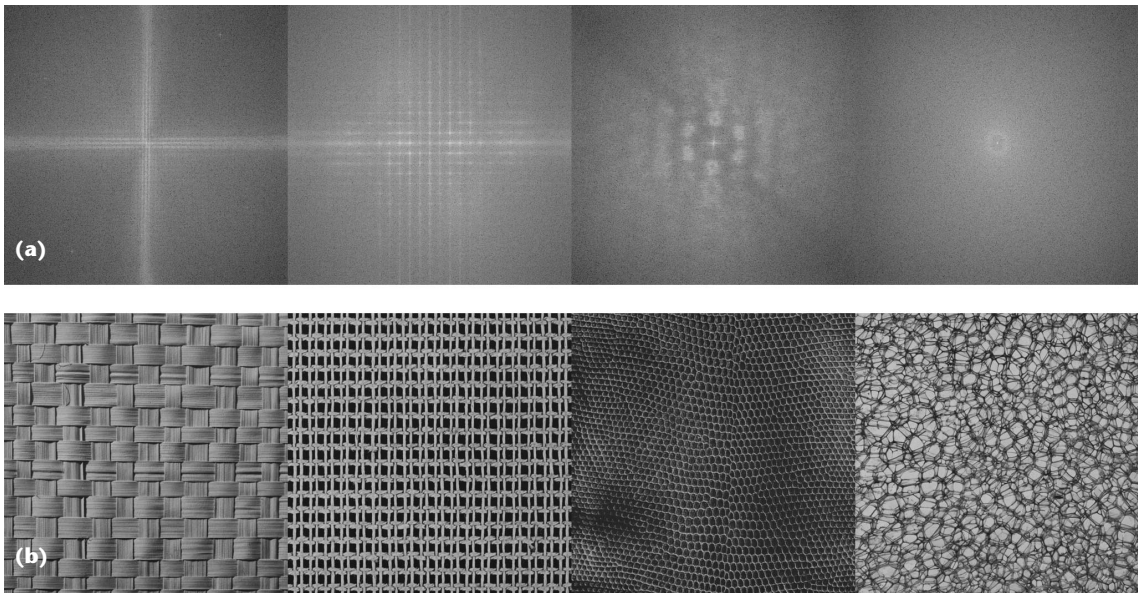
In addition to determining which textures tend to cluster, it's important for creating a perceptually linear texture space to quantify the perceptual distances between individual texture patterns. It's also necessary to estimate the magnitude of the perceived distance due to the differences along each of the feature dimensions.

One possible approach is to estimate the magnitude of the change required to enable a "just noticeable difference" between images along individually selected texture dimensions such as scale, contrast, orientation, regularity, and so on using psychophysical methods.

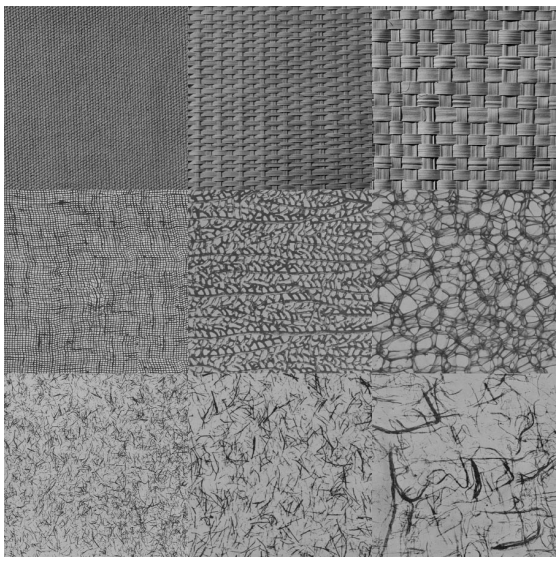
Another possibility, which may be more appropriate for judging the kinds of differences that cannot be easily brought down to threshold levels, is to measure the pre-attentive discriminability or salience of differences in features of individual texture patterns randomly embedded in homogeneous and heterogeneous fields of distracters, as illustrated in Figure 1. The objective in this case then is to determine how large of a difference is required to allow the effortless identification of the "odd man out" in brief, masked stimulus presentations. Studies using individual element arrays have found that salience (or the tendency to "pop-out") tends to increase when the targets are characterized by redundant, unique properties such as luminance and hue or color and orientation.[9] Similarly, the salience of the target tends to decrease as the heterogeneity of the distracter elements increases, even when the heterogeneity occurs along a different perceptual dimension.

It may additionally be of interest to determine how many different texture types people can simultaneously discriminate, using a methodology similar to what Healey employed for studying color.[10]

The long and rich history of research on texture classification algorithms in the image processing community also offers valuable resources for constructing a texture palette. Although the extent and variety of the possible computational approaches for classifying texture patterns is somewhat overwhelming, most of the successful methods work by extracting a finite set of features from the texture patterns (via transforms similar to those shown in Figure 2) and then calculating various statistics across these feature sets. Rubner and



2 (a) Statistics of texture pattern features, such as those evident in this set of power spectra images, form the basis for many computational texture characterization methods. (b) It's easy when looking at the original images to form an intuitive understanding of the types of information carried by the various features of the Fourier coefficients.



3 A small potential texture palette. Scale increases along the horizontal axis, regularity increases along the vertical axis, and intensity increases along the left-to-right descending diagonal.

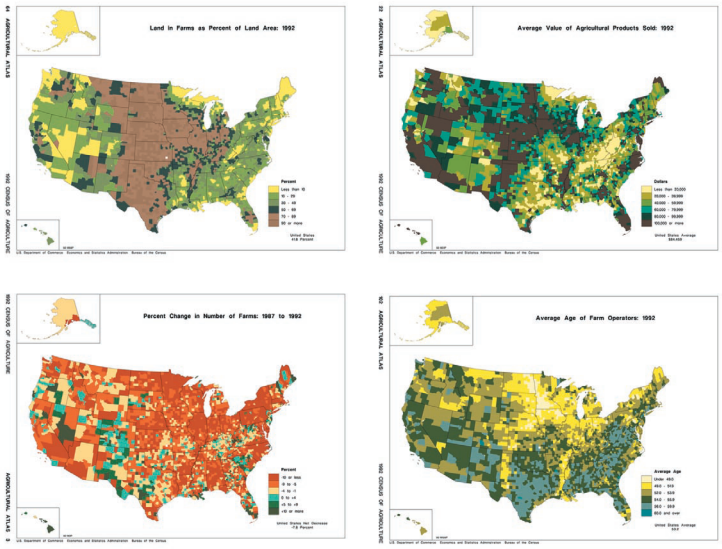
Tomasi[8] proposed quantifying texture similarity by using the Earth Mover's Distance as a metric for the goodness of fit between histograms of these features. A key issue for us is ensuring that the computer vision results and the human observer criteria agree.

Creating a texture palette

One approach to creating a texture palette (Figure 3) is to begin with a collection of well-chosen input images. You can then objectively determine where they lie in the best-fitting multidimensional texture feature space that they span and appropriately fill in the remaining open space with intermediate textures that lie at equal perceptual distances along each of the dimensions.

Some of the difficulties in creating a texture palette are that the texture space may not be orthogonal, there may be interaction among certain dimensions (such as contrast and spatial frequency), and some texture type mixtures may not be meaningful. However, my intuition is that the closer we can get to aesthetically filling out the space with real acquired images, the easier it will be to patch the holes.

Texture synthesis methods such as those that Portilla and Simoncelli[11] and Zhu et al.[12] have proposed may hold the greatest potential for creating a set of sample textures that fill out a multidimensional palette. Working from these frameworks, we may have the possibility to deterministically create natural-looking intermediate textures that interpolate the characteristic properties of their neighboring swatches in texture space or that



4 A set of four agricultural maps,[14] representing distributions of four different variables across the US counties.

conform to a prespecified set of desired criteria. However, these methods currently invoke too many features (roughly 800 plus) and are not quite mature enough to guarantee the realism that we seek.

Efros and Leung[13] suggest a very different synthesis method for seamlessly generating highly realistic samples of "more of the same" textures from a given sample. The success of this method is subject to the assumption of stationarity and a reasonable estimate of the extent of the lowest spatial frequency detail that must be preserved. However the feasibility of extending this approach to texture interpolation has yet to be shown.

Another issue we face is how to properly deal with the lighting consistency problems that will inevitably arise when we try to combine acquired textures that exhibit relief. It may be necessary to solve for the surface relief, allowing the material intensity texture to be handled separately. In the 2D case, the most important consideration is simply to maintain consistency. Additionally, it's desirable to avoid orienting textures so that to preserve convexity, observers must envision the light as coming from below.

Feasibility issues

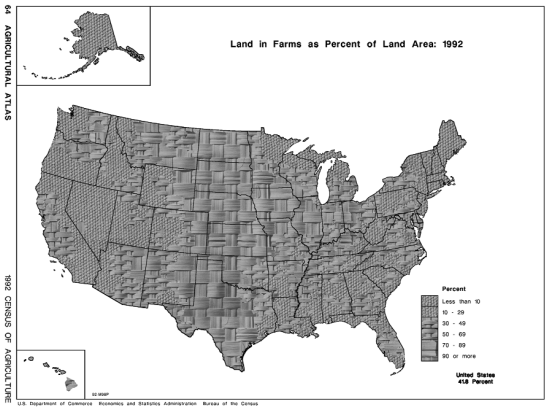
Some of the questions that we need to address are

- What does a reasonable partitioning of a natural texture space look like?
- Would it be feasible to try to choose exemplars at the endpoints of each perceptually relevant texture dimension, characterize them statistically, then interpolate to obtain intermediate textures that fill out the space?
- To what extent do we need to guarantee that different textures will meld continuously into each other at the transitions between level set regions?
- How can we most effectively combine color with texture to convey yet more information in a meaningful way?

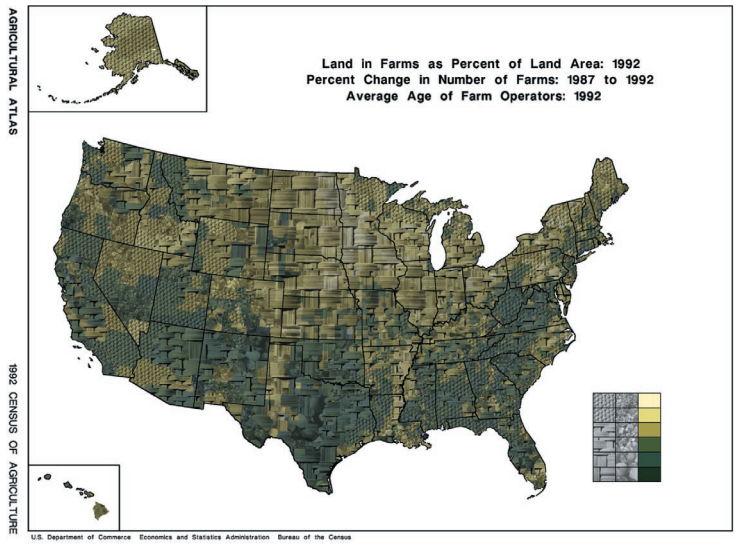
Clearly, texture has the greatest potential to be effective as a tool for visual data representation when it conveys local values of an underlying function across homogeneous cells of sufficient size to allow the characteristic detail of the resident texture to be discriminated.

The philosophy behind my attempt to more effectively harness the potential of texture for multivariate data visualization is that we should begin from a vision of what we want to achieve and work from there to figure out how to accomplish the desired results. To more freely explore the possibilities before becoming bogged down in implementation issues, I mocked up several tests by hand using Adobe Photoshop and scanned images from the Brodatz texture album[4] to represent multivariate agricultural data[14] at the county level (Figures 4 through 7).

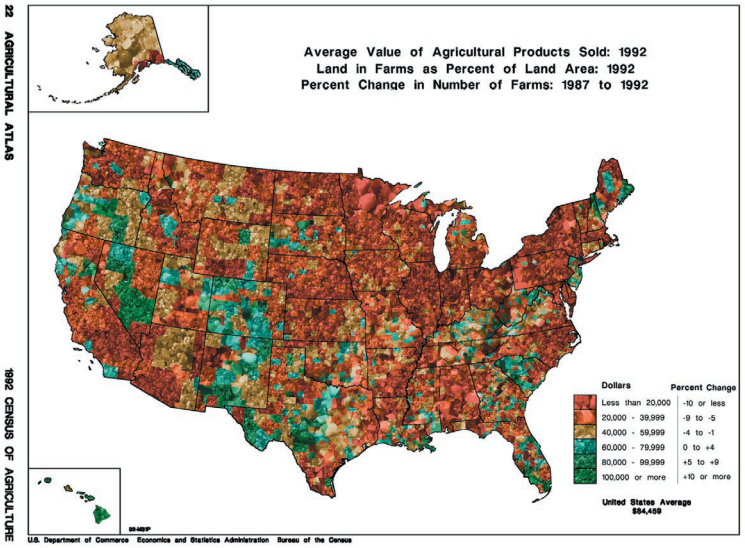
In Figure 7, texture scale represents one of three



5 A simple hand-crafted demonstration of a display in which three natural textures of increasing scale convey three discrete ranges in the value of a variable along a single



6 A hand-crafted example in which one dimension of color and two dimensions of texture overlap to represent three different variables in each cell.



7 A second hand-crafted example in which one dimension of color and two dimensions of texture represent three different variables in each cell.

ranges of the average value of the agricultural products produced in each county (with the largest scale corresponding to the lowest value, evoking the metaphor of inhospitable terrain). The texture type represents the direction of change in the amount of land in farms (with rocks representing areas with an overall loss of farmland and weaves representing areas with an overall increase in the amount of land used for farming). Color labels the percentage of the total land area in each county used for farming (with the green tones indicating the higher percentages and the brown tones representing the lower percentages of land occupied by farms).

Representing uncertainty

Both color and texture admit intriguing possibilities for representing uncertainty in data measurements. Texture regularity has particularly good potential as an intuitive marker for certainty, with texture pattern irregularities increasing in prominence where measurement reliability is lower. Color regularity has somewhat weaker potential as a marker for measurement certainty, as textures that contain more balanced or restricted distributions of hues may appear less distinct or vibrant (implying decreased noteworthiness) than textures in which the hues vary more widely across the spectrum. Figures 8 and 9 provide examples of these two different configuration series. It remains to be seen how easily irregularity of the kind shown in Figure 9 can be incorporated into a computational texture synthesis definition for such patterns.

Conclusions

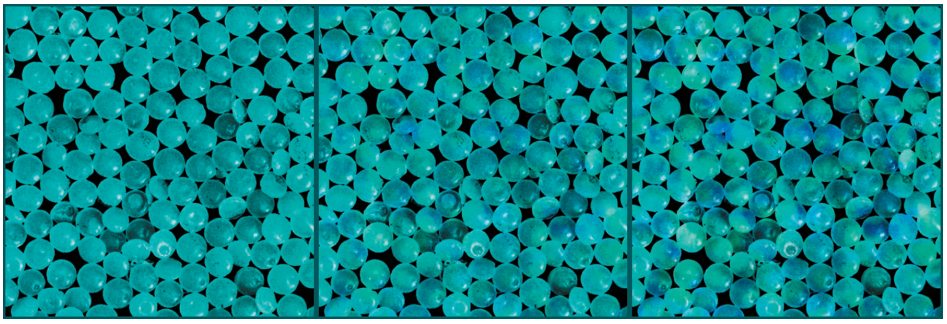
Despite the excellent progress made in recent years,[1-3] I believe that there remains great untapped potential for the effective use of texture in multivariate visualization. I have proposed that we might take important steps towards realizing more of this potential by attempting to harness the power of rich natural textures. I envision that a successful approach will begin from a fundamental understanding of visual texture perception and progress toward an understanding of how to synthesize a multidimensional palette of detailed texture samples whose variations evoke an intrinsic appreciation of the local and global relationships between multiple quantities across a 2D domain. ■

Acknowledgments

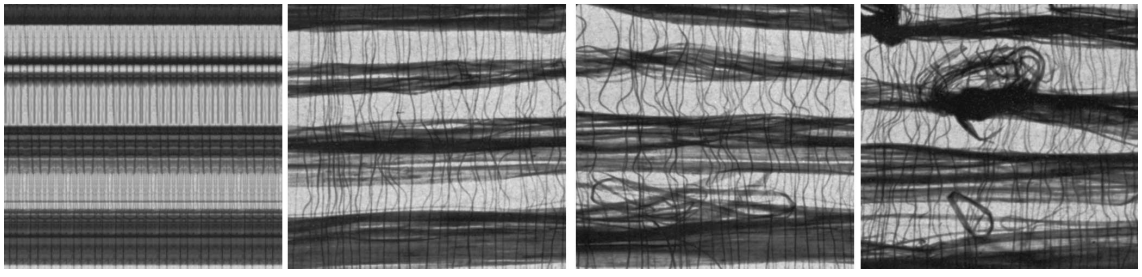
In framing this work, I have benefited from informal discussions with a number of different people. I am particularly grateful to Tomás Filsinger for providing valuable criticism and insights into understanding effective uses of texture from an artist/graphic designer's perspective, and to Guillermo Sapiro and Dan Kersten for pointers into the computer vision and texture synthesis literature. My fledgling efforts are partially supported by a 1999-2000 University of Minnesota Grant-in-Aid of Research, Artistry, and Scholarship and an NSF Presidential Early Career Award for Scientists and Engineers (Pecase) award.

References

1. C.G. Healey and J.T. Enns, "Large Datasets at a Glance: Combining Textures and Colors in Scientific Visualization," *IEEE Trans. Visualization and Computer Graphics*, Vol. 5, No. 2, 1999, pp. 145-167.
2. D.H. Laidlaw et al., "Visualizing Diffusion Tensor Images of the Mouse Spinal Cord," *Proc. IEEE Visualization 98*, ACM Press, New York, 1998, pp. 127-134.
3. C. Ware and W. Knight, "Using Visual Texture for Information Display," *ACM Trans. Graphics*, Vol. 14, No. 1, 1995, pp. 3-20.
4. P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover Publications, New York, 1966.
5. E.R. Tufte, *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, Conn., 1983, p. 111.
6. A.R. Rao and G.L. Lohse, "Identifying High Level Features of Texture Perception," *CVGIP: Graphical Models and Image Processing*, Vol. 55, No. 3, 1993, pp. 218-233.



8 Variability in hue might be used to convey the range of imprecision or uncertainty in the data value recorded for a region.



9 Increasing irregularity in a texture pattern can intuitively represent increasing amounts of uncertainty.

7. J.S. Payne, L. Hepplewhite, and T.J. Stonham, *Texture Similarity: Using Human Studies to Interpret Retrieval Results*, Tech. Report TR-685, Dept. Electronic and Computer Engineering, Brunel University, UK, 1997.
8. Y. Rubner and C. Tomasi, "Texture Metrics," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics*, IEEE Press, Piscataway, N.J., 1998, pp. 4601-4607.
9. H-C. Nothdurft "Salience from Feature Contrast: Additivity Across Dimensions," *Vision Research*, Vol. 40, 2000, pp. 1183-1201.
10. C.G. Healey, "Choosing Effective Colors for Data Visualization," *Proc. IEEE Visualization 96*, ACM Press, New York, 1996, pp. 263-270.
11. J. Portilla and E.P. Simoncelli, "A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients," *Int'l J. of Computer Vision*, 2000, to appear; currently available on the Web at http://www.cns.nyu.edu/ftp/eero/portilla99.pdf.
12. S.-C. Zhu et al., "Exploring Texture Ensembles by Efficient Markov Chain Monte Carlo—Toward a 'Trichromacy' Theory of Texture," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 6, 2000, pp. 554-569.
13. A.A. Efros and T.K. Leung, "Texture Synthesis by Nonparametric Sampling," *IEEE Int'l Conf. Computer Vision, Volume II*, IEEE Press, Piscataway, N.J., 1999, pp. 1033-1038.
14. *1992 Agricultural Atlas of the United States*, U.S. Census Bureau Geography Div. Map Gallery, http://www.census.gov/geo/www/mapGallery.

Readers may contact Interrante at the Dept. of Computer Science and Engineering, University of Minnesota, 4-192 EE/CS Bldg., 200 Union St. SE, Minneapolis, MN 55455, e-mail interrante@cs.umn.edu.

Readers may contact department editors Rhyne and Treinish by e-mail at rhyne.theresa@epamail.epa.gov and lloydt@us.ibm.com

Correction

In the September/October issue, the Visualization Viewpoints article "Visualizing Visualizations: User Interfaces for Managing and Exploring Scientific Visualization Data" by Kwan-Liu Ma incorrectly implied that Fritz Hasler (NASA Goddard) alone worked on the Distributed Image Spreadsheet (p. 17). In fact, the work was done by both Hasler and Kannappan Palaniappan (University of Missouri-Columbia).

Investigating the Effect of Texture Orientation on the Perception of 3D Shape

Victoria Interrante and Sunghee Kim

Department of Computer Science and Engineering
University of Minnesota

ABSTRACT

Perception of the 3D shape of a smoothly curving surface can be facilitated or impeded by the use of different surface texture patterns. In this paper we report the results of a series of experiments intended to provide insight into how to select or design an appropriate texture for shape representation in computer graphics. In these experiments, we examine the effect of the presence and direction of luminance texture pattern anisotropy on the accuracy of observers' judgments of 3D surface shape. Our stimuli consist of complicated, smoothly curving level surfaces from a typical volumetric dataset, across which we have generated four different texture patterns via 3D line integral convolution: one isotropic and three anisotropic, with the anisotropic patterns oriented across the surface either in a single uniform direction, in a coherently varying direction, or in the first principal direction at every surface point. Observers indicated shape judgements via manipulating an array of local probes so that their circular bases appeared to lie in the tangent plane to the surface at the probe's center, and the perpendicular extensions appeared to point in the direction of the local surface normal. Stimuli were displayed as binocularly viewed flat images in the first trials, and in stereo during the second trials. Under flat viewing, performance was found to be better in the cases of the isotropic pattern and the anisotropic pattern that followed the first principal direction than in the cases of the other two anisotropic pattems. Under stereo viewing, accuracy increased for all texture types, but was still greater for the isotropic and principal direction patterns than for the other two. Our results are consistent with a hypothesis that texture pattern anisotropy impedes surface shape perception in the case that the direction of the anisotropy does not locally follow the direction of greatest normal curvature.

Keywords: texture, shape representation, principal directions, shape perception.

1 INTRODUCTION

A key objective in the field of visualization is to design and implement algorithms for effectively communicating information through images. Given a set of data, we must design a visual representation for that data which facilitates its understanding. The investigations reported in this paper were motivated by applications in which one needs to be able to accurately and intuitively convey the three-dimensional shape of large, smooth, arbitrarily curving surfaces. Previous studies [cf. Interrante *et al.* 97] have indicated that shape perception can facilitated by the addition of surface texture markings, but the question of how to characterize the kind of surface texture that will show shape best remains open. Texture can also be used to *mask* surface shape features, as was shown by Ferwerda *et al.* [1997]. It has been suggested [Cumming *et al.* 1993] that the perception of shape from texture can be impeded when the texture pattern is highly anisotropic, consisting of elements that are systematically elongated in a specific direction. However a wide variety of textures consisting of line-like elements have been shown to indicate surface curvature [Todd and Reichel 1990]. Knill [1999] shows that across developable surfaces[1] any homogeneous texture pattern will appear to flow along parallel geodesics[2], and suggests that our visual system uses shape-from-contour[3] to infer shape from the systematic projective distortion or flow of the pattern. Inspired by the considerable amount of research [Stevens 1983, Mamassian and Landy 1998, Li and Zaidi 2000] that seems to imply that surface shape may be perceived most accurately from line-like markings when they follow the lines of curvature., we sought in the series of experiments described in this paper to further experimentally investigate the effect of the direction of surface texture pattern anisotropy on the accuracy of observers' shape judgments. Specifically we wanted to know: can an anisotropic pattern that follows the principal directions show shape more effectively than a pattern in which the direction of anisotropy follows some other path? Than an isotropic pattern? Are the effects the same in the case of shaded displacement texture? To what extent are these effects mitigated by stereo viewing?

Correspondence: Interrante; Other author info: VI; email: interran@cs.umn.edu; web: www.cs.umn.edu/~interran; SK; email: skim@cs.umn.edu; web: www.cs.umn.edu/~skim.

[1] surfaces that can be unrolled to lay out flat in the plane;

[2] curves that do not turn in the surface

[3] the set of all points at which the surface normal is orthogonal to the line of sight

2 METHODS

We conducted a series of two experiments intended to investigate the effect of the presence and direction of texture pattern anisotropy on the ability of observers to accurately perceive the 3D shape of a smoothly curving surface. The goal of these experiments was to gain insight that might facilitate our efforts to use texture most effectively to facilitate the accurate perception of surface shape in renderings of scientific data. In the following sections we provide the details of the experimental set up and design.

2.1 Stimuli

The stimuli that we used in our experiments were cropped images of the front-facing portions of textured level surfaces rendered in perspective projection using a hybrid renderer [Interrante *et al.* 97] that uses raycasting [Levoy 88] together with a Marching Cubes algorithm [Lorensen and Cline 87] for surface localization. The volumetric test data from which we extract these surfaces is a three dimensional dose distribution calculated for a radiation therapy treatment plan. We chose to use the radiation data as our testbed, rather than a more restricted type of analytically-defined surface, because this data is typical of the kind of data whose shape features we seek to be able to more effectively portray through the use of surface texture.

The first step in image generation was to define the solid texture patterns that would appear on the level surfaces. We used a high-quality three-dimensional line integral convolution algorithm [Stalling and Hege 95] to synthesis the textures in the vicinity of the selected level surface. Beginning with a three-dimensional array of binary noise, line integral convolution produces an output texture in which the input values are correlated along the directions indicated by an accompanying vector field. We defined four different vector fields to produce four different types of texture patterns.

The procedure that we used to obtain the principal direction vector field is fully described in [Interrante 97], but is briefly restated here for completeness. We begin by computing an orthogonal frame at each sample point in the 433x357x325 voxel 3D volumetric dataset. We define the third frame vector to be in the direction of the grey-level gradient, which is the normal to the level surface that passes through the sample. We compute the gradient using Gaussian-weighted central differences in the axial directions over the 3x3x3 area surrounding the sample point. We next choose an arbitrary point in the tangent plane to define the direction of the first frame vector and then take the cross product of these two vectors to obtain the remaining orthogonal direction. Finally, we estimate the 2^{nd} Fundamental Form [Koenderink 90] from the Gaussian-weighted central differences of the gradients trilinearly interpolated at sample positions over a 3x3x3 grid aligned with the local frame, diagonalize to obtain the 2D principal directions (eigenvectors) and principal curvatures (eigenvectors) in the tangent plane, and convert to 3D object space coordinates. The direction corresponding to the eigenvalue with the greatest unsigned magnitude is saved in the 3D principal direction vector array and used to create the first anisotropic texture ('pdir').

The remaining 3D vector fields are obtained by simpler means. First, we obtain the vector field of uniform directions by taking at each point the direction given by the intersection of the tangent plane with the plane orthogonal to the z axis that passes through the sample point: $udir_x = -n_y$, $udir_y = n_x$, $udir_z = 0$, where (n_x, n_y, n_z) is the surface normal or gradient. Then, we obtain the vector field of random directions that is used to create the isotropic texture pattern by rotating the uniform direction previously obtained at each point by a random angle θ_1 about the surface normal, $-\pi/2 \le \theta_1 \le \pi/2$. Finally, we obtain the vector field of coherently varying directions that is used to create the anisotropic texture pattern that contains lines with non-zero geodesic curvature by rotating the original uniform direction about the surface normal by an angle $\theta_2 = 10\pi(x+y+z)/n$ where (x,y,z) is the index of the sample point in the volume and n is the total number of sample points in the 3D array.

Figure 1 illustrates the process of texture synthesis, showing a single slice (z=263) from the 3D input texture volume and from each of the different 3D output texture volumes. Not all of the values are filled in, because we have elected to initiate the streamlines that are used to compute the output texture values only at the voxels that are in the vicinity of the level surface that is being used to create the test image.

During rendering, the intensity value interpolated from the 3D texture at the ray/surface intersection point is taken as the base color of the surface at the ray surface intersection point, and Phong shading is then applied to obtain the final surface color. We rendered 48 test images for the experiment, 24 for the left eye views and 24 for the right eye views, using the four different textures applied to views from six different vantage points around a single level surface. Figure 2 shows three of these images, all computed for the same viewing position. In order to avoid the potentially confounding influence of shape-from-contour information, as a last step we cropped each image to a 400x400 pixel region that did not contain any points on the silhouette edges of the object.

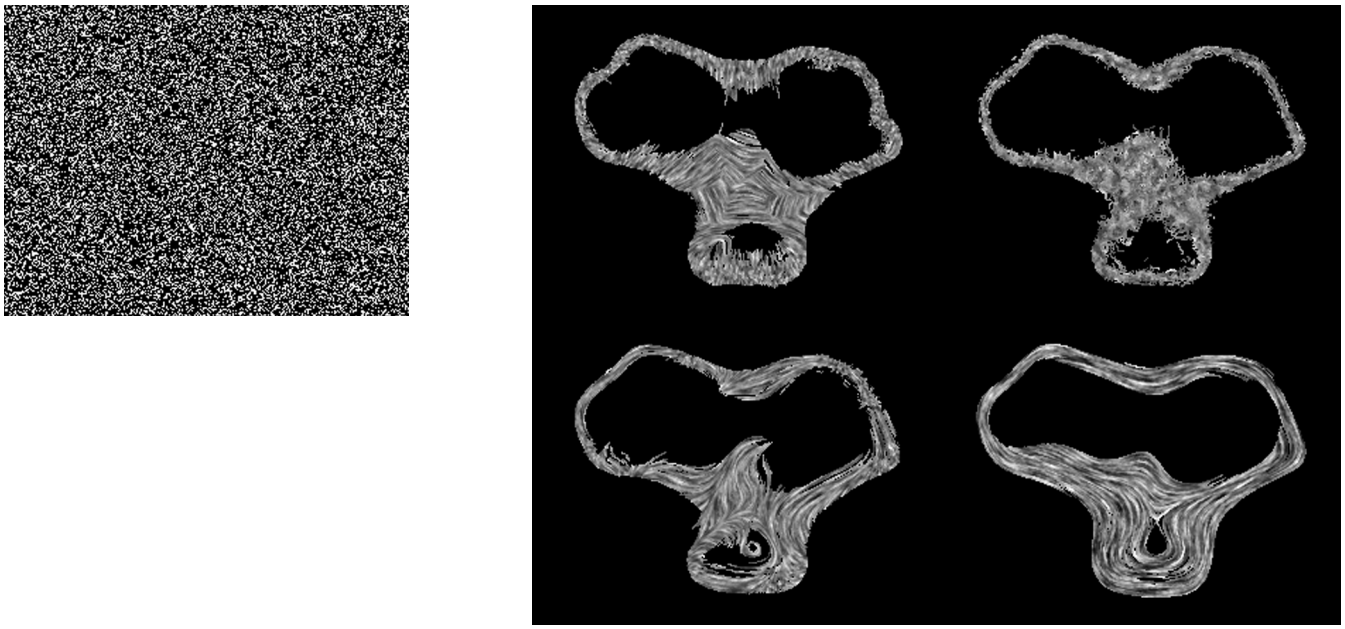


Figure 1: Slices from the 3D solid textures. Left: The slice z=263 before line integral convolution; Right: The same slice after line integral convolution along (in clockwise order) first principal directions (pdir), random directions (rdir), uniform directions (udir) and coherently varying or swirling directions (sdir) computed at each sample point.

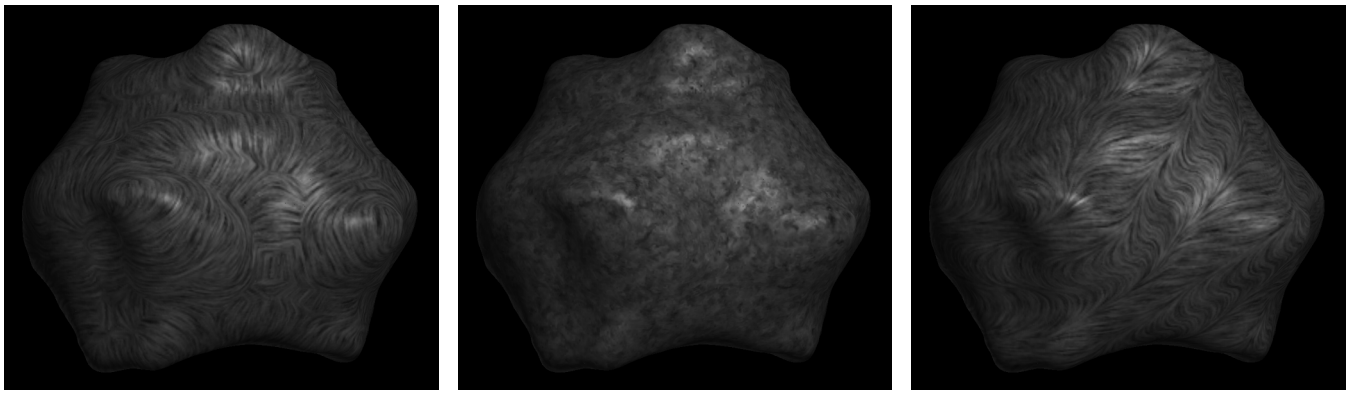


Figure 2: Examples of the 3D textured surfaces. From left to right: pdir, rdir and sdir. Note that informal assessment of the potential impact of texture type on shape judgments is complicated in these images by the prominence of shape-from-contour cues, which tend to dominate when other information about shape is less readily accessible. Because we are most interested in studying how the presence of texture might facilitate shape judgments across non-trivially structured interior regions where shape-from-contour information is not available, we cropped all of the images to eliminate the edge cues before testing.

2.2 Task

In originally planning these investigations, we had hoped to be able to design an experimental task that could reveal the effect of different texture types on the accuracy and efficiency of an observer's perception of the global 3D shape of a displayed object (shape from a glance). However we had great difficulty coming up with a means to evaluate observers' immediate global impressions of surface shape in a way that avoided confounding influences such as isolated 2D feature recognition or partial picture matching. Hence we decided to proceed with estimates of surface shape perception accumulated from individual judgments of the orientation of the surface at local points. Because it is well known that our visual system does not build up an estimate of shape from the accumulation of isolated individual local estimates of surface heading, but rather obtains shape understanding from the comparative relationships between nearby points, we decided to present an array of probes [Koenderink *et al.* 1992] that completely covered the central area of the presented surface and to ask observers not

only to adjust each probe by pulling on its handle until the circular base appeared to lie in the tangent plane to the surface at its central point and the perpendicular extension appeared to point in the surface normal direction, but also before proceeding to the next trial to verify that the shape of the surface they had implicitly indicated through the collective orientations of all of the probes appeared to faithfully match the shape of the underlying textured surface at all points.

Unfortunately, we neglected to recognize, before beginning the experiments, that our decision to place the probes at exactly evenly spaced intervals over a rectangular grid would interfere with observers' ability perceive all of the probes as lying in the surface at the same time, due to violation of the generic viewpoint assumption. (If the probes did all lie in a smooth surface that varied in depth, and still appeared to be evenly spaced in a single view, then any tiny translation of the viewing position would have to break the symmetry of the spacing. Our visual system hence preferentially adopts the more likely interpretation that the probes are arrayed on a transparent flat plane in front of the underlying curved surface.) Our subjects did not report an inability to see the probes as lying in the surface on an individual basis, but, as will be discussed later, certain of the individual responses appeared to indicate that the probes were not always consistently visualized as a coherent unit across each image. Figure 3 shows the user interface at the beginning of the 5th trial.

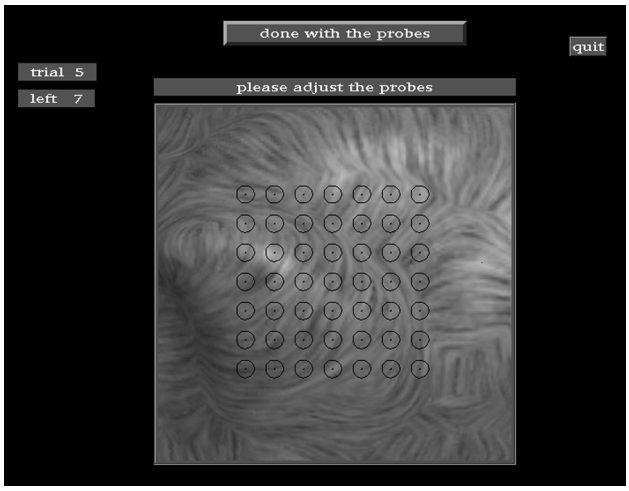


Figure 3: The graphical user interface with all probes displayed in their starting positions.

In designing the experiment we were particularly concerned about avoiding a situation in which differences dues to texture type might be confounded with differences due to other unanticipated or uncontrolled factors such as individual differences, or particular surface shape configurations. Ideally, we would have liked to present identical views of each surface under all four texture conditions, and to have all subjects make judgements on all of the images. However we were also concerned about the possibility of subjects' current shape judgments being biased by information they obtained from previous trials in which the same surface had been shown under a different texture condition. With only 24 binocular images (6 views x 4 texture types), and an small anticipated subject pool size, we had to make some difficult tradeoffs. What we did was to divide the subjects and the stimuli into two different groups, so that each subject made shape judgments at the 49 probe locations on only half of the data (12 images). Each set contained each view and each texture type, in equal proportions, but did not contain all of the possible combinations. Within each set, the stimuli were further grouped into two lots, in which each lot had no surface repeated. Figures 4 and 5 show the complete set of stimuli presented to each group of observers.

The six images within each lot were presented in random order, and subjects were required to take a 10 minute rest break after finishing the 6th trial, thereby avoiding the possibility that any two differently textured but identically shaped surfaces might be presented immediately in sequence, and minimizing the likelihood of surface recognition and any consequent possible learning effects. After adjusting the probes on the 12 images in the flat viewing condition, subjects repeated the entire process under conditions of stereo viewing. To facilitate estimation of the effects of viewing condition, subjects were presented with the same stimuli, differently ordered, in the two viewing conditions. The entire process took about two hours for most of the subjects.

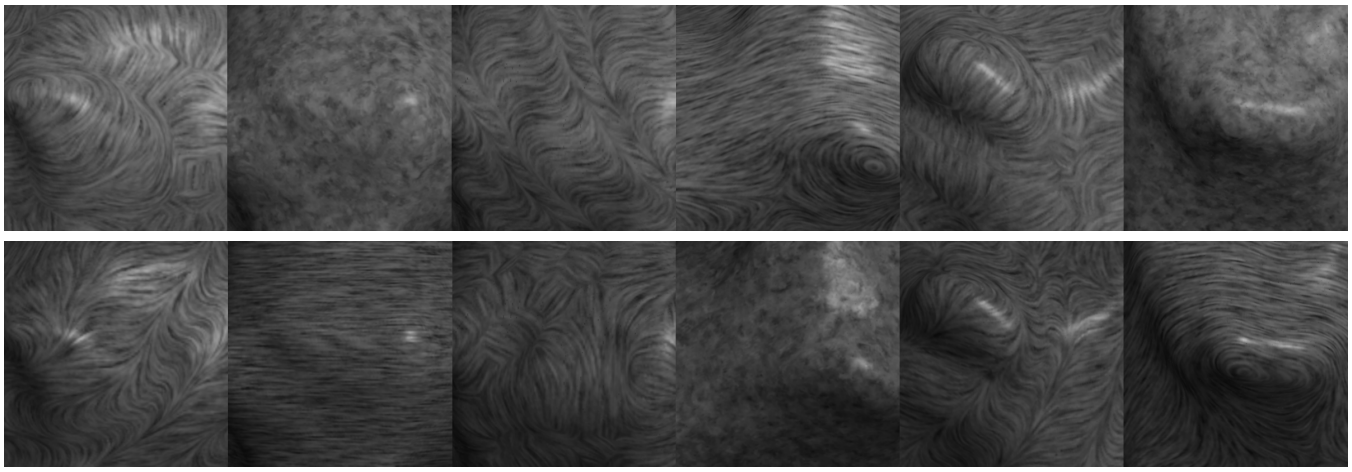


Figure 4: The set of stimuli seen by group A. First row: lot 1; second row: lot 2. The presentation order was randomly determined and was different for each subject.

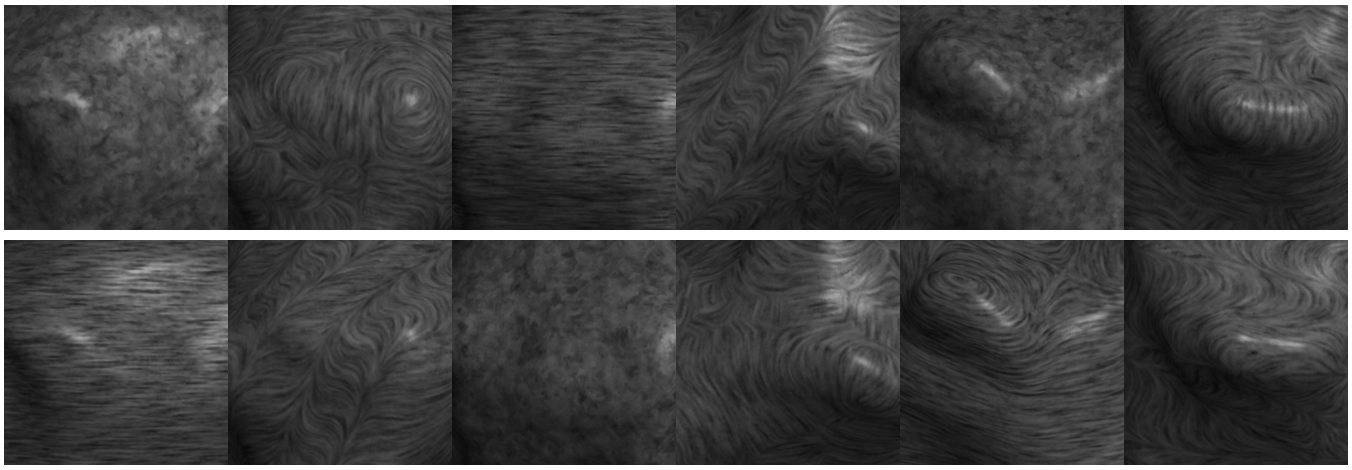


Figure 5: The set of stimuli seen by group B. First row: lot 1; second row: lot 2. The presentation order was randomly determined and was different for each subject.

2.3 Observers

We had five subjects participate in the experiments. All of the subjects were male EE and CS graduate students from the University of Minnesota, who agreed to participate as a favor to the second author and for compensation in the form of gift certificates to local coffee shops and/or eateries. All subjects were kept fairly naïve to the purposes of this experiment, though some of the subjects were certainly aware of the authors' previous work with principal direction texture. We informed the subjects that we were conducting experiments to evaluate peoples' ability to accurately perceive 3D shape in images but we specifically did not mention anything about texture. Our goal in doing this was to keep the subjects as free as possible of any potential biases and to avoid leading them into certain behaviors (such as lining up the direction of probe base elongation with the direction of the texture pattern) that they might not otherwise have considered. Before beginning the experiment, the subjects were asked to read a set of written instructions which described the probe positioning task. We used written rather than verbal instructions in an effort to maintain consistency. Subjects were also shown a single "training" image (figure 6) that portrayed ground truth answers in the form of correctly positioned probes for a seventh surface not included in the test data and rendered without texture. Note that several of these probes appear to point straight out of the screen. We showed them this image in order to give them an idea of what a set of exactly correctly positioned probes might look like. We were fairly selective in attempting to obtain participants that we hoped would be diligent in their efforts, and in the written instructions we stressed the importance of trying hard to do a consistently good job on all of the images, even if the shape was difficult to perceive. As an extra incentive, however, we told the subjects that after all results had been tallied, we would give a $20 bonus certificate to the student who gave the most accurate answers, overall.

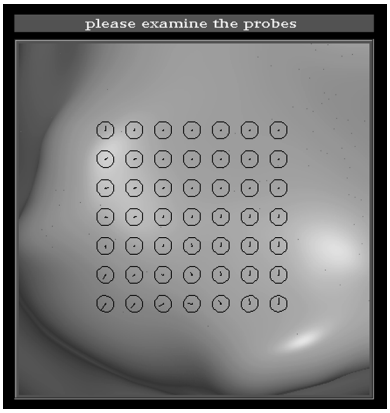
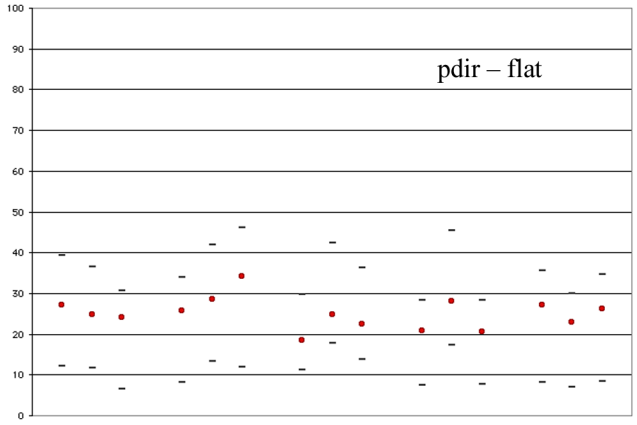
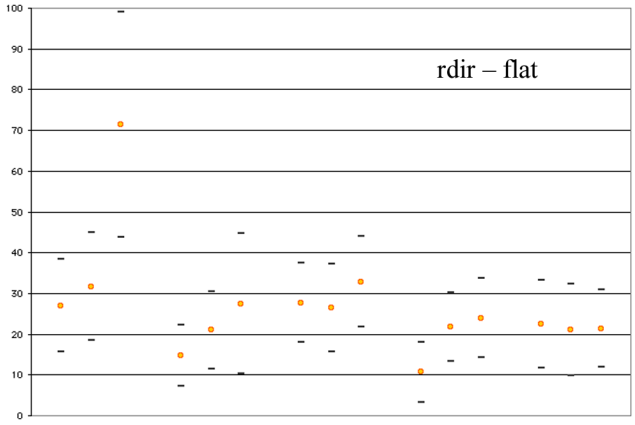


Figure 6: Training image, showing ground truth answers (correct probe orientations) at points across an untextured surface.

3 RESULTS

Having observed that other investigators studying shape perception using local probes analyze the perceived surface orientation in terms of slant and tilt, where slant is the angle of rotation out of the fronto-parallel plane, and the tilt is the angle of rotation about the viewing direction, we had initially hoped to be able to do the same in our studies – measuring the accuracy of observers' estimates of local heading in terms of the deviation in slant and tilt from the ground truth answers. While fairly satisfied with the indication of error provided by deviations in slant, we had several serious problems interpreting the magnitude of the errors due to incorrect estimates of tilt. The root of our difficulties was that too many of the points on our surfaces were too near to being parallel with the image plane. In numerous incidences the angular deviation in tilt was degenerate, because the estimated normal projected to a single point, and it was not clear how to appropriately handle these cases. We could not simply exclude these samples from our error calculations, because their occurrence was not uniform but tended to predominate in "bad texture" conditions, where the cues to shape were inadequate and subjects reverted to the default assumption that the surface lay in the plane of the image, or subjects simply gave up in frustration and left the probes untouched at their default original positions. Furthermore, even in the cases where the tilt angle was not degenerate, the lengths of the projected normal vectors could be exceedingly small, on the order of one or two pixels, and it was therefore possible to register huge estimated errors in the tilt component in places where the observer had merely misplaced the endpoint of the vector by two or three pixels (less than 1mm on the screen) in a particularly unfortunate direction. We therefore reluctantly decided to break with tradition and simply use as an error metric the angle in $\Re^3$ between the estimated normal direction specified by the probe and the true surface normal direction at the probe center. Figure 7 shows the mean angular error and standard deviations computed over the 49 probe positions at which estimates were made by each subject for each image, with each texture type, under conditions of binocular flat viewing. The results are grouped into different images by texture type, and then grouped within each image by test subject. Figure 8 shows the results under conditions of stereo viewing.





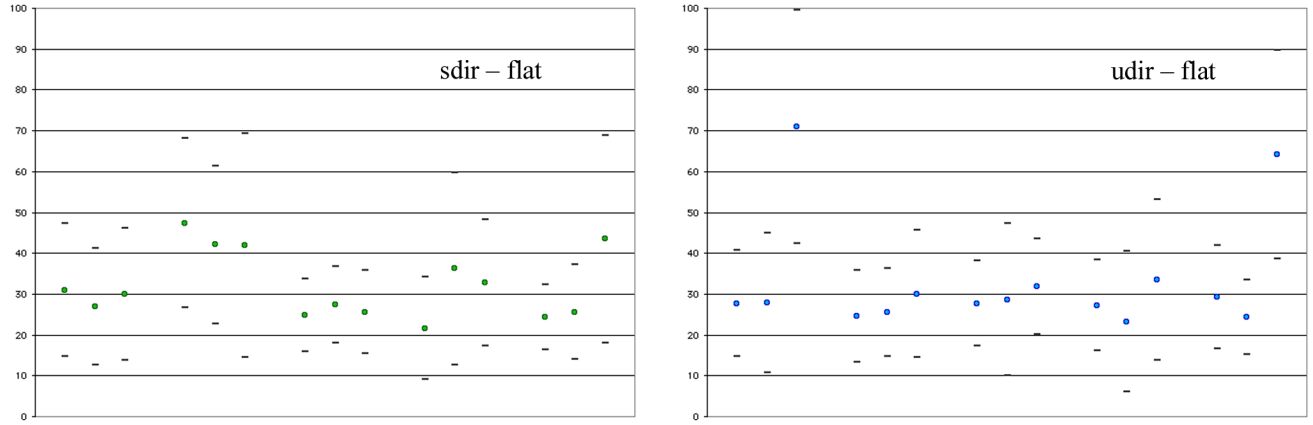


Figure 7: Individual results for the flat viewing condition. The height of each point represents mean angular error over the 49 probe locations per image. Subject number is the unspecified independent variable along the horizontal axis. Judgements from a single subject for different surfaces rendered with the same texture type are grouped by proximity along this direction. The textures are (clockwise from the top left): principal direction (pdir), isotropic (rdir), uniform (udir), and swirling (sdir).

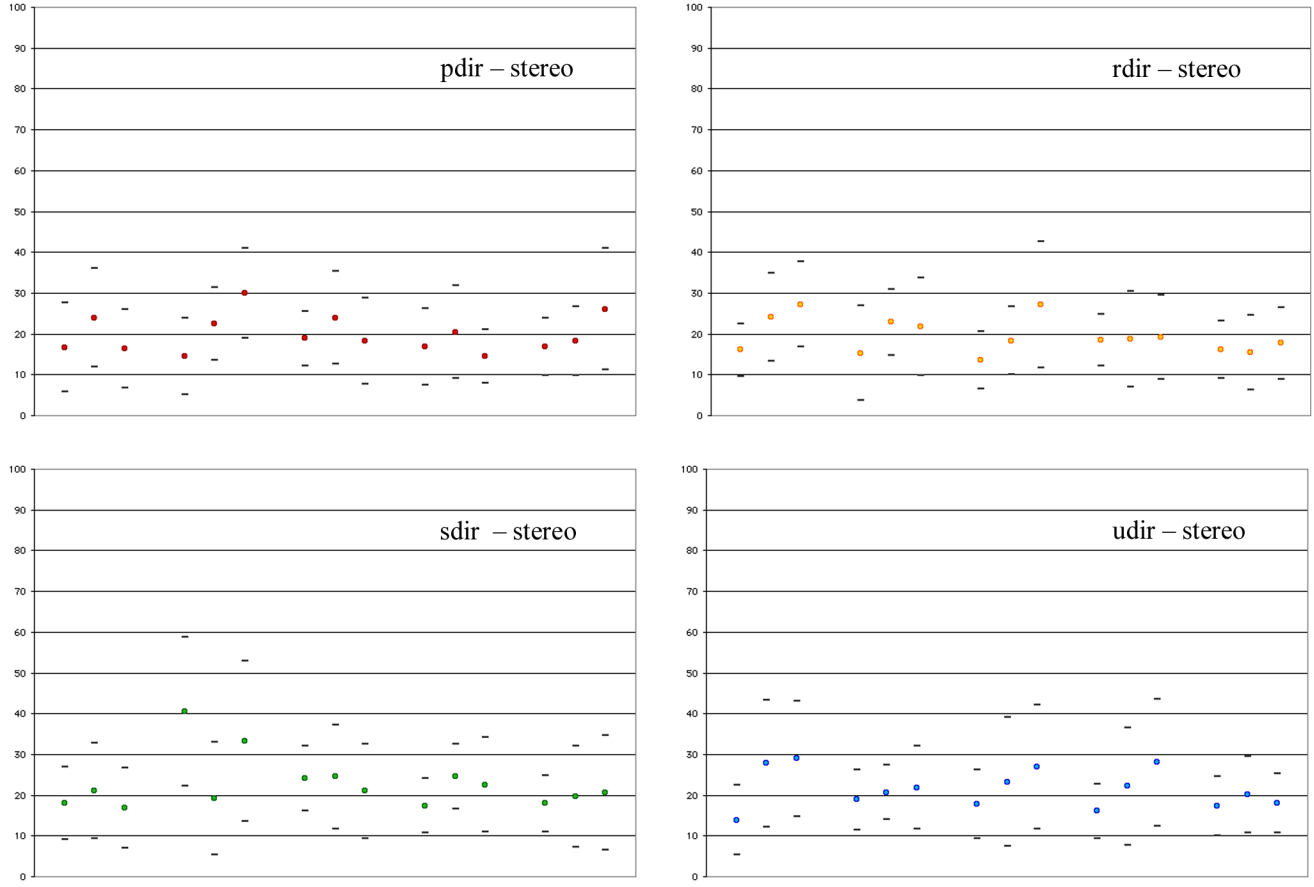


Figure 8: Individual results for the stereo viewing condition. Each point represents mean angular error over the 49 probe locations per image. Clockwise from top left: principal direction (pdir), isotropic (rdir), uniform (udir), swirling (sdir).

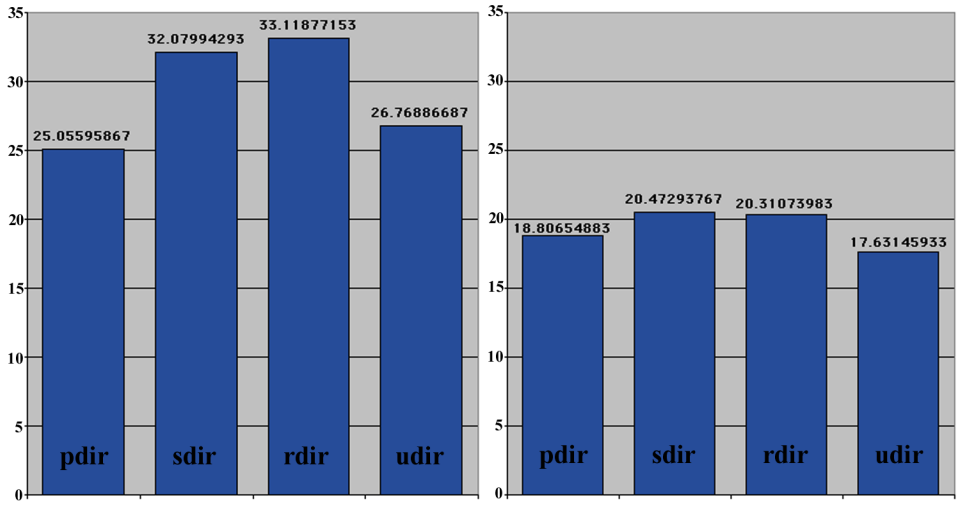


Figure 9: Pooled results (mean angle error) for all subjects, all surfaces, by texture type. Left: flat presentation; Right stereo.

4 DISCUSSION

A definitive statement of the results is hampered by the fact that we have not yet succeeded in doing a thorough statistical analysis of the data and hence cannot make any claims about the statistical significance of the differences in the mean angular errors observed under the different texture conditions. Overall, subjects seem to do somewhat better in the principal direction oriented and isotropic texture conditions than in either of the other two. It appears, from inspection of the individual results, that subjects may be less prone to making catastrophic errors when stimuli are viewed as flat images if the surfaces are rendered with the pdir texture. However, closer inspection of the pattern of errors is needed. A preliminary inspection suggests the presence of two different types of errors: coherent errors due to perceived depth inversion, and incoherent errors, as shown in figure 10. Errors appear to accumulate in the principal direction texture around discontinuities in the pattern where the first and second principal directions switch places. We had anticipated the possibility of an advantage in using an anisotropic texture in which the direction of the anisotropy followed lines of curvature over the surface, but this interpretation is not strongly supported by the experimental results. Most subjects appeared to perform equally well or better with the purely isotropic pattern. However some subjects were clearly misled in some places by the anisotropic patterns that followed directions different from the principal direction, suggesting that if one must use an anisotropic pattern, one must be careful about how it is applied over the object.

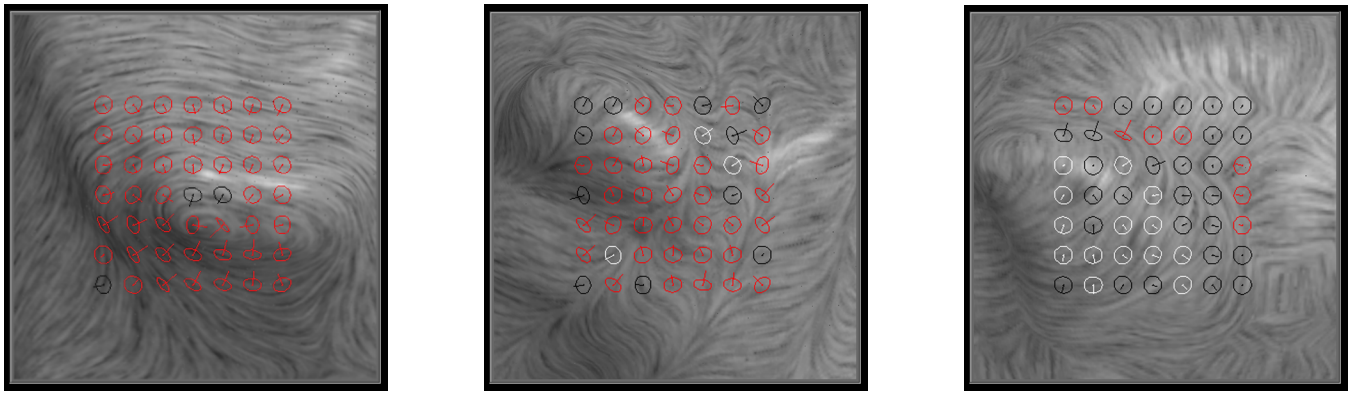


Figure 10: Some detailed individual results: Left: coherent errors due to depth inversion; Middle: incoherent errors apparently due to shape misperception; Right: errors tend to pile up at texture flow discontinuities, where the first and second principal directions switch places.

5 FUTURE WORK

There is considerable room for future work. One of the primary factors motivating this research was the desire to gain insight into how to select or define a texture pattern that could be used to facilitate the accurate and intuitive appreciation of 3D shape of a rendered surface. It appears clear that the principal direction textures defined above leave something to be desired in this respect. Shape representation from line orientation seems to be good in places where one of the two principal curvature values is high, but errors accumulate in the flatter areas where the directional information is less useful and less reliable. One direction for future work is to develop a more effective texture model that combines the strengths of several different texture definition approaches. A perhaps more immediate direction for future work is the investigation of the effect of texture orientation on surface shape judgments when the texture pattern is defined by surface relief rather than surface luminance. Does texture orientation affect shape perception in the same way in the two cases? Examples of some preliminary stimuli for subsequent experiments on this subject are shown in figure 11.

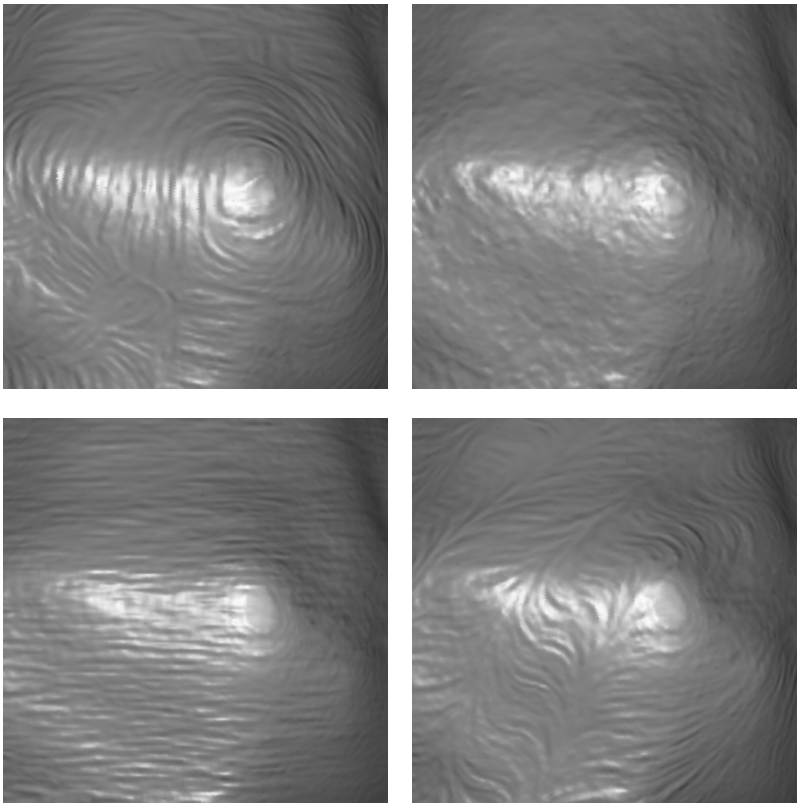


Figure 11: The same stimuli with the same textures, this time rendered as shaded relief rather than as luminance patterns.

6 ACKNOWLEDGMENTS

This work was partially supported by an NSF Presidential Early Career Award for Scientists and Engineers (PECASE). The images were rendered using parts of volume rendering software originally written by Marc Levoy. The dose data was provided by Dr. Julian Rosenman. Jeremy Leboy assisted with the implementation of the probe. We are grateful to our subjects TU, GG, MH, MM and HN for their dedicated efforts.

7 REFERENCES

Cumming, Bruce G., Elizabeth B. Johnston and Andrew J. Parker. (1993) Effects of Different Texture Cues on Curved Surfaces Viewed Stereoscopically, *Vision Research*, **33**(5/6): 827–838.

Ferwerda, James A., Sumanta N. Pattanaik, Peter Shirley and Donald P. Greenberg. (1997) A Model of Visual Masking for Computer Graphics, *ACM SIGGRAPH Proceedings*, pp.143–152.

Interrante, Victoria, Henry Fuchs and Stephen Pizer. (1997) Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture, *IEEE Computer Graphics and Applications*, **3**(2): 98–117.

Interrante, Victoria. (1997) Illustrating Smooth Surfaces in Volume Data via Principal Direction-Driven 3D Line Integral Convolution, *ACM SIGGRAPH Proceedings*, pp. 109–116.

Knill, David C. (1999) Contour into Texture: The Information Content of Surface Contours and Texture Flow, *www.cvs.rochester.edu/~knill/papers/postscript/texture_flow.pdf.*

Koenderink, Jan J. (1990) Solid Shape, MIT Press.

Koenderink, Jan J., Andrea van Doorn and Astrid M. L. Kappers. (1992) Surface Perception in Pictures, *Perception*, **52**, pp. 487–496.

Levoy, Marc. (1988) Display of Surfaces in Volume Data. *IEEE Computer Graphics and Applications*, **8**(3): 29–37.

Li, Andrea and Qasim Zaidi. (2000) Perception of Three-Dimensional Shape from Texture is Based on Patterns of Oriented Energy, *Vision Research*, **40**, pp. 217–242.

Lorensen, William E. and Harvey E. Cline (1987) Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *ACM SIGGRAPH Proceedings*, pp.163–169.

Mamassian, Pascal and Michael P. Landy. (1998) Observer Biases in the 3D Interpretation of Line Drawings, *Vision Research*, **38**, pp. 2817–2832.

Stalling, Detlev and Hans-Christian Hege. (1995) Fast and Resolution Independent Line Integral Convolution. *ACM SIGGRAPH Proceedings*, pp. 249–256.

Stevens, Kent (1983) The Line of Curvature Constraint and the Interpretation of 3-D Shape from Parallel Surface Contours, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp.1057–1061.

Todd, James T. and Francene D. Reichel. (1990) The Visual Perception of Smoothly Curved Surfaces from Double Projected Contour Patterns, *Journal of Experimental Psychology: Human Perception and Performance*, **16**(3): 665-674.

Growing Fitted Textures on Surfaces

Gabriele Gorla[†], Victoria Interrante[‡] and Guillermo Sapiro[†]

[†]*Electrical and Computer Engineering,* [‡]*Computer Science and Engineering*
University of Minnesota

Abstract

In this paper, we address the problem of how to, seamlessly and without visible projective distortion, automatically cover the surface of a polygonally-defined model with a texture pattern derived from an acquired 2D image such that the dominant orientation of the pattern everywhere follows the surface shape in an aesthetically pleasing way and repetition artifacts in the texture pattern are completely avoided. We propose an efficient, automatic method for synthesizing, from a small sample swatch, patches of perceptually similar texture in which the pattern orientation may locally follow a specified 3D vector field on the surface, such as the principal directions of curvature, at a per-pixel level, and in which the continuity of large and small scale features of the pattern is generally preserved across adjacent patches. We demonstrate the results of our method with a variety of texture swatches applied to standard graphics datasets.

CR categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism — Texture.
Keywords: Texture synthesis, texture mapping, shape perception, shape representation.

1. Introduction

Adding texture to the surface of a polygonal model can profoundly enhance its visual richness. In addition to contributing abundant detail and complexity at minimal computational expense, texture has the potential, depending upon the characteristics of the pattern, and how it is applied, to affect our perception of an object's geometry, masking faceting artifacts [8], or enhancing our appreciation of the curvature of the form [14]. Given a texture pattern and a surface model, the historical challenge has been to determine how to apply the pattern to the surface in an appropriate manner, minimizing the visual impact of seams and projective distortion while orienting the pattern so that it flows over the shape in a desirable way.

Many different approaches to this basic problem are possible. Our solution focuses on the case in which the desired texture is defined by a provided 2D image. The method that we propose has the advantages of being essentially automatic (requiring no manual intervention), reasonably efficient, fairly straightforward to implement, and applicable across a wide variety of texture types and models. In addition, the resulting textured objects can be easily displayed at interactive frame rates using a conventional renderer on a standard PC with texture mapping hardware.

Figure 1: An example of a synthesized surface texture produced by our method. No manual intervention of any kind was employed. This texture was grown from an original 92x92 swatch [4], pre-rotated to 63 orientations each cropped to 64x64 pixels, to cover 291 surface patches at 128x128 resolution following a vector field locally defined by the projection of (0,1,0) onto the tangent plane at each point. The entire process required approximately 12 minutes.

Our technique consists of the following main steps:

- Partition the polygons of the model into contiguous patches, as nearly planar (to prevent distortion) and as nearly similarly sized (to simplify texture map handling) as reasonably possible.
- Compute a vector field over the object, or read a pre-defined field from a file.
- Synthesize the texture pattern over each patch, maintaining pattern continuity across the boundaries with neighboring patches, using an efficient, orientation-adaptive variation of the non-parametric sampling method proposed by Efros and Leung [7].

An example of the results of our algorithm is presented in figure 1.

2. Previous Work

A variety of methods have been previously proposed for texturing polygonal models with patterns that are as free as possible of seams and distortion artifacts.

One method is solid texturing [20,21,29], in which the texture pattern is defined over a 3D volume. Particularly good results have been achieved with this method for water, as well as for objects made of wood and stone. However, there are significant challenges in synthesizing 3D textures modeled after sampled materials [13, 6], and current methods for creating custom-fitted 3D textures whose features follow a surface's shape [14] are severely limited in scope and applicability.

Methods for applying 2D image-based texture to arbitrary polygonal models for the most part must balance the inherent trade-off between seams and distortion (one cannot in general apply a 2D image to a non-developable surface without incurring one or the other), employing piecewise flattening in the case of arbitrary parametric [2] or polygonally-defined [17] models, or using careful surface parameterization [15], or pre-distortion of the texture [1,28] to achieve desired results in other particular cases. Conformal mapping [11] offers a global solution that preserves angles, but not lengths or areas.

Closer to our objectives, Neyret and Cani [19] proposed an excellent technique for achieving seamless and virtually distortion-free mapping of 2D isotropic texture patterns on arbitrary objects via custom-defined triangular texture tiles that are continuous with one another across various of their boundaries. Unfortunately an extension of this method to anisotropic texture patterns is not obvious. Last year, Praun et al. [24] proposed "lapped textures", which provides capabilities that are the most similar to those towards which our method aspires, although the approach that we take is very different. The lapped texture method repeatedly pastes copies of a sample texture swatch onto overlapping patches across a surface after some subtle warping and reorientation to align the pattern with a user-defined vector field. This method produces very good results when used with texture patterns that contain enough high frequency detail and natural irregularity in feature element sizes and relative positions. This is needed to perceptually mask artifacts due to the partial overlap or misalignment of feature elements across patch boundaries. As currently formulated, the lapped texture approach is not particularly well-suited for rigidly structured patterns, such as a checkerboard, or textures such as netting, which are characterized by the global continuity of specific elongated elements. It is also less

well-suited for use with vector fields that contain significant high frequency variation. In addition, the lapped texturing process as described in [24] involves considerable amounts of user interaction.

The method that we describe in this paper provides capabilities beyond those offered by previous methods.[1]. It achieves nearly seamless and distortion-free texturing of arbitrary polygonally-defined models with a texture pattern derived from a provided sample. Most importantly, the method is suitable for use with anisotropic patterns. It generally preserves larger scale texture pattern continuity across patch boundaries, does not require manual user intervention, and allows the orientation of the applied pattern to locally follow a specified vector field on a per-pixel basis.

Our method falls into the category of methods that achieve texture pattern continuity without distortion by in effect synthesizing the texture "in place" over the surface of the object. Previous methods in this category include direct painting [12], and reaction-diffusion texture synthesis [26], which yield excellent results for hand-crafted textures and textures modeled after organic processes. We want to achieve similarly good results with automatically synthesized textures that are perceptually equivalent to a given sample swatch.

Our work is perhaps most fundamentally motivated by the impressive advances in texture synthesis methods [13,5,23,30,7,27] which make it possible to create, for an increasingly wide range of patterns, unlimited quantities of a texture that is perceptually equivalent to a small provided sample.

Leveraging research in human texture perception, Heeger and Bergen [13] developed a highly successful method for synthesizing textures that capture the essential perceptual properties of a variety of homogeneous stochastic sample patterns. Their method works by modifying a random noise image so that its intensity histogram matches the histogram of the sample texture across each of the subbands in a steerable pyramid representation of each image. De Bonet [5] developed a related method based on interchanging elements in the Laplacian pyramid representation of a self-tiling pattern where possible, while preserving the joint-occurrence relationships of features across multiple resolutions. This method yields impressive results for an even wider variety of patterns, though some difficulties remain in preserving larger scale globally significant structure. Several other highly sophisticated statistically based approaches have been subsequently proposed for texture synthesis [23, 30], however none of these methods, while quite inspiring, is exactly well-suited for our needs.

Rather we follow the statistical texture synthesis approach recently proposed by Efros and Leung [7]. In this method, a new texture pattern is grown, pixel-by-pixel, by sampling into a provided template pattern and choosing randomly from among the pixels whose neighborhoods are close matches to the yet partially-defined neighborhood of the pixel to be filled in, in the pattern being synthesized. A serious concern with this method however is speed, as addressed last year by Wei and Levoy [27] who proposed a more efficient approach to the problem of finding the set of good matches from among all of the possible neighborhoods based on tree-

[1] Note to the reviewers: After completing the work described in this paper[TR], we were made aware of the fact that efforts similar to ours had been undertaken at the same time by other groups. At the present moment we have no specific information about any of this parallel work, neither its exact scope or nature nor the identity of any but one of the groups undertaking it. However we will look forward to including discussion of these other efforts in this paper as soon as such information is made publicly available.

structured vector quantization. Wei and Levoy suggest as a direction for future work the possibility of extending their TSVQ method to synthesize texture over surface meshes. However because the speed advantage of their method is obtained via the imposition of constraints on the configuration of the causal neighborhood (*a priori* knowledge of the number and locations of the pixels at which the texture has already been synthesized) there are significant complications for such an extension. This constraint also limits the class of textures that can be accurately synthesized.

In the remainder of this paper we describe the method that we have developed and the details of its implementation, talk about some of the issues that arise in automatically determining a good way to orient a texture pattern over a surface, show representative results, and conclude with a discussion of the current limitations of our implementation and directions for future work.

3. Proposed Method

Our proposed method is basically a two step process. First the surface is partitioned into small, almost flat patches. Then, the texture is grown over the planar projection of each individual patch taking into consideration the proper boundary conditions to maintain the continuity of the texture pattern across seams at the patch boundaries. During the synthesis of an anisotropic pattern, the texture is locally constrained into alignment with a specified vector field over the surface. For simplicity, we store the resulting synthesized texture as a collection of separate small images for each patch, although other approaches are certainly possible.

3.1 Partitioning

The goal of the first stage is to partition the mesh into a minimum number of approximately planar patches (collections of triangles). Obviously these are conflicting goals for any closed surface and a suitable tradeoff must be found. In order to keep our implementation as simple as possible, we restrict patches to be of approximately the same size. Maintaining relatively consistent patch sizes simplifies texture memory management by allowing us to allocate and synthesize texture maps of a consistent fixed resolution for each patch.

Two input parameters define the maximum patch size (which influences the scale at which the texture appears over the surface) and the maximum projective distortion that the user is willing to tolerate. The initial partitioning is done with a greedy algorithm, after which an optimization step is performed to reduce the average projection error.

The process for the initial partitioning can be summarized by the following pseudo code:

```
while (unassigned_triangles > 0) {
    pick an arbitrary unassigned triangle T;
    assign T to a new group G;
    add to group G all connected triangles C that satisfy:
        - Normal(C) • Normal(T) > min_cosine_displacement;
        - distance from the center of C to the farthest vertex of T is less than max_dist;
}
```

The image on the left side of figure 2 shows a representative result after the first stage in the splitting. The green triangles are the reference triangles that define the plane onto which the patch will ultimately be flattened. It is easy to notice that some of the patches obtained at this point are very small and/or contain triangles that are relatively far from being aligned with the reference plane. A refinement pass is used to reduce both the number of patches and the number of triangles that are oriented at a sharp angle to the plane into which they will ultimately be projected. Two simple experimental rules are iteratively applied until no significant improvement is observed:

- remove a patch if it is very small, and its triangles can be added to a neighboring patch without violating the distance constraint;
- reassign a triangle T from patch P1 to a neighboring patch P2 if T borders P2 and is more closely aligned with the reference plane of that patch than with its own.

The image on the right side of figure 2 shows the results after iterative refinement. Although the simple refinement procedure that we use is not guaranteed to converge to the theoretically optimal result, we have found that the results are consistently good and quite sufficient for our purposes. Most importantly from a practical standpoint, this method is very easy to code and extremely fast. The greedy splitting step takes between 1/2 to 2 seconds, depending upon the size of the model, and the refinement no more than 2-10 seconds in the worst cases. In the rare event that acceptable results are not achieved, the splitting process can be repeated using a tighter limit on the acceptable normal error (which will result in more, smaller patches), or by specifying a smaller maximum patch size directly. We have found that increasing the number of patches causes only a marginal increase in the computational expense in the subsequent texture synthesis step. More significant is the issue that with very small patches comes an increased risk that the texture synthesis process will run into difficulties and "grow garbage", either due to the paucity of available contextual information along the shortened boundary, or to the near proximity of mutually incompatible pre-defined boundary conditions.

3.2 Parameterization

After the model has been partitioned into contiguous patches, the triangles comprising each patch are projected onto their common reference plane, and the texture coordinates are defined at each vertex according the coordinates of the projected vertices in the reference plane coordinate system. Adjacent triangles from the neighboring patches, which provide the boundary conditions for maintaining the continuity of the texture pattern during synthesis, are then rotated about their shared edges into the reference plane. We use rotation for these triangles rather than projection in order to minimize the projective distortion of the texture that we will need to refer to for reference purposes. However, it is necessary to check for the very infrequently encountered cases where it is not possible to rotate each of the adjacent triangles of a particular patch into the projection plane without causing some of these triangles to overlap.

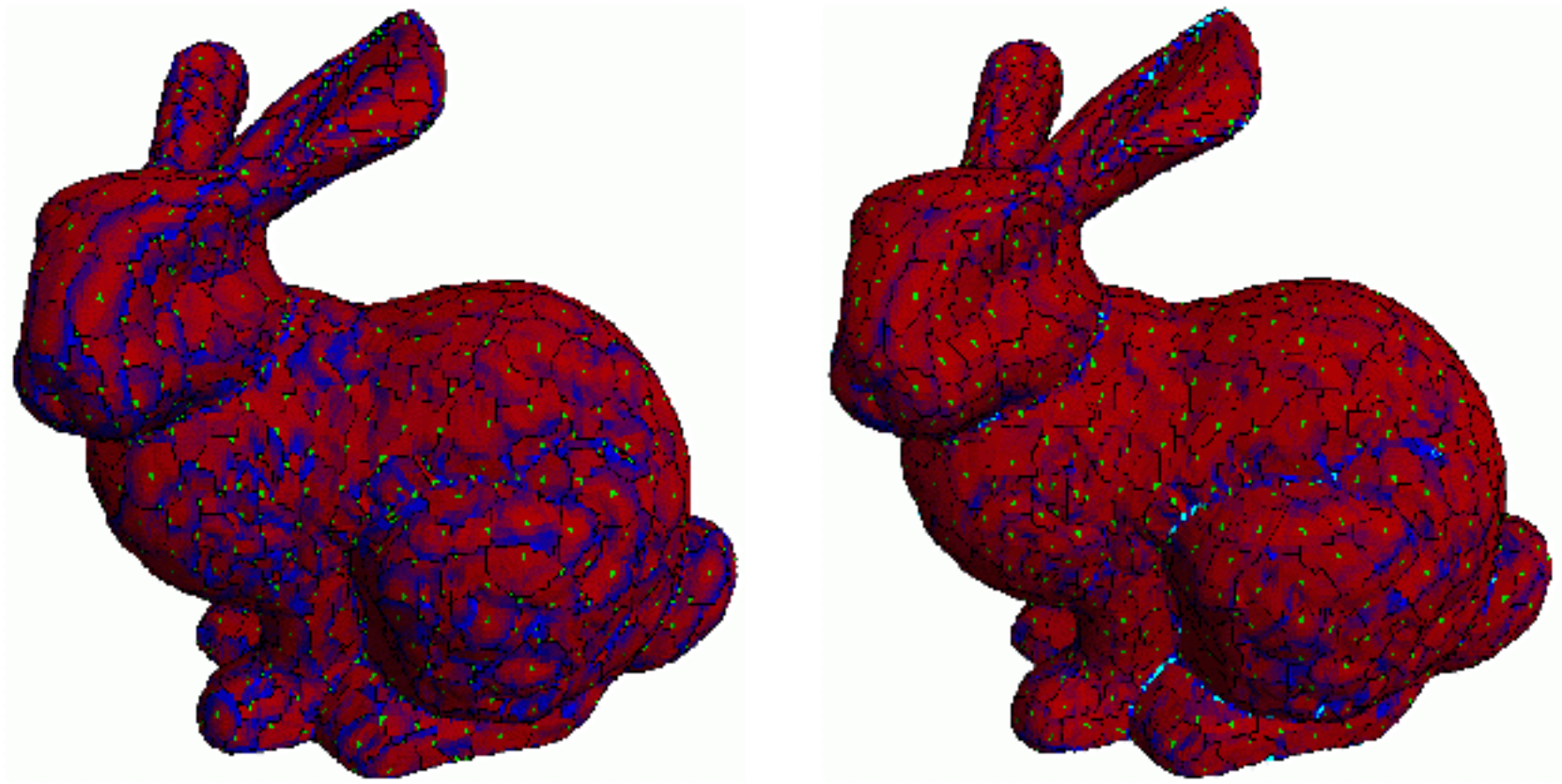


Figure 2: The partitioned bunny, after the first stage of our “greedy” splitting algorithm (left) and after iterative refinement (right), which decreased the number of patches from 988 to 699. Patch boundaries are outlined in black on each model. The green triangles are the seed triangles, which also define the reference plane for each patch. Triangles whose normal directions differ by less than 18 degrees from the direction of the normal to the reference plane, corresponding to a 5% error in the linear projection, are colored red. Triangles up to 25 degrees out of alignment with the reference plane for their patch, corresponding to a 10% error in the projection, are tinted blue. Triangles rotated more than 25 degrees from the reference plane (>10% error) are colored cyan.

3.3 Synthesis

At this point, we have created a 2D image for each patch containing:

- an area, defined by the projection of the triangles of the patch onto the reference plane, which contains the pixels to be filled by the synthesized texture; and
- an area, defined by the rotation into the reference plane of the neighboring triangles from the adjacent patches, that will hold any previously synthesized texture and provide the boundary conditions necessary to avoid seams due to discontinuities between texture element features in adjacent patches.

It is important to note that the partitioning process described in the immediately previous sections is completely independent from the synthesis algorithm. Any constrained synthesis method that can fill arbitrary regions with arbitrary boundary conditions could potentially be used, although none of the existing algorithms we reviewed appeared to provide both the flexibility and the speed required for this project. The work of Efros and Leung [7] came closest to meeting our needs and thus we elected to follow their general approach, which has been shown to produce good results for a wide range of texture patterns that can be modeled by Markov random fields (i.e. textures whose characteristics are fairly consistent under translation over the image and in which the value at any given point can be fully characterized by the values at its closely neighboring points over a limited range).

In order make tractable the problem of efficiently synthesizing enough texture to cover a standard model of arbitrary topology at a reasonable resolution, the first objective of our proposed method is to achieve results that are of the same caliber as those demonstrated by Efros and Leung but that require significantly less time, while also preserving the flexible applicability of their approach. To do this, we use a new two-pass search strategy. The first pass, which is exhaustive, is done using a very small unweighted neighborhood (usually 5x5 for sample textures of sizes between 64x64 to 128x128). The *n* best matches, where *n* is a user-definable parameter, are saved in a list to be processed by the second pass. This two pass approach presumes strong locality in the input textures (which holds true for many natural texture patterns) and has the effect of rapidly eliminating most of the uninteresting part of the search space. The size of this preselect list ultimately determines the overall speed of the synthesis algorithm. We found that some textures produced excellent results with preselect lists of as few elements as the number contained in one scanline of the original image; these we considered easy to synthesize. Others required 4 or even 8 times more elements in the preselect list, and these we considered hard to synthesize. In the second pass, each of the pixels in the saved list is tested against the full size weighted neighborhood (this size is also user selectable) and the error metrics are updated. Among the best 10 or 10% of matches (whichever is greater) a random pixel is chosen and used in the synthesized image. Figure 3 shows a sample result of this synthesis algorithm in the 2D case. The speed of this method does not match the speed of Wei and Levoy's tree-structured vector quantization [27] for the synthesis of rectangular swatches of texture, but unlike their method it can be directly used to fill in regions with arbitrarily configured boundaries, and the results it produces are of consistently high quality. The

proposed method is still fast enough to make feasible our goal of growing of a fitted surface texture via the MRF sampling approach.

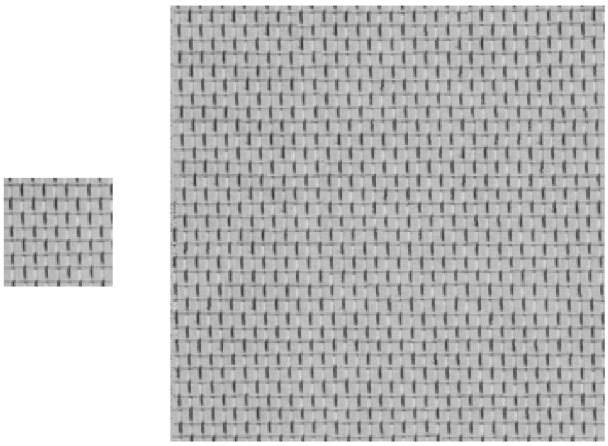


Figure 3: An example of the results of our two-pass texture synthesis method using pattern D06 from the Brodatz album [4]. The speed of the synthesis approach varies from pattern to pattern depending on the sizes of the match-defining neighborhoods and the length of the preselect list. In this case we used a first pass neighborhood of 5x5, a maximum preselect list length of 64, and a second pass neighborhood of 12x12 to synthesize the 256x256 patch on the right from the 64x64 pixel sample on the left in about 73s.

3.3.1 Isotropic Textures

As basically formulated, the approach we have just described can be used to cover the surface of a model with an isotropic texture pattern in an orientation-insensitive way. In other words, if we assume that our sample texture pattern is perfectly rotationally symmetric, we can directly use this approach, in its most basic form, to seamlessly synthesize the texture pattern across all of the patches in the model without any special considerations apart from the boundary conditions. However, as we quickly found, there are very few acquired textures that can be used with good results without regard to orientation. Even patterns which initially appeared to be isotropic in the original 2D sample revealed unexpected orientation dependencies. This is due to such things as the subtle structuring that stems from nearly imperceptible correlations due to shading when applied without regard to orientation, as is evident in figure 4.

3.3.2 Directional Textures

In the vast majority of cases, it is necessary to control the orientation of the texture over the surface. For greatest flexibility, we allow a directional texture to follow any specified direction field. In the next section some examples will be discussed. We note that in the lapped textures method, Praun et al. [24] also align textures on a per patch basis, slightly distorting the parameterization to achieve good local continuity within the patch with the underlying directional specification. In the case of sparse triangulation, they reduce undersampling of the vector field by locally subdividing the mesh.

Figure 4: A texture ('wool.bw', from SGI) that originally appeared to be isotropic, reveals its anisotropic nature (due to the effects of shading) when synthesized over the Stanford bunny dataset via an approach in which the texture orientation is allowed to vary arbitrarily between each patch.

In our presented method the synthesis algorithm has been enhanced to allow per-pixel texture re-orientation. We pre-rotate the original texture into a quantified number of orientations, and during synthesis perform the search for best-matching neighborhoods in the pre-rotated image that is most closely aligned to the direction locally specified by the vector field. Figure 5 shows a sample of one quadrant of pre-rotated brick texture. If the number of pre-rotated images is sufficiently high, the synthesized texture will follow the vector field smoothly. For the examples in this paper, we used between 64 to 128 different rotations of the input texture. Although it is of course possible to specify the use of any arbitrary number of pre-rotated images, we did not notice an appreciable increase in the quality of the results when finer quantizations were used.

For efficiency and ease of implementation, we use the PC's graphics hardware to perform the rotations and resampling. Searching for matches in pre-rotated texture images allows considerably faster synthesis than would be possible if we had to perform the rotation on-the-fly for each pixel of the texture

during synthesis. While it is relatively fast to compute any arbitrary rotation of the original image, there is a non-trivial cost associated with having to read the results from the frame buffer.

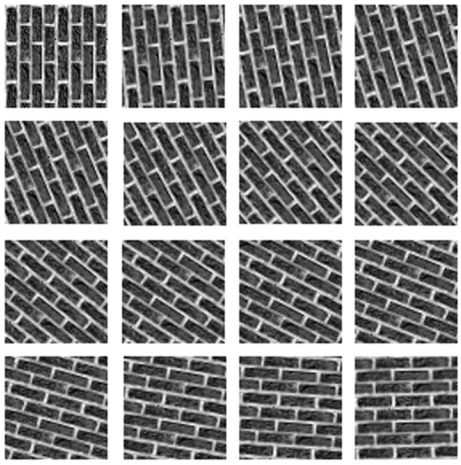


Figure 5: A quadrant of pre-rotated brick texture samples.

3.3.2.1 Constant Direction Fields

We originally began this work with the intent to explore the possibility of applying textures along the principal directions of curvature. Despite the latent potential in that approach, it is not without its difficulties, which we will discuss in greater detail in the following section. We quickly discovered that aesthetic results could be achieved for a wide range of models using other, much simpler, vector field definitions. Notably, the field of "up" directions, locally projected onto the tangent plane at each point, appears to yield particularly nice results for many textures and datasets, as shown in figures 6 and 7. It is worth mentioning in the context of these two images that we worked hard to challenge our texture synthesis method, testing its performance on difficult texture patterns such as the crocodile skin, which contains potentially problematic sets of features spanning a wide range of spatial frequencies (from 3–21 pixels in diameter), and the square glass blocks pattern, which is a highly structured checkerboard-style design in which irregularities in the size, shape and/or positioning of any of the elements have the potential to stand out especially prominently.

3.3.2.2 Principal Direction Fields

The constant direction field produces good results in many but not all cases. Specifically, it tends to fail for models that do not have a single well-defined intrinsic orientation, or across which we are interested in emphasizing local shape features. Of greatest intrinsic interest to our ongoing research is the possibility of applying an oriented texture pattern to the surface of an object such that it will be everywhere aligned with the principal directions of curvature.

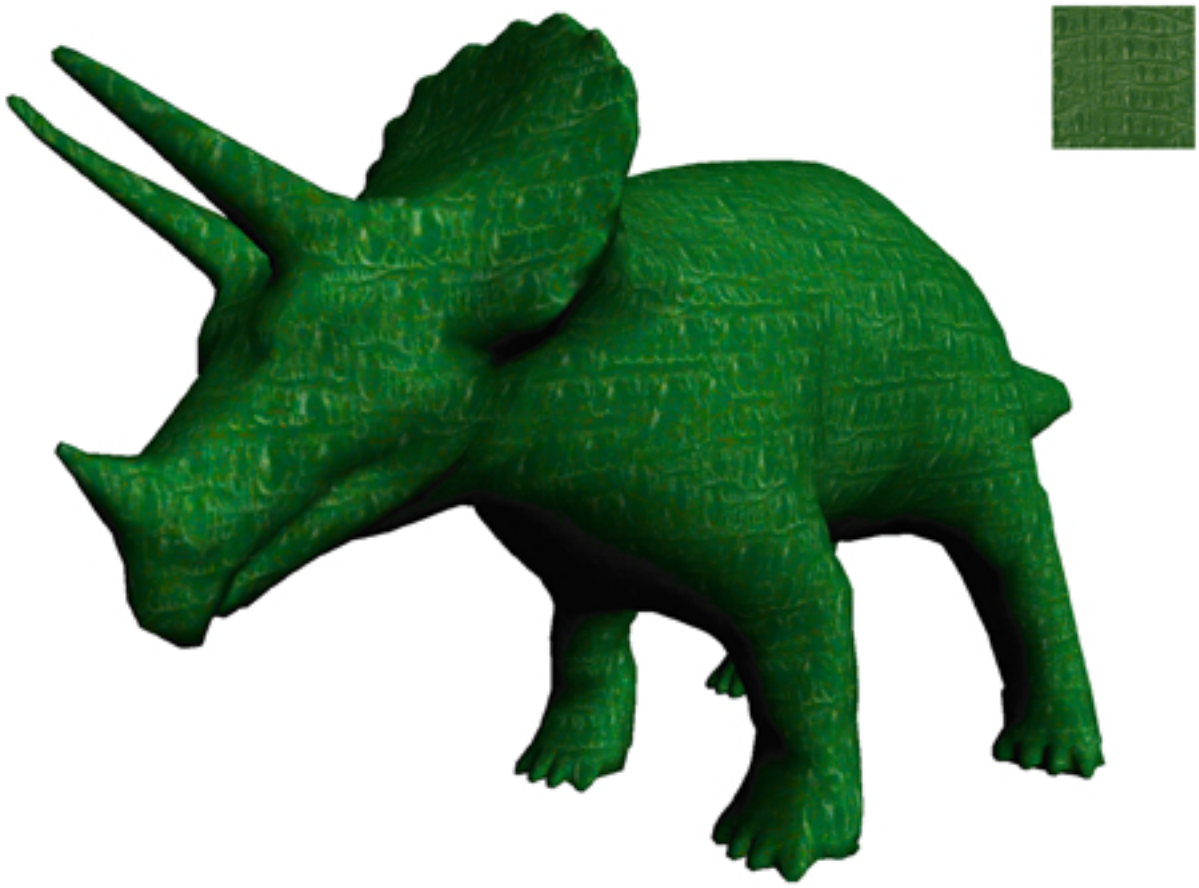


Figure 6: The crocodile skin texture (D10) synthesized over the triceratops model following the direction field (0,1,0) locally projected onto the tangent plane.

Recent results in biological vision research support the idea that the principal directions play an important role in surface shape understanding, and we are interested in probing these ideas further through controlled studies of the effects of texture pattern orientation on observers' perception of the 3D shapes of complicated underlying models. Mammassian and Landy [18] have shown that observers' interpretations of line drawings of simple patches are consistent with an inherent bias, among other things, towards perceiving lines on objects as being oriented in the principal directions, supporting an observation made by Stevens [25] nearly 20 years ago. Li and Zaidi [16] examined observers' ability to estimate the relative curvatures of developable surfaces textured with various implicitly or explicitly plaid-like patterns, and concluded that shape perception depends critically upon the observation of changes in oriented energy along lines corresponding to the principal directions. However these ideas remain to be examined in the context of more complicated, arbitrary surfaces, where the first and second principal directions can switch places numerous times. A significant challenge in this effort is to obtain accurate computations of the principal direction vector fields.

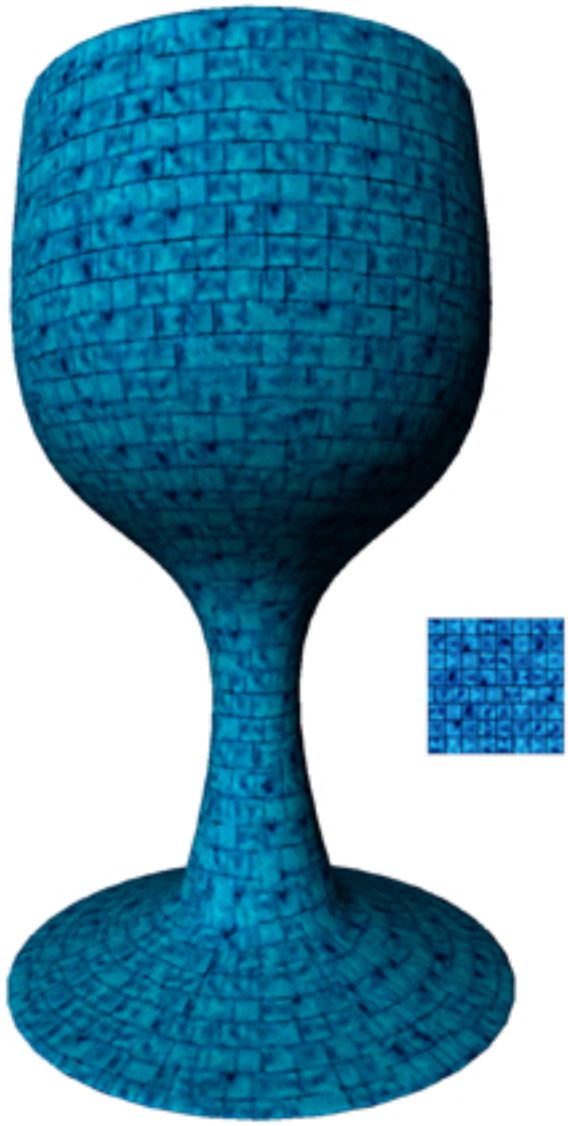


Figure 7: Glass block texture applied to a simple model with a constant directional field. Note the general preservation of continuity in the texture pattern and the relative consistency of the bricks' shapes and sizes, despite scattered artifacts. The direction field is locally given by the projection of the central axis of the object onto the tangent plane of each patch. Some of the patches in this model were resynthesized in a postprocess.

Figure 8: The orientation of a directed pattern over a curved surface can influence our perception of the surface's 3D shape. On the left, the bricks are oriented in the direction of greatest signed normal curvature; in the middle they are oriented in the direction of least signed normal curvature, and on the right they are oriented in the same constant "up" direction used for the models in figures 6-7.

We are currently working with Dr. Jack Goldfeather to develop robust methods for computing smooth, accurate principal direction vector fields across arbitrary polygonally-defined objects [9]. A complementary approach being developed by Bertalmio *et al.* [3] has the potential to facilitate the anisotropic smoothing of these fields. Our present results in applying an anisotropic texture over the surface of an object such that its dominant orientation is everywhere aligned with the first and second principal directions are shown in figure 8, and contrasted there with the results obtained using a constant "up" direction. Despite the preliminary state of work in shape representation via texture, we believe that there is significant inherent potential for a method capable of synthesizing a variety of principal direction textures over arbitrary curving forms.

4. Implementation

Our system is fully automatic, and does not require user interaction during either the splitting or texture synthesis process. The system has several parameters which can be adjusted by the user in order to increase the likelihood of obtaining optimal results with different kinds of textures or models. Within the splitting stage, these parameters include: an upper bound on the size of any single patch during splitting and an upper bound on the angle that the normal of any member triangle can make with the reference direction for a patch. Within the texture synthesis stage, the user may first choose among several possible texture orientation options: to have the texture follow the direction of greatest or least signed or unsigned normal curvature, to follow the projection onto the local tangent plane of a constant specified direction, or to follow no specific direction (in which case the pattern is assumed to be invariant under rotation). With regard to texture synthesis, the user may also control: the sizes of the neighborhoods used in the each pass of the texture synthesis (larger neighborhoods generally increase the computational expense of the synthesis but are sometimes necessary in order to preserve features across a range of different scales); the number of first-round preselected locations to be tested for a match on the second pass; and the weighting scheme used over the neighborhood during the matching process.

In most cases, it is sufficient to define the direction of texture synthesis across a patch according to the distribution pattern of previously textured pixels in the boundary region, under the assumption that starting from the side containing the greatest number of previously filled pixels will provide the most stable seed for the synthesis. Unfortunately, this is not always true (see the section on errors below). Certain strongly directional patterns seem to yield better results when the synthesis is performed in a particular set of directions (left/right or top/down). For quickly varying direction fields, starting from areas in which the direction field is most calm seems to improve the quality of the synthesized patch.

We use one of two different methods to determine the direction in which the synthesis proceeds from patch to patch. The simple method that works well on many vector fields is to begin with a randomly chosen patch and proceed to any blank connected patch, filling in any holes at the end. For principal direction vector fields we get somewhat better results choosing the blank connected patch in which the difference between the

two principal curvatures is greatest. This favors working first in areas over which the principal directions are clearly defined, providing a stable seed for the synthesis of the other patches.

The most computationally expensive part of the algorithm is the texture synthesis, which accounts for 95 to 99% of the running time. Partitioning and optimizing the patches takes only about 1-3 seconds for simple meshes such as the Venus and triceratops, 10-12 seconds for larger meshes such as the 70,000 triangle bunny.

The synthesis time depends on the required number of pixels, which determines the final resolution of the texture on the object, and the size of the pre-select list needed to correctly synthesize the texture. Growing the weave texture on the Venus model in figure 1 took slightly less than 12 min. The goblet/glass block combination required about 20 minutes. These timings refer to a C++ implementation compiled with gcc[2] running on a standard linux (2.2.16) 933MHz Pentium III PC with a 32Mb GeForce 2 GTS.

4.1.1 Limitations

In its current form, our implementation is limited to the synthesis of grayscale and pseudo-color mapped images; this is common for many of the algorithms discussed in section 2. While this may not be an optimal final choice, beginning our algorithmic development in the greyscale domain allowed us to sidestep initially the problems of choosing an appropriate color space and finding a good weighting function for the three channels, and to focus immediately on the central critical issues in the synthesis of artifact-free fitted textures. We achieved the colorized textures shown in this paper by the following method. An optimal color map is extracted from a true color image using a paint program such as "The GIMP". The (YIQ) colormap is then sorted by Y and the color indices in the picture used directly as a greyscale image. When the synthesis is finished, the sorted colormap is applied to the greyscale synthesized texture using the gray value again directly as an index into the colormap. This index-based method is not formally correct, as very different hues could end up to be nearly adjacent in the colormap because only the color *order* is defined by the Y values. However, we did not observe appreciable differences in the results obtained using this approach compared with using a correct greyscale image formed from the Y values only. The disadvantage in using the exact Y values directly, as opposed to relying on the ordered color indices, is that it would force the use of floats very early in the program, making things more complicated.

A second limitation that bears mentioning is that we found that the large amounts of texture generated by the synthesis caused problems for certain architectures, specifically those that had hard built-in limitations on the amount of texture memory that could be used. When a high level of detail is desired, without any possibility for pattern repetition, the total amount of texture required to cover the mesh can easily exceed the total size of the texture memory. All of the machines that did not allow storage of textures in main memory failed to display the more complex models (e.g. the crocodile skinned triceratops). Using a texture atlas similar to the one described in [24] might help reduce the texture memory usage, at the cost of incorrect

[2] optimization flags -O2 -funroll-loops -fstrict-aliasing -march=i686

mipmapping. In our current implementation each patch is stored as a single texture, producing a texture memory waste of up to 25% to 50% depending on the model. Thanks to OpenGL texture compression extensions, all of the models presented in this paper can maintain interactive frame rates on our standard PC however, even in cases where the amount of uncompressed texture exceeds 100Mb.

The method that we have proposed is currently designed to be applied to static models, and we have not thought about how to extend it to the case of deforming animated objects. Modifications to make the texture synthesis process deterministic are an obvious first step toward satisfying this requirement, but it is not immediately clear how one could guarantee that independently synthesized patterns will not differ profoundly from frame to frame as the mesh defining the object is globally deformed.

To achieve good results with the shape-following textures, the direction field must be band-limited: for any given texture there is a maximum spatial frequency that can be followed in the synthesis process and still produce correct results. Nevertheless we found that the method did a good job at singular points on the brick-textured lava lamp object, as can be seen in figure 8.

Sometimes, as described in [7], the synthesis algorithm fails and produces undesirable results across all or part of a patch. Depending on the texture, we find that 0-5% of the patches typically contain some synthesis errors. Unfortunately, areas as small as 5x5 pixels (in a 128x128 pixel patch) are easily noticed. The difficulties in automatically detecting such small areas prevented the implementation of a mechanism for automatic correction. However, the current implementation allows the user to interactively select and re-synthesize the unaesthetic patches after the main synthesis process has finished. Figs 7-11 in this paper show models in which parts of the texture were resynthesized across one or more patches. Figs 1, 4 and 6 show results that were obtained without any such postprocessing.

5. Applications And Future Work

There are many promising applications for this system and many directions for future work. One of the most interesting of these is multi-texturing. On a per-pixel basis it is possible to change not only the direction of the synthesized texture but even the texture itself according to any arbitrary function. Figures 10 and 11 are made using the illumination equation and two and four different textures respectively, each one with 63 rotations. These models, like all of the others in this paper, can be displayed at interactive frames rates on our standard PC.

The multi-texturing methods described in this paper have the potential to be useful for important applications in scientific visualization, for example in encoding a scalar distribution using texture type variations across an arbitrary domain in 2D or 3D. Other direction fields, such as gradient descent, hold promise for different applications, such as non-photorealistic rendering of terrain models (esp. in the case when it is desired to see through the surface). The methods that we have proposed can also be used for the visualization of scientifically computed vector fields over surfaces. An intriguing possible use for an extension of this work is in defining texture mixtures.

Figure 10: A demonstration of multi-texturing, in which the search for matches is performed within an array of different texture types. The texture type index can be defined by any function. In this example, we used the illumination function. In a real application one would want to use something more meaningful, such as soil type.

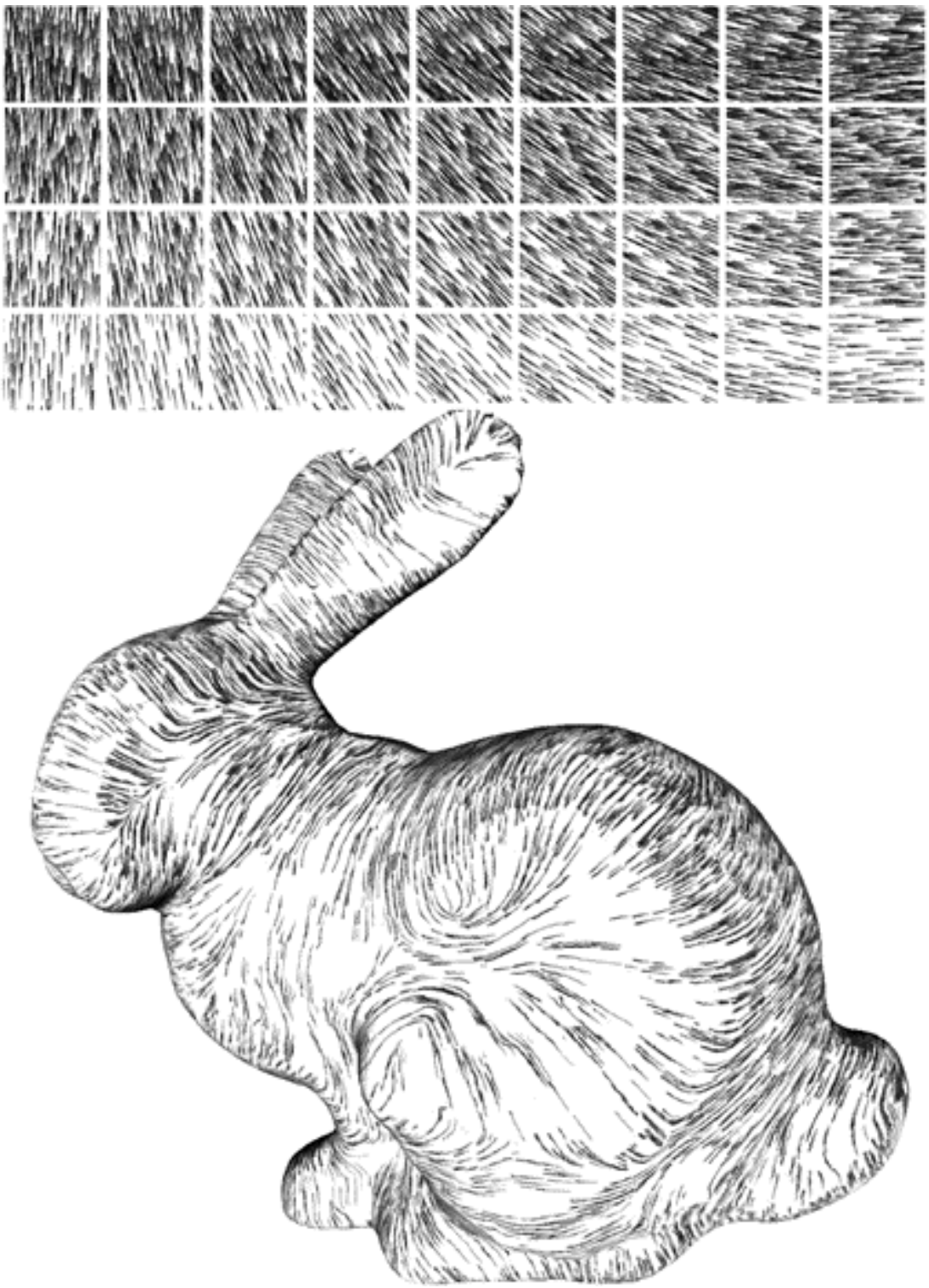


Figure 11: Multiple textures containing lines of different widths applied to an automatically-defined smooth vector field approximating the first principal direction over the Stanford bunny. Indexing along the dimension of varying width was done as a function of the illumination.

6. Conclusions

In this paper we describe a fully automatic method for synthesizing, from a small provided sample, patches of a perceptually equivalent texture pattern that can be fit over contiguous sets of polygons in an object. This is done without appreciable seams or projective distortion, and the texture pattern can be locally oriented to follow any computed vector field over the surface. Our method is straightforward and computationally efficient, and demonstrates good results with a variety of different texture types and models.

7. Acknowledgments

This work was supported by a University of Minnesota Grant-in-Aid of Research, Artistry and Scholarship. Additional support was provided by an NSF Presidential Early Career Award for Scientists and Engineers (PECASE) 9875368, NSF CAREER award CCR-9873670, ONR-N00014-97-2-0509, an Office of Naval Research Young Investigator Award, Presidential Early Career Award for Scientists and Engineers (PECASE) Navy/N00014-98-1-0631, and the NSF Learning and Intelligent Systems Program.

8. References

[1] Nur Arad and Gershon Elber. Isomeric Texture Mapping for Free-form Surfaces, *Computer Graphics Forum*, *1997*, **16**(5): 247– 256.

[2] Chakib Bennis, Jean-Marc Vézien and Gérard Iglésias. Piecewise Surface Flattening For Non-Distorted Texture Mapping, *Proceedings of SIGGRAPH 91*, pp. 237 – 246.

[3] Marcelo Bertalmio, Guillermo Sapiro, Li-Tien Cheng and Stanley Osher. A Framework for Solving Surface Differential Equations for Computer Graphics Applications, UCLA-CAM report 00-43, December 2000 (www.math.ucla.edu).

[4] Phil Brodatz. Textures: A Photographic Album for Artists and Designers, Dover Publications, 1966.

[5] Jeremy S. De Bonet. Multiresolution Sampling Procedure for Analysis and Synthesis of Texture Images, *Proceedings of SIGGRAPH 97*, pp. 361 - 368

[6] J.-M. Dischler, D. Ghazanfarpour and R. Freydier. Anisotropic Solid Texture Synthesis Using Orthogonal 2D Views, Computer Graphics Forum, **17**(3), September 1998.

[7] Alexei A. Efros and Thomas K. Leung. Texture Synthesis by Non-Parametric Sampling, *Proceedings of the International Conference on Computer Vision*, vol. 2, 1999, pp. 1033 -1038.

[8] James A. Ferwerda, Sumanta N. Pattanaik, Peter Shirley and Donald P. Greenberg. A Model of Visual Masking for Computer Graphics, *Proceedings of SIGGRAPH 97*, pp. 143 – 152.

[9] Jack Goldfeather. "Understanding Errors in Approximating Principal Direction Vectors", TR01-006, Dept. of Computer Science and Engineering, Univ. of Minnesota, Jan. 2001.

[10] Gabriele Gorla, Victoria Interrante and Guillermo Sapiro. Growing Fitted Textures, *IMA preprint 1748*, Feb. 2001.

[11] Steven Haker, Sigurd Angenent, Allen Tannenbaum, Ron Kikinis, Guillermo Sapiro and Michael Halle. Conformal Surface Parameterization for Texture Mapping, *IEEE Transactions on Visualization and Computer Graphics*, **6**(2), April/June 2000, pp. 181-189.

[12] Pat Hanrahan and Paul Haeberli. Direct WYSIWYG Painting and Texturing on 3D Shapes, *Proceedings of SIGGRAPH 90*, pp. 215 – 223.

[13] David J. Heeger and James R. Bergen. Pyramid-Based Texture Analysis/Synthesis, *Proceedings of SIGGRAPH 95*, pp. 229 – 238.

[14] Victoria Interrante. Illustrating Surface Shape in Volume Data via Principal Direction-Driven 3D Line Integral Convolution, *Proceedings of SIGGRAPH 97,* pp. 109 – 116.

[15] Bruno Lévy and Jean-Laurent Mallet. Non-distorted Texture Mapping for Sheared Triangulated Meshes, *Proceedings of SIGGRAPH 98*, pp. 343 – 352.

[16] Andrea Li and Qasim Zaidi. Perception of Three-Dimensional Shape From Texture is Based on Patterns of Oriented Energy, *Vision Research*, **40**(2), January 2000, pp. 217 – 242.

[17] Jérôme Maillot, Hussein Yahia and Anne Verroust. Interactive texture mapping, *Proceedings of SIGGRAPH 93*, pp. 27 – 34

[18] Pascal Mamassian and Michael S. Landy. Observer Biases in the 3D Interpretation of Line Drawings, *Vision Research*, **38**(18), September 1998, pp. 2817 – 2832.

[19] Fabrice Neyret and Marie-Paul Cani. Pattern-Based Texturing Revisited, *Proceedings of SIGGRAPH 99*, pp. 235-242.

[20] Darwyn R. Peachey. Solid texturing of Complex Surfaces, *Proceedings of SIGGRAPH 85*, pp. 279–286.

[21] Ken Perlin. An Image Synthesizer, *Proceedings of SIGGRAPH 85*, pp. 287 – 296.

[22] Dan Piponi and George Borshukov. Seamless Texture Mapping of Subdivision Surfaces by Model Pelting and Texture Blending, *Proceedings of SIGGRAPH 2000*, pp. 471 – 478.

[23] Javier Portilla and Eero P. Simoncelli. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients, *International Journal of Computer Vision*, **40**(1), October 2000, pp. 49-70.

[24] Emil Praun, Adam Finkelstein and Hughes Hoppe. Lapped Textures, *Proceedings of SIGGRAPH 2000*, pp. 465-470.

[25] Kent A. Stevens. The Visual Interpretation of Surface Contours, *Artificial Intelligence*, **17**(1-3), August 1981, pp. 47-73.

[26] Greg Turk. Generating Textures for Arbitrary Surfaces Using Reaction-Diffusion, *Proceedings of SIGGRAPH 91*, pp. 289 – 298.

[27] Li-Yi Wei and Marc Levoy. Fast Texture Synthesis Using Tree-structured Vector Quantization, *Proceedings of SIGGRAPH 2000*, pp. 479 – 488.

[28] Andrew Witkin and Michael Kass. Reaction-Diffusion Textures, *Proceedings of SIGGRAPH 91*, pp. 299– 308.

[29] Steven Worley. A Cellular Texture Basis Function, *Proceedings of SIGGRAPH 96*, pp. 291– 294.

[30] Song Chun Zhu, Ying Nian Wu and David Mumford. Filters, Random Fields and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling, *International Journal of Computer Vision*, **27**(2), March 1998, pp. 107-126.

Opacity-modulating Triangular Textures for Irregular Surfaces

Penny Rheingans
Department of Computer Science
University of Mississippi

Abstract

Many scientific and medical visualization techniques produce irregular surfaces whose shape and structure need to be understood. Examples include tissue and tumor boundaries in medical imaging, molecular surfaces and force thresholds in chemical and pharmaceutical applications, and isosurfaces in a wide range of 3D domains. The 3D shape of such surfaces can be particularly difficult to interpret because of the unfamiliar, irregular shapes, the potential concavities and bulges, and the lack of parallel lines and right angles to provide perspective depth cues. Attempts to display multiple irregular surfaces by making some or all of them transparent further complicates the problem. Texture can provide valuable cues to aid in the interpretation of irregular surfaces. Opacity-modulating textures offer a mechanism for the display of multiple surfaces without the extreme loss of clarity of multiple transparent surfaces. This paper presents a method for creating simple repeating textures and mapping them onto irregular surfaces.

1. Introduction

Many applications of visualization require the display of irregularly shaped surfaces. Understanding terrain undulations, cloud shapes, molecular surfaces, or tissue masses requires the interpretation of surfaces with irregular, and often unfamiliar, shapes. Display of the 3D statistical surface defined by scalar values in a 2D domain, commonly called a mountain plot, can involve similar irregular surfaces. Additionally, visualization operations often create additional irregular surfaces for interpretation. Isosurface generation is the most common of these, defining the boundaries of 3D regions in a wide variety of applications. Isosurfaces are used to delineate 3D regions where certain conditions hold, such as places where air pollutant levels exceed legislated limits, tissue density deviates from normal values, toxin levels in river sediment are unacceptably high, or air flow creates high pressure values. In such situations, observers use these isosurfaces to judge whether additional pollution restrictions should be imposed, a biopsy should be performed to investigate a suspected tumor, the river bottom should be dredged to remove contaminated sediment, or the building should be redesigned to improve air flow. Such critical decision-making depends on accurate interpretation of isosurface shapes.

Arbitrary surfaces are particularly hard to understand because they can lack many of the visual cues observers generally use to interpret three-dimensional shape. The lack of parallel lines and right angles reduce the power of perspective depth cues. Unfamiliar shapes limit interpretation based on past experiences with similar shapes. Irregular surfaces can self-shadow and self-obscure themselves in complex ways, reducing the effectiveness of lighting and obscuration cues. When surfaces are made transparent, in order to show what lies behind or within them, their shape becomes even more difficult to perceive accurately.

The application of texture to irregular surfaces can make them easier to interpret by providing additional shape cues. Texture can help disambiguate obscuration cues by clarifying how different parts of the surface meet and hide one another. The effects of perspective, and from them additional depth cues, are visible in the texture gradient. Specifically, the apparent scale of the projected texture decreases as the surface recedes from the observer. The texture in distant patches appears to be compressed, a denser pattern with smaller features. These perspective effects are likely to be more obvious on the texture than on the object itself. This is because texture shape in one area can be compared to texture shape in another, with differences indicating depth differences. Similar comparisons among geometric features of the object would not be as enlightening because object shape in different areas cannot necessarily be expected to be similar. The texture gradient also provides information about the orientation of surface segments, as textures undergo convergence on surfaces that drop away from the observer. More information about surface perception from texture can be found in the perception literature [for instance, Gibson50, Cutting84, Todd87, Cumming93].

Adding texture to transparent surfaces results in even greater perceptual benefits, since perception of shape from completely transparent surfaces is so difficult to begin with. For maximum benefit, the texture should combine transparent parts which show what lies behind with opaque parts which improve shape perception. I call such textures *opacity-modulating*, since they specify variations in the opacity of the surface as well as, or instead of, the color. The visual features of such textures can be either direct changes in color and intensity or indirect changes that occur when the identity of the visible surface changes as a result of changes in opacity.

Published in *Proceedings IEEE Visualization '96*, IEEE Computer Society Press, Los Alamitos CA, October 1996, pp. 219-225..

2. Related Work

Schweitzer first proposed that artificial textures, those not necessarily mimicking any realistic texture, could be used to improve the perception of surface shape from rendered images [Schweitzer83]. He generated screen space texture elements with feature density and shape determined by the surface depth and orientation.

Previous uses of transparent textures to improve shape perception have included screen-space textures for automatic illustration [Dooley90], solid grid textures in medical imaging [Levoy90], and solid equation textures for mathematical visualization [Wejchert92]. Interrante [Interrante95] addressed the problem of accurately communicating the shape of transparent skin surfaces by adding opaque ridge and valley lines reminiscent of artists' renderings. This technique works very well on familiar shapes with relatively sparse ridge and valley features, such as the medical applications for which it was developed. The unfamiliar shapes and multiple self-obscurations of arbitrary surfaces limit the usefulness of ridge and valley lines.

Reaction diffusion textures, such as those developed by Turk [Turk91], can be generated directly on a surface, guaranteeing that the texture will follow surface contours. Unfortunately, systems of reaction-diffusion equations are generally tricky to define and slow to compute. Although texture values can be stored directly as vertex colors once they have been computed, making texture mapping solely a preprocessing step which does not impact view-time rendering speed, this forces texture repeats to be several times the size of typical polygons if texture details are to be visible.

In molecular visualization, texture mapping has increasingly been used to accelerate the mapping from molecular properties to surface characteristics [Teschner94]. In this property-based texture mapping, molecular properties at vertices are used to generate texture coordinates, so surface texture follows the distribution of property values across the surface. Duncan and Olson [Duncan95] describe location-based texture mapping for parametric molecular surfaces, characterized by a one-to-one correspondence between surface points and texture elements. In both sorts of molecular texture mapping, the objective is the display of additional information, rather than the improvement of surface shape perception. The textures used are generally opaque.

3. Requirements

What characteristics should a texture have in order to give strong shape cues on irregular surfaces? Appropriate textures are fine-grain repeating patterns which follow the surface of the object. Small, regular textures provide visual features without drawing undue attention to themselves. They must be able to be tiled without visible seams, since such seams are artifacts of the polygonal decomposition or specific texture mapping and might draw the observer's attention without conveying any useful information about surface shape. These stringent pattern requirements practically preclude that suitable textures can simply be scanned in; they need to be generated algorithmically.

Additionally, textures should be easy to generate and quick to render. Generally, this means that textures should be *image-based*, a precomputed image stored in texture memory, rather than *procedural*, defined by a procedure evaluated during the shading calculations made at each pixel. This allows applications to exploit the texture-mapping hardware available on common graphics workstations. Rendering speed is particularly important, since interactive viewing of complex three-dimensional shapes is a source of strong shape cues. *Kinetic depth effect* occurs when an object is rotated or translated to reveal previously obscured portions. The manner in which the surface appears from, and disappears to, behind the surface reveals much about the 3D structure underlying the 2D projection. Conversely, *head-motion parallax* results when the observer's virtual position in the environment changes, revealing different parts of the object. This position can be controlled in a variety of ways, but the strongest effects occur when head position is tracked to automatically control view position [vanDamme94].

Mapping a two-dimensional surface texture onto an arbitrary surface in 3D poses a problem. Arbitrary surfaces are not *developable*; that is, they cannot be laid out flat without distortion. By way of example, a cylinder is developable; a sphere is not. The difficulty lies in assigning texture coordinates to vertices in a way that accurately represents the surface geometry. A common method of mapping a texture onto a surface is to project the texture orthogonally or cylindrically onto the surface. The more the object is like a plane, or a cylinder, the better this works. Unfortunately, in these sorts of projections depth information (or the distance from the surface to the texture plane) is ignored. This distorts the texture in areas where the surface is not orthogonal to the direction of projection. Since we wish to use the apparent texture gradient distortions from the viewing and perspective transformations to convey shape information, texture distortions from the mapping process are clearly undesirable.

Alternatively, solid texture could be used to circumvent the problems of 2D-to-3D texture mapping. The appropriate texture value at each location on the object surface would simply be the value at the corresponding location in the 3D texture map. The result can be thought of as an object of the specified shape carved out of a block of the material described by the 3D texture. Unfortunately, solid textures do not provide the desired shape cues since the texture does not follow the surface contours. While texture gradient

changes due to perspective will still be apparent, those due to orientation will not. In order to provide the appropriate shape cues, we require a texture that is generated directly on the surface; that is, one which is based on the shape of the surface itself, not on the underlying coordinate system.

4. Triangular Texture Space

A certain class of textures generated in a triangular texture space can greatly simplify the problem of mapping a 2D texture onto an arbitrary polygonal surface. Assume the surface is composed entirely of triangles. If not, an arbitrary polygon mesh can easily be transformed into a triangle mesh. For simplicity, use extreme points of (0,0), (1,0) and (0.5, $\sqrt{3}/2$) to create an equilateral triangular texture element. Now the problem has become that of mapping one triangle onto another. The assignment of texture coordinates to triangle vertices has been simplified to the assignment of the coordinates of one of the texture element extreme points to each triangle vertex.

Next, if the sequence of texture values along each side is identical, and consequently the values at the texture element extreme points are the same, the problem is further simplified. Now, any side of the texture element will match seamlessly with any other side, so the texture element may be placed on the triangle in any orientation. Simply assign coordinates to one vertex randomly and then assign the other coordinates in turn, walking in the same direction around the texture element and triangle.

Clearly this simplification of the texture mapping process is not suitable for the creation of realistic textures on familiar objects, but it suffices for the creation of small scale, regular texture patterns on unfamiliar surfaces for the purpose of enhancing shape perception.

Triangular textures were created using *Pdbq*, a special-purpose language for molecular visualization developed by Palmer [Palmer92]. Pdbq is a C-like interpreted language which provides basic and geometric data types, control structures, overloaded operators, streams to support both simple and geometric file I/O, and built-in functions to perform standard, molecular, and geometric tasks. Simple, symmetric triangular textures [Figure 1] can be generated by such methods as:

```
{spots at texel center -- Fig 1a}
for all points (x,y)
   opacity(x,y)   =   1   -   dist((x,y),
   centroid)
```

or:

```
{spots at vertices (v1,v2,v3) -- Fig 1b}
for all points (x,y)
   opacity(x,y) =
      1       -       min(dist((x,y),v1),
         dist((x,y),v2), dist((x,y),v3)
```

or:

```
{ring texture: RAD sets radius -- Fig 1c
}
for all points (x,y)
   opacity(x,y) =
      1-MAX(ABS(RAD-min(dist((x,y),v1),
      dist((x,y),v2),dist((x,y),v3)
```

Designing textures which modulate opacity offer specific challenges. An effectiveß texture will have both enough transparent parts for the objects behind to be seen clearly and enough opaque parts for the surface on which it lies to be clearly perceived. Unfortunately, these requirements are directly at odds with one another. The right balance must be struck on a case-by-case basis, depending on clarity of the inside objects, the irregularity of the outside surface, and the relative importance of the various objects.

5. Tessellation

Obviously, the most regular texture patterns will be formed over surfaces with uniformly sized triangles. On such surfaces, texture elements are replicated at a constant size across the surface. Additionally, since the texture element

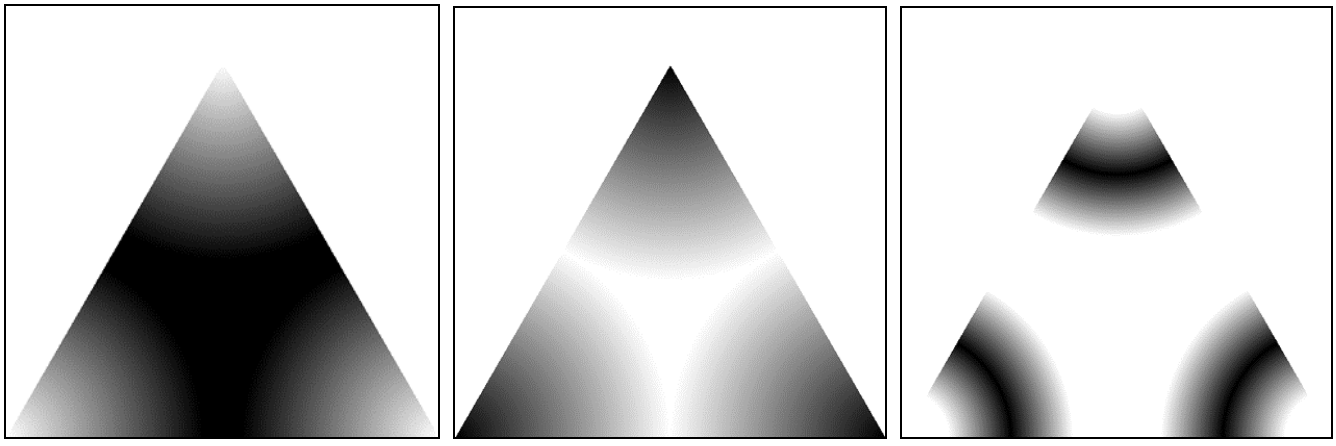


Figure 1. Three simple triangular textures with opaque area a) at element centroid, b) at vertices, and c) in rings around vertices. Dark areas represent high opacity.

is defined on an equilateral triangle, the textured surface will appear most regular when triangles of the surface are also equilateral. Clearly, an arbitrary surface will not necessarily be comprised of uniformly sized equilateral triangles. On irregular surfaces, triangles which are unusually large, small, or skinny result in disruptions of the texture pattern. Figure 2a shows the initial tessellation resulting from the computation of the solvent-accessible surface of a molecule. Figure 2b shows the result of texturing this surface with a triangular texture. Disruptions of the texture regularity are especially visible in the lower left part of the molecule. The textures used in Figure 2 are completely opaque to show the effects of the tessellation more clearly.

Texture pattern regularity can be improved by a preprocessing step which regularizes the polygonal tessellation, producing triangles which are more uniformly sized and closer to equilateral. This re-tiling was performed using an automatic method developed by Greg Turk [Turk92]. First, a new set of points is chosen at random locations on the surface. These points will later become the vertices of the regularized tessellation. Then, a relaxation procedure is applied to move points away from neighboring points by way of a simulated repulsion force. The result of this process is a set of candidate points uniformly distributed across the polygonal surface. Next, each polygon of the original triangulation is replaced by a set of triangles which uses the original polygon vertices and the

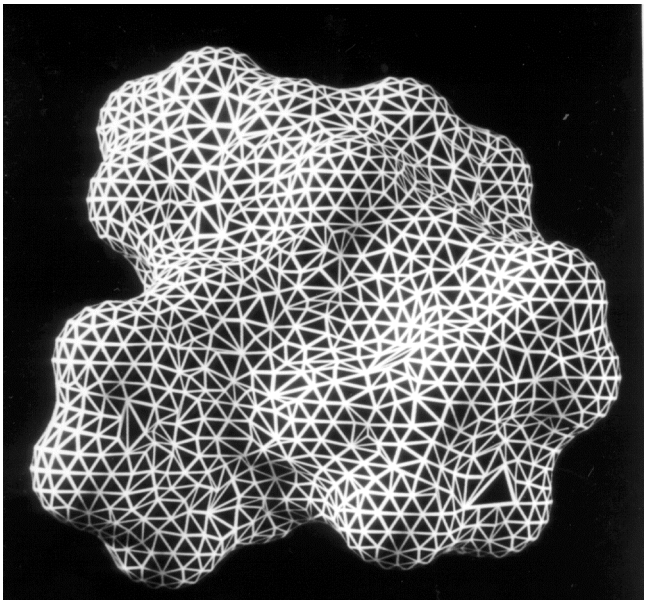
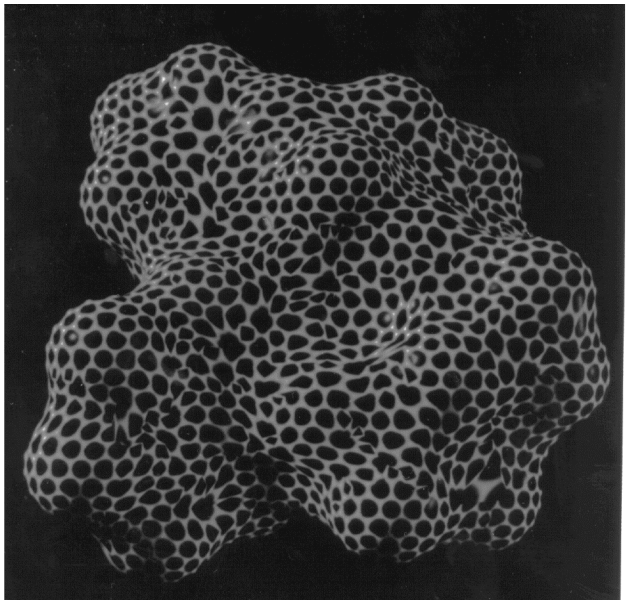
a)  b) 

Figure 2. a) Initial tessellation of solvent-accessible surface and b) result of applying triangular texture to that surface.

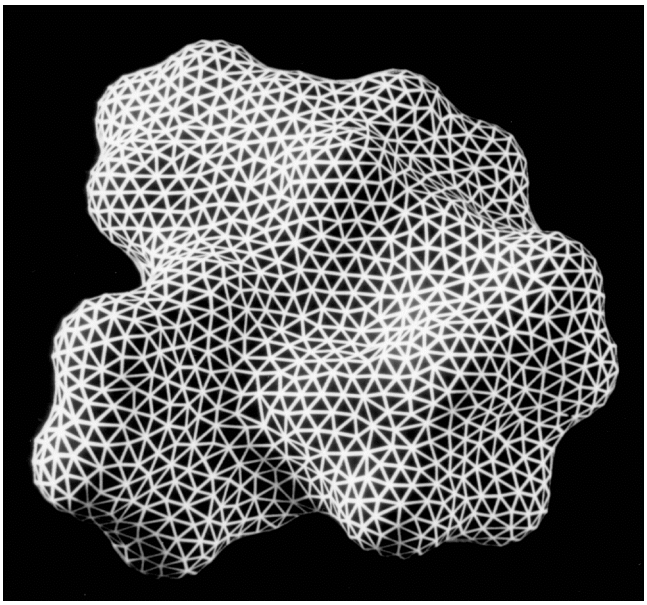
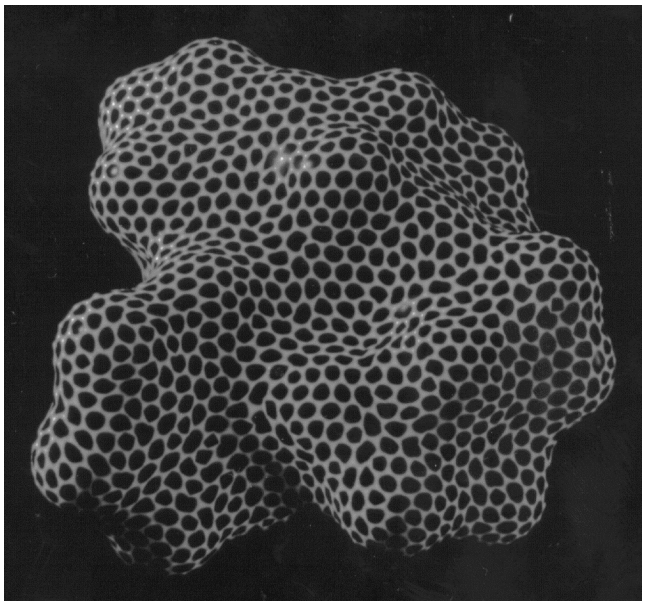
a)  b) 

Figure 3. a) Regularized tessellation of solvent-accessible surface and b) result of applying triangular texture to the new surface.

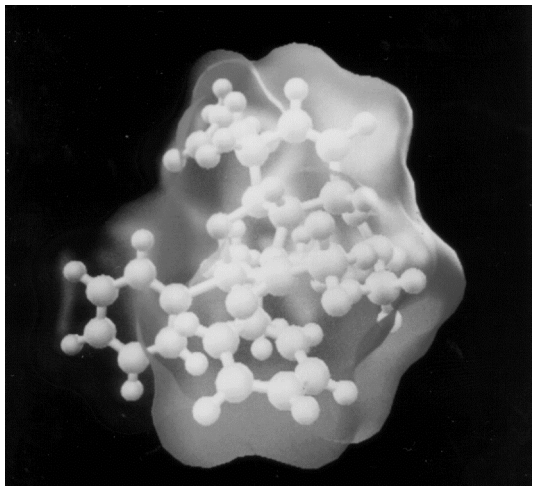
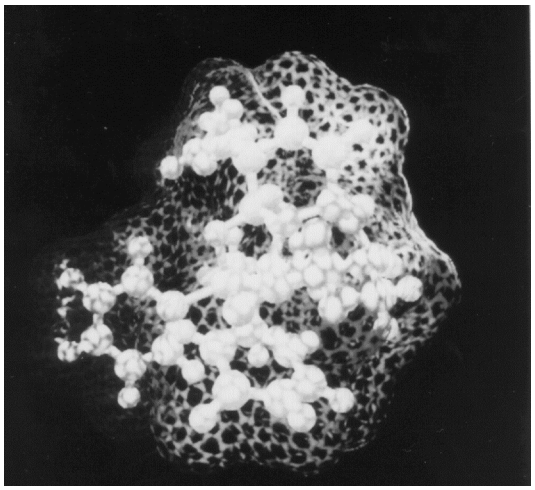
a)  b) 

Figure 4. Comparison of a) transparent surface to b) opacity-modulating surface for molecular application. Color represents an additional variable (in this case, simply position in x) evaluated at the solvent-accessible surface. See color plates.

candidate vertices which lie in the polygon to exactly tile the polygon. This mutual tessellation preserves the connectedness of the original surface. Finally, the original vertices are removed from the mutual tessellation, resulting in a polygonal surface containing only the new vertices. Figure 3 shows the result of regularizing the molecular surface shown in Figure 2. Notice that triangles are distributed more evenly, resulting in a more regular texture pattern.

In fact, even just the hidden surface view of the new tessellation (Figure 3a) shows the molecule shape clearly, since the polygon boundaries form a relatively small scale, regular pattern. This image was rendered by the same method as the textured surfaces, in this case a triangular texture with high value near the boundaries of the texture element was used. This pattern, however, would not work well for an opacity-modulating texture, since the opaque and transparent areas are not properly balanced.

Although these same methods can be used to create polygonal models at various levels of detail, the molecular surfaces used in the figures were re-tiled with approximately the same number of polygons as the original tessellation, creating a more regular tessellation at the same level of detail.

6. Results

Details of the shape of the molecular surface are visible in the textured image. For instance, the vertically compressed texture pattern near the center of the image in Figure 3b indicates a “shelf” in the molecular surface. A similar feature is visible near the indentation on the right side of the molecule. In an untextured surface, the visibility of these features would be dependent on the position of lights in the scene. For many lighting configurations, these features would be extremely difficult to see.

Figure 4 shows the solvent-accessible surface of the same molecule represented with a transparent surface (Figure 4a) and an opacity-modulating textured surface (Figure 4b). In both cases the ball-and-stick representation of the molecule’s atoms and bonds can clearly be seen within the surface. The textured image gives better clues to the shape of the surface, particularly in places where the surface is nearly orthogonal to the view direction. Both images require close examination to understand the shape of such a complex, unfamiliar object, but at some places on the surface of the transparent object no amount of examination seems to reveal structural details. In both images color has been used to represent the values of another variable at points on the surface. While the overall distribution of color values is more striking in the transparent surface, in areas where elements of the ball-and-stick representation underlie the surface, the color is very difficult to discern. In the textured surface, color values are apparent at opaque parts of the surface. However, if the opaque bands are narrow, judging color values may require careful examination. While color appearance of transparent surfaces is always changed by the color of underlying surfaces, the color appearance of opaque parts of an opacity-modulating surface is only influenced by simultaneous contrast effects with nearby areas, including those visible through holes in the surface.

Statistical surfaces textured with opacity-modulating patterns are also more comprehensible than those rendered as transparent surfaces. Figure 5 shows a statistical surface of ozone concentration represented using both methods. In both images, ozone level is mapped redundantly to height and color. The surface in Figure 5b is textured with a rectangular opacity-modulating texture, but the same general principles as for triangular textures apply. Both techniques show the map outline lying below the surface, but texturing conveys more information about surface shape. In the textured image, the slight valley between the two ridges in the foreground is more visible. In the

transparent image, the ridges themselves are visible primarily from color changes and not from shape cues. Similarly, subtle undulations of the far left part of the surface are clearly visible in the textured image but appear to be a formless jumble in the transparent image.

The shape of multiple irregular molecular surfaces is virtually incomprehensible when displayed using transparent surfaces [Figure 6a]. In these images, the solvent-accessible surfaces for solvent atoms of two different radii are displayed along with the ball-and-stick representation. Surface structure where the screen projections of the different surfaces overlap is particularly difficult to discern. Display of the same surfaces using opacity-modulating textures is somewhat clearer [Figure 6b], especially in places where surface orientation causes substantial distortions of the texture gradient. Still, a clear understanding of the details of surface shape is difficult to glean from either image. When the image is animated, however, the improvement in clarity resulting from the textured surface is striking. The more abundant surface features provided by the repeating texture facilitate perception of the surface shape. Relative movement of these small features creates a strong sensation of depth relationships, improving surface perception.

7. Extensions

Stereo pairs of textured irregular surfaces should offer many of the benefits of rotation of the surfaces at a much lower computational cost. Just as moving textured surfaces offer compelling shape cues not present in either static

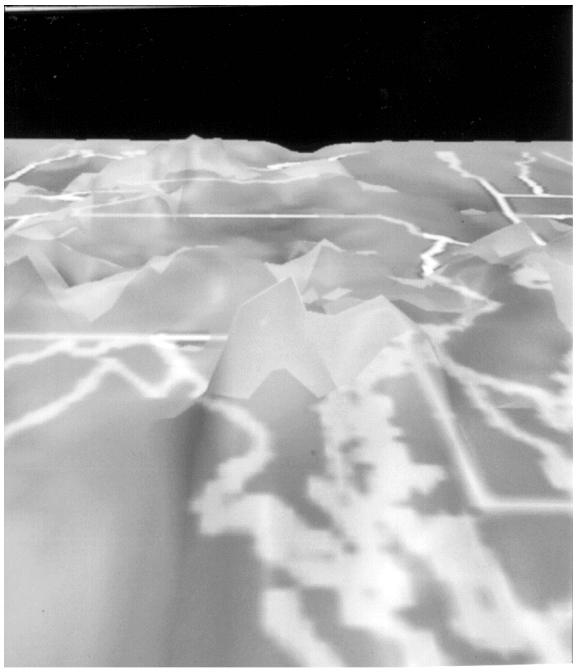
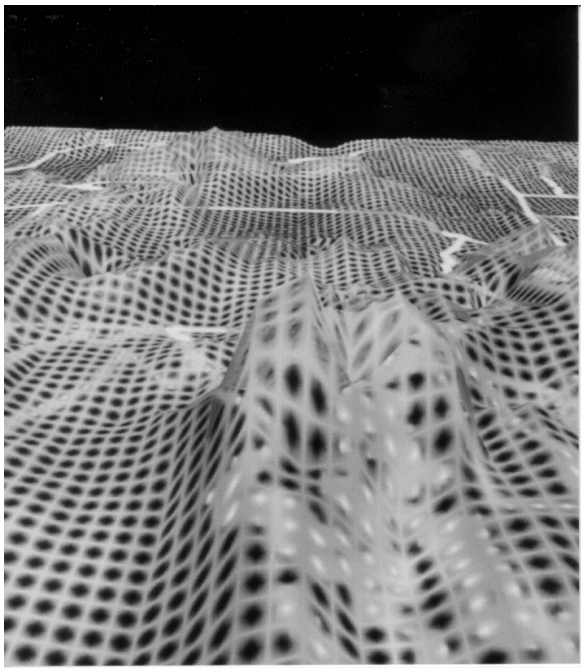
a)  b) 

Figure 5. Comparison of a) transparent and b) textured statistical surface. Ozone concentrations over the Eastern seaboard of the US are redundantly mapped to height and color in both images. See color plates.

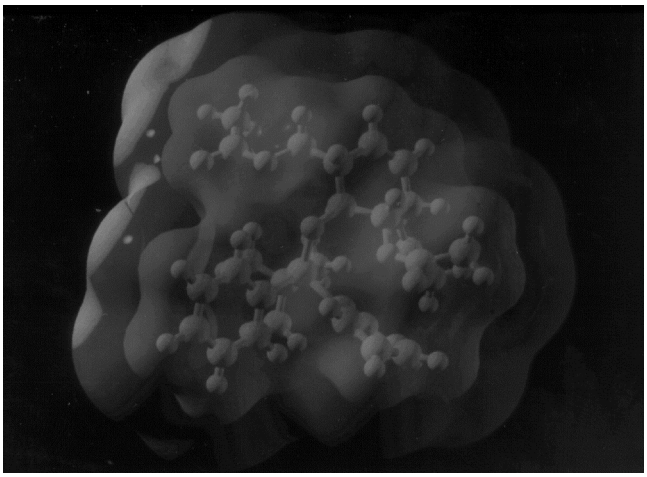
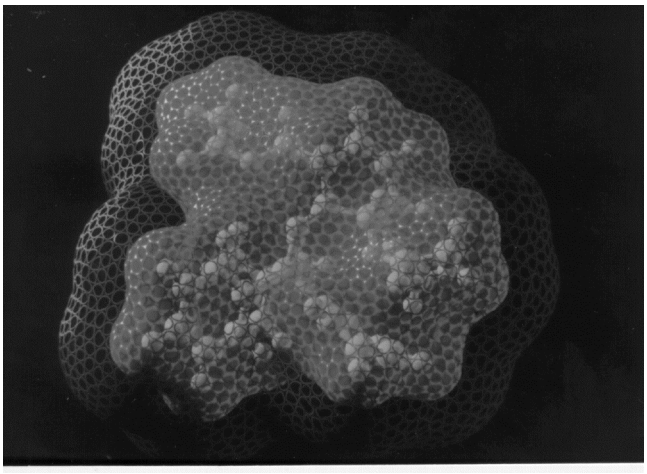
a)  b) 

Figure 6. Comparison of a) multiple transparent surfaces to b) multiple opacity-modulating surfaces. See color plates.

views or moving transparent surfaces, so should stereo views of textured surfaces. The richness of textural detail provides abundant features for matching locations between the two views. Since it is the positional disparity of features between the two views which creates the sensation of depth, the presence of adequate textural features ensures a strong perception of depth from positional disparity, just as it ensures a strong perception of depth from relative motion. This expectation is supported by the fact that the mechanisms of stereo perception are similar to those of motion perception, and by and large the two are performed by the same channels in the human visual system.

Another interesting question which remains is whether the texture pattern can be used to carry information of its own without disrupting its role in perception improvement. For instance, type of texture element could encode some classifier on the surface, such as tissue type or land use. As long as texture elements joined seamlessly at polygon boundaries, either because classification of adjoining polygons was guaranteed to match or because all texture elements had identical values along their boundaries, no complications would be introduced to the texture mapping process. Alternatively, texture density or opacity drop-off rate could represent confidence in the data. Such a technique would require that the texture be procedurally generated, unless the number of distinct values represented by texture characteristics was fairly small. Otherwise, a potentially prohibitive number of different texture elements would need to be generated and stored in order to represent the range of values (or combinations of values).

Acknowledgments

This work performed at the North Carolina Supercomputer Center and the US EPA Visualization Center. Greg Turk contributed ideas, encouragement, and the source code for his polygonal re-tiling methods. Tom Palmer provided valuable discussions and the molecular visualization language Pdbq. The manufactured carbohydrate named 'Wilma' was provided by Mark Zottola. Ozone concentration data came from the US EPA Atmospheric Research and Exposure Assessment Laboratory.

References

[Cumming93] Cumming, Bruce, Elizabeth Johnston, and Andrew Parker, Effects of Different Texture Cues on Curved Surfaces Viewed Stereoscopically. *Vision Research*, vol. 33, no 5/6, 1993, pp. 827-838.

[Cutting84] Cutting, James, and Robert Millard, Three Gradients and the Perception and Flat and Curved Surfaces. *Journal of Experimental Psychology: General*, vol. 113, no. 2, 1984, pp. 198-216.

[Dooley90] Dooley, Debra, an Michael Cohen, Automatic Illustration of 3D Geometric Models: Surfaces. *Proceedings of Visualization '90*, October 1990, pp. 307-314.

[Duncan95] Duncan, Bruce, and Arthur Olson, Texture mapping parametric molecular surfaces. *Journal of Molecular Graphics*, vol. 13, no. 4, August 1995, pp. 258-264.

[Gibson50] Gibson, James, *The Perception of the Visual World*, Houghton Mifflin, 1950.

[Interrante95] Interrante, Victoria, Enhancing Transparent Skin Surfaces with Ridge and Valley Lines. *Proceedings of Visualization '95*, October 1995, pp. 52-59.

[Levoy90] Levoy, Marc, Henry Fuchs, Stephen Pizer, Julian Rosenman, Edward Chaney, George Sherouse, Victoria Interrante, and Jeffrey Kiel, Volume Rendering in Radiation Treatment Planning. *First Conference on Visualization in Biomedical Computing*, 1990, pp. 4-10.

[Palmer92] Palmer, Thomas C., A Language for Molecular Visualization. *IEEE Computer Graphics and Applications*, vol. 12, no. 3, May 1992, pp. 23-32.

[Schweitzer83] Schweitzer, Dino, Artificial Texturing: An Aid to Surface Visualization. In Computer Graphics (SIGGRAPH 83 Proceedings], vol. 17, no. 3, July 1983, pp. 23-27.

[Teschner94] Teschner, Michael, Christian Henn, Horst Vollhardt, Stefan Reiling, and Jurgen Brickman, Texture mapping: A new tool for molecular graphics. *Journal of Molecular Graphics*, vol. 12, no. 2, June 194, pp. 98-105.

[Todd87] Todd, James, and Robin Akerstrom, Perception of Three-Dimensional Form from Patterns of Optical Texture. *Journal of Experimental Psychology: Human Perception and Performance*, vol. 13, no. 2, 1987, pp. 242-255.

[Turk92] Turk, Greg, Re-Tiling Polygonal Surfaces. In *Computer Graphics (SIGGRAPH 92 Proceedings)*, vol. 26, no. 2, August 1992, pp. 55-64.

[Turk91] Turk, Greg, Generating Textures on Arbitrary Surfaces Using Reaction-Diffusion. In *Computer Graphics (SIGGRAPH 91 Proceedings)*, vol. 25, no. 4, July 1991, pp. 289-298.

[vanDamme94] van Damme, Wim, *Active Vision: exploration of three-dimensional structure*. Ph.D. thesis, University of Utrecht, 1994, ISBN 90-393-0803-0.

[Wejchert92] Wejchert, Jakub, Textures and Equation Visualization. *IEEE Computer Graphics and Applications*, vol. 12, no. 6, November 1992, pp. 10-12.

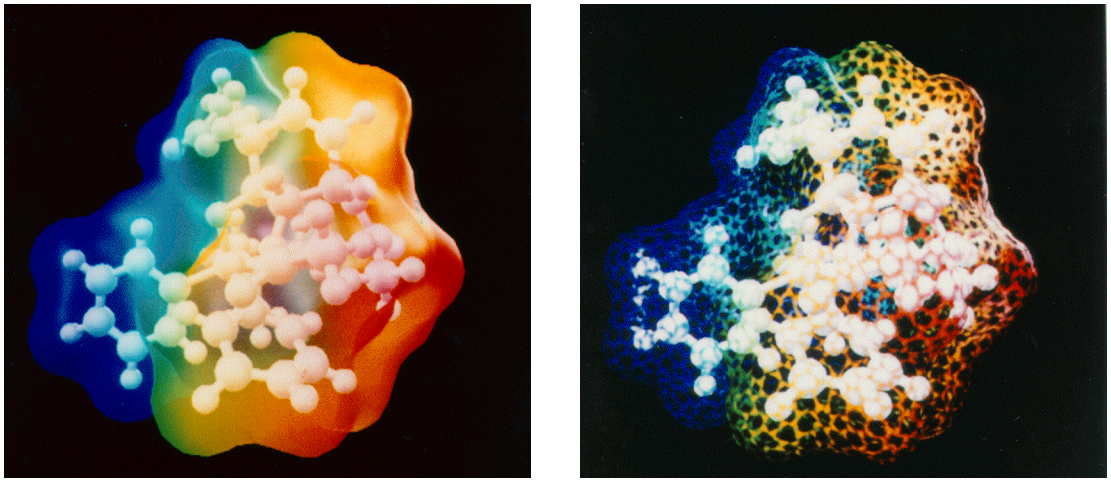


Figure 4. Comparison of a) transparent surface to b) opacity-modulating surface for molecular application. Color represen variable (in this case, simply position in x) evaluated at the solvent-accessible surface.

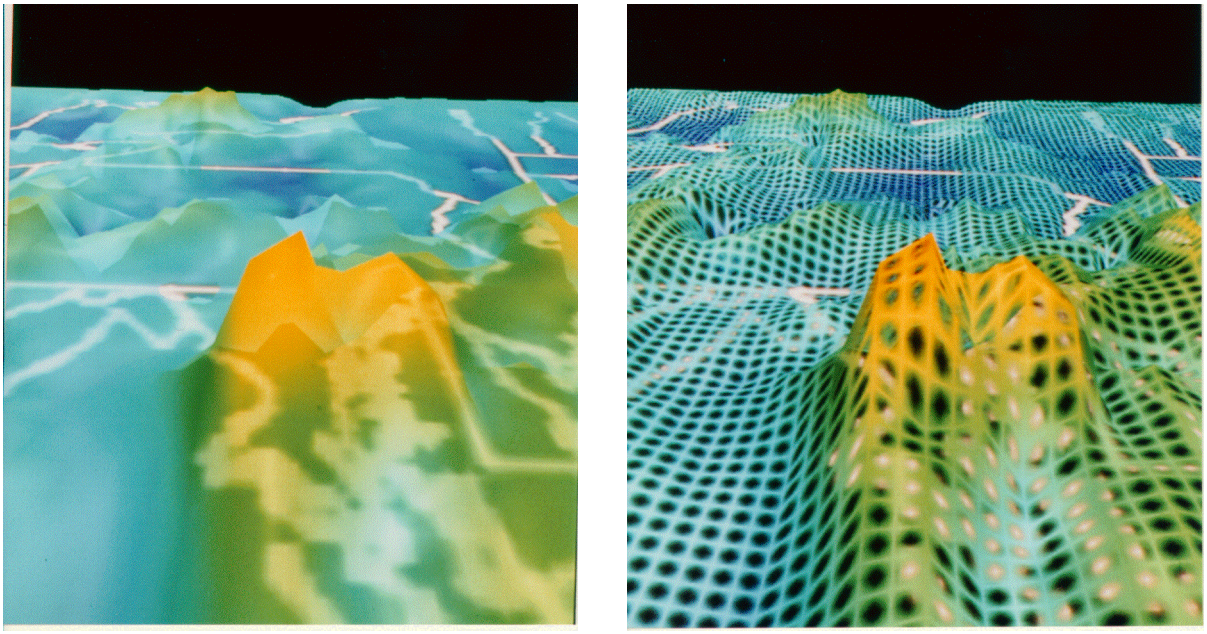


Figure 5. Comparison of a) transparent and b) textured statistical surface. Ozone concentrations over the Eastern seaboar mapped to height and color in both images.

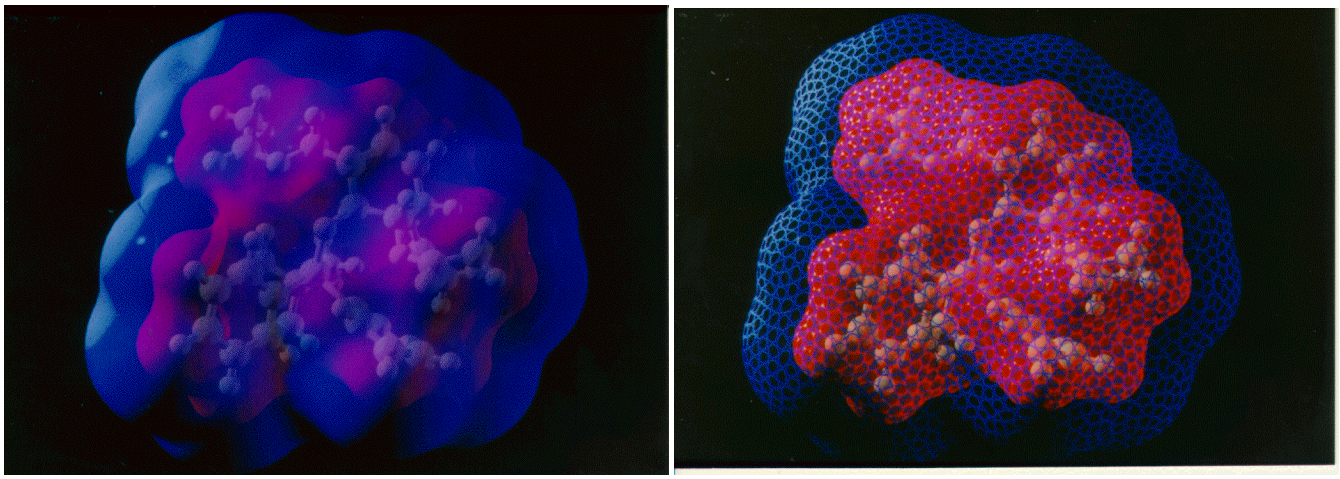


Figure 6. Comparison of a) multiple transparent surfaces to b) multiple opacity-modulating surfaces.

Appears in *Proceedings of IEEE Visualization '00* (October 2000, Salt Lake City, UT), IEEE Computer Society Press, pp. 195-202.

Volume Illustration: Non-Photorealistic Rendering of Volume Models

David Ebert Penny Rheingans
Computer Science and Electrical Engineering
University of Maryland Baltimore County
Baltimore MD 21250
[ebert | rheingan]@cs.umbc.edu

Abstract

Accurately and automatically conveying the structure of a volume model is a problem not fully solved by existing volume rendering approaches. Physics-based volume rendering approaches create images which may match the appearance of translucent materials in nature, but may not embody important structural details. Transfer function approaches allow flexible design of the volume appearance, but generally require substantial hand tuning for each new data set in order to be effective. We introduce the volume illustration approach, combining the familiarity of a physics-based illumination model with the ability to enhance important features using non-photorealistic rendering techniques. Since features to be enhanced are defined on the basis of local volume characteristics rather than volume sample value, the application of volume illustration techniques requires less manual tuning than the design of a good transfer function. Volume illustration provides a flexible unified framework for enhancing structural perception of volume models through the amplification of features and the addition of illumination effects.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – color, shading, and texture; I.3.8 [Computer Graphics]: Applications.
Keywords: Volume rendering, non-photorealistic rendering, illustration, lighting models, shading, visualization.

1 Introduction

For volume models, the key advantage of direct volume rendering over surface rendering approaches is the potential to show the structure of the value distribution throughout the volume, rather than just at selected boundary surfaces of variable value (by isosurface) or coordinate value (by cutting plane). The contribution of each volume sample to the final image is explicitly computed and included. The key challenge of direct volume rendering is to convey that value distribution clearly and accurately. In particular, showing each volume sample with full opacity and clarity is impossible if volume samples in the rear of the volume are not to be completely obscured.

Traditionally, volume rendering has employed one of two approaches. The first attempts a physically accurate simulation of a process such as the illumination and attenuation of light in a gaseous volume or the attenuation of X-rays through tissue [Kajiya84, Drebin88]. This approach produces the most realistic and familiar views of a volume data set, at least for data that has an appropriate physical meaning. The second approach is only loosely based on the physical behavior of light through a volume, using instead an arbitrary transfer function specifying the appearance of a volume sample based on its value and an accumulation process that is not necessarily based on any actual accumulation mechanism [Levoy90]. This approach allows the designer to create a wider range of appearances for the volume in the visualization, but sacrifices the familiarity and ease of interpretation of the more physics-based approach.

We propose a new approach to volume rendering: the augmentation of a physics-based rendering process with non-photorealistic rendering (NPR) techniques [Winkenbach94, Salisbury94] to enhance the expressiveness of the visualization. NPR draws inspiration from such fields as art and technical illustration to develop automatic methods to synthesize images with an illustrated look from geometric surface models. Non-photorealistic rendering research has effectively addressed both the illustration of surface shape and the visualization of 2D data, but has virtually ignored the rendering of volume models. We describe a set of NPR techniques specifically for the visualization of volume data, including both the adaptation of existing NPR techniques to volume rendering and the development of new techniques specifically suited for volume models. We call this approach *volume illustration*.

The volume illustration approach combines the benefits of the two traditional volume rendering approaches in a flexible and parameterized manner. It provides the ease of interpretation resulting from familiar physics-based illumination and accumulation processes with the flexibility of the transfer function approach. In addition, volume illustration provides flexibility beyond that of the traditional transfer function, including the capabilities of local and global distribution analysis, and light and view direction specific effects. Therefore, volume illustration techniques can be used to create visualizations of volume data that are more effective at conveying the structure within the volume than either of the traditional approaches. As the name suggests, volume illustration is intended primarily for illustration or presentation situations, such as figures in textbooks, scientific articles, and educational video.

2 Related Work

Traditional volume rendering spans a spectrum from the accurate to the ad hoc. Kajiya's original work on volume ray tracing for generating images of clouds [Kajiya84] incorporated a physics-based illumination and atmospheric attenuation model. This work in realistic volume rendering techniques has been extended by numerous researchers [Nishita87, Ebert90, Krueger91, Williams92, Max95, Nishita98]. In contrast, traditional volume rendering has relied on the use of transfer functions to produce artificial views of the data to highlight regions of interest [Drebin88]. These transfer functions, however, require in-depth knowledge of the data and need to be adjusted for each data set.

Appears in *Proceedings of IEEE Visualization '00* (October 2000, Salt Lake City, UT), IEEE Computer Society Press, pp. 195-202.

The design of effective transfer functions is still an active research area [Fang98, Kindlmann98, Fujishiro99]. While transfer functions can be effective at bringing out the structure in the value distribution of a volume, they are limited by their dependence on voxel value as the sole transfer function domain.

In contrast, there has been extensive research for illustrating surface shape using non-photorealistic rendering techniques. Adopting a technique found in painting, Gooch et al. developed a tone-based illumination model that determined hue, as well as intensity, from the orientation of a surface element to a light source [Gooch98]. The extraction and rendering of silhouettes and other expressive lines has been addressed by several researchers [Saito90, Salisbury94, Gooch99, Interrante95]. Expressive textures have been applied to surfaces to convey surface shape [Rheingans96, Salisbury97, Interrante97].

A few researchers have applied NPR techniques to the display of data. Laidlaw used concepts from painting to create visualizations of 2D data, using brushstroke-like elements to convey information [Laidlaw98] and a painterly process to compose complex visualizations [Kirby99]. Treavett has developed techniques for pen-and-ink illustrations of surfaces within volumes [Treavett00]. Interrante applied principles from technical illustration to convey depth relationships with halos around foreground features in flow data [Interrante98]. Saito converted 3D scalar fields into a sampled point representation and visualized selected points with a simple primitive, creating an NPR look [Saito94]. With the exceptions of the work of Saito and Interrante, the use of NPR techniques has been confined to surface rendering.

3 Approach

We have developed a collection of volume illustration techniques that adapt and extend NPR techniques to volume objects. Most traditional volume enhancement has relied on functions of the volume sample values (e.g., opacity transfer functions), although some techniques have also used the volume gradient (e.g., [Levoy90]). In contrast, our volume illustration techniques are fully incorporated into the volume rendering process, utilizing viewing information, lighting information, and additional volumetric properties to provide a powerful, easily extensible framework for volumetric enhancement. Comparing Diagram 1, the traditional volume rendering system, and Diagram 2, our volume illustration rendering system, demonstrates the difference in our approach to volume enhancement. By incorporating the enhancement of the volume sample's color, illumination, and opacity into the rendering system, we can implement a wide range of enhancement techniques. The properties that can be incorporated into the volume illustration procedures include the following:

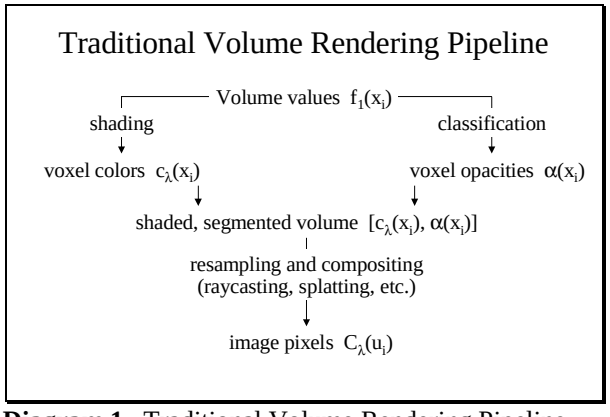


Diagram 1. Traditional Volume Rendering Pipeline.

- Volume sample location and value
- Local volumetric properties, such as gradient and minimal change direction
- View direction
- Light information

The view direction and light information allows global orientation information to be used in enhancing local volumetric features. Combining this rendering information with user selected parameters provides a powerful framework for volumetric enhancement and modification for artistic effects.

Volumetric illustration differs from surface-based NPR in several important ways. In NPR, the surfaces (features) are well defined, whereas with volumes, feature areas within the volume must be determined through analysis of local volumetric properties. The volumetric features vary continuously throughout three-dimensional space and are not as well defined as surface features. Once these volumetric feature volumes are identified, user selected parametric properties can be used to enhance and illustrate them.

We begin with a volume renderer that implements physics-based illumination of gaseous phenomena. The opacity transfer function that we are using is the following simple power function:

$$o_v = (k_{os} v_i)^{k_{oe}}$$

where v_i is the volume sample value and k_{os} is the scalar controlling maximum opacity. Exponent k_{oe} values less than 1 soften volume differences and values greater than 1 increase the contrast within the volume.

Figure 1 shows gaseous illumination of an abdominal CT volume of 256×256×128 voxels. In this image, as in others of this dataset, the scene is illuminated by a single light above the volume and slightly toward the viewer. The structure of tissues and organs is difficult to understand. In Figure 2, a transfer function has been used to assign voxel colors which mimic those found in actual tissue. The volume is illuminated as before. Organization of tissues into organs is clear, but the interiors of structures are still unclear. We chose to base our examples on an atmospheric illumination model, but the same approach can be easily applied to a base renderer using Phong illumination and linear accumulation.

In the following two sections, we describe our current collection of volume illustration techniques. These techniques can be applied in almost arbitrary amounts and combinations, becoming a flexible toolkit for the production of expressive images of volume models. The volume illustration techniques we

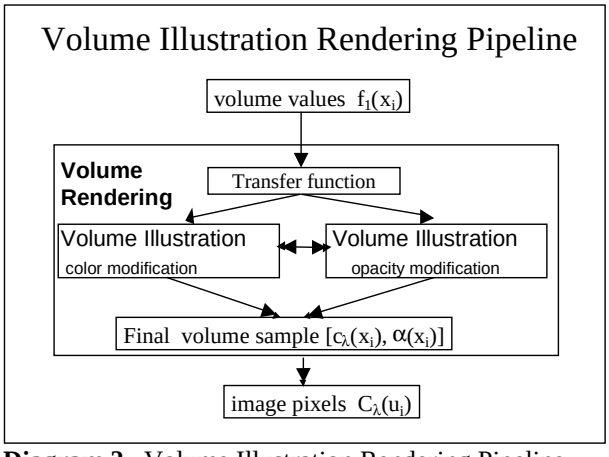


Diagram 2. Volume Illustration Rendering Pipeline.

propose are of two basic types: feature enhancement, and depth and orientation cues.

4 Feature Enhancement

In a surface model, the essential feature is the surface itself. The surface is explicitly and discretely defined by a surface model, making "surfaceness" a boolean quality. Many other features, such as silhouettes or regions of high curvature, are simply interesting parts of the surface. Such features can be identified by analysis of regions of the surface.

In a volume model, there are no such discretely defined features. Volume characteristics and the features that they indicate exist continuously throughout the volume. However, the boundaries between regions are still one feature of interest. The local gradient magnitude at a volume sample can be used to indicate the degree to which the sample is a boundary between disparate regions. The direction of the gradient is analogous to the surface normal. Regions of high gradient are similar to surfaces, but now "surfaceness" is a continuous, volumetric quality, rather than a boolean quality. We have developed several volume illustration techniques for the enhancement of volume features based on volume gradient information.

4.1 Boundary Enhancement

Levoy [Levoy90] introduced gradient-based shading and opacity enhancement to volume rendering. In his approach, the opacity of each voxel was scaled by the voxel's gradient magnitude to emphasize the boundaries between data (e.g., tissue) of different densities and make areas of constant density transparent (e.g., organ interiors). We have adapted this idea to allow the user to selectively enhance the density of each volume sample by a function of the gradient. Assume a volume data set containing a precomputed set of sample points. The value at a location P_i is a scalar given by:

$$v_i = f(P_i) = f(x_i, y_i, z_i)$$

We can also calculate the value gradient $\nabla_f (P_i)$ at that location. In many operations we will want that gradient to be normalized. We use ∇_{fn} to indicate the normalized value gradient.

Before enhancement, voxel values are optionally mapped through a standard transfer function which yields color value c_v and opacity o_v for the voxel. If no transfer function is used, these values can be set to constants for the whole volume. The inclusion of a transfer function allows artistic enhancements to supplement, rather than replace, existing volume rendering mechanisms.

We can define a boundary-enhanced opacity for a voxel by combining a fraction of the voxel's original opacity with an enhancement based on the local boundary strength, as indicated by the voxel gradient magnitude. The gradient-based opacity of the volume sample becomes:

$$o_g = o_v \left(k_{gc} + k_{gs} \left(\left\| \nabla_f \right\| \right)^{k_{ge}} \right)$$

where o_v is original opacity and ∇_f is the value gradient of the volume at the sample under consideration. This equation allows the user to select a range of effects from no gradient enhancement (k_{gc}=1, k_{gs}=0) to full gradient enhancement (k_{gs} >=1) to only showing areas with large gradients (k_{gc}=0), as in traditional volume rendering. The use of the power function with exponent k_{ge} allows the user to adjust the slope of the opacity curve to best highlight the dataset.

Figure 3 shows the effect of boundary enhancement in the medical volume. The edges of the lungs and pulmonary vasculature can be seen much more clearly than before, as well as some of the internal structure of the kidney. Parameter values used in Figure 3 are k_{gc} = 0.7, k_{gs} = 10, k_{ge} = 2.0.

4.2 Oriented Feature Enhancement: Silhouettes, Fading, and Sketch Lines

Surface orientation is an important visual cue that has been successfully conveyed by artists for centuries through numerous techniques, including silhouette lines and orientation-determined saturation effects. Silhouette lines are particularly important in the perception of surface shape, and have been utilized in surface illustration and surface visualization rendering [Salisbury94, Interrante95]. Similarly, silhouette volumes increase the perception of volumetric features.

In order to strengthen the cues provided by silhouette volumes, we increase the opacity of volume samples where the gradient nears perpendicular to the view direction, indicated by a dot product between gradient and view direction which nears zero. The silhouette enhancement is described by:

$$o_s = o_v \left(k_{sc} + k_{ss} \left(1 - abs\left(\nabla_{fn} \cdot V \right) \right)^{k_{se}} \right)$$

where k_{sc} controls the scaling of non-silhouette regions, k_{ss} controls the amount of silhouette enhancement, and k_{se} controls the sharpness of the silhouette curve.

Figure 4 shows the result of both boundary and silhouette enhancement in the medical volume. The fine honeycomb structure of the liver interior is clearly apparent, as well as additional internal structure of the kidneys. Parameter values used in Figure 4 are k_{gc} = 0.8, k_{gs} = 5.0, k_{ge} = 1.0; k_{sc} = 0.9, k_{ss} = 50, k_{se} = 0.25.

Decreasing the opacity of volume features oriented toward the viewer emphasizes feature orientation, and in the extreme cases, can create sketches of the volume, as illustrated in Figure 5. Figure 5 shows a black and white sketch of the medical dataset by using a white sketch color and making non volumetric silhouettes transparent. To get appropriate shadowing of the sketch lines, the shadows are calculated based on the original volume opacity. Using a black silhouette color can also be effective for outlining volume data.

Orientation information can also be used effectively to change feature color. For instance, in medical illustration the portions of anatomical structures oriented toward the viewer are desaturated and structures oriented away from the view are darkened and saturated [Clark99]. We simulate these effects by allowing the volumetric gradient orientation to the viewer to modify the color, saturation, value, and transparency of the given volume sample. The use of the HSV color space allows the system to easily utilize the intuitive color modification techniques of painters and illustrators. Figure 10 shows oriented changes in the saturation and value of the medical volume. In this figure, the color value (V) is decreased as the angle between the gradient and the viewer increases, simulating more traditional illustration techniques of oriented fading.

5 Depth and Orientation Cues

Few of the usual depth cues are present in traditional rendering of translucent volumes. Obscuration cues are largely missing since there are no opaque objects to show a clear depth ordering. Perspective cues from converging lines and texture compression are also lacking, since few volume models contain straight lines

or uniform textures. The dearth of clear depth cues makes understanding spatial relationships of features in the volume difficult. One common approach to this difficulty is the use of hard transfer functions, those with rapidly increasing opacity at particular value ranges of interest. While this may increase depth cues by creating the appearance of surfaces within the volume, it does so by hiding all information in some regions of the volume, sacrificing a key advantage of volume rendering.

Similarly, information about the orientation of features within the volume is also largely missing. Many volume rendering systems use very simple illumination models and often do not include the effect of shadows, particularly volume self-shadowing to improve performance, even though many volume shadowing algorithms have been developed [Ebert90, Kajiya84]. Accurate volumetric shadowing often produces subtle effects which do not provide strong three-dimensional depth cues. As a result, the shape of individual structures within even illuminated volumes is difficult to perceive.

We have developed several techniques for the enhancement of depth and orientation cues in volume models, inspired by shading concepts in art and technical illustration.

5.1 Distance color blending

Intensity depth-cuing is a well known technique for enhancing the perception of depth in a scene [Foley96]. This technique dims the color of objects far from the viewer, creating an effect similar to viewing the scene through haze. We have adapted this technique for volume rendering, dimming volume sample colors as they recede from the viewer. In addition, we have augmented the standard intensity depth-cuing with a subtle color modulation. This color modulation increases the amount of blue in the colors of more distant volume samples, simulating techniques used for centuries by painters, such as aerial perspective [daVinci1506, Beirstadt1881]. This technique exploits the tendency of cool colors (such as blue) to recede visually while warm colors (such as red) advance.

Depth-cued colors start as the voxel color at the front of the volume, decreasing in intensity and moving toward the background color as depth into the volume increases. The progression of depth-cuing need not be linear; we use an exponential function to control the application of depth-cuing. The distance color blending process can be described by:

$$c_d = \left(1 - k_{ds} d_v^{\,k_{de}}\right) c_v + k_{ds} d_v^{\,k_{de}} c_b$$

where k_{ds} controls the size of the color blending effect, k_{de} controls the rate of application of color blending, d_v is the fraction of distance through the volume, and c_b is a defined background color. When c_b is a shade of grey ($c_b = (a, a, a)$ for some value of a), only standard intensity depth-cuing is performed. Using a background color that is a shade of blue ($c_b = (a, b, c)$ for $c > a$, b), introduces a cool shift in distant regions. Other color modulation effects are clearly possible, but make less sense perceptually.

Figure 6 shows the effect of distance color blending. The ribs behind the lungs fade into the distance and the region around the kidneys seems to recede slightly. Color blending parameters used in Figure 6 are $c_b = (0, 0, 0.15)$, $k_{ds} = 1.0$, $k_{se} = 0.5$.

5.2 Feature halos

Illustrators sometimes use null halos around foreground features to reinforce the perception of depth relationships within a scene. The effect is to leave the areas just outside surfaces empty, even if an accurate depiction would show a background object in that place. Interrante [Interrante98] used a similar idea to show depth relationships in 3D flow data using Line Integral Convolution (LIC). She created a second LIC volume with a larger element size, using this second volume to impede the view. Special care was required to keep objects from being obscured by their own halos. The resulting halos achieved the desired effect, but the method depended on having flow data suitable for processing with LIC.

We introduce a more general method for creating halo effects during the illumination process using the local spatial properties of the volume. Halos are created primarily in planes orthogonal to the view vector by making regions just outside features darker and more opaque, obscuring background elements which would otherwise be visible. The strongest halos are created in empty regions just outside (in the plane perpendicular to the view direction) of a strong feature.

The halo effect at a voxel is computed from the distance weighted sum of haloing influences in a specified neighborhood. In order to restrict halos to less interesting regions, summed influences are weighted by the complement of the voxel's gradient. The size of the halo effect is given by:

$$h_i = \left(\sum_{n}^{neighbors} \frac{h_n}{\left\| P_i - P_n \right\|^2} \right) \left(1 - \left\| \nabla_f (P_i) \right\| \right)$$

where h_n is the maximum potential halo contribution of a neighbor. The haloing influence of a neighbor is inversely related to its distance and the tendency of a location to be a halo is inversely related to its gradient magnitude.

The maximum potential halo contribution of each neighbor is proportional to the product of the alignment of the neighbor's gradient with the direction to the voxel under consideration (calculated from the dot product between them) and the degree to which the neighbor's gradient is aligned perpendicular to the view direction (also calculated as a dot product). The halo potential (h_n) is given by:

$$h_n = \left(\nabla_{fn}(P_n) \cdot \left(\frac{(P_i - P_n)}{\left\| P_i - P_n \right\|} \right) \right)^{k_{hpe}} \left(1 - \nabla_{fn}(P_n) \cdot V\right)^{k_{hse}}$$

where k_{hpe} controls how directly the neighbor's gradient must be oriented toward the current location, and k_{hse} controls how tightly halos are kept in the plane orthogonal to the view direction. The most strong halo effects will come from neighbors that are displaced from the volume sample of interest in a direction orthogonal to the view direction and that have a large gradient in the direction of this sample.

Once the size of the halo effect has been determined, parameters control the visual appearance of halo regions. The most common adjustment to the halo region is to decrease the brightness by a scalar times the halo effect and increase the opacity by another scalar times the halo effect. This method produces effects similar to those of Interrante, but can be applied to any type of data or model during the illumination process. Since the halos generated are inherently view dependent, no special processing must be done to keep features from casting a halo on themselves.

Figure 6 shows the effectiveness of adding halos to the medical volume. Structures in the foreground, such as the liver and kidneys, stand out more clearly. Halo parameters used in Figure 6 are $k_{hpe} = 1.0$ and $k_{hse} = 2.0$.

5.3 Tone shading

Another illustrative technique used by painters is to modify the tone of an object based on the orientation of that object relative to the light. This technique can be used to give surfaces facing the light a warm cast while surfaces not facing the light get a cool cast, giving effects suggestive of illumination by a warm light source, such as sunlight. Gooch et al. proposed an illumination model based on this technique [Gooch98], defining a parameterized model for effects from pure tone shading to pure illuminated object color. The parameters define a warm color by combining yellow and the scaled fully illuminated object color. Similarly, a cool color combines blue and the scaled ambient illuminated object color. The final surface color is formed by interpolation between the warm and cool color based on the signed dot product between the surface normal and light vector. The model assumes a single light source, generally located above the scene.

We implemented an illumination model similar to Gooch tone shading for use with volume models. As with Gooch tone shading, the tone contribution is formed by interpolation between the warm and cool colors based on the signed dot product between the volume sample gradient and the light vector. Unlike Gooch tone shading, the illuminated object contribution is calculated using only the positive dot product, becoming zero at orientations orthogonal to the light vector. This more closely matches familiar diffuse illumination models.

The color at a voxel is a weighted sum of the illuminated gaseous color (including any traditional transfer function calculations) and the total tone and directed shading from all directed light sources. The new tone illumination model is given by:

$$c = k_{ta} I_G + \sum_{i}^{N_L} \left(I_t + k_{td} I_o \right)$$

where k_{ta} controls the amount of gaseous illumination (I_G) included, N_L is the number of lights, k_{td} controls the amount of directed illumination included, I_t is the tone contribution to volume sample color, and I_o is the illuminated object color contribution. Although this model allows for multiple light sources, more than a few is likely to result in confusing images, since we are not used to interpreting complex illumination coming from many lights.

The tone contribution from a single light source is interpolated from the warm and cool colors, depending on the angle between the light vector and the sample gradient. It is given by:

$$I_t = \left((1.0 + \nabla_{fn} \cdot L)/2 \right) c_w + \left(1 - (1.0 + \nabla_{fn} \cdot L)/2 \right) c_c$$

where L is the unit vector in the direction of the light and

$c_w = (k_{ty}, k_{ty}, 0)$, $c_c = (0, 0, k_{tb})$

describe the warm and cool tone colors. Samples oriented toward the light become more like the warm color while samples oriented away from the light become more like the cool color.

The directed illumination component is related to the angle between the voxel gradient and the light direction, for angles up to 90 degrees. It is given by:

$$I_o = \begin{cases} k_{td} I_i (\nabla_{fn} \cdot L) : \nabla_{fn} \cdot L > 0 \\ 0 : \nabla_{fn} \cdot L \le 0 \end{cases}$$

where k_{td} controls how much directed illumination is added.

Figure 7 shows modified tone shading applied to the uncolored medical volume. The small structure of the liver shows clearly, as does the larger structures of the kidney. The bulges of intestine at the lower right are much more clearly rounded 3D shapes than with just boundary and silhouette enhancement (Figure 4). Figure 8 shows tone shading applied together with colors from a transfer function. The tone effects are subtler, but still improve shape perception. The basic tissue colors are preserved, but the banded structure of the aorta is more apparent than in a simple illuminated and color-mapped image (Figure 2). Tone shading parameters used in Figures 7 and 8 are $k_{ty} = 0.3$, $k_{tb} = 0.3$, $k_{ta} = 1.0$, $k_{td} = 0.6$.

6 Application Examples

We have also applied the techniques in the previous sections to several other scientific data sets. Figures 10 and 11 are volume rendered images from a 256x256x64 MRI dataset of a tomato from Lawrence Berkeley National Laboratories. Figure 10 is a normal gas-based volume rendering of the tomato where a few of the internal structures are visible. Figure 11 has our volume illustration gradient and silhouette enhancements applied, resulting in a much more revealing image showing the internal structures within the tomato. Parameters used in Figure 11 are $k_{gc} = 0.5$, $k_{gs} = 2.5$, $k_{ge} = 3.0$; $k_{sc} = 0.4$, $k_{ss} = 500$, $k_{se} = 0.3$.

Figure 12 shows a 512x512x128 element flow data set from the time series simulation of unsteady flow emanating from a 2D rectangular slot jet. The 2D jet source is located at the left of the image and the flow is to the right. Flow researchers notice that both Figures 12 and 13 resemble Schlieren photographs that are traditionally used to analyze flow. Figure 13 shows the effectiveness of boundary enhancement, silhouette enhancement, and tone shading on this data set. The overall flow structure, vortex shedding, and helical structure are much easier to perceive in Figure 13 than in Figure 12.

Figures 14 and 15 are volume renderings of a 64x64x64 high-potential iron protein data set. Figure 14 is a traditional gas-based rendering of the data. Figure 15 has our tone shading volume illustration techniques applied, with parameters $k_{ty} = 0.15$, $k_{tb} = 0.15$, $k_{ta} = 1.0$, $k_{td} = 0.6$. The relationship of structure features and the three-dimensional location of the features is much clearer with the tone-based shading enhancements applied.

7 Conclusions

We have introduced the concept of volume illustration, combining the strengths of direct volume rendering with the expressive power of non-photorealistic rendering techniques. Volume illustration provides a powerful unified framework for producing a wide range of illustration styles using local and global properties of the volume model to control opacity accumulation and illumination. Volume illustration techniques enhance the perception of structure, shape, orientation, and depth relationships in a volume model. Comparing standard volume rendering (Figures 2, 10, 12, 14) with volume illustration images (Figures 3, 4, 5, 6, 7, 8, 9, 11, 13, 15) clearly shows the power of employing volumetric illustration techniques to enhance 3D depth perception and volumetric feature understanding.

8 Future Directions

We plan on extending our collection of NPR techniques and exploring suitability of these volume illustration techniques for data exploration and diagnosis.

9 Acknowledgements

We would like to thank researchers at the Mississippi State University NSF Computational Field Simulation Engineering

Appears in *Proceedings of IEEE Visualization '00* (October 2000, Salt Lake City, UT), IEEE Computer Society Press, pp. 195-202.

Research Center and the Armed Forces Institute of Pathology for help in evaluating the effectiveness of these technique and guiding our research. We would also like to thank Dr. Elliott Fishman of Johns Hopkins Medical Institutions for the abdominal CT dataset. The iron protein dataset came from the vtk website (www.kitware.com/vtk.html). Christopher Morris generated some to the pictures included in this paper. This work supported in part by the National Science Foundation under Grants ACIR 9996043 and ACIR 9978032.

References

[Bierstadt1881] Albert Bierstadt. "Near Salt Lake City, Utah," Museum of Art, Brigham Young University, 1881.

[Clark99] John O.E. Clark. *A Visual Guide to the Human Body, Barnes and Noble Books, 1999.*

[daVinci1506] Leonardo daVinci, "The Virgin of the Rocks," National Gallery, London, 1503-1506.

[Drebin88] Robert A. Drebin and Loren Carpenter and Pat Hanrahan. Volume Rendering, *Computer Graphics (Proceedings of SIGGRAPH 88)*, 22(4), pp. 65-74 (August 1988, Atlanta, Georgia). Edited by John Dill.

[Ebert90] David S. Ebert and Richard E. Parent. Rendering and Animation of Gaseous Phenomena by Combining Fast Volume and Scanline A-buffer Techniques, *Computer Graphics (Proceedings of SIGGRAPH 90)*, 24 (4), pp. 357-366 (August 1990, Dallas, Texas). Edited by Forest Baskett. ISBN 0-201-50933-4.

[Fang98] Shiaofen Fang and Tom Biddlecome and Mihran Tuceryan. Image-Based Transfer Function Design for Data Exploration in Volume Visualization, *IEEE Visualization '98*, pp. 319-326 (October 1998). IEEE. Edited by David Ebert and Hans Hagen and Holly Rushmeier. ISBN 0-8186-9176-X.

[Foley96] James Foley, Andries van Dam, Steven Feiner, and John Hughes, *Computer Graphics: Principles and Practice, Second Edition in C*, Addison Wesley 1996.

[Fujishiro99] Issei Fujishiro and Taeko Azuma and Yuriko Takeshima. Automating Transfer Function Design for Comprehensible Volume Rendering Based on 3D Field Topology Analysis, *IEEE Visualization '99*, pp. 467-470 (October 1999, San Francisco, California). IEEE. Edited by David Ebert and Markus Gross and Bernd Hamann. ISBN 0-7803-5897-X.

[Gooch98] Amy Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A Non-photorealistic Lighting Model for Automatic Technical Illustration. In *Proceedings of SIGGRAPH '98* (Orlando, FL, July 1998), Computer Graphics Proceedings, Annual Conference Series, pp. 447-452, ACM SIGGRAPH, ACM Press, July 1998.

[Gooch99] Bruce Gooch and Peter-Pike J. Sloan and Amy Gooch and Peter Shirley and Rich Riesenfeld. Interactive Technical Illustration, *1999 ACM Symposium on Interactive 3D Graphics*, pp. 31-38 (April 1999). ACM SIGGRAPH. Edited by Jessica Hodgins and James D. Foley. ISBN 1-58113-082-1 .

[Interrante95] Victoria Interrante, Henry Fuchs, and Stephen Pizer. Enhancing Transparent Skin Surfaces with Ridge and Valley Lines, IEEE Visualization '95, pp. 52-59 (October 1995, Atlanta GA). IEEE. Edited by Gregory Nielson and Deborah Silver. ISBN 0-8186-7187-4.

[Interrante97] Victoria Interrante and Henry Fuchs and Stephen M. Pizer. Conveying the 3D Shape of Smoothly Curving Transparent Surfaces via Texture, *IEEE Transactions on Visualization and Computer Graphics*, 3(2), (April - June 1997). ISSN 1077-2626.

[Interrante98] Victoria Interrante and Chester Grosch. Visualizing 3D Flow, *IEEE Computer Graphics & Applications*, 18(4), pp. 49-53 (July - August 1998). ISSN 0272-1716.

[Kajiya84] James T. Kajiya and Brian P. Von Herzen. Ray Tracing Volume Densities, *Computer Graphics (Proceedings of SIGGRAPH 84)*, 18(3), pp. 165-174 (July 1984, Minneapolis, Minnesota). Edited by Hank Christiansen.

[Kindlmann98] Gordon Kindlmann and James Durkin. Semi-Automatic Generation of Transfer Functions for Direct Volume Rendering, In *Proceedings of 1998 IEEE Symposium on Volume Visualization*, pp. 79-86.

[Kirby99] R.M. Kirby, H. Marmanis, and D.H. Laidlaw. Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting, *IEEE Visualization '99*, pp. 333-340 (October 1999, San Francisco, California). IEEE. Edited by David Ebert and Markus Gross and Bernd Hamann. ISBN 0-7803-5897-X.

[Krueger91] Wolfgang Krueger, The Application of transport theory to the visualization of 3D scalar fields, *Computers in Physics*, pp. 397-406, July 1991.

[Laidlaw98] David H. Laidlaw and Eric T. Ahrens and David Kremers and Matthew J. Avalos and Russell E. Jacobs and Carol Readhead. Visualizing Diffusion Tensor Images of the Mouse Spinal Cord, *IEEE Visualization '98*, pp. 127-134 (October 1998). IEEE. Edited by David Ebert and Hans Hagen and Holly Rushmeier. ISBN 0-8186-9176-X.

[Levoy90] Marc Levoy. Efficient Ray Tracing of Volume Data, *ACM Transactions on Graphics*, 9 (3), pp. 245-261 (July 1990). ISSN 0730-0301.

[Max95] Nelson Max. Optical models for direct volume rendering, *IEEE Transactions on Visualization and Computer Graphics*, 1 (2), pp. 99-108 (June 1995). ISSN 1077-2626.

[Nishita87] Tomoyuki Nishita and Yasuhiro Miyawaki and Eihachiro Nakamae. A Shading Model for Atmospheric Scattering Considering Luminous Intensity Distribution of Light Sources, *Computer Graphics (Proceedings of SIGGRAPH 87)*, 21 (4), pp. 303-310 (July 1987, Anaheim, California). Edited by Maureen C. Stone.

[Nishita98] Tomoyuki Nishita. Light Scattering Models for the Realistic Rendering of Natural Scenes, *Eurographics Rendering Workshop 1998*, pp. 1-10 (June 1998, Vienna, Austria). Eurographics. Edited by George Drettakis and Nelson Max. ISBN3-211-83213-0.

[Rheingans96] Penny Rheingans. Opacity-modulating Triangular Textures for Irregular Surfaces, Proceedings of IEEE Visualization '96, pp. 219-225 (October 1996, San Francisco CA). IEEE. Edited by Roni Yagel and Gregory Nielson. ISBN 0-89791-864-9.

[Saito90] Takafumi Saito and Tokiichiro Takahashi. Comprehensible Rendering of 3-D Shapes, *Computer Graphics (Proceedings of SIGGRAPH 90)*, 24 (4), pp. 197-206 (August 1990, Dallas, Texas).

[Saito94] Takafumi Saito. Real-time Previewing for Volume Visualization. In *Proceedings of 1994 IEEE Symposium on Volume Visualization*, pp. 99-106.

[Salisbury94] Michael P. Salisbury and Sean E. Anderson and Ronen Barzel and David H. Salesin. Interactive Pen-And-Ink Illustration, *Proceedings of SIGGRAPH 94, Computer Graphics Proceedings, Annual Conference Series*, pp. 101-108 (July 1994, Orlando, Florida). ACM Press. Edited by Andrew Glassner. ISBN 0-89791-667-0.

Appears in *Proceedings of IEEE Visualization '00* (October 2000, Salt Lake City, UT), IEEE Computer Society Press, pp. 195-202.

[Salisbury97] Michael P. Salisbury and Michael T. Wong and John F. Hughes and David H. Salesin. Orientable Textures for Image-Based Pen-and-Ink Illustration, *Proceedings of SIGGRAPH 97*, Computer Graphics Proceedings, Annual Conference Series, pp. 401-406 (August 1997, Los Angeles, California). Addison Wesley. Edited by Turner Whitted. ISBN 0-89791-896-7.

[Treavett00] S.M.F. Treavett and M. Chen. Pen-and-Ink Rendering in Volume Visualisation, *Proceedings of IEEE Visualization 20000*, October 2000, ACM SIGGRAPH Press.

[Williams92] Peter L. Williams and Nelson Max. A Volume Density Optical Model, *1992 Workshop on Volume Visualization*, pp. 61-68 (1992). ACM.

[Winkenbach94] Georges Winkenbach and David H. Salesin. Computer-Generated Pen-And-Ink Illustration, *Proceedings of SIGGRAPH 94, Computer Graphics Proceedings, Annual Conference Series*, pp. 91-100 (July 1994, Orlando, Florida). ACM Press. Edited by Andrew Glassner. ISBN 0-89791-667-0.

Figure 1. Gaseous illumination of medical CT volume. Voxels are a constant color.

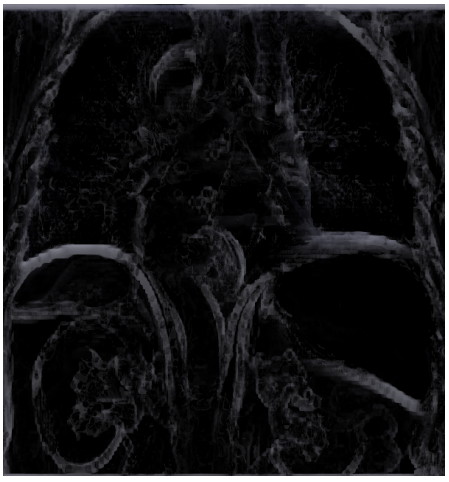


Figure 5. Volumetric sketch lines on CT volume. Lines are all white.

Figure 12. Atmospheric volume rendering of square jet. No illustration enhancements.

Appears in *Proceedings of IEEE Visualization '00* (October 2000, Salt Lake City, UT), IEEE Computer Society Press, pp. 195-202.

Figure 2. Gaseous illumination of color-mapped CT volume.

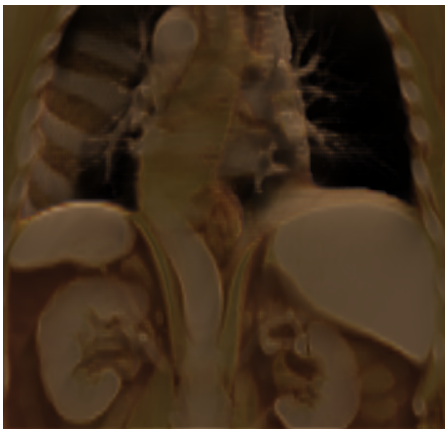


Figure 3. Color-mapped gaseous illumination with boundary enhancement.

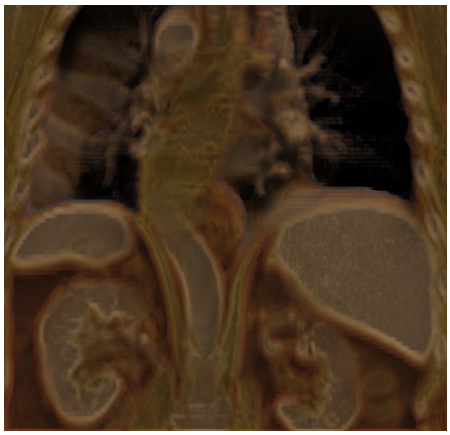


Figure 4. Silhouette and boundary enhancement of CT volume.

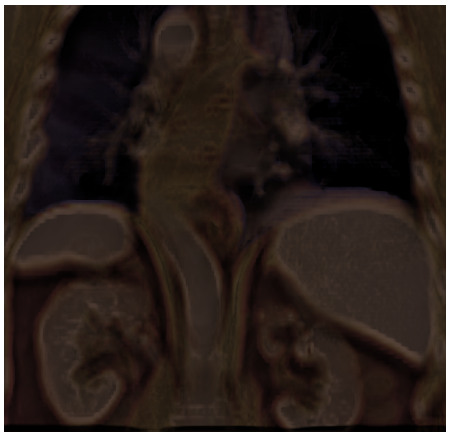


Figure 6. Distance color blending and halos around features of CT volume.

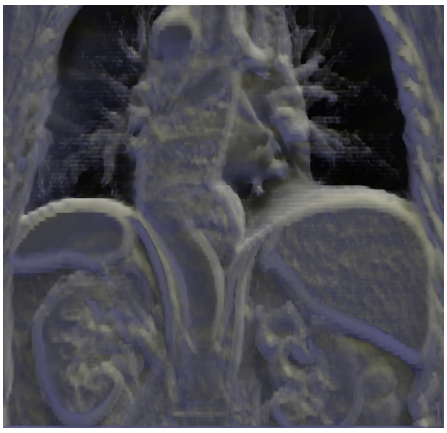


Figure 7. Tone shading in CT volume. Surfaces toward light receive warm color.

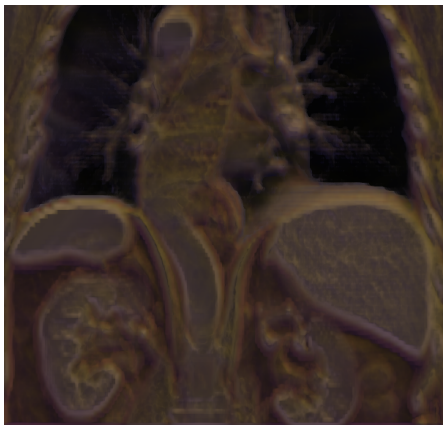


Figure 8. Tone shading in colored volume. Surfaces toward light receive warm color.

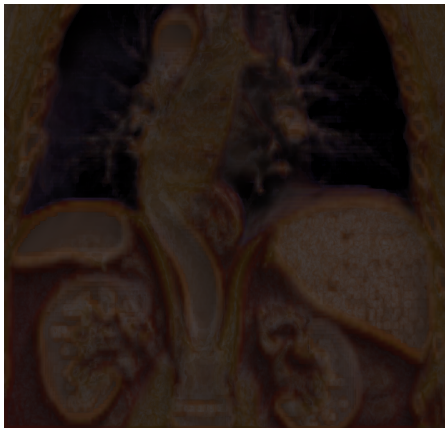


Figure 9. Orientation fading. Surfaces toward viewer are desaturated.

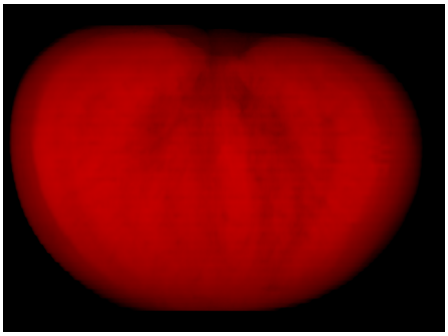


Figure 10. Standard atmospheric volume rendering of tomato.

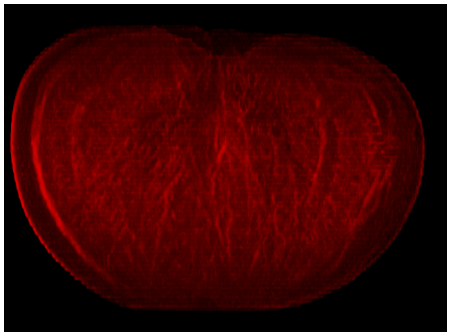


Figure 11. Boundary and silhouette enhanced tomato.

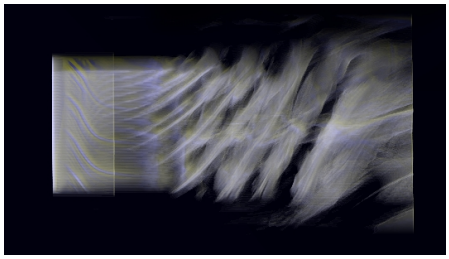


Figure 13. Square jet with boundary and silhouette enhancement, and tone shading.

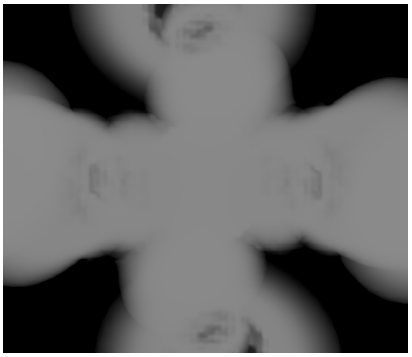


Figure 14. Atmospheric rendering of iron protein.

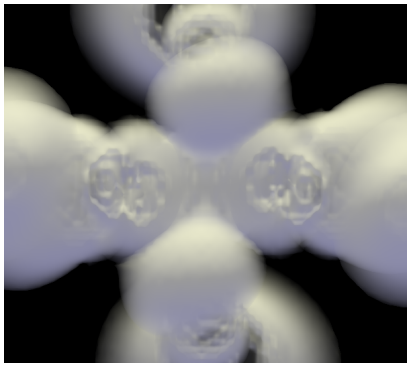


Figure 15. Tone shaded iron protein.

Formalizing Artistic Techniques and Scientific Visualization for Painted Renditions of Complex Information Spaces

Christopher G. Healey
Department of Computer Science, North Carolina State University
1010 Main Campus Drive, Raleigh, NC, 27695-7534
healey@csc.ncsu.edu

Abstract

This paper describes a new method for *visualizing* complex information spaces as painted images. Scientific visualization converts data into pictures that allow viewers to "see" trends, relationships, and patterns. We introduce a formal definition of the correspondence between traditional visualization techniques and painterly styles from the Impressionist art movement. This correspondence allows us to apply perceptual guidelines from visualization to control the presentation of information in a computer-generated painting. The result is an image that is visually engaging, but that also allows viewers to rapidly and accurately explore and analyze the underlying data values. We conclude by applying our technique to a collection of environmental and weather readings, to demonstrate its viability in a practical, real-world visualization environment.

1 Introduction

This paper describes a formal method for constructing visual representations of complex information spaces that support rapid and accurate exploration and analysis. Our technique falls within the domain of *scientific visualization*, the conversion of collections of strings and numbers (or datasets, as they are often called) into images that allow viewers to *explore*, *analyze*, *validate*, and *discover* within their data. We focused on two important issues [Smith and Van Rosendale, 1998] during our investigations:

1. *Multidimensional displays:* Our technique must support the visualization of multiple overlapping data fields together in the same display. This is much more difficult than representing a single data field in isolation. Designing techniques to effectively represent multidimensional datasets is an open area of research in visualization.
2. *Aesthetic displays:* We also seek to create images that are visually engaging. We believe this will motivate viewers to study a visualization in more detail. It will draw viewers into an image, and can be used to emphasize areas of importance in a dataset.

We address these goals by: (1) applying results from human perception to create images that harness the strengths of our low-level visual system, and (2) using artistic techniques from the Impressionist movement to produce *painterly renditions* that are both beautiful and engaging. From an AI perspective, the contribution of this work is the identification of a close relationship between specific painterly techniques and the performance properties of human perception; our formalization lays the groundwork for the generation of scientific visualizations that are effective and aesthetically pleasing.

Our technique converts a collection of data values into a painterly image as follows. First, one or more computer generated "brush strokes" are attached to each data element in the collection. A brush stroke has style properties (*e.g.*, color, length, or direction) that we can vary to modify its visual appearance. Data values in the data element are used to select specific states for the different properties. The result is a stroke that represents its corresponding data element. Rendering all of the strokes for every data element produces a painterly image whose stroke properties visualize the underlying dataset.

The remainder of this paper describes in detail how each step in this process is managed and controlled. We begin by defining formalisms for: (1) a multidimensional dataset and its visualization, and (2) the brush strokes that make up a painterly image. We next present a set of perceptual rules on the use of color and texture in visualization that we extend via our formalisms to the painterly domain. These rules ensure the images we produce represent a dataset in a perceptually salient manner. Finally, we discuss how our techniques were used to visualize a real-world collection of environmental and weather readings for the continental United States. We conclude with a summary and a short description of future work.

2 Formalisms

We began our investigation by identifying methods for building painterly images that we can use to represent multidimensional datasets. A key insight is that many painterly styles correspond closely to perceptual features that are detected by the human visual system. In some sense this is not surprising. Artistic masters understood intuitively which properties of a painting would capture a viewer's gaze, and their styles naturally focused on harnessing these features. Moreover, certain movements used scientific studies of the visual system

to help them understand how viewers would perceive their work. The overlap of artistic styles and perception offers a important starting point: the body of knowledge on the use of perception during visualization will help us to predict how corresponding painterly styles might perform in the same environment.

In order to make use of this advantage, we define a relationship between traditional visualization techniques and painted images. This is done by constructing a correspondence between formal specifications of the two environments. The correspondence can then be used to extend our perceptual guidelines to a painterly domain.

2.1 Multidimensional Visualization

A simple formalization of a multidimensional visualization consists of two parts: a description of the dataset, and a definition of the mapping function used to convert it into an image. A multidimensional dataset $D = \{e_1, \ldots, e_n\}$ contains n samples points or data elements e_i. D represents two or more data attributes $A = \{A_1, \ldots, A_m\}$, $m > 1$; data elements encode values for each attribute, that is, $e_i = \{a_{i,1}, \ldots, a_{i,m}\}, a_{i,j} \in A_j$.

Visualization begins with the construction of a data-feature mapping $M(V, \phi)$ that converts the raw data into images that are presented to the viewer. $V = \{V_1, \ldots, V_m\}$ identifies a visual feature V_j to use to display data attribute A_j. $\phi_j : A_j \rightarrow V_j$ maps the domain of A_j to the range of displayable values in V_j. Based on these definitions, visualization is the selection of M and a viewer's interpretation of the images produced by M. An effective visualization chooses M to support the exploration and analysis tasks the viewer wants to perform.

3 Painterly Styles

Our investigation of painterly styles is directed by two separate criteria. First, we restrict our search to a particular movement in art known as Impressionism. Second, we attempt to pair each style with a corresponding visual feature that has proven to be effective in a perceptual visualization environment. There are no technical reasons for why Impressionism was chosen over any other movement. In fact, we expect the basic theories behind our technique will extend to other types of artistic presentation. For our initial work, however, we felt it was important to narrow our focus to a set of fundamental goals in the context of a single type of painting style.

The term "Impressionism" was attached to a small group of French artists (initially including Monet, Degas, Manet, Renoir, and Pissarro, and later Cézanne, Sisley, and Van Gogh, among others) who broke from the traditional schools of the time to approach painting from a new perspective. The Impressionist technique was based on a number of underlying principles (see also [Schapiro, 1997]):

1. *Object and environment interpenetrate.* Outlines of objects are softened or obscured (*e.g.*, Monet's water lilies); objects are bathed and interact with light; shadows are colored and movement is represented as unfinished outlines.
2. *Color acquires independence.* There is no constant hue for an object, atmospheric conditions and light moderate color across its surface; objects may be reduced to swatches of color.
3. *Show a small section of nature.* The artist is not placed in a privileged position relative to nature; the world is shown as a series of close-up details.
4. *Minimize perspective.* Perspective is shortened and distance reduced to turn 3D space into a 2D image.
5. *Solicit a viewer's optics.* Study the retinal system; divide tones as separate brush strokes to vitalize color rather than graying with overlapping strokes; harness simultaneous contrast; use models from color scientists like Chevreul [Chevreul, 1967] or Rood [Rood, 1879].

Although these general characteristics are perhaps less precise than we might prefer, we can still draw a number of important conclusions. Properties of hue, luminance, and lighting were explicitly controlled and even studied in a scientific fashion by some of the Impressionists. Rather than using an "object-based" representation, the artists appear to be more concerned with subdividing a painting based on the interactions of light with color and other surface properties. Additional painterly styles can be identified by studying the paintings themselves. These styles often varied dramatically between individual artists, acting to define their unique painting techniques. Examples include:

- *path:* the path or direction a brush stroke follows; Van Gogh made extensive use of curved paths to define boundaries and shape in his paintings; other artists favored simpler, straighter strokes,
- *length:* the length of individual strokes on the canvas, often used to differentiate between contextually different parts of a painting,
- *density:* the number of strokes laid down in a fixed area of canvas,
- *coarseness:* the coarseness of the brush used to apply a stroke; a coarser brush causes visible bristle lines and surface roughness, and
- *weight:* the amount of paint applied during each stroke; heavy strokes highlight brush coarseness and produce ridges of paint that cause underhanging shadows when lit from the proper direction.

In this context, a painting P can be seen as a collection of n brush strokes $P = \{b_1, \ldots, b_n\}$, with each stroke made up of p style properties S_j, that is, $b_i = \{s_{i,1}, \ldots, s_{i,j}\}$, $s_{i,j} \in S_j$.

Although it would be tedious (and perhaps uninformative) to characterize a real painting in this manner, these definitions provide an effective way to relate the visualization process to a painted image. First, we can match many of the painterly styles to visual features used during visualization. For example, color and lighting in Impressionism has a direct correspondence to the use of hue and luminance in perceptual visualization. Other styles (*e.g.*, path, density, and length) have similar partners in perception (*e.g.*, orientation, contrast, and size). Second, data elements e_i in a dataset are analogous to

brush strokes b_i in a painting. Attribute values $a_{i,j}$ in element e_i could therefore be used to select specific $s_{i,j}$ for each style in b_i.

Consider a data-feature mapping $M(V, \phi)$ in this context. The visual features $V_j \in V$ can be converted to their corresponding painterly styles S_j. M now describes how to convert a data element e_i into painted brush stroke b_i whose visual appearance represents the attribute values $a_{i,j}$ embedded in e_i. The close correspondence $V_j \leftrightarrow S_j$ between perceptual features and many of the painterly styles we hope to apply is particularly advantageous. Since numerous controlled experiments on the use of perceptual features have already been conducted, we have a large body of evidence to use to predict how we expect painterly styles to react in a multidimensional visualization environment.

4 Perceptual Characteristics

Past research has studied methods for applying rules of perception during visualization [Bergman *et al.*, 1995; Healey and Enns, 1999; Rheingans and Tebbs, 1990]. The cognitive abilities of the low-level human visual system have been studied extensively in the area of human psychophysics. One interesting result is the identification of a limited set of visual features that are detected rapidly, accurately, and relatively effortlessly by a human viewer [Triesman, 1985; Wolfe, 1994]. These features are similar to the ones we display during multidimensional visualization (*e.g.*, hue, luminance, orientation, size, and motion). When combined properly, they can be used to perform exploratory analysis tasks like searching for data elements with a particular attribute value, identifying boundaries between groups of elements with similar values, tracking elements as they move in time and space, and estimating the number of elements with common values. The ability to harness the low-level visual system during visualization through the use of these features is especially attractive, since:

- analysis is rapid and accurate, often requiring no more than 200 milliseconds,
- task completion time is constant and independent of the number of elements in a display, and
- different visual features can interact with one another to mask information; psychophysical experiments allow us to identify and avoid these interference patterns.

A data-feature mapping that builds on a perceptual foundation can support high-level exploration and analysis of large amounts of data in a relatively short period of time. Our recent work focuses on the combined use of fundamental properties of color and texture to encode multiple attributes in a single display. We draw on three specific areas of research in perception and visualization to guide the construction of our brush strokes: color selection, texture selection, and feature hierarchies that can cause visual interference and masking.

4.1 Color Selection

Color is a common feature used in many visualization designs. Some techniques attempt to measure and control the color difference viewers perceive between pairs of colors. This allows:

- *perceptual balance:* a unit step anywhere along the color scale produces a perceptually uniform difference in color,
- *distinguishability:* within a discrete collection of colors, every color is equally distinguishable from all the others (*i.e.*, no color is "easier" or "harder" to identify), and
- *flexibility:* colors can be selected from any part of color space.

Standard color models like CIE LUV or CIE Lab use Euclidean distance to approximate perceived color difference. More complex techniques extend this basic idea. For example, Rheingans and Tebbs [Rheingans and Tebbs, 1990] plotted a path through a color model, a allowed a viewer to vary how colors are selected along the path. Ware constructed color scales that spiral up around the luminance axis [Ware, 1988]; such a scale maintains a uniform simultaneous contrast error along its length. Healey and Enns [Healey and Enns, 1999] showed that color distance, linear separation, and color category must all be controlled to select discrete collections of equally distinguishable colors.

Our color selection technique combines different aspects of each of these methods. A single loop spiraling up around the luminance axis is plotted in the region of CIE LUV space that contains our monitor's color gamut. The path is subdivided into r named color regions (*e.g.*, a blue region, a green region, and so on). n colors are then selected by choosing $\frac{n}{r}$ colors uniformly spaced along each of the r color regions. The result is a set of colors selected from a perceptually balanced color model, each with a roughly constant simultaneous contrast error, and chosen such that color distance and linear separation are constant within each named color region.

4.2 Texture Selection

Although texture is often viewed as a single visual feature, it can be decomposed into fundamental perceptual dimensions. Research in computer vision has used properties like regularity, directionality, and contrast to perform automatic texture segmentation and classification. Results from psychophysics have shown that many of these properties can also be detected by the low-level visual system.

One promising approach in visualization has been to use the perceptual dimensions of a texture pattern to represent multiple data attributes. Individual values in a data element control a corresponding texture dimension, producing a texture pattern that changes its visual appearance based on the underlying dataset. Grinstein et al. [Grinstein *et al.*, 1989] built "stick-man" icons to represent high dimensional data elements; the orientation of each limb encodes a value for one particular attribute. Ware and and Knight [Ware and Knight, 1995] displayed Gabor filters that modified their orientation, size, and contrast based on the values of three independent data attributes. Healey and Enns [Healey and Enns, 1999] constructed perceptual texture elements (or pexels) that varied in height, density, and regularity; their results showed that both height and density were perceptually salient, but regularity was not. More recent work [Weigle *et al.*, 2000] found

that an orientation difference of 15° is sufficient to rapidly distinguish visual elements.

4.3 Feature Hierarchy

A third factor that must be considered is visual interference, that is, a situation where one visual feature masks another. Although the need to rank each brush stroke style's perceptual strength is not necessary in a painting, this information is critical for effective visualization design. The most important data attributes (as defined by the viewer) should be displayed using the most salient features. Secondary data should never be visualized in a way that masks the information a viewer is most interested in seeing.

Perceptual features are ordered in a hierarchy by the low-level visual system. Results reported in both the psychophysical and visualization literature have confirmed a luminance–hue–texture interference pattern. Variations in luminance can slow a viewer's ability to identify the presence of individual hues or the spatial patterns they form [Callaghan, 1990]. The interference is asymmetric: random variations in hue have no effect on a viewer's ability to see luminance patterns. A similar asymmetric hue on texture interference has also been shown to exist [Healey and Enns, 1999; Triesman, 1985]; random variations in hue interfere with the identification of texture patterns, but not vice-versa. These results suggest that luminance, then hue, then various texture properties should be used to display attributes in order of importance.

5 Painterly Visualization

Based on the perception guidelines discussed above, and on our formal correspondence between visualization techniques and painterly images, we decided to build a system that varied brush stroke color, size, spatial density, and orientation to encode up to four independent data attributes (in addition to the two spatial values used to position each stroke). The presence of feature hierarchies suggest color should be used to represent the most important attribute, followed by the texture properties. The results of [Healey and Enns, 1999] further refine this to applying color, size, density, and orientation in order of attribute importance.

The brush strokes in our current prototype are constructed using a simple texture mapping scheme. A real painted stroke was digitized and converted into a texture map. This texture map is attached to a polygon to produce a reasonable approximation of a generic brush stroke. The stroke's position, color, size, and orientation are controlled by modifying the texture map and transforming the polygon. Density is varied by changing the number of strokes rendered in a unit area of screen space. Fig. 1 shows an example of brush strokes with four different greyscales, sizes, densities, and orientations (greyscale is used only for the printed figures; on-screen images are displayed in full color).

5.1 Practical Applications

One of the application testbeds for our visualization techniques is a dataset of monthly environmental and weather conditions collected and recorded by the Intergovernmental Panel on Climate Change. This dataset contains mean

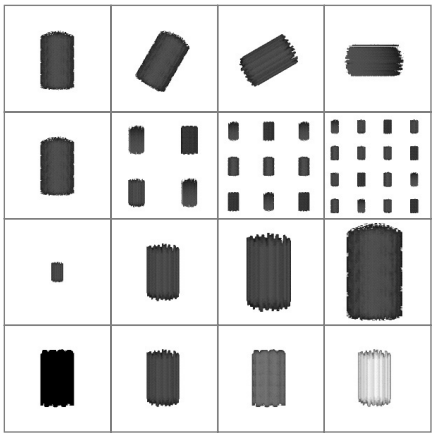


Figure 1: Examples of texture mapped brush strokes with different orientations (top row), densities (second row), sizes (third row), and greyscales (fourth row)

monthly surface climate readings in $\frac{1}{2}^\circ$ latitude and longitude steps for the years 1961 to 1990 (*e.g.*, readings for January averaged over the years 1961-1990, readings for February averaged over the years 1961-1990, and so on). We chose to visualize temperature, precipitation, pressure, and windspeed. Based on this order of importance, we built a data-feature mapping M that assigns brush stroke greyscale (or color for on-screen images), size (or coverage), density, and orientation, respectively, to our four attributes. Temperature is represented by greyscales selected uniformly from a perceptually balanced luminance path. This path runs from dark (for cold temperatures) to bright (for hot temperatures). Precipitation is represented size (*i.e.*, the amount of an element's spatial region its brush stroke covers). Sizes range exponentially from very small coverage (for little or no precipitation) to full coverage (for heavy precipitation). Windspeed is represented by orientations ranging from 0° or upright (for weak winds) to 90° or flat (for strong winds). Finally, pressure is represented by four increasingly dense arrays of brush strokes: a single stroke, a 2×2 array of strokes, a 3×3 array, and a 4×4 array; continuous pressure values are discritized into four uniform ranges, then mapped to the appropriate density (sparse for low pressure, dense for high pressure).

Fig. 2 shows a visualization of data for February in the eastern half of the continental United States. Although unlikely to be mistaken for a real Impressionist painting, we feel the image contains important aesthetic qualities that make it stand out from a traditional visualization. The top four images (the top row of Fig. 2) use a perceptual greyscale ramp to show the individual variation in temperature, precipitation, pressure, and windspeed. M was used to construct the painterly visualization of all four attributes shown in the bottom image of Fig. 2. Various luminance and texture patterns representing different weather phenomena are noted in this image.

We have applied our painterly visualizations to a number of additional real-world environments including scientific simulations, e-commerce activity logs, and medical scans. Anecdotal feedback from domain experts collaborating on our ef-

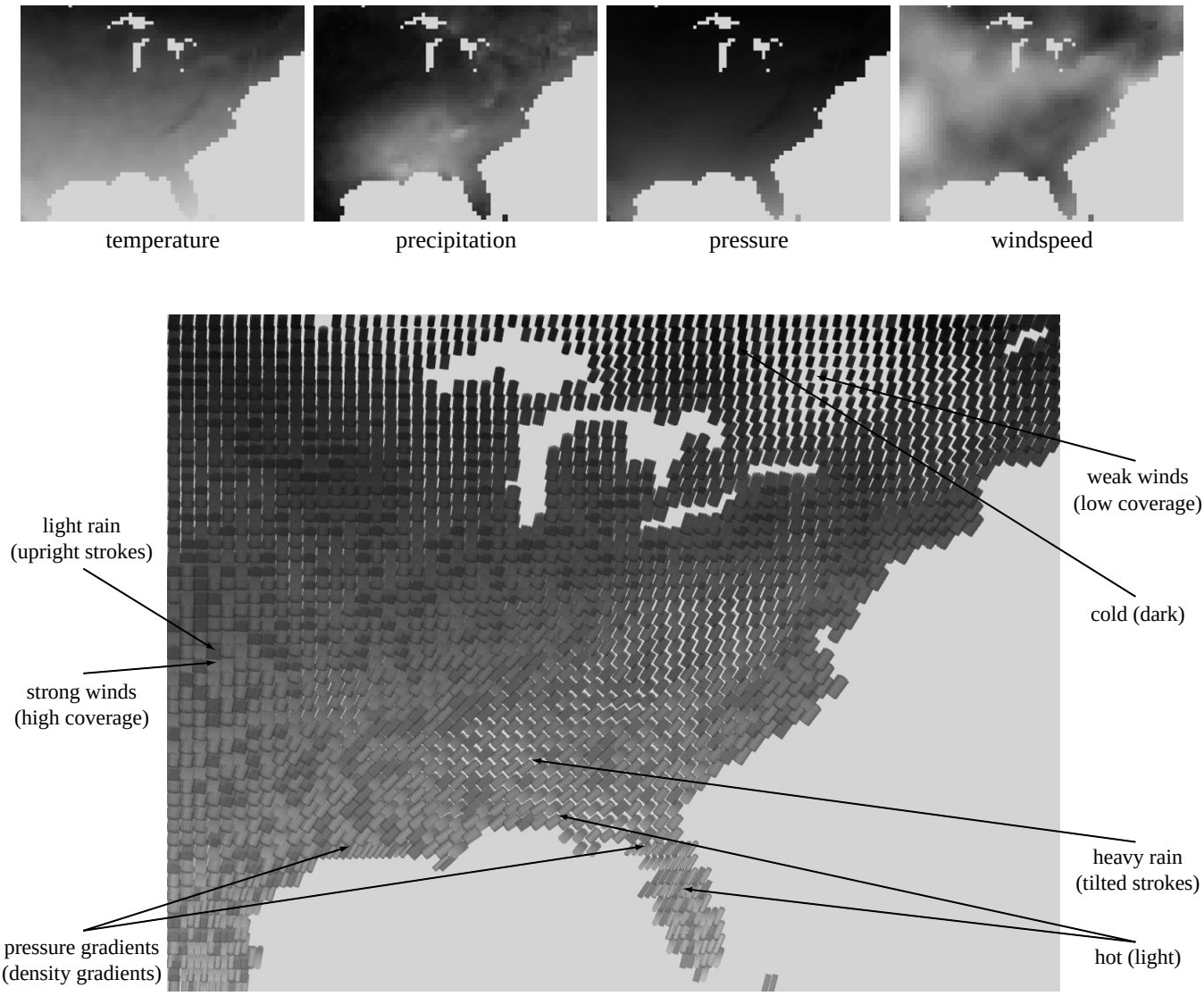


Figure 2: A painterly visualization of environment conditions for February over the eastern United States: (top row) greyscale ramps (dark for small values to light for large values) of temperature in isolation, precipitation in isolation, pressure in isolation, and windspeed in isolation; (bottom image) a painterly visualization of all four attributes represented with the brush stroke properties greyscale, size (or coverage), density, and orientation

forts suggests that our technique achieves its goal of producing images that: (1) represent multidimensional datasets in a clear and effective manner, and (2) contain many of the aesthetic and visually engaging properties of a real painting.

6 Conclusions and Future Work

This paper describes a new method of visualization that uses painted brush strokes to represent multidimensional data elements. Our strokes support the variation of visual properties based in large part on styles from the Impressionist school of painting. Each attribute in a dataset is mapped to a specific painterly style; a data element's attribute values can then be used to vary the visual appearance of its brush stroke. The styles we chose are closely related to perceptual features detected by the low-level human visual system. Research studying the use of these features during visualization allows us to optimize the selection and application of the corresponding painterly styles. The result is a "painted image" whose color and texture patterns are used to explore, analyze, verify, and discover information stored in a multidimensional dataset.

One important area of future work is the construction of new brush stroke models. Texture maps are common in most graphics APIs, and are often rendered using hardware acceleration. Unfortunately, certain styles (*e.g.*, stroke coarseness or weight) are not easy to manipulate using texture maps. It may also be difficult to animate textured brush strokes during real-time visualization. We are currently investigating three potential solutions to this problem: (1) building a library of texture maps that explicitly differ across certain styles; (2) using mathematical spline surfaces to model more sophisticated brush stroke properties, and (3) using a physical simulation system to construct realistic strokes. Early results suggest a combination of models (*e.g.*, a texture map library whose entries are precomputed or dynamically updated) may be most appropriate.

We are also working to identify new painterly styles, and to integrate them into our stroke models. Two promising candidates we have already discussed are coarseness and weight. We are reviewing literature on technical and artistic properties in Impressionism, while at the same time searching for perceptual features that may correspond to new painterly styles. Increasing the number of styles we can encode in each brush stroke will allow us to represent larger datasets with higher dimensionality.

We note one final advantage we can derive from the correspondence between perceptual features and painterly styles. We measure the perceptual salience of a visual feature using controlled psychophysical experiments. Exactly the same technique can be used to investigate the strengths and limitations of new painterly styles. Just as research in perception helps us to identify and control brush stroke properties during painterly visualization, work on new styles may offer insight into how the low-level visual system "sees" certain combinations of visual properties.

Acknowledgments

This work is supported by the National Science Foundation research grant NSF-ACI-0083421.

References

[Bergman *et al.*, 1995] Lawrence D. Bergman, Bernice E. Rogowitz, and Lloyd A. Treinish. A rule-based tool for assisting colormap selection. In *Proceedings Visualization '95*, pages 118–125, Atlanta, Georgia, 1995.

[Callaghan, 1990] T. C. Callaghan. Interference and dominance in texture segregation. In D. Brogan, editor, *Visual Search*, pages 81–87. Taylor & Francis, New York, New York, 1990.

[Chevreul, 1967] Michel Eugène Chevreul. *The Principles of Harmony and Contrast of Colors and Their Applications to the Arts*. Reinhold Publishing Corporation, New York, New York, 1967.

[Grinstein *et al.*, 1989] G. Grinstein, R. Pickett, and M. Williams. EXVIS: An exploratory data visualization environment. In *Proceedings Graphics Interface '89*, pages 254–261, London, Canada, 1989.

[Healey and Enns, 1999] Christopher G. Healey and James T. Enns. Large datasets at a glance: Combining textures and colors in scientific visualization. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):145–167, 1999.

[Rheingans and Tebbs, 1990] Penny Rheingans and Brice Tebbs. A tool for dynamic explorations of color mappings. *Computer Graphics*, 24(2):145–146, 1990.

[Rood, 1879] Ogden Nicholas Rood. *Modern Chromatics, with Applications to Art and Industry*. Appleton, New York, New York, 1879.

[Schapiro, 1997] Meyer Schapiro. *Impressionism: Reflections and Perceptions*. George Brazillier, Inc., New York, New York, 1997.

[Smith and Van Rosendale, 1998] P. H. Smith and J. Van Rosendale. Data and visualization corridors report on the 1998 CVD workshop series (sponsored by DOE and NSF). Technical Report CACR-164, Center for Advanced Computing Research, California Institute of Technology, 1998.

[Triesman, 1985] A. Triesman. Preattentive processing in vision. *Computer Vision, Graphics and Image Processing*, 31:156–177, 1985.

[Ware and Knight, 1995] Colin Ware and William Knight. Using visual texture for information display. *ACM Transactions on Graphics*, 14(1):3–20, 1995.

[Ware, 1988] C. Ware. Color sequences for univariate maps: Theory, experiments, and principles. *IEEE Computer Graphics & Applications*, 8(5):41–49, 1988.

[Weigle *et al.*, 2000] Christopher Weigle, William G. Emigh, Geniva Liu, Russell M. Taylor, James T. Enns, and Christopher G. Healey. Oriented texture slivers: A technique for local value estimation of multiple scalar fields. In *Proceedings Graphics Interface 2000*, pages 163–170, Montréal, Canada, 2000.

[Wolfe, 1994] Jeremy M. Wolfe. Guided Search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review*, 1(2):202–238, 1994.

Large Datasets at a Glance: Combining Textures and Colors in Scientific Visualization

Christopher G. Healey and James T. Enns

Abstract— **This paper presents a new method for using texture and color to visualize multivariate data elements arranged on an underlying height field. We combine simple texture patterns with perceptually uniform colors to increase the number of attribute values we can display simultaneously. Our technique builds multicolored perceptual texture elements (or pexels) to represent each data element. Attribute values encoded in an element are used to vary the appearance of its pexel. Texture and color patterns that form when the pexels are displayed can be used to rapidly and accurately explore the dataset. Our pexels are built by varying three separate texture dimensions: height, density, and regularity. Results from computer graphics, computer vision, and human visual psychophysics have identified these dimensions as important for the formation of perceptual texture patterns. The pexels are colored using a selection technique that controls color distance, linear separation, and color category. Proper use of these criteria guarantees colors that are equally distinguishable from one another. We describe a set of controlled experiments that demonstrate the effectiveness of our texture dimensions and color selection criteria. We then discuss new work that studies how texture and color can be used simultaneously in a single display. Our results show that variations of height and density have no effect on color segmentation, but that random color patterns can interfere with texture segmentation. As the difficulty of the visual detection task increases, so too does the amount of color on texture interference increase. We conclude by demonstrating the applicability of our approach to a real-world problem, the tracking of typhoon conditions in Southeast Asia.**

Keywords—**Color, color category, experimental design, human vision, linear separation, multivariate dataset, perception, pexel, preattentive processing, psychophysics, scientific visualization, texture, typhoon**

I. Introduction

THIS paper investigates the problem of visualizing multivariate data elements arrayed across an underlying height field. We seek a flexible method of displaying effectively large and complex datasets that encode multiple data values at a single spatial location. Examples include visualizing geographic and environmental conditions on topographical maps, representing surface locations, orientations, and material properties in medical volumes, or displaying rigid and rotational velocities on the surface of a three-dimensional object. Currently, features like hue, intensity, orientation, motion, and isocontours are used to represent these types of datasets. We are investigating the simultaneous use of perceptual textures and colors for multivariate visualization. We believe an effective combination of these features will increase the number of data values that can be shown at one time in a single display. To do this, we must first design methods for building texture and color patterns that support the rapid, accurate, and effective visualization of multivariate data elements.

We use multicolored perceptual texture elements (or pexels) to represent values in our dataset. Our texture elements are built by varying three separate texture dimensions: height, density, and regularity. Density and regularity have been identified in the computer vision literature as being important for performing texture classification [39], [40], [50]. Moreover, results from psychophysics have shown that all three dimensions are encoded in the low-level human visual system [1], [28], [51], [58]. Our pexels are colored using a technique that supports rapid, accurate, and consistent color identification. Three selection criteria are used to choose appropriate colors: color distance, linear separation, and named color category. All three criteria have been identified as important for measuring perceived color difference [3], [4], [14], [31], [60].

One of our real-world testbeds is the visualization of simulation results from studies being conducted in the Department of Zoology. Researchers are designing models of how they believe salmon feed and move in the open ocean. These simulated salmon are placed in a set of known environmental conditions, then tracked to see if their behavior mirrors that of the real fish. A method is needed for visualizing the simulation system. This method will be used to display both static (*e.g.,* environmental conditions for a particular month and year) and dynamic results (*e.g.,* a real-time display of environmental conditions as they change over time, possibly with the overlay of salmon locations and movement). We have approached the problems of dataset size and dimensionality by trying to exploit the power of the low-level human visual system. Research in computer vision and human visual psychophysics provides insight on how the visual system analyzes images. One of our goals is to select texture and color properties that will allow rapid visual exploration, while at the same time minimizing any loss of information due to interactions between the visual features being used.

Fig. 1 shows an example of our technique applied to the oceanographic dataset: environmental conditions in the northern Pacific Ocean are visualized using multicolored pexels. In this display, color represents open-ocean plankton density, height represents ocean current strength (taller for stronger), and density represents sea surface temperature (denser for warmer). Fig. 1 is only one frame from a much larger time-series of historical ocean conditions. Our choice of visual features was guided by experimental re-

C. G. Healey is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534. E-mail: healey@csc.ncsu.edu.

J. T. Enns is with the Department of Psychology, University of British Columbia, Vancouver, British Columbia, Canada, V6T 1Z4. E-mail: jenns@psych.ubc.ca.

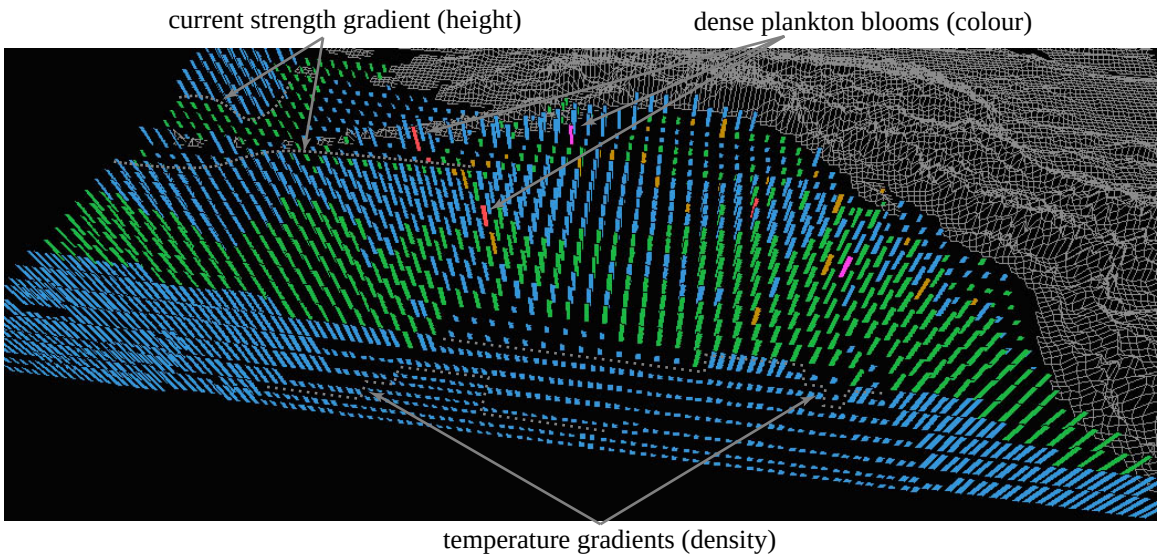


Fig. 1. Color, height, and density used to visualize open-ocean plankton density, ocean current strength, and sea surface temperature, respectively; low to high plankton densities represented with blue, green, brown, red, and purple, stronger currents represented with taller pexels, and warmer temperatures represented with denser pexels

sults that show how different color and texture properties can be used in combination to represent multivariate data elements.

Work described in this paper is an extension of earlier texture and color studies reported in [22], [23], [25]. We began our investigation by conducting a set of controlled experiments to measure user performance and identify visual interference that may occur during visualization. Individual texture and color experiments were run in isolation. The texture experiments studied the perceptual salience of and interference between the perceptual texture dimensions height, density, and regularity. The color experiments measured the effects of color distance, linear separation, and named color category on perceived color difference. Positive results from both studies led us to conduct an additional set of experiments that tested the combination of texture and color in a single display. Results in the literature vary in their description of this task: some researchers have reported that random color variation can interfere significantly with a user's ability to see an underlying texture region [8], [9], [49], while others have found no impact on performance [53], [58]. Our investigation extends this earlier work on two-dimensional texture patterns into an environment that displays height fields using perspective projections. To our knowledge, no one has studied the issue of color on texture or texture on color interference during visualization. Results from our experiments showed that we could design an environment in which color variations caused a small but statistically reliable interference effect during texture segmentation. The strength of this effect depends on the difficulty of the analysis task: tasks that are more difficult are more susceptible to color interference. Texture variation, on the other hand, caused no interference during color segmentation. We are using these results to build a collection of pexels that allow a viewer to visually explore a multivariate dataset in a rapid, accurate, and relatively effortless fashion.

We begin with a description of results from computer vision, computer graphics, and psychophysics that discuss methods for identifying and controlling the perceptual properties of texture and color. Next, we describe an area of human psychophysics concerned with modeling control of visual attention in the low-level human visual system. We discuss how the use of visual stimuli that control attention can lead to significant advantages during visualization. Section 4 gives an overview of the experiments we used to build and test our perceptual texture elements. In Section 5, we discuss how we chose to select and test our perceptual colors. Following this, we describe new experiments designed to study the combined use of texture and color. Finally, we report on practical applications of our research in Section 7, then discuss avenues for future research in Section 8.

II. Related Work

Researchers from many different areas have studied texture and color in the context of their work. Before we discuss our own investigations, we provide an overview of results in the literature that are most directly related to our interests.

A. Texture

The study of texture crosses many disciplines, including computer vision, human visual psychophysics, and computer graphics. Although each group focuses on separate problems (texture segmentation and classification in computer vision, modeling the low-level human visual system in psychophysics, and information display in graphics) they each need ways to describe accurately the texture patterns being classified, modeled, or displayed. [41] describes two

general classes of texture representation: statistical models that use convolution filters and other techniques to measure variance, inertia, entropy, or energy, and perceptual models that identify underlying perceptual texture dimensions like contrast, size, regularity, and directionality. Our current texture studies focus on the perceptual features that make up a texture pattern. In our work we demonstrate that we can use texture attributes to assist in visualization, producing displays that allow users to rapidly and accurately explore their data by analyzing the resulting texture patterns.

Different methods have been used to identify and investigate the perceptual features inherent in a texture pattern. Bela Julész [27], [28] has conducted numerous psychophysical experiments to study how a texture's first, second, and third-order statistics affect discrimination in the low-level visual system. This led to the texton theory [29], which proposes that early vision detects three types of features (or textons, as Julész called them): elongated blobs with specific visual properties (*e.g.,* hue, orientation, or length), ends of line segments, and crossing of line segments. Other psychophysical researchers have studied how visual features like color, orientation, and form can be used to rapidly and accurately segment collections of elements into spatially coherent regions [7], [8], [52], [58], [59].

Work in psychophysics has also been conducted to study how texture gradients are used to judge an object's shape. Cutting and Millard discuss how different types of gradients affect a viewer's perception of the flatness or curvature of an underlying 3D surface [13]. Three texture gradients were tested: perspective, which refers to smooth changes in the horizontal width of each texture element, compression, which refers to changes in the height to width ratio of a texture element, and density, which refers to changes in the number of texture elements per unit of solid visual angle. For most surfaces the perspective and compression gradients decrease with distance, while the density gradient increases. Cutting and Millard found that viewers use perspective and density gradients almost exclusively to identify the relative slant of a flat surface. In contrast, the compression gradient was most important for judging the curvature of undulating surfaces. Later work by Aks and Enns on overcoming perspective foreshortening in early vision also discussed the effects of texture gradients on the perceived shape of an underlying surface [1].

Work in computer vision is also interested in how viewers segment images, in part to try to develop automated texture classification and segmentation algorithms. Tamura et al. and Rao and Lohse identified texture dimensions by conducting experiments that asked observers to divide pictures depicting different types of textures (Brodatz images) into groups [39], [40], [50]. Tamura et al. used their results to propose methods for measuring coarseness, contrast, directionality, line-likeness, regularity, and roughness. Rao and Lohse used multidimensional scaling to identify the primary texture dimensions used by their observers to group images: regularity, directionality, and complexity. Haralick built grayscale spatial dependency matrices to identify features like homogeneity, contrast, and linear dependency [21]. These features were used to classify satellite images into categories like forest, woodlands, grasslands, and water. Liu and Picard used Wold features to synthesize texture patterns [35]. A Wold decomposition divides a 2D homogeneous pattern (*e.g.,* a texture pattern) into three mutually orthogonal components with perceptual properties that roughly correspond to periodicity, directionality, and randomness. Malik and Perona designed computer algorithms that use orientation filtering, nonlinear inhibition, and computation of the resulting texture gradient to mimic the discrimination ability of the low-level human visual system [37].

Researchers in computer graphics are studying methods for using texture to perform tasks such as representing surface shape and extent, displaying flow patterns, identifying spatially coherent regions in high-dimensional data, and multivariate visualization. Schweitzer used rotated discs to highlight the shape and orientation of a three-dimensional surface [47]. Grinstein et al. created a system called EXVIS that uses "stick-men" icons to produce texture patterns that show spatial coherence in a multivariate dataset [19]. Ware and Knight used Gabor filters to construct texture patterns; attributes in an underlying dataset are used to modify the orientation, size, and contrast of the Gabor elements during visualization [57]. Turk and Banks described an iterated method for placing streamlines to visualize two-dimensional vector fields [54]. Interrante displayed texture strokes to help show three-dimensional shape and depth on layered transparent surfaces; principal directions and curvatures are used to orient and advect the strokes across the surface [26]. Salisbury et al. used texturing techniques to build computer-generated pen-and-ink drawings that convey a realistic sense of shape, depth, and orientation [46]. Finally, Laidlaw described two methods for displaying a 2D diffuse tensor image with seven values at each spatial location [32]. The first method used the shape of normalized ellipsoids to represent individual tensor values. The second used techniques from oil painting [38] to represent all seven values simultaneously via multiple layers of varying brush strokes.

Visualization techniques like EXVIS [19] are sometimes referred to as "glyph-based" methods. Glyphs are graphical icons with visual features like shape, orientation, color, and size that are controlled by attributes in an underlying dataset. Glyph-based techniques range from representation via individual icons to the formation of texture and color patterns through the overlay of many thousands of glyphs. Initial work by Chernoff suggested the use of facial characteristics to represent information in a multivariate dataset [6], [10]. A face is used to summarize a data element; individual data values control features in the face like the nose, eyes, eyebrows, mouth, and jowls. Foley and Ribarsky have created a visualization tool called Glyphmaker that can be used to build visual representations of multivariate datasets in an effective, interactive fashion [16]. Glyphmaker uses a glyph editor and glyph binder to create glyphs, to arrange them spatially, and to bind attributes to their visual

properties. Levkowitz described a prototype system for combining colored squares to produce patterns to represent an underlying multivariate dataset [33]. Other techniques such as the normalized ellipsoids of Laidlaw [32], the Gabor elements of Ware [57], or even the pexels described in this paper might also be classified as glyphs, although we prefer to think of them as texture-based visualization methods.

B. Color

As with texture, color has a rich history in the areas of computer graphics and psychophysics. In graphics, researchers have studied issues related to color specification, color perception, and the selection of colors for information representation during visualization. Work in psychophysics describes how the human visual system mediates color perception.

A number of different color models have been built in computer graphics to try to support the unambiguous specification of color [60]. These models are almost always three-dimensional, and are often divided into a device-dependent class, where a model represents the displayable colors of a given output device, and a device-independent class, where a model provides coordinates for colors from the visible color spectrum. Common examples of device-dependent models include monitor RGB and CMYK. Common examples of device-independent models include CIE XYZ, CIE LUV, and CIE Lab. Certain models were designed to provide additional functionality that can be used during visualization. For example, both CIE LUV and CIE Lab provide rough perceptual uniformity; that is, the Euclidean distance between a pair of colors specified in these models roughly corresponds to perceived color difference. These models also provide a measure of isoluminance, since their L-axis is meant to correspond to perceived brightness.

Previous work has also addressed the issue of constructing color scales for certain types of data visualization. For example, Ware and Beatty describe a simple color visualization technique for displaying correlation in a five-dimensional dataset [56]. Ware has also designed a method for building continuous color scales that control color surround effects [55]. The color scales use a combination of luminance and hue variation that allows viewers to determine the value associated with a specific color, and to identify the spatial locations of peaks and valleys (*i.e.,* to see the shape) in a 2D distribution of an attribute's values. Controlling color surround ensures a small, near-constant perceptual error effect from neighboring colors across the entire range of the color scale. Robertson described user interface techniques for visualizing the range of colors a display device can support using perceptual color models [44]. Rheingans and Tebbs have built a system that allows users to interactively construct a continuous color scale by tracing a path through a 3D color model [43]. This technique allows users to vary how different values of an attribute map onto the color path. Multiple colors can be used to highlight areas of interest within an attribute, even when those areas constitute only a small fraction of the attribute's full range of allowable values. Levkowitz and Herman designed a locally optimal color scale that maximizes the just-noticeable color difference between neighboring pairs of colors [34]. The result is a significantly larger number of just-noticeably different colors in their color scales, compared to standard scales like red-blue, rainbow, or luminance.

Recent work at the IBM Thomas J. Watson Research Center has focused on a rule-based visualization tool [45]. Initial research addressed the need for rules that take into account how a user perceives visual features like hue, luminance, height, and so on. These rules are used to guide or restrict a user's choices during data-feature mapping. The rules use various metadata, for example, the visualization task being performed, the visual features being used, and the spatial frequency of the data being visualized. A specific example of one part of this system is the colormap selection tool from the IBM Visualization Data Explorer [5]. The selection tool uses perceptual rules and metadata to limit the choice of colormaps available to the user.

Finally, psychophysicists have been working to identify properties that affect perceived color difference. Two important discoveries include the linear separation [3], [4], [14] and color category [31] effects. The linear separation theory states that if a target color can be separated from all the other background colors being displayed with a single straight line in color space, it will be easier to detect (*i.e.,* its perceived difference from all the other colors will increase) compared to a case where it can be formed by a linear combination of the background colors. D'Zmura's initial work on this phenomena [14] showed that a target color could be rapidly identified in a sea of background elements uniformly colored one of two colors (*e.g.,* an orange target could be rapidly identified in a sea of red elements, or in a sea of yellow elements). The same target, however, was much more difficult to find when the background elements used both colors simultaneously (*e.g.,* an orange target could not be rapidly identified in a sea of red and yellow elements). This second case is an example of a target color (orange) that is a linear combination of its background colors (red and yellow). The color category effect suggests that the perceived difference between a pair of colors increases when the two colors occupy different named color regions (*i.e.,* one lies in the "blue" region and one lies in the "purple" region, as opposed to both in blue or both in purple). We believe both results may need to be considered to guarantee perceptual uniformity during color selection.

C. Combined Texture and Color

Although texture and color have been studied extensively in isolation, much less work has focused on their combined use for information representation. An effective method of displaying color and texture patterns simultaneously would increase the number of attributes we can represent at one time. The first step towards supporting this goal is the determination of the amount of visual interference that occurs between these features during visualization.

Experiments in psychophysics have produced interesting but contradictory answers to this question. Callaghan reported asymmetric interference of color on form during texture segmentation: a random color pattern interfered with the identification of a boundary between two groups of different forms, but a random form pattern had no effect on identifying color boundaries [8], [9]. Triesman, however, showed that random variation of color had no effect on detecting the presence or absence of a target element defined by a difference in orientation (recall that directionality has been identified as a fundamental perceptual texture dimension) [53]. Recent work by Snowden [49] recreated the differing results of both Callaghan and Triesman. Snowden ran a number of additional experiments to test the effects of random color and stereo depth variation on the detection of a target line element with a unique orientation. As with Callaghan and Triesman, results differed depending on the target type. Search for a single line element was rapid and accurate, even with random color or depth variation. Search for a spatial collection of targets was easy only when color and depth were fixed to a constant value. Random variation of color or depth produced a significant reduction in detection accuracy. Snowden suggests that the visual system wants to try to group spatially neighboring elements with common visual features, even if this grouping is not helpful for the task being performed. Any random variation of color or depth interferes with this grouping process, thereby forcing a reduction in performance.

These results leave unanswered the question of whether color variation will interfere with texture identification during visualization. Moreover, work in psychophysics studied two-dimensional texture segmentation. Our pexels are arrayed over an underlying height field, then displayed in 3D using a perspective projection. Most of the research to date has focused on color on texture interference. Less work has been conducted to study how changes in texture dimensions like height, density, or regularity will affect the identification of data elements with a particular target color. The question of interference in this kind of height-field environment needs to be addressed before we can recommend methods for the combined use of color and texture.

III. Perceptual Visualization

An important requirement for any visualization technique is a method for rapid, accurate, and effortless visual exploration. We address this goal by using what is known about the control of human visual attention as a foundation for our visualization tools. The individual factors that govern what is attended in a visual display can be organized along two major dimensions: bottom-up (or stimulus driven) versus top-down (or goal directed).

Bottom-up factors in the control of attention include the limited set of features that psychophysicists have identified as being detected very quickly by the human visual system, without the need for search. These features are often called preattentive, because their detection occurs rapidly and accurately, usually in an amount of time independent of the total number of elements being displayed. When applied properly, preattentive features can be used to perform different types of exploratory analysis. Examples include searching for data elements with a unique visual feature, identifying the boundaries between groups of elements with common features, tracking groups of elements as they move in time and space, and estimating the number of elements with a specific feature. Preattentive tasks can be performed in a single glance, which corresponds to 200 milliseconds (ms) or less. As noted above, the time required to complete the task is independent of the number of data elements being displayed. Since the visual system cannot choose to refocus attention within this timeframe, users must complete their task using only a "single glance" at the image.

Fig. 2 shows examples of both types of target search. In Fig. 2a-2d the target, a red circle, is easy to find. Here, the target contains a preattentive feature unique from the background distracters: color (red versus blue) or shape (circle versus square). This unique feature is used by the low-level visual system to rapidly identify the presence or absence of the target. Unfortunately, an intuitive combination of these results can lead to visual interference. Fig. 2e and 2f simulate a two-dimensional dataset where one attribute is encoded with color (red or blue), and the other is encoded with shape (circle or square). Although these features worked well in isolation, searching for a red circle target in a sea of blue circles and red squares is significantly more difficult. In fact, experiments have shown that search time is directly proportional to the number of elements in the display, suggesting that viewers are searching small subgroups of elements (or even individual elements themselves) to identify the target. In this example the low-level visual system has no unique feature to search for, since circular elements (blue circles) and red elements (red squares) are also present in the display. The visual system cannot integrate preattentively the presence of multiple visual features (circular and red) at the same spatial location. This is a very simple example of a situation where knowledge of preattentive vision would have allowed us to avoid displays that actively interfere with our analysis task.

In spite of the perceptual salience of the target in Fig. 2a-2d, bottom-up influences cannot be assumed to operate independently of the current goals and attentional state of the observer. Recent studies have demonstrated that many of the bottom-up factors only influence perception when the observer is engaged in a task in which they are expected or task-relevant (see the review by [15]). For example, a target defined as a color singleton will "pop out" of a display only when the observer is looking for targets defined by color. The same color singleton will not influence perception when observers are searching exclusively for luminance defined targets. Sometimes observers will fail completely to see otherwise salient targets in their visual field, either because they are absorbed in the performance of a cognitively-demanding task [36], there are a multitude of other simultaneous salient visual events [42], or because the salient event occurs during an eye movement or other change in viewpoint [48]. Therefore, the control of atten-

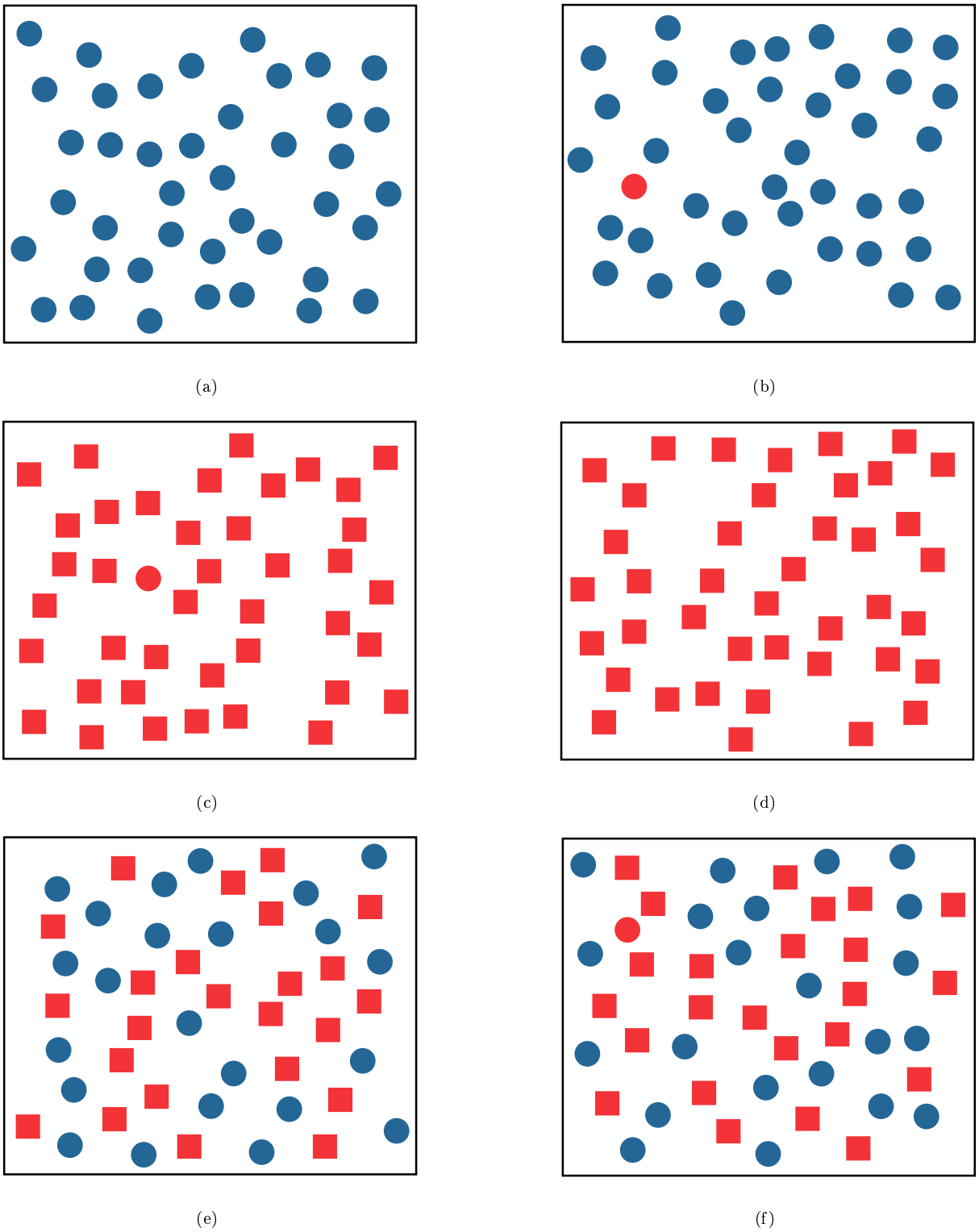


Fig. 2. Examples of target search: (a, b) identifying a red target in a sea of blue distracters is rapid and accurate, target absent in (a), present in (b); (c, d) identifying a red circular target in a sea of red square distracters is rapid and accurate, target present in (c), absent in (d); (e, f) identifying the same red circle target in a combined sea of blue circular distracters and red square distracters is significantly more difficult, target absent in (e), present in (f)

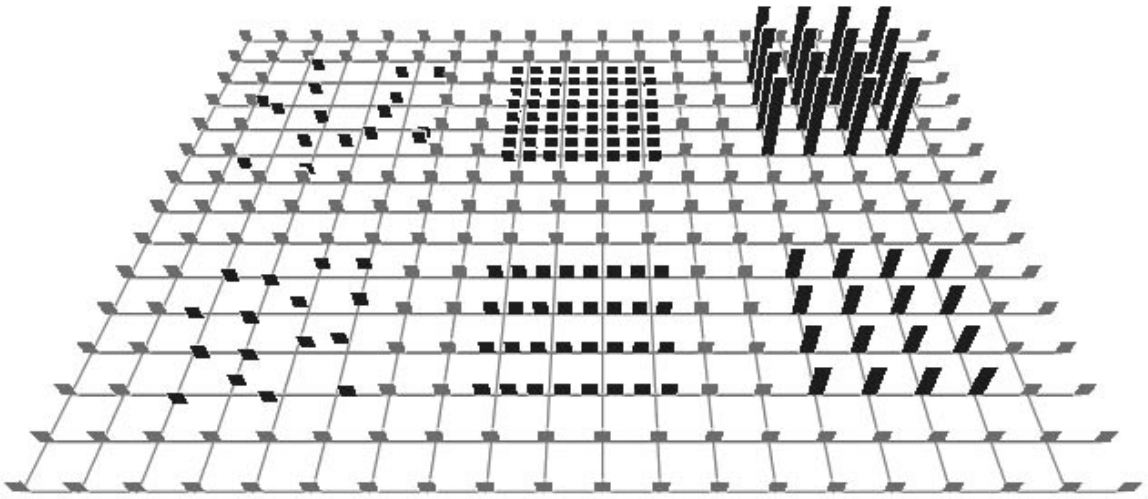


Fig. 3. A background array of short, sparse, regular pexels; the lower and upper groups on the left contain irregular and random pexels, respectively; the lower and upper groups in the center contain dense and very dense pexels; the lower and upper groups to the right contain medium and tall pexels

tion must always be understood as an interaction between bottom-up and top-down mechanisms.

Our research is focused on identifying relevant results in the vision and psychophysical literature, then extending these results and integrating them into a visualization environment. Tools that make use of preattentive vision offer a number of important advantages during multivariate visualization:

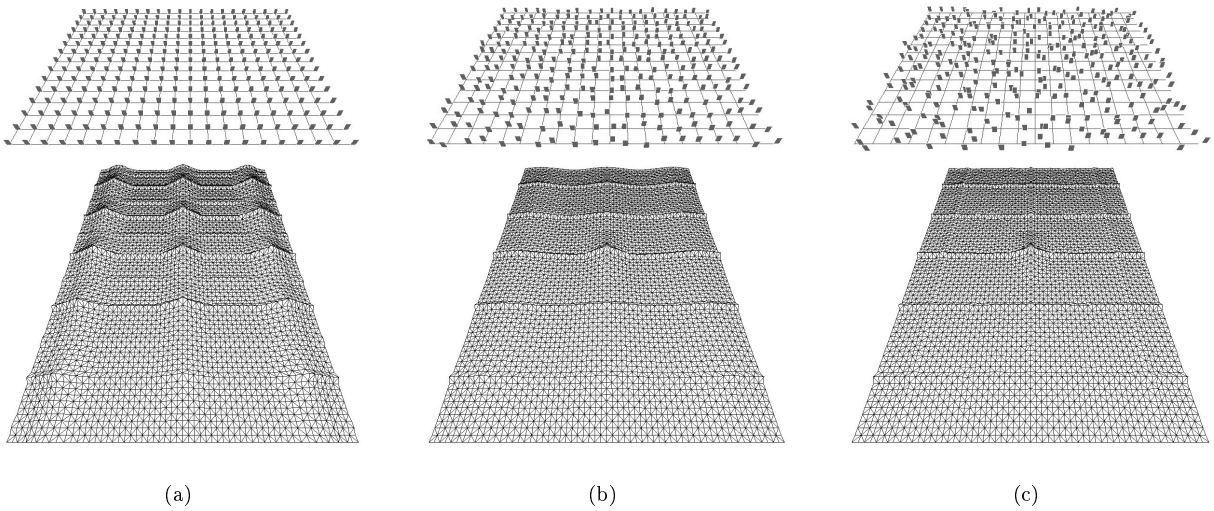
1. Visual analysis is rapid, accurate, and relatively effortless since preattentive tasks can be completed in 200 ms or less. We have shown that tasks performed on static displays extend to a dynamic environment where data frames are shown one after another in a movie-like fashion [24] (*i.e.,* tasks that can be performed on an individual display in 200 ms can also be performed on a sequence of displays shown at five frames a second).
2. The time required for task completion is independent of display size (to the resolution limits of the display). This means we can increase the number of data elements in a display with little or no increase in the time required to analyze the display.
3. Certain combinations of visual features cause interference patterns that mask information in the low-level visual system. Our experiments are designed to identify these situations. This means our visualization tools can be built to avoid data-feature mappings that might interfere with the analysis task.

Since preattentive tasks are rapid and insensitive to display size, we believe visualization techniques that make use of these properties will support high-speed exploratory analysis of large, multivariate datasets. Care must be taken, however, to ensure that we choose data-feature mappings that maximize the perceptual salience of all the data attributes being displayed.

We are currently investigating the combined use of two important and commonly used visual features: texture and color. Previous work in our laboratory has identified methods for choosing perceptual textures and colors for multivariate visualization. Results from vision and psychophysics on the simultaneous use of both features are mixed: some researchers have reported that background color patterns mask texture information, and vice-versa, while others claim that no interference occurs. Experiments reported in this paper are designed to test for color-texture interactions during visualization. A lack of interference would suggest that we could combine both features to simultaneously encode multiple attributes. The presence of interference, on the other hand, would place important limitations on the way in which visual attributes should be mapped onto data attributes. Visualization tools based on these findings would then be able to display textures with the appropriate mapping of data dimensions to visual attributes.

IV. Pexels

One of the main goals of multivariate visualization is to display multiple attribute values at a single spatial location without overwhelming the user's ability to comprehend the resulting image. Researchers in vision, psychophysics, and graphics have been studying how the visual system analyzes texture patterns. We wanted to know whether perceptual texture dimensions could be used to represent multivariate data elements during visualization. To this end, we designed a set of perceptual texture elements, or pexels, that support the variation of three separate texture dimensions: density, regularity, and height. Density and regularity have been identified in the literature as primary texture dimensions [39], [40], [50]. Although height might not be considered an "intrinsic textural cue", we note that height is one aspect of element size, and that size is an important property of a texture pattern. Results from psychophysical experiments have shown that differences in



(a) (b) (c)

Fig. 4. Three displays of pexels with different regularity and a 5×3 patch from the center of the corresponding autocorrelation graphs: (a) a completely regular display, resulting in sharp peaks of height 1.0 at regular intervals in the autocorrelation graph; (b) a display with irregularly-spaced pexels, peaks in the graph are reduced to a maximum height between 0.3 and 0.4; (c) a display with randomly-spaced pexels, resulting in a completely flat graph except at (0,0) and where underlying grid lines overlap

height are detected preattentively [51], moreover, viewers properly correct for perspective foreshortening when they perceive that elements are being displayed in 3D [1]. We wanted to build three-dimensional pexels that "sit up" on the underlying surface. This allows for the possibility of applying various orientations (another important texture dimension) to a pexel.

Our pexels look like a collection of one or more upright paper strips. Each element in the dataset is represented by a single pexel. The user maps attributes in their dataset to density (which controls the number of strips in a pexel), height, and regularity. The attribute values for a particular element can then control the appearance of its pexel. When all the pexels for a particular data frame are displayed, they form texture patterns that represent the underlying structure of the dataset.

Fig. 3 shows an example of regularity, density, and height varied across three discrete values. Each pexel in the original array (shown in gray) is short, sparse, and regular. The lower and upper patches on the left of the array (shown in black) contain irregular and random pexels, respectively. The lower and upper patches in the middle of the array contain dense and very dense pexels. The lower and upper patches on the right contain medium and tall pexels.

A. Ordering Texture Dimensions

In order to use height, density, and regularity during visualization, we needed an ordinal ranking for each dimension. Height and density both have a natural ordering: shorter comes before taller, and sparser comes before denser.

Although viewers can often order regularity intuitively, we required a more formal method for measurement. We chose to use autocorrelation to rank regularity. This technique measures the second-order statistic of a texture pattern. Psychophysicists have reported that a change in regularity produces a corresponding change in a texture's second order statistic [27], [28], [30]. Intuitively, autocorrelating an image shifts two copies of the image on top of one another, to see how closely they can be matched. If the texture is made up of a regular, repeating pattern it can be shifted in various ways to exactly overlap with itself. As more and more irregularity is introduced into the texture, the amount of overlap decreases, regardless of where we shift the copies. Consider two copies of an image A and B, each with a width of N and a height of M pixels. The amount of autocorrelation that occurs when A is overlaid onto B at offset (t, u) is:

$$C(t,u) = \frac{1}{K} \sum_{x=1}^{N} \sum_{y=1}^{M} (A[x,y] - \overline{A})(B[x+t, y+u] - \overline{B}) \quad (1)$$

$$K = NM \sqrt{\sigma^2(A)} \sqrt{\sigma^2(B)} \quad (2)$$

$$\overline{A} = \frac{1}{NM} \sum_{x=1}^{N} \sum_{y=1}^{M} A[x,y] \quad (3)$$

$$\sigma^2(A) = \frac{1}{NM} \sum_{x=1}^{N} \sum_{y=1}^{M} (A[x,y] - \overline{A})^2 \quad (4)$$

with $\overline{B}$ and $\sigma^2(B)$ computed in a similar fashion. Elements in A that do not overlap with B are wrapped to the opposite side of B (*i.e.,* elements in A lying above the top of B wrap back to the bottom, elements lying below the bottom of B wrap back to the top, similarly for elements to the left or right of B).

As a practical example, consider Fig. 4a (pexels on a regular underlying grid), Fig. 4b (pexels on an irregular grid),

and Fig. 4c (pexels on a random grid). Irregular and random pexels are created by allowing each strip in the pexel to walk a random distance (up to fixed maximum) in a random direction from its original anchor point. Autocorrelation was computed on the orthogonal projection of each image. A 5×3 patch from the center of the corresponding autocorrelation graph is shown beneath each of the three grids. Results in the graphs mirror what we see in each display, that is, as randomness increases, peaks in the autocorrelation graph decrease in height. In Fig. 4a peaks of height 1.0 appear at regular intervals in the graph. Each peak represents a shift that places pexels so they exactly overlap with one another. The rate of increase towards each peak differs in the vertical and horizontal directions because the elements in the graph are rectangles (*i.e.*, taller than they are wide), rather than squares. In Fig. 4b, the graph has the expected sharp peak at (0,0). It also has gentle peaks at shift positions that realign the grid with itself. The peaks are not as high as for the regular grid, because the pexels no longer align perfectly with one another. The sharp vertical and horizontal ridges in the graph represent positions where the underlying grid lines exactly overlap with one another (the grid lines showing the original position of each pexel are still regular in this image). The height of each gentle peak ranges between 0.3 and 0.4. Increasing randomness reduces again the height of the peaks in the correlation graph. In Fig. 4c, no peaks are present, apart from (0,0) and the sharp ridges that occur when the underlying grid lines overlap with one another. The resulting correlation values suggests that this image is "more random" than either of its predecessors. Informal tests with a variety of regularity patterns showed a near-perfect match between user-chosen rankings and rankings based on our autocorrelation technique. Autocorrelation on the perspective projections of each grid could also be computed. The tall peaks and flattened results would be similar to those in Fig. 4, although the density of their spacing would change near the top of the image due to perspective compression and foreshortening.

B. Pexel Salience and Interference

We conducted experiments to test the ability of each texture dimension to display effectively an underlying data attribute during multivariate visualization. To summarize, our experiments were designed to answer the following three questions:

1. Can the perceptual dimensions of density, regularity, and height be used to show structure in a dataset through the variation of a corresponding texture pattern?
2. How can we use the dataset's attributes to control the values of each perceptual dimension?
3. How much visual interference occurs between each of the perceptual dimensions when they are displayed simultaneously?

C. Experiments

We designed texture displays to test the detectability of six different target types: taller, shorter, denser, sparser, more regular, and more irregular. For each target type, a number of parameters were varied, including exposure duration, texture dimension salience, and visual interference. For example, during the "taller" experiment, each display showed a 20×15 array of pexels rotated 45° about the X-axis. Observers were asked to determine whether the array contained a group of pexels that were taller than all the rest. The following conditions varied:

- *target-background pairing:* some displays showed a medium target in a sea of short pexels, while others showed a tall target in a sea of medium pexels; this allowed us to test whether some target defining attributes were generally more salient than others,
- *secondary texture dimension:* displays contained either no background variation (every pexel was sparse and regular), a random variation of density across the array, or a random variation of regularity across the array; this allowed us to test for background interference during target search,
- *exposure duration:* displays were shown for 50, 150, or 450 ms; this allowed us to test for a reduction in performance when exposure duration was decreased, and
- *target patch size:* target groups were either 2×2 pexels or 4×4 pexels in size; this allowed us to test for a reduction in performance for smaller target patches.

The heights, densities, and regularities we used were chosen through a set of pilot studies. Two patches were placed side-by-side, each displaying a pair of heights, densities, or regularities. Viewers were asked to answer whether the patches were different from one another. Response times for correct answers were used as a measure of performance. We tested a range of values for each dimension, although the spatial area available for an individual pexel during our experiments limited the maximum amount of density and irregularity we were able to display. The final values we chose could be rapidly and accurately differentiated in this limited setting.

The experiments that tested the other five target types (shorter, denser, sparser, regular, and irregular) were designed in a similar fashion, with one exception. Experiments testing regularity had only one target-background pairing: a target of regular pexels in a sea of random pexels (for the regular experiment), or a target of random pexels in a sea of regular pexels (for the irregular experiment). The pilot studies used to select values for each dimension showed that users had great difficulty discriminating an irregular patch from a random patch. This was due in part to the small spatial area available to each pexel.

Our pilot studies produced experiments that tested three separate heights (short, medium, and tall), three separate densities (sparse, dense, and very dense) and two separate regularities (regular and random). Examples of two display types (taller and regular) are shown in Fig. 5. Both displays include target pexels. Fig. 5a contains a 2×2 target group of medium pexels in a sea of short pexels. The density of each pexel varies across the array, producing an underlying density pattern that is clearly visible. This display type simulates two dimensional data elements being

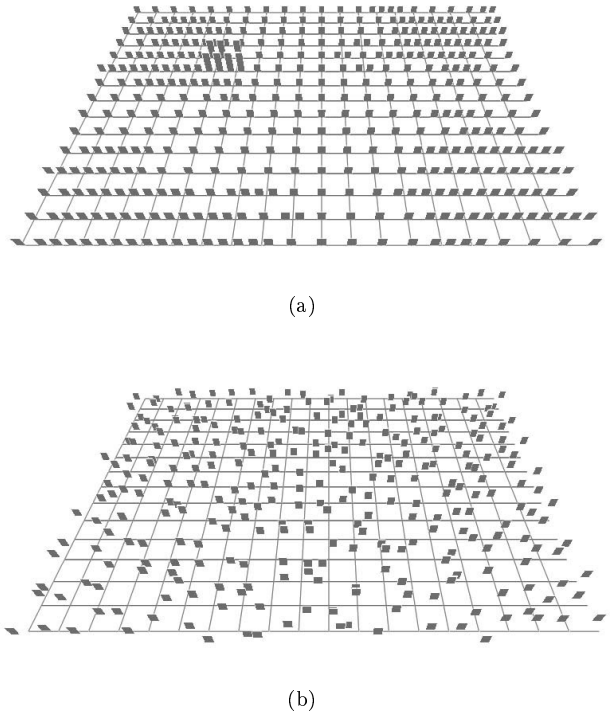


Fig. 5. Two display types from the taller and regular pexel experiments: (a) a target of medium pexels in a sea of short pexels with a background density pattern (2×2 target group located left of center); (b) a target of regular pexels in a sea of irregular pexels with no background texture pattern (2×2 target group located 3 grid steps right and 7 grid steps up from the lower-left corner of the array)

visualized with height as the primary texture dimension and density as the secondary texture dimension. Recall that the number of paper strips in a pexel depends on its density. Since three of the target pexels in Fig. 5a are dense, they each display two strips. The remaining pexel is sparse, and therefore displays a only single strip. Fig. 5b contains a 2×2 target group of regular pexels in a sea of random pexels, with a no background texture pattern. The taller target in Fig. 5a is very easy to find, while the regular target in Fig. 5b is almost invisible.

D. Results

Detection accuracy data were analyzed using a multifactor analysis of variance (ANOVA). A complete description of our analysis and statistical findings is available in [22], [23], [25]. In summary, we found:

1. Taller target regions were identified rapidly (*i.e.,* 150 ms or less) with very high accuracy (approximately 93%); background density and regularity patterns produced no significant interference.
2. Shorter, denser, and sparser targets were more difficult to identify than taller targets, although good results were obtained at both 150 and 450 ms (82.3%, 94.0%, and 94.7% for shorter, denser, and sparser targets with no background variation at 150 ms). This was not surprising, since similar results have been documented by [51] and [1] using displays of texture on a two-dimensional plane.
3. Background variation in non-target attributes produced small, but statistically significant, interference effects. These effects tended to be largest when target detectability was lowest. For example, density and regularity interfered with searching for shorter targets; height and regularity interfered with searching for sparser targets; but only height interfered with the (easier to find) denser targets.
4. Irregular target regions were difficult to identify at 150 and 450 ms, even with no secondary texture pattern (approximately 76%). Whether this accuracy level is sufficiently high will depend on the application. In contrast, regular regions were invisible under these conditions; the percentage of correct responses approached chance (*i.e.,* 50%) in every condition.

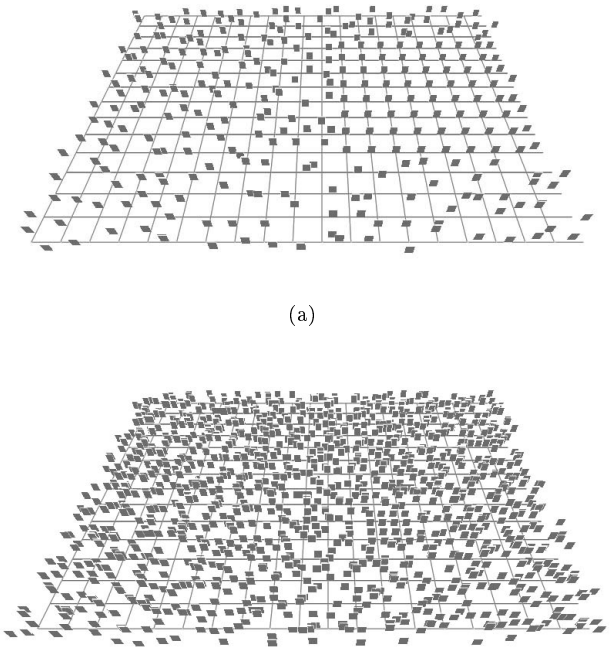


Fig. 6. Two displays with a regular target, both displays should be compared with the target shown in Fig. 5b: (a) larger target, an 8×8 target in a sea of sparse, random pexels; (b) denser background, a 2×2 target in a sea of dense, random pexels (target group located right of center)

Our poor detection results for regularity were unexpected, particularly since vision algorithms that perform texture classification use regularity as one of their primary decision criteria [35], [39], [40], [50]. We confirmed that our results were not due to a difference in our definition of regularity; the way we produced irregular patches matches the methods described by [20], [28], [30], [39], [40], [50]. It may be that regularity is important for classifying different textures, but not for the type of texture segmentation that we are performing. Informal post-experiment investigations showed that we could improve the salience of a reg-

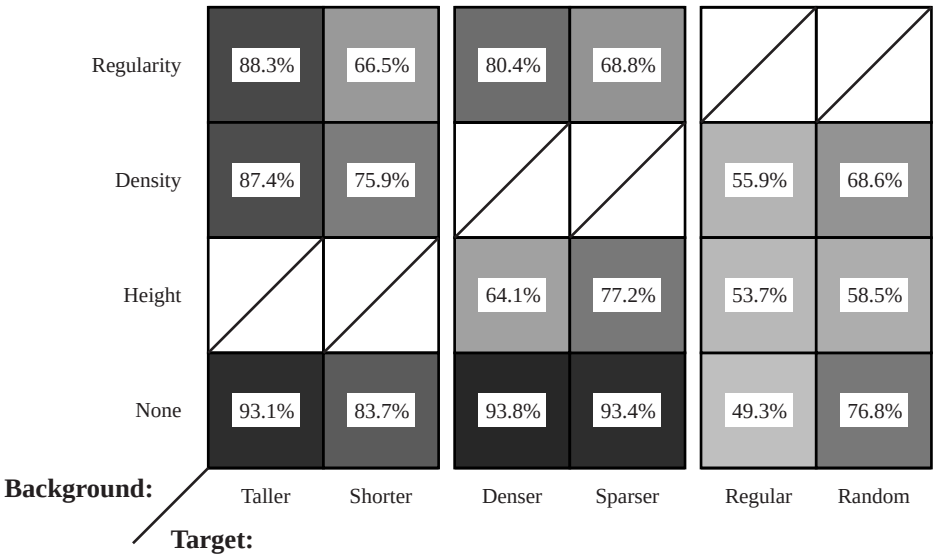


Fig. 7. A table showing the percentage of correct responses for each target-background pairing; target type along the horizontal axis, background type along the vertical axis; darker squares represent pairings with a high percentage of correct responses; blank entries with diagonal slashes indicate target-background pairings that do not exist

ular (or irregular) patch by increasing its size (Fig. 6a), or by increasing the minimum pexel density to be very dense (Fig. 6b). However, neither of these solutions is necessarily useful. There is no way to guarantee that data values will cluster together to form the large spatial regions needed for regularity detection. If we constrain density to be very dense across the array, we lose the ability to vary density over an easily identifiable range. This reduces the dimensionality of our pexels to two (height and regularity), producing a situation that is no better than when regularity is difficult to identify. Because of this, we normally choose to display an attribute with low importance using regularity. While differences in regularity cannot be detected consistently by the low-level visual system, in many cases users may be able to see the changes when areas of interest in the dataset are identified and analyzed in a focused or attentive fashion.

Fig. 7 shows average subject performance as a table representing each target-background pair. Target type varies along the horizontal axis, while background type varies along the vertical axis. Darker squares represent target-background pairings with highly accurate subject performance. The number in the center of each square reports the percentage of correct responses averaged across all subjects.

V. Perceptual Colors

In addition to our study of pexels, we have examined methods for choosing multiple individual colors. These experiments were designed to select a set of n colors such that:

1. Any color can be detected preattentively, even in the presence of all the others.
2. The colors are equally distinguishable from one another; that is, every color is equally easy to identify.

We also tested for the maximum number of colors that can be displayed simultaneously, while still satisfying the above requirements. Background research suggested that we needed to consider three separate selection criteria: color distance, linear separation, and color category.

A. Color Distance

Perceptually balanced color models are often used to measure perceived color difference between pairs of colors. Examples include CIE LUV, CIE Lab, Munsell, and the Optical Society of America Uniform Color Space. We used CIE LUV to measure color distance. Colors are specified in this model using three axes: L^*, u^*, and v^*. L^* encodes luminance, while u^* and v^* encode chromaticity (u^* and v^* correspond roughly to the red-green and blue-yellow opponent color channels). CIE LUV provides two important properties. First, colors with the same L^* are isoluminant, that is, they have roughly the same perceived brightness. Second, the Euclidean distance between a pair of colors corresponds roughly to their perceived color difference. Given two colors x and y in CIE LUV space, the perceived difference measured in ΔE^* units is:

$$\Delta E^*_{xy} = \sqrt{(\Delta L^*_{xy})^2 + (\Delta u^*_{xy})^2 + (\Delta v^*_{xy})^2} \qquad (5)$$

Our techniques do not depend on CIE LUV; we could have chosen to use any perceptually balanced color model. We picked CIE LUV in part because it is reasonably well known, and in part because it is recommended by the Commission Internationale de L'Éclairage (CIE) as the appropriate model to use for CRT displays [11].

B. Linear Separation

Results from vision and psychophysics suggest that colors that are linearly separable are significantly easier to

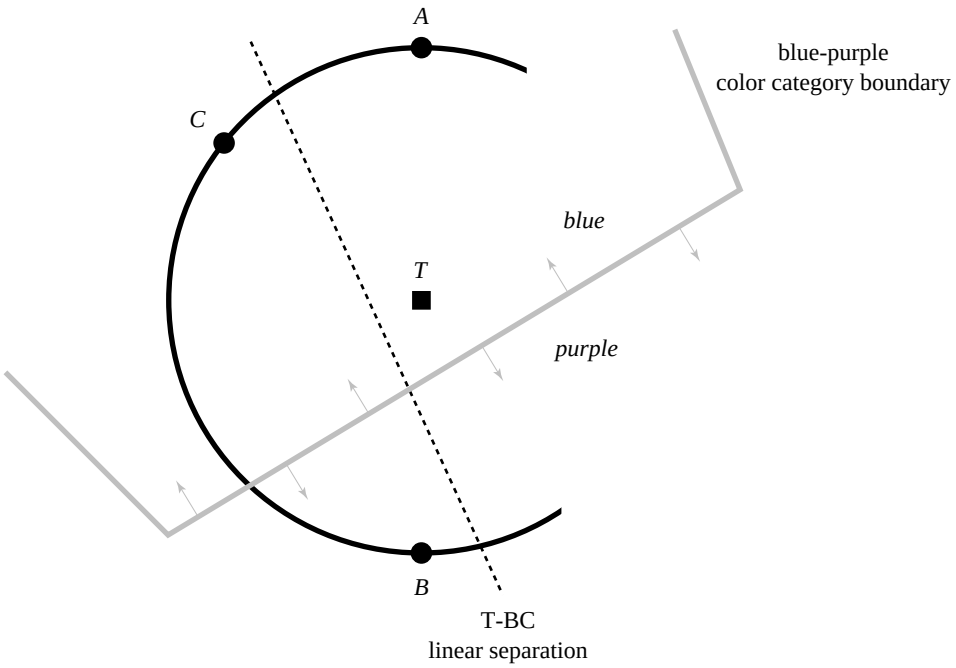


Fig. 8. A small, isoluminant patch within the CIE LUV color model, showing a target color T and three background distracter colors A, B, and C; note that T is collinear with A and B, but can be separated by a straight line from B and C; note also that T, A, and C occupy the "blue" color region, while B occupies the "purple" color region

distinguish from one another. Initial work on this problem was reported in [14]. These results were subsequently confirmed and strengthened by [3], [4] who showed that a perceptually balanced color model could not be used to overcome the linear separation effect.

As an example, consider a target color T and three background distracter colors A, B, and C shown in CIE LUV space in Fig. 8. Since the Euclidean distances TA, TB, and TC are equal, the perceived color difference between T and A, B, and C should also be roughly equal. However, searching for a target element colored T in a sea of background elements colored A and B is significantly more difficult than searching for T in a sea of elements colored B and C. Experimental results suggest that this occurs because T is collinear with A and B, whereas T can be separated by a straight line in color space from B and C. Linear separation increases perceived color difference, even when a perceptual color model is used to try to control that difference.

C. Color Category

Recent work reported by Kawai et al. showed that, during their experiments, the named categories in which people place individual colors can affect perceived color difference [31]. Colors from different named categories have a larger perceived color difference, even when Euclidean distance in a perceptually balanced color model is held constant.

Consider again the target color T and two background distracter colors A and B shown in CIE LUV space in Fig. 8. Notice also that this region of color space has been divided into two named color categories. As before, the Euclidean distances TA and TB are equal, yet finding an element colored T in a sea of background elements colored A is significantly more difficult than finding T in a sea of elements colored B. Kawai et al. suggest this is because both T and A lie within a color category named "blue", while B lies within a different category named "purple". Colors from different named categories have a higher perceived color difference, even when a perceptual color model is used to try to control that difference.

D. Color Selection Experiments

Our first experiment selected colors by controlling color distance and linear separation, but not color category. The reasons for this were twofold. First, traditional methods for subdividing a color space into named color regions are tedious and time-consuming to run. Second, we were not convinced that results from [31] were important for our color selection goals. If problems occurred during our initial experiment, and if those problems could be addressed by controlling color category during color selection, this would both strengthen the results of [31] and highlight their applicability to the general color selection task.

We selected colors from the boundary of a maximum-radius circle embedded in our monitor's gamut. The circle was located on an isoluminant slice through the CIE LUV color model. Previous work reported in [7], [9] showed that a random variation of luminance can interfere with the identification of a boundary between two groups of differently colored elements. Holding the perceived luminance of each color constant guaranteed variations in performance would not be the result of a random luminance effect. Fig. 9 shows an example of selecting five colors about the circumference of the maximum-radius circle inscribed within our

monitor's gamut at $L^* = 61.7$. Since colors are located equidistant around the circle, every color has a constant distance d to its two nearest neighbors, and a constant distance l to the line that separates it from all the other colors.

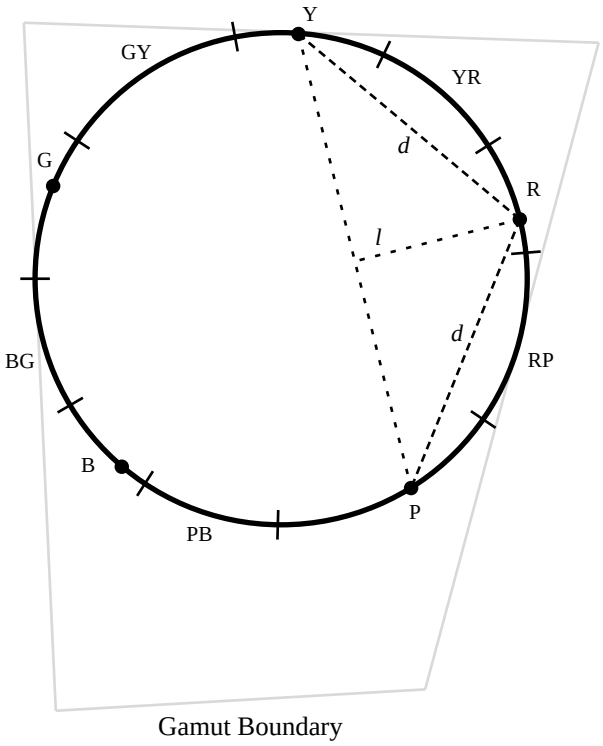


Fig. 9. Choosing colors from the monitor's gamut, the boundary of the gamut at $L^* = 61.7$ represented as a quadrilateral, along with the maximum inscribed circle centered at $(L^*, u^*, v^*) = (67.1, 13.1, -0.98)$, radius $70.5\Delta E^*$; five colors chosen around the circle's circumference; each element has a constant color distance d with its two neighbors, and a constant linear separation l from the remaining (non-target) elements; the circle's circumference has been subdivided into ten named categories, corresponding to the ten hue names from the Munsell color model

We split the experiment into four studies that displayed three, five, seven, and nine colors simultaneously. This allowed us to test for the maximum number of colors we could show while still supporting preattentive identification. Displays in each study were further divided along the following conditions:

- *target color:* each color being displayed was tested as a target, for example, during the three-color study some observers searched for a red target in a sea of green and blue distracters, others search for a blue target in a sea of red and green distracters, and the remainder searched for a green target in a sea of red and blue distracters; asymmetric performance (that is, good performance for some colors and poor performance for others) would indicate that constant distance and separation are not sufficient to guarantee equal perceived color difference, and
- *display size:* experiment displays contained either 17, 33, or 49 elements; any decrease in performance when display size increased would indicate that the search task is not preattentive.

At the beginning of an experiment session observers were asked to search a set of displays for an element with a particular target color. Observers were told that half the displays would contain an element with the target color, and half would not. They were then shown a sequence of experiment displays that contained multiple colored squares randomly located on an underlying 9×9 grid. Each display remained onscreen until the observer indicated via a keypress whether a square with the given target color was present or absent. Observers were told to answer as quickly as possible without making mistakes.

E. Results

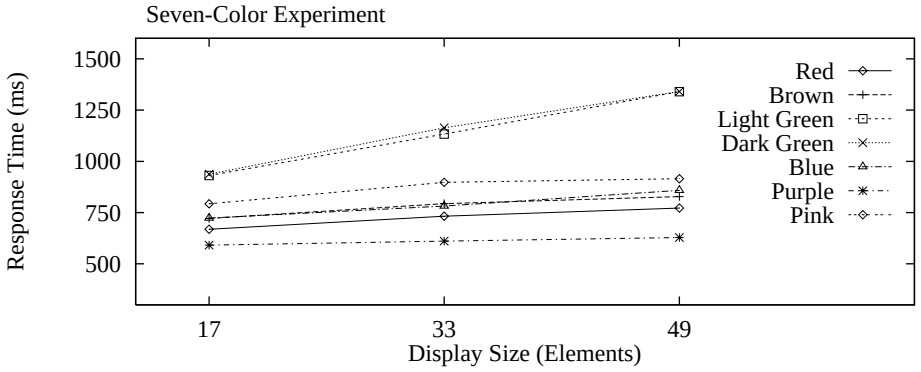
Observers were able to detect all the color targets rapidly and accurately during both the three-color and five-color studies; the average error rate was 2.5%, while the average response times ranged from 459 to 661 ms (response times exceeded the normal preattentive limit of 200 ms because they include the time required for observers to enter their responses on the keyboard). Increasing the display size had no significant effect on response time.

Observers had much more difficulty identifying certain colors during the seven-color (Fig. 10a) and nine-color studies. Response times increased and accuracy decreased during both studies. More importantly, the time required to detect certain colors (*e.g.,* light green and dark green in the seven-color study) was directly proportional to display size. This indicates observers are searching serially through the display to find the target element. Other colors exhibited relatively flat response time curves. These asymmetric results suggest that controlling color distance and linear separation alone is not enough to guarantee a collection of equally distinguishable colors.

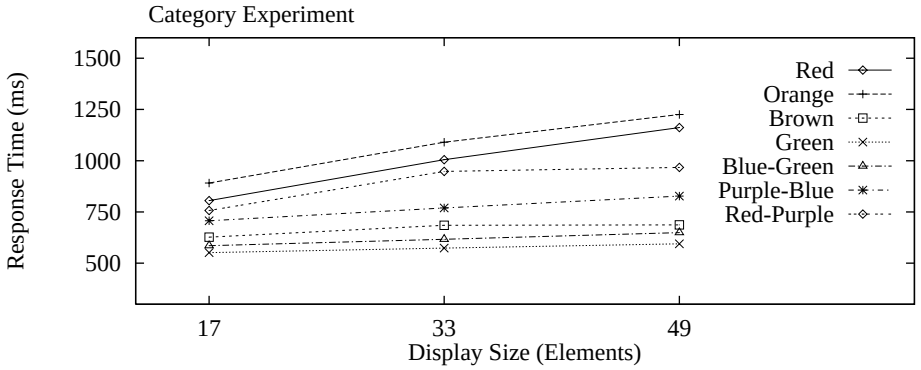
F. Color Category Experiments

We decided to try to determine whether named color categories could be used to explain the inconsistent results from our initial experiment. To do this, we needed to subdivide a color space (in our case, the circumference of our maximum radius circle) into named color regions. Traditional color naming experiments divide the color space into a fine-grained collection of color samples. Observers are then asked to name each of the samples. We chose to use a simpler, faster method designed to measure the amount of overlap between a set of named color regions. Our technique runs in three steps:

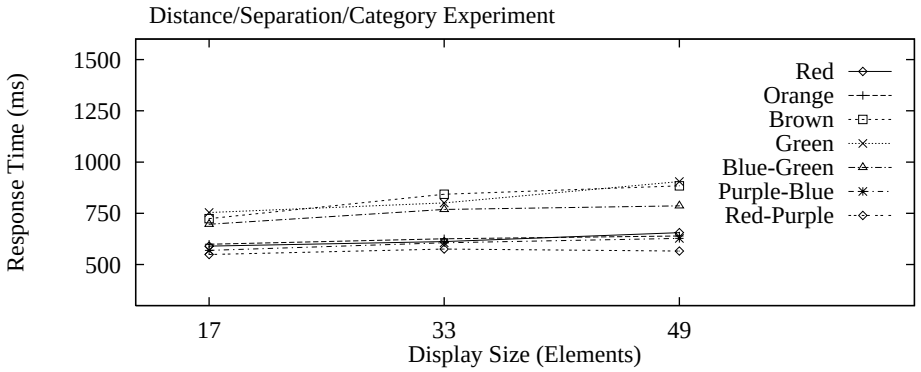
1. The color space is automatically divided into ten named color regions using the Munsell color model. The hue axis of the Munsell model is specified using the ten color names red, yellow-red, yellow, green-yellow, green, blue-green, blue, purple-blue, purple, and red-purple (or R, YR, Y, GY, G, BG, B, PB, P, and RP). Colors are converted to Munsell space, then assigned their hue name within that space (Fig. 9).
2. Representative colors from each of the ten named regions are selected. We chose the color at the center of each region to act as the representative color for that region.



(a)



(b)



(c)

Fig. 10. Graphs showing averaged subject response times for three of the six studies: (a) response time as a function of display size (*i.e.*, total number of elements shown in each display) for each target from the seven-color study; (b) response times for each target from the color category study; (c) response times for each target from the combined distance-separation-category study

3. Observers are asked to name each of the representative colors. The amount of overlap between the names chosen for the representative colors for each region defines the amount of "category overlap" that exists between the regions.

Consider Table I, which lists the percentage of observers who selected a particular name for six of the representative colors. For example, the table shows that representative colors from P and R overlap only at the "pink" name. Their overlap is not that strong, since neither P nor R are strongly classified as pink. The amount of overlap is computed by multiplying the percentages for the common name, giving a P-R overlap of $5.2\% \times 26.3\% = 0.014$. A closer correspondence of user-chosen names for a pair of regions results in a stronger category similarity. For example, nearly all observers named the representative colors from the G and GY regions as "green". This resulted in an overlap of 0.973. Representative colors that overlap over multiple names are combined using addition, for example, YR and Y overlapped in both the "orange" and "brown"

TABLE I
A LIST OF SIX REPRESENTATIVE COLORS FOR THE COLOR REGIONS PURPLE, RED, YELLOW-RED, YELLOW, GREEN-YELLOW, AND GREEN, AND THE PERCENTAGE OF OBSERVERS WHO CHOSE A PARTICULAR NAME FOR EACH REPRESENTATIVE COLOR

	purple	*magenta*	*pink*	*red*	*orange*	*brown*	*yellow*	*green*
P	86.9%	2.6%	5.2%					
R			26.3%	71.0%				
YR				5.3%	86.8%	7.9%		
Y					2.6%	44.7%	47.4%	
GY								97.3%
G								100.0%

names, resulting in a YR-Y overlap of (86.8% × 2.6%) + (7.9% × 44.7%) = 0.058.

G. Color Category Results

When we compared the category overlap values against results from our seven and nine-color studies, we found that the amount of overlap between the target color and its background distracters provided a strong indication of performance. Colors that worked well as targets had low category overlap with all of their distracter colors. Colors that worked poorly had higher overlap with one or more of their distracter colors. A measure of rank performance to total category overlap produced correlation values of 0.821 and 0.762 for the seven and nine-color studies, respectively. This suggests that our measure of category overlap is a direct predictor of subject performance. Low category overlap between the target color and all of its background distracters produces relatively rapid subject performance. High category overlap between the target color and one or more background distracters results in relatively slow subject performance.

These results might suggest that color category alone can be used to choose a set of equally distinguishable colors. To test this, we selected seven new colors that all had low category overlap with one another, then reran the experiments. Results from this new set of colors were as poor as the original seven-color study (Fig. 10b). The seven new colors were located at the centers of their named categories, so their distances and linear separations varied. The colors with the longest response times had the smallest distances and separations. This suggests that we need to maintain at least a minimum amount of distance and separation to guarantee acceptable identification performance.

In our last experiment, we chose a final set of seven colors that tried to satisfy all three selection criteria. The categories in which the colors were located all had low overlap with one another. Colors were shifted within their categories to provide as large a distance and linear separation as possible. We also tried to maintain constant distances and linear separations for all the colors. Results from this final experiment were encouraging (Fig. 10c). Response times for each of the colors acting as a target were similar, with little or no effect from increased display size. The mean response error was also significantly lower than during the previous two seven-color experiments. We concluded that up to seven isoluminant colors can be displayed simultaneously while still allowing for rapid and accurate identification, but only if the colors satisfy proper color distance, linear separation, and color category guidelines.

VI. Combining Texture and Color

Previous work in our laboratory focused on selecting perceptual textures and colors in isolation. Clearly, we would like to use multicolored pexels during visualization. The ability to combine both features effectively would increase the number of attributes we can visualize simultaneously. Results in the literature are mixed on how this might be achieved. Some researchers have reported that task irrelevant variation in color has no effect on texture discrimination [51], [58], while others have found exactly this kind of interference [8], [9], [49]. Moreover, we are not aware of any studies that address whether there is interference from random variation in texture properties when discrimination is based on color. Experiments are therefore needed that examine possible interference effects in both directions, that is, effects of color variation on texture discrimination and effects of texture variation on color discrimination.

A. Experiments

In order to investigate these issues, we designed a new set of psychophysical experiments. Our two specific questions were:

1. Does random variation in pexel color influence the detection of a region of target pexels defined by height or density?
2. Does random variation in pexel height or density influence the detection of a region of target pexels defined by color?

We chose to ignore regularity, since it performed poorly as a target defining property during all phases of our original texture experiments [23], [25]. We chose three different colors using our perceptual color selection technique [22], [23]. Colors were initially selected in the CIE LUV color space, then converted to our monitor's RGB gamut. The three colors corresponded approximately to red (monitor RGB = 246, 73, 50), green (monitor RGB = 49, 144, 21) and blue (monitor RGB = 82, 109, 168). Our new experiments were constructed around a set of conditions similar to those used during the original texture experiments.

For color targets, we varied:

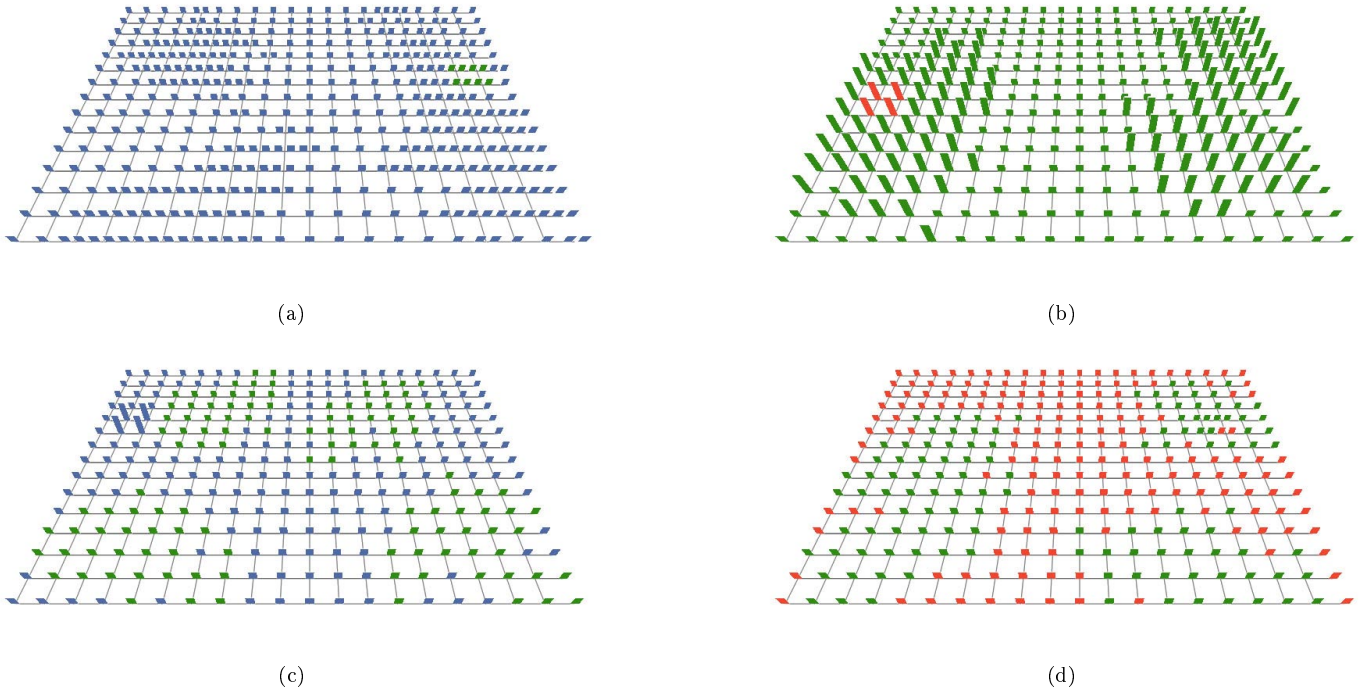


Fig. 11. Four displays from the combined color-texture experiments, printed colors may not match exactly on-screen colors used during our experiments: (a) a green target in a sea of blue pexels with background density variation; (b) a red target in a sea of green pexels with background height variation; (c) a tall target with background blue-green color variation; (d) a dense target with background green-red color variation

- *target-background pairing:* some displays contained a green target region in a sea of blue pexels, while others contained a red target region in a sea of green pexels (Fig. 11a and 11b); two different pairings were used to increase the generality of the results,
- *secondary dimension:* displays contained either no background variation (*e.g.,* every pexel was sparse and short), a random variation of density across the array, or a random variation of height across the array; this allowed us to test for interference from two different texture dimensions during target detection based on color,
- *exposure duration:* displays were shown for either 50, 150, or 450 ms; this allowed us to see how detection accuracy was influenced by the exposure duration of the display, and
- *target patch size:* target regions were either 2×2 pexels or 4×4 pexels in size. This allowed us to examine the influence of all the foregoing factors at both relatively difficult (2×2) and easy (4×4) levels of target detectability.

Two texture dimensions (height and density) were studied, and each involved two different target types: taller and shorter (for height) and denser and sparser (for density). For each type of target, we designed an experiment that tested a similar set of conditions. For example, in the taller experiment we varied:

- *target-background pairing:* half the displays contained a target region of medium pexels in a sea of short pexels, while the other half contained a target region of tall pexels in a sea of medium pexels; two different pairings were used to increase the generality of the results,
- *secondary dimension:* the displays contained pexels that were either a constant gray or that varied randomly between two colors; when color was varied, half the displays contained blue and green pexels, while the other half of the displays contained green and red pexels (Fig. 11c),
- *exposure duration:* displays were shown for 50, 150, or 450 ms, and
- *target patch size:* target groups were either 2×2 pexels or 4×4 pexels in size.

Fig. 11 shows examples of four experiment displays. Fig. 11a and 11b contain a green target in a sea of blue pexels, and a red target in a sea of green pexels, respectively. Density varies in the background in Fig. 11a, while height varies in Fig. 11b. Fig. 11c contains a tall target with a blue-green background color pattern. Fig. 11d contains a dense target with a green-red background color pattern. Any background variation that is present can pass through a target. This occurs in Fig. 11d, where part of the target is red and part is green. Note also that, as described for Fig. 5, the number of paper strips in an individual pexel depends on its density.

The colors we used during our experiments were chosen in CIE LUV color space. A simple set of formulas can be used to convert from CIE LUV to CIE XYZ (a standard device-independent color model), and from there to our monitor's color gamut. To move from LUV to XYZ:

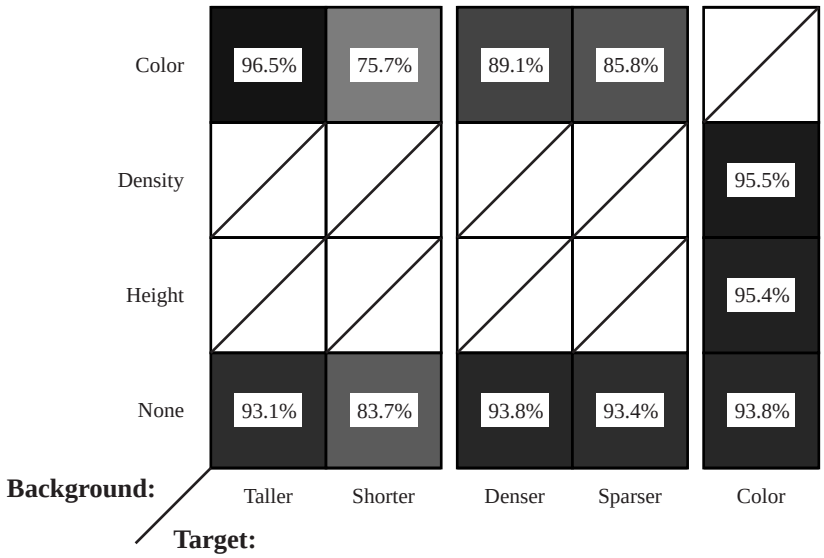


Fig. 12. A table showing the percentage of correct responses for each target-background pairing; target type along the horizontal axis, background type along the vertical axis; darker squares represent pairings with a high percentage of correct responses; results for taller, shorter, denser, and sparser with no background variation are from the original texture experiments; blank entries with diagonal slashes indicate target-background pairings that did not exist during the combined color-texture experiments

$$\begin{aligned} Y &= \left(\frac{L^* + 16}{116}\right)^3 Y_w \\ X &= \frac{9u'}{4v'}Y \\ Z &= \frac{3}{v'}Y - 5Y - \frac{3u'}{4v'}Y \end{aligned} \tag{6}$$

where L^*, u', and v' are used to specify a color in CIE LUV (u' and v' are simple respecifications of u^* and v^*), and Y_w represents the luminance of a reference white point. We then built a conversion matrix to map colors from CIE XYZ into our monitor's color gamut. This was done by obtaining the chromaticities of our monitor's red, green, and blue triads, then measuring the luminance of the monitor's maximum intensity red, green, and blue with a spot photometer. These values are needed to convert colors from a device-independent space (*i.e.,* CIE XYZ) into device-dependent coordinates (*i.e.,* our monitors RGB color space). All of our experiments were displayed on a Sony Trinitron monitor with CIE XYZ chromaticities $(x_r, y_r) = (0.625, 0.340)$, $(x_g, y_g) = (0.280, 0.595)$, and $(x_b, y_b) = (0.155, 0.070)$. The luminances of maximum intensity red, green, and blue were $Y_r = 5.5$, $Y_g = 16.6$, $Y_b = 2.8$. This produced an XYZ to monitor RGB conversion matrix of:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 0.131 & 0.057 & 0.021 \\ -0.044 & 0.081 & 0.002 \\ 0.003 & 0.008 & 0.033 \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{7}$$

For a complete description of how the conversion formulas are built, we refer the reader to any of [17], [18], [60].

Ten users participated as observers in each of the two color and four texture experiments. Each observer had normal or corrected acuity. Observers who completed the color experiments were also tested for color blindness [12]. Observers were provided with an opportunity to practice before each experiment. This helped them become familiar with the task and the duration of the displays. Before each testing session began, observers were told that half the displays would contain a target, and half would not. We used a Macintosh computer with an 8-bit color lookup table to run our experiments. Responses (either "target present" or "target absent") for each display an observer was shown were recorded for later analysis.

B. Results

Mean percentage target detection accuracy was the measure of performance. Observer responses were collected, averaged, and analyzed using multi-factor ANOVA. In summary, we found:

1. Color targets were detected rapidly (*i.e.,* at 150 ms) with very high accuracy (96%). Background variation in height and density produced no interference effects in this detection task.
2. Detection accuracy for targets defined by density or height were very similar to results from our original texture experiments [23], [25]. When there was no background variation in color, excellent detection accuracy was obtained for density defined targets (*i.e.,* denser and sparser targets) at 150 ms (94%). Height defined targets (*i.e.,* taller and shorter) were detected somewhat less accurately at 150 ms (88%) but were highly detectable at 450 ms (93%). As we had also found previously, taller targets were generally easier to detect than shorter targets, and denser targets were easier than sparser targets.
3. In all four texture experiments, background variation in color produced a small but significant interference effect, averaging 6% in overall accuracy reduction.

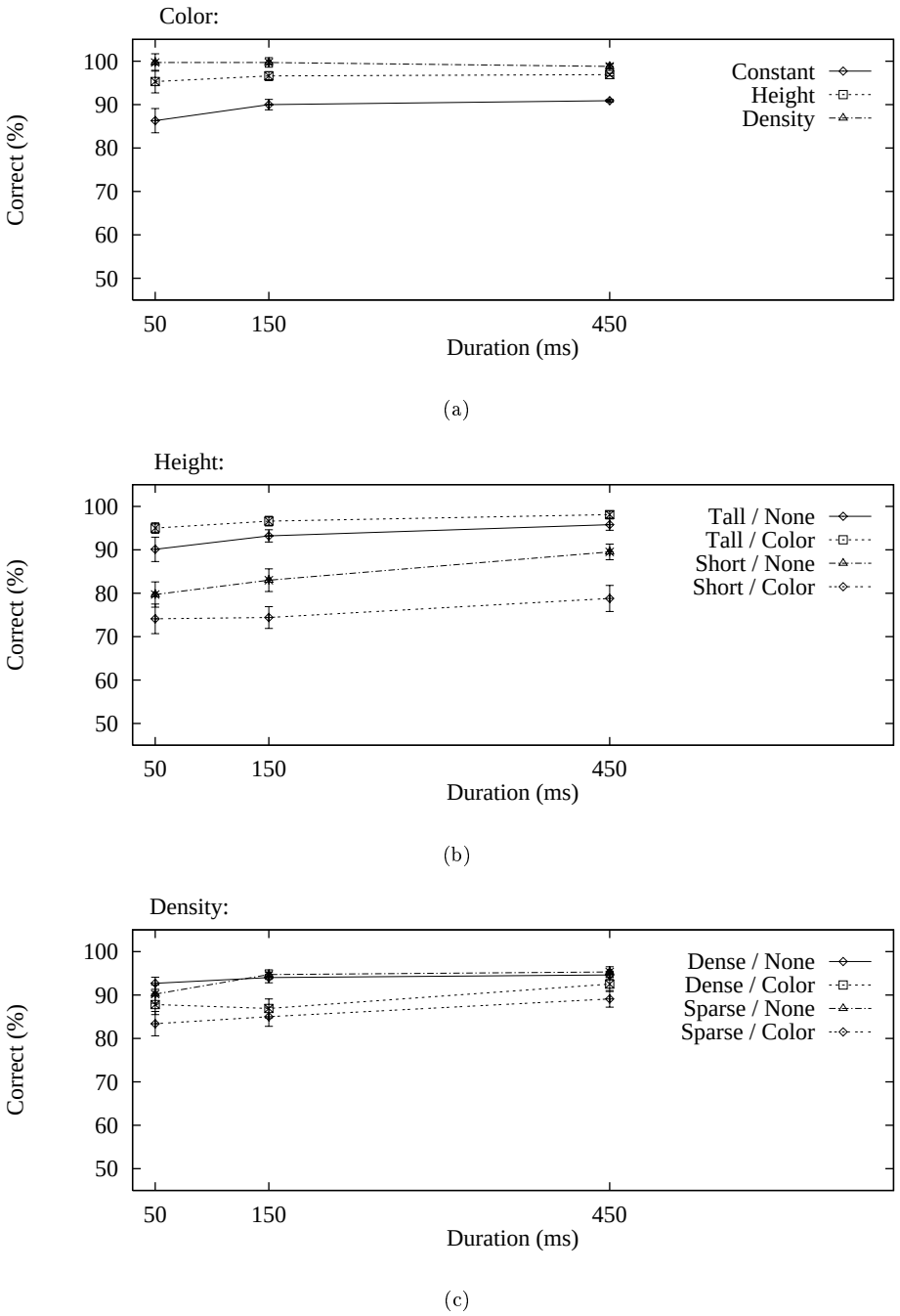


Fig. 13. Graphs showing averaged subject results for color, height, and density trials: (a) results for color trials, horizontal axis plots exposure duration, vertical axis plots percentage of correct responses, each line corresponds to one of the three different background conditions (no variation, height variation, or density variation); (b) results for height trials; (c) results for density trials

4. The absolute reduction in accuracy due to color interference depended on the difficulty of the detection task. Specifically, color interfered more with the less visible target values (shorter and sparser targets yielded a mean accuracy reduction of 8%) than with the more visible targets (taller and denser targets yield a mean accuracy reduction of 4%).

Fig. 12 shows average subject performance as a table representing each target-background pair. Target type varies along the horizontal axis, while background type varies along the vertical axis. Darker squares represent target-background pairings with highly accurate subject performance. The number in the center of each square reports the percentage of correct responses averaged across all subjects.

Target regions defined by a particular pexel color were identified rapidly and accurately in all cases. At a 150 ms exposure duration mean accuracy was approximately 96%. The small increase in accuracy from shorter to longer exposure durations was significant, $F(2,36) = 41.03, p < .001$.

However, variation in the background height or density of pexels caused no significant reduction in performance (mean accuracy of 95.3% for constant background, 96.6% for varying height, and 96.9% for varying density; see also the graph in Fig. 13a). In fact, the graphs in Figure 13a report that absolute performance was slightly better for conditions with background variations of height or density. We suspect that geometric regularity in the texture pattern may produce a gestalt or configurational effect that interferes with target detection based on color. If so, this would be similar to previous reports in the psychophysical literature [2] showing inhibitory effects of gestalt grouping on target detection.

Detection accuracy for targets defined by texture properties were very similar to results from our previous texture experiments [22], [23]. Both kinds of targets benefited from longer exposure durations (density, $F(2,58) = 9.24, p < .001$; height, $F(2,58) = 10.66, p < .001$), with small but significant increases in accuracy with each increase in duration. With regard to the four kinds of targets, denser and taller target regions were easiest to identify, followed by sparser and shorter target regions (Fig. 13b and 13c). However, only the difference between taller versus shorter targets was statistically significant, $F(1,29) = 67.14, p < .001$. These effects were not unexpected, since they have been reported in other psychophysical studies [1], [51]. In the target present displays, accuracy for shorter targets seemed to be compromised even more than usual because of occlusion: a group of shorter pexels was often partially occluded by a group of taller pexels placed in front of them. A group of taller pexels, on the other hand, tended to stand out among the shorter pexels that surrounded them. Sparser targets suffer from a different problem: the need for a minimum amount of physical space to become perceptually salient. Since dense targets "add information to" their target region, rather than "take information away", they were less susceptible to this problem. This asymmetry contributed to a significant target type by region size interaction, $F(1,29) = 11.14, p < .01$. This was reflected in a dramatic reduction in the performance gap between dense and sparse targets when 2×2 and 4×4 target patches are compared. In displays with 2×2 target regions and background color variation, dense targets outperform sparse targets by approximately 7%. For 4×4 target regions, however, dense and sparse displays were nearly equal in accuracy (less than 1% difference).

For targets defined by texture, random color variation tended to interfere with detection, causing accuracy to be lower for both denser and sparser targets in the density displays ($F(1,29) = 9.12, p < .01$) and by interacting with target type in the case of height ($F(1,29) = 10.61, p < .01$, see also Fig. 13b and 13c). This interaction resulted from color variation having a greater influence on accuracy for short targets ($F(1,15) = 6.73, p < .03$), which were generally more difficult to see, than for tall targets, which were detected with uniformly high accuracy (greater than 90%). These results suggest that color interference can be limited when color and texture are combined, but only in cases where the detection task is relatively effortless prior to the addition of color variation. As can be seen in Fig. 13b and 13c, the interference effect of color variation tends to be greatest when the target detection task is most difficult.

Several other miscellaneous effects were worthy of note. Detection accuracy was generally higher on displays with a target present than when no target was present (color, $F(1,18) = 37.32, p < .001$; density, $F(1,29) = 5.09, p < .04$; height, $F(1,29) = 6.64, p < .02$). This was a small difference overall (an average of 4%) but it reflected a slight bias on the part of users to guess "target present" when they were uncertain what they had seen. Large target regions (4×4) were generally easier to identify than small regions (2×2) (color, $F(1,18) = 15.38, p < .001$; density, $F(1,29) = 94.24, p < .001$; height, $F(1,29) = 24.78, p < .001$), due to the greater visibility associated with a larger target region.

Taken together, these results are consistent with studies based on textures arrayed in a two-dimensional plane and reported in the psychophysical literature. As described by [49], we found that color produces a small but statistically reliable interference effect during texture segmentation. Moreover, we found color and texture form a "feature hierarchy" that produces asymmetric interference: color variation interferes with an observer's ability to see texture regions based on height or density, but variation in texture has no effect on region detection based on color. This is similar to reports by [8], [9], who reported asymmetric color on shape interference in a boundary detection task involving two-dimensional textures.

VII. Practical Applications

Although our theoretical results provide a solid design foundation, it is equally important to ensure that these results can be applied to real-world data. Our initial goal was a technique for visualizing multivariate data on an underlying height field. We decided to test our perceptual visualization technique by analyzing environmental conditions on a topographic map. Specifically, we visualized typhoons in the Northwest Pacific Ocean during the summer and fall of 1997.

A. Visualizing Typhoons

The names "typhoon" and "hurricane" are region-specific, and refer to the same type of weather phenomena: an atmospheric disturbance characterized by low pressure, thunderstorm activity, and a cyclic wind pattern. Storms of this type with windspeeds below 17m/s are called "tropical depressions". When windspeeds exceed 17m/s, they become "tropical storms". This is also when storms are assigned a specific name. When windspeeds reach 33m/s, a storm becomes a typhoon (in the Northwest Pacific) or a hurricane (in the Northeast Pacific and North Atlantic).

We combined information from a number of different sources to collect the data that we needed. A U.S. Navy elevation dataset[1] was used to obtain land elevations at

[1] http://grid2.cr.usgs.gov/dem/

ten minute latitude and longitude intervals. Land-based weather station readings collected from around the world and archived by the National Climatic Data Center[2] provided daily measurements for eighteen separate environmental conditions. Finally, satellite archives made available by the Global Hydrology and Climate Center[3] contained daily open-ocean windspeed measurements at thirty minute latitude and longitude intervals. The National Climatic Data Center defined the 1997 typhoon season to run from August 1 to October 31. Each of our datasets contained measurements for this time period.

We chose to visualize three environmental conditions related to typhoons: windspeed, pressure, and precipitation. All three values were measured on a daily basis at each land-based weather station, but only daily windspeeds were available for open-ocean positions. In spite of the missing open-ocean pressure and precipitation, we were able to track storms as they moved across the Northwest Pacific Ocean. When the storms made landfall the associated windspeed, sea-level pressure, and precipitation were provided by weather stations along their path.

Based on our experimental results, we chose to represent windspeed, pressure, and precipitation with height, density, and color, respectively. Localized areas of high windspeed are obvious indicators of storm activity. We chose to map increasing windspeed to an increased pexel height. Although our experimental results showed statistically significant interference from background color variation, the absolute effect was very small. Taller and denser pexels were easily identified in all other cases, suggesting there should be no changes in color interference due to an increase in task difficulty. Windspeed has two important boundaries: 17m/s (where tropical depressions become tropical storms) and 33m/s (where storms become typhoons). We mirrored these boundaries with height discontinuities. Pexel height increases linearly from 0-17m/s. At 17m/s, height approximately doubles, then continues linearly from 17-33m/s. At 33m/s another height discontinuity is introduced, followed by a linear increase for any windspeeds over 33m/s.

Pressure is represented with pexel density. Since our results showed it was easier to find dense pexels in a sea of sparse pexels (as opposed to sparse in dense), an increase in pressure is mapped to a decrease in pexel density (*i.e.,* very dense pexels represent the low pressure regions associated with typhoons). Three different texture densities were used to represent three pressure ranges. Pressure readings less than 996 millibars, between 996 and 1014 millibars, and greater than 1014 millibars produce very dense, dense, and sparse pexels, respectively.

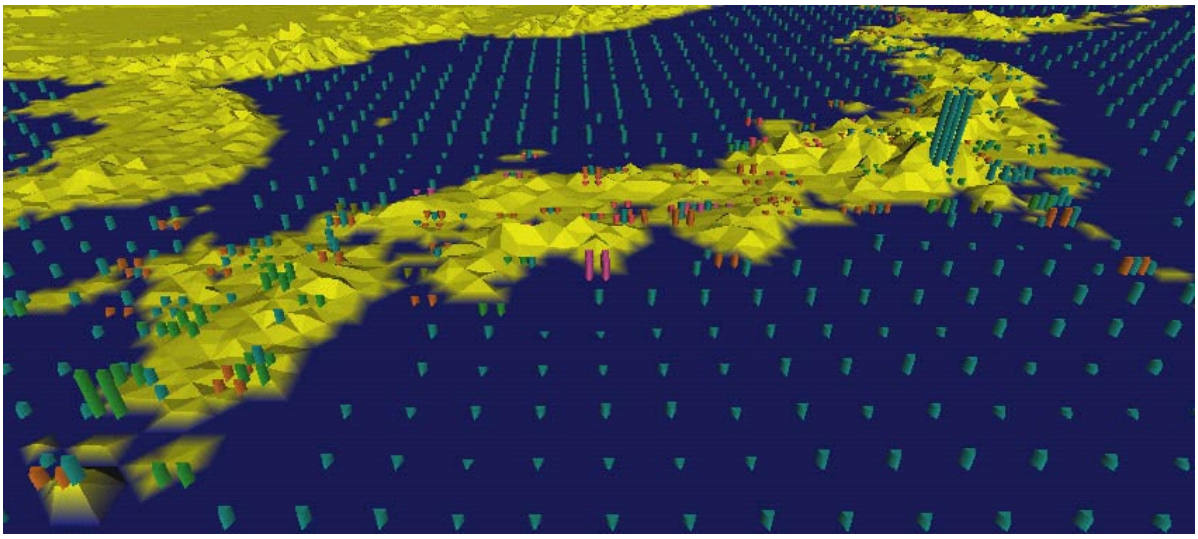
Precipitation is represented with color. We used our perceptual color selection technique to choose five perceptually uniform colors. Daily precipitation readings of zero, 0–0.03 inches, 0.03–0.4 inches, 0.4–1.0 inches, and 1.0–10.71 inches were colored green, yellow, orange, red, and purple, respectively (each precipitation range had an equal number of entries in our typhoon dataset). Pexels on the open ocean or at weather stations where no precipitation values were reported were colored blue-green. Our experimental results showed no texture-on-color interference. Moreover, our color selection technique is designed to produce colors that are equally distinguishable from one another. Our mapping uses red and purple to highlight the high-precipitation areas associated with typhoon activity.

We should note that our data-feature mapping is designed to allow viewers to rapidly and accurately identify and track the locations of storms and typhoons as spatial collections of tall, dense, red and purple pexels. Our visualization system is not meant to allow users to determine exact values of windspeed, pressure, and precipitation from an individual pexel. However, knowing the range of values that produce certain types of height, density, and color will allow a viewer to estimate the environmental conditions at a given spatial location.

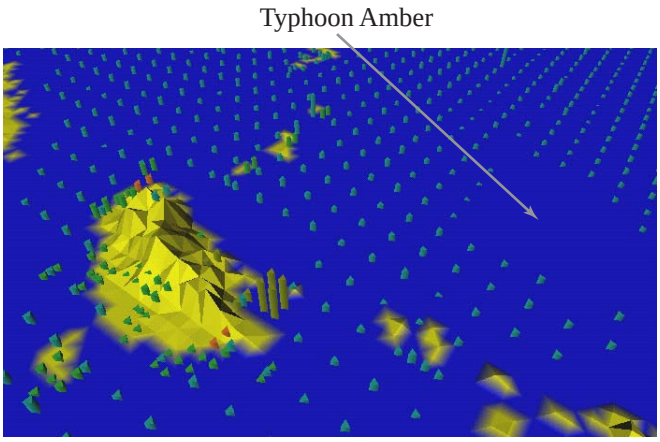
We built a simple visualization tool that maps windspeed, pressure, and precipitation to their corresponding height, density, and color. Our visualization tool allows a user to move forwards and backwards through the dataset day-by-day. One interesting result was immediately evident when we began our analysis: typhoon activity was not represented by high windspeed values in our open-ocean dataset. Typhoons normally contain severe rain and thunderstorms. The high levels of cloud-based water vapor produced by these storms block the satellites that are used to measure open-ocean windspeeds. The result is an absence of any windspeed values within a typhoon's spatial extent. Rather than appearing as a local region of high windspeeds, typhoons on the open-ocean are displayed as a "hole", an ocean region without any windspeed readings (see Fig. 14b and 14d). This absence of a visual feature (*i.e.,* a hole in the texture field) is large enough to be salient in our displays, and can be preattentively identified and tracked over time. Therefore, users have little difficulty finding storms and watching them as they move across the open ocean. When a storm makes landfall, the weather stations along the storm's path provide the proper windspeed, as well as pressure and precipitation. Weather stations measure windspeed directly, rather than using satellite images, so high levels of cloud-based water vapor cause no loss of information.

Fig. 14 shows windspeed, pressure, and precipitation around Japan, Korea, and Taiwan during August 1997. Fig. 14a looks north, and displays normal summer conditions across Japan on August 7, 1997. Fig. 14b, looking northeast, tracks typhoon Amber (one of the region's major typhoons) approaching along an east to west path across the Northwest Pacific Ocean on August 27, 1997. Fig. 14c shows typhoon Amber one day later as it moves through Taiwan. Weather stations within the typhoon show the expected strong winds, low pressure, and high levels of rainfall. These results are easily identified as tall, dense, red and purple pexels. Compare these images to Fig. 14d and 14e, where windspeed was mapped to regularity, pressure to height, and precipitation to density (a mapping without color that our original texture experiments predict will

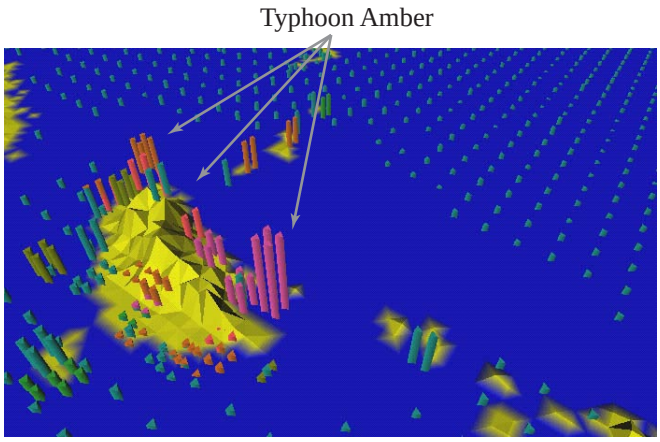
[2] http://www.ncdc.noaa.gov/ol/climate/online/gsod.html
[3] http://ghrc.msfc.nasa.gov/ghrc/list.html



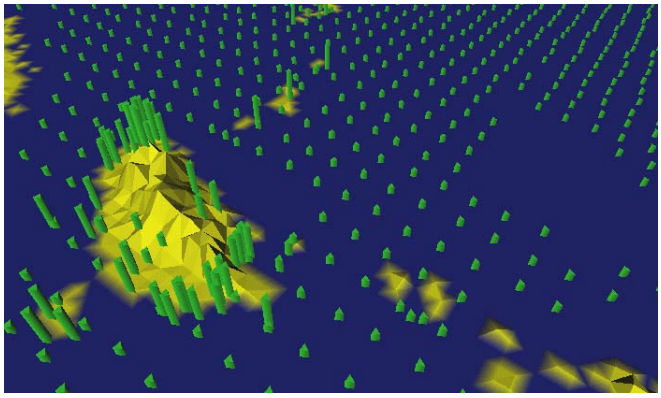
(a)



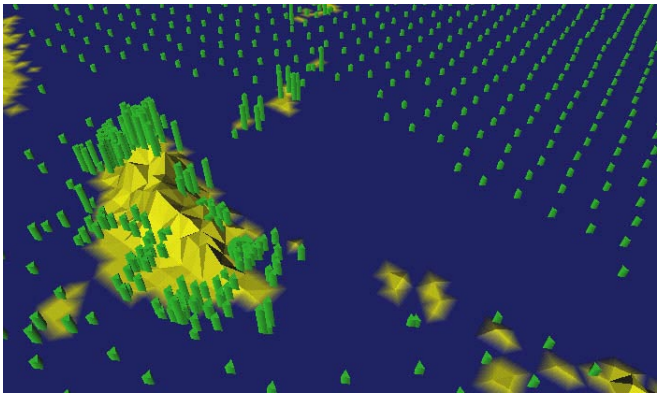
(b)



(c)



(d)



(e)

Fig. 14. Typhoon conditions across Southeast Asia during the summer of 1997: (a) August 7, 1997, normal weather conditions over Japan; (b) August 27, 1997, typhoon Amber approaches the island of Taiwan from the southeast; (c) August 28, 1997, typhoon Amber strikes Taiwan, producing tall, dense pexels colored orange, red, and purple (representing high precipitation); (d, e) the same data as in (b) and (c) but with windspeed represented by regularity, pressure by height, and precipitation by density

perform poorly). Although viewers can identify areas of lower and higher windspeed (*e.g.,* on the open ocean and over Taiwan), it is difficult to identify *a change* in lower or higher windspeeds (e.g., the change in windspeed as typhoon Amber moves onshore over Taiwan). In fact, viewers often searched for an increase in density that represents an increase in precipitation, rather than an increase in irregularity; pexels over Taiwan become noticeably denser between Fig. 14d and 14e.

VIII. Conclusions and Future Work

This paper describes a method for combining perceptual textures and colors for multivariate data visualization. Our pexels are built by varying three perceptual texture dimensions: height, density, and regularity. Our perceptual colors are selected by controlling the color distance, linear separation, and color category of each color. Both experimental and real-world results showed that colored pexels can be used to rapidly, accurately, and effortlessly analyze large, multi-element displays. Care must be taken, however, to ensure that the data-feature mapping builds upon the fundamental workings of the low-level human visual system. An ad-hoc mapping will often introduce visual artifacts that actively interfere with a user's ability to perform their visual analysis task. Our initial texture experiments showed that taller, shorter, denser, and sparser pexels can be easily identified, but that certain background patterns must be avoided to ensure accurate performance. During our color selection experiments we found that color distance, linear separation, and color category must all be considered to ensure a collection of equally distinguishable colors. New results on the combined use of texture and color showed that background color variation causes a small but statistically significant interference effect during a search for targets based on height or density. The size of the effect is directly related to the difficulty of the visual analysis task; tasks that are more difficult result in more color interference. Variation of height and density, on the other hand, had no effect on identifying color targets. These results are similar to reports in the psychophysical literature [8], [9], [49], although to our knowledge no one has studied perceptual textures and colors displayed in 3D using perspective projections.

Our results were further validated when we applied them to real-world applications like typhoon visualization. Our tools were designed to satisfy findings from our experiments. For example, attributes were mapped in order of importance to height, density, and color. In cases where an subject analyzed height or density patterns, we tried to ensure an effortless search task (*i.e.,* looking for taller or denser rather than shorter or sparser) to minimize any color on texture interference that might occur.

One important area of future work is a comparison of our visualization techniques against other methods that might be used to represent information in our real-world applications. For example, it would be useful to test a user's ability to track storm activity in our visualization environment against other standard techniques for representing weather activity. Although we have yet to conducted these kinds of practical experiments, we hope to initiate them in the near future as part of our perceptual visualization studies.

We are now working to integrate our colored pexels with other visual features. One candidate is orientation; in fact, our pexels were initially designed to "stand up" off the underlying height field to support variation of orientation. Another visual property with significant potential is apparent motion. This technique can be used to make individual strips in a pexel "walk" within their spatial extent. It may be possible to tie direction and speed of motion to two underlying attribute values, thereby increasing the dimensionality of our visualization techniques. We are designing experiments to investigate the effectiveness of each of these features for encoding information. We will also study any interactions that occur when multiple texture, color, orientation, and motion dimensions are displayed simultaneously.

Acknowledgments

We would like to thank the National Climatic Data Center, and Sherry Harrison and the Global Hydrology Resource Center for generously providing typhoon-related weather data. We would also like to thank Jeanette Lum for coordinating and running our experiment sessions. This research was funded in part by the National Science and Engineering Research Council of Canada, and by the Office of Naval Research (Grant N00014-96-1120) and the Ballistic Missile Defense Organization through the Multiuniversity Research Initiative.

References

[1] Aks, D. J., and Enns, J. T. Visual search for size is influenced by a background texture gradient. *Journal of Experimental Psychology: Perception and Performance 22*, 6 (1996), 1467–1481.

[2] Banks, W. P., and Prinzmetal, W. Configurational effects in visual information processing. *Perception & Psychophysics 19* (1976), 361–367.

[3] Bauer, B., Jolicoeur, P., and Cowan, W. B. Visual search for colour targets that are or are not linearly-separable from distractors. *Vision Research 36* (1996), 1439–1446.

[4] Bauer, B., Jolicoeur, P., and Cowan, W. B. The linear separability effect in color visual search: Ruling out the additive color hypothesis. *Perception & Psychophysics 60*, 6 (1998), 1083–1093.

[5] Bergman, L. D., Rogowitz, B. E., and Treinish, L. A. A rule-based tool for assisting colormap selection. In *Proceedings Visualization '95* (Atlanta, Georgia, 1995), pp. 118–125.

[6] Bruckner, L. A. On Chernoff faces. In *Graphical Representation of Multivariate Data*, P. C. C. Wang, Ed. Academic Press, New York, New York, 1978, pp. 93–121.

[7] Callaghan, T. C. Dimensional interaction of hue and brightness in preattentive field segregation. *Perception & Psychophysics 36*, 1 (1984), 25–34.

[8] Callaghan, T. C. Interference and domination in texture segregation: Hue, geometric form, and line orientation. *Perception & Psychophysics 46*, 4 (1989), 299–311.

[9] Callaghan, T. C. Interference and dominance in texture segregation. In *Visual Search*, D. Brogan, Ed. Taylor & Francis, New York, New York, 1990, pp. 81–87.

[10] Chernoff, H. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association 68*, 342 (1973), 361–367.

[11] CIE. *CIE Publication No. 15, Supplement Number 2 (E-1.3.1): Official Recommendations on Uniform Color Spaces, Color-Difference Equations, and Metric Color Terms.* Commission Internationale de L'Èclairge, 1976.

[12] Coren, S., and Hakstian, A. R. Color vision screening without the use of technical equipment: Scale development and cross-validation. *Perception & Psychophysics 43* (1988), 115–120.

[13] Cutting, J. E., and Millard, R. T. Three gradients and the perception of flat and curved surfaces. *Journal of Experimental Psychology: General 113*, 2 (1984), 198–216.

[14] D'Zmura, M. Color in visual search. *Vision Research 31*, 6 (1991), 951–966.

[15] Egeth, H. E., and Yantis, S. Visual attention: Control, representation, and time course. *Annual Review of Psychology 48* (1997), 269–297.

[16] Foley, J., and Ribarsky, W. Next-generation data visualization tools. In *Scientific Visualization: Advances and Challenges*, L. Rosenblum, Ed. Academic Press, San Diego, California, 1994, pp. 103–127.

[17] Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. *Computer Graphics: Principles and Practice.* Addison-Wesley Publishing Company, Reading, Massachusetts, 1990.

[18] Glassner, A. S. *Principles of Digital Image Synthesis.* Morgan Kaufmann Publishers, Inc., San Francisco, California, 1995.

[19] Grinstein, G., Pickett, R., and Williams, M. EXVIS: An exploratory data visualization environment. In *Proceedings Graphics Interface '89* (London, Canada, 1989), pp. 254–261.

[20] Hallett, P. E. Segregation of mesh-derived textures evaluated by resistance to added disorder. *Vision Research 32*, 10 (1992), 1899–1911.

[21] Haralick, R. M., Shanmugam, K., and Dinstein, I. Textural features for image classification. *IEEE Transactions on System, Man, and Cybernetics SMC-3*, 6 (1973), 610–621.

[22] Healey, C. G. Choosing effective colours for data visualization. In *Proceedings Visualization '96* (San Francisco, California, 1996), pp. 263–270.

[23] Healey, C. G. Building a perceptual visualisation architecture. *Behaviour and Information Technology (in press)* (1998).

[24] Healey, C. G., Booth, K. S., and Enns, J. T. Real-time multivariate data visualization using preattentive processing. *ACM Transactions on Modeling and Computer Simulation 5*, 3 (1995), 190–221.

[25] Healey, C. G., and Enns, J. T. Building perceptual textures to visualize multidimensional datasets. In *Proceedings Visualization '98* (Research Triangle Park, North Carolina, 1998), pp. 111–118.

[26] Interrante, V. Illustrating surface shape in volume data via principle direction-driven 3d line integral convolution. In *SIGGRAPH 97 Conference Proceedings* (Los Angeles, California, 1997), T. Whitted, Ed., pp. 109–116.

[27] Julész, B. Textons, the elements of texture perception, and their interactions. *Nature 290* (1981), 91–97.

[28] Julész, B. A theory of preattentive texture discrimination based on first-order statistics of textons. *Biological Cybernetics 41* (1981), 131–138.

[29] Julész, B. A brief outline of the texton theory of human vision. *Trends in Neuroscience 7*, 2 (1984), 41–45.

[30] Julész, B., and Bergen, J. R. Textons, the fundamental elements in preattentive vision and perception of textures. *The Bell System Technical Journal 62*, 6 (1983), 1619–1645.

[31] Kawai, M., Uchikawa, K., and Ujike, H. Influence of color category on visual search. In *Annual Meeting of the Association for Research in Vision and Ophthalmology* (Fort Lauderdale, Florida, 1995), p. #2991.

[32] Laidlaw, D. H., Ahrens, E. T., Kremers, D., Avalos, M. J., Jacobs, R. E., and Readhead, C. Visualizing diffusion tensor images of the mouse spinal cord. In *Proceedings Visualization '98* (Research Triangle Park, North Carolina, 1998), pp. 127–134.

[33] Levkowitz, H. Color icons: Merging color and texture perception for integrated visualization of multiple parameters. In *Proceedings Visualization '91* (San Diego, California, 1991), pp. 164–170.

[34] Levkowitz, H., and Herman, G. T. Color scales for image data. *IEEE Computer Graphics & Applications 12*, 1 (1992), 72–80.

[35] Liu, F., and Picard, R. W. Periodicity, directionality, and randomness: Wold features for perceptual pattern recognition. In *Proceedings 12th International Conference on Pattern Recognition* (Jerusalem, Israel, 1994), pp. 1–5.

[36] Mack, A., and Rock, I. *Inattentional Blindness.* MIT Press, Menlo Park, California, 1998.

[37] Malik, J., and Perona, P. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A 7*, 5 (1990), 923–932.

[38] Meier, B. J. Painterly rendering for animation. In *SIGGRAPH 96 Conference Proceedings* (New Orleans, Louisiana, 1996), H. Rushmeier, Ed., pp. 477–484.

[39] Rao, A. R., and Lohse, G. L. Identifying high level features of texture perception. *CVGIP: Graphics Models and Image Processing 55*, 3 (1993), 218–233.

[40] Rao, A. R., and Lohse, G. L. Towards a texture naming system: Identifying relevant dimensions of texture. In *Proceedings Visualization '93* (San Jose, California, 1993), pp. 220–227.

[41] Reed, T. R., and Hans Du Buf, J. M. A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image Understanding 57*, 3 (1993), 359–372.

[42] Rensink, R. A., O'Regan, J. K., and Clark, J. J. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science 8* (1997), 368–373.

[43] Rheingans, P., and Tebbs, B. A tool for dynamic explorations of color mappings. *Computer Graphics 24*, 2 (1990), 145–146.

[44] Robertson, P. K. Visualizing color gamuts: A user interface for the effective use of perceptual color spaces in data displays. *IEEE Computer Graphics & Applications 8*, 5 (1988), 50–64.

[45] Rogowitz, B. E., and Treinish, L. A. An architecture for rule-based visualization. In *Proceedings Visualization '93* (San Jose, California, 1993), pp. 236–243.

[46] Salisbury, M., Wong, M. T., Hughes, J. F., and Salesin, D. H. Orientable textures for image-based pen-and-ink illustration. In *SIGGRAPH 97 Conference Proceedings* (Los Angeles, California, 1997), T. Whitted, Ed., pp. 401–406.

[47] Schweitzer, D. Artificial texturing: An aid to surface visualization. *Computer Graphics (SIGGRAPH 83 Conference Proceedings) 17*, 3 (1983), 23–29.

[48] Simon, D. J., and Levin, D. T. Change blindness. *Trends in Cognitive Science 1* (1997), 261–267.

[49] Snowden, R. J. Texture segregation and visual search: A comparison of the effects of random variations along irrelevant dimensions. *Journal of Experimental Psychology: Human Perception and Performance 24*, 5 (1998), 1354–1367.

[50] Tamura, H., Mori, S., and Yamawaki, T. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics SMC-8*, 6 (1978), 460–473.

[51] Triesman, A. Preattentive processing in vision. *Computer Vision, Graphics and Image Processing 31* (1985), 156–177.

[52] Triesman, A. Search, similarity, and integration of features between and within dimensions. *Journal of Experimental Psychology: Human Perception & Performance 17*, 3 (1991), 652–676.

[53] Triesman, A., and Gormican, S. Feature analysis in early vision: Evidence from search asymmetries. *Psychological Review 95*, 1 (1988), 15–48.

[54] Turk, G., and Banks, D. Image-guided streamline placement. In *SIGGRAPH 96 Conference Proceedings* (New Orleans, Louisiana, 1996), H. Rushmeier, Ed., pp. 453–460.

[55] Ware, C. Color sequences for univariate maps: Theory, experiments, and principles. *IEEE Computer Graphics & Applications 8*, 5 (1988), 41–49.

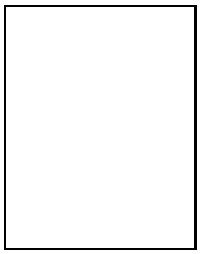
[56] Ware, C., and Beatty, J. C. Using colour dimensions to display data dimensions. *Human Factors 30*, 2 (1988), 127–142.

[57] Ware, C., and Knight, W. Using visual texture for information display. *ACM Transactions on Graphics 14*, 1 (1995), 3–20.

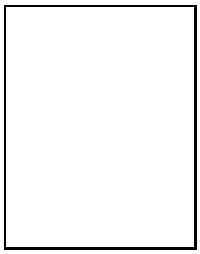
[58] Wolfe, J. M. Guided Search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review 1*, 2 (1994), 202–238.

[59] Wolfe, J. M., Yu, K. P., Stewart, M. I., Shorter, A. D., Friedman-Hill, S. R., and Cave, K. R. Limitations on the parallel guidance of visual search: Color × color and orientation × orientation conjunctions. *Journal of Experimental Psychology: Human Perception & Performance 16*, 4 (1990), 879–892.

[60] Wyszecki, G., and Stiles, W. S. *Color Science: Concepts and Methods, Quantitative Data and Formulae, 2nd Edition.* John Wiley & Sons, Inc., New York, New York, 1982.



Christopher G. Healey received a PhD in computer science in 1996 from the University of British Columbia, an MSc in 1992 from the University of British Columbia, and a BMath in 1990 from the University of Waterloo. Following graduation he completed a two-year postdoctoral fellowship in computer graphics with Dr. Carlo Séquin at the University of California, Berkeley. He is currently working as an assistant professor in the Department of Computer Science at North Carolina State University. His dissertation studied methods for displaying effectively large, multivariate datasets during scientific visualization. This work investigated techniques for exploiting the low-level human visual system for information representation. His current research focuses on the use of visual features like color, texture, and apparent motion for visually exploring multivariate data. He is also investigating automated data-feature mapping techniques and data management issues in an effort to design a flexible, robust perceptual visualization architecture.



James T. Enns received a PhD in psychology from Princeton University (1984) and a BA (honours) from the University of Winnipeg (1980). Following graduation, he was appointed an assistant professor at Dalhousie University, before moving to the Unviersity of British Columbia in 1987, where he is now appointed as a professor in both the Department of Psychology and the Graduate Program in Neuroscience. A central focus of his research over the years has been the role of attention in perception. This has included studies of how perception and attention change with development, how the visual world is represented outside the focus of attention, and how attention changes the perceptions that form the basis for consciousness. Along with the publication of these studies in *Science, Psychological Review, Perception & Psychophysics,* and *The Journal of Experimental Psychology,* he has edited two research volumes (*The Development of Attention,* 1990; *Attention, Development, & Psychopathology,* 1997) and coauthored two textbooks (*Analysis of Variance,* 1986; *Sensation & Perception*, fifth edition, 1999).