# 30 Years of Multidimensional Multivariate Visualization

Pak Chung Wong          R. Daniel Bergeron

*pcw@cs.unh.edu*          *rdb@cs.unh.edu*

Department of Computer Science

University of New Hampshire

Durham, New Hampshire 03824, USA

**Abstract**

We present a survey of multidimensional multivariate (mdmv) visualization techniques developed during the last three decades. This subfield of scientific visualization deals with the analysis of data with multiple parameters or factors, and the key relationships among them. The course of development is roughly organized into four stages, within which major milestones are discussed. Recently developed techniques are explored with examples.

## 1   Introduction

Multidimensional multivariate visualization is an important subfield of scientific visualization. It was studied separately by statisticians and psychologists long before computer science was deemed a discipline. The appearance of low-priced personal computers and workstations during the 1980's breathed new life into graphical analysis of mdmv data. This research topic was among one of the short-term goals included in the 1987 National Science Foundation (NSF) sponsored workshop on *Visualization in Scientific Computing* [MDB87]. The quest for effective and efficient mdmv visualization techniques has expanded since then.

This paper attempts to trace three decades of intensive development in this visualization field. It is by no means a comprehensive survey. We provide a brief history along with a description of the principal concepts of some mdmv visualization techniques. Recently developed mdmv visualization techniques are discussed in detail with examples. A remark of the trends of mdmv visualization research is given.

## 2   Four Stages of Multidimensional Multivariate Visualization Development

The last three decades of mdmv visualization development can be roughly characterized into four stages. The classic exploratory data analysis (EDA) book by Tukey [Tuk77], the 1987 NSF workshop on Visualization in

Scientific Computing [MDB87], and the IEEE Visualization '91 conference [NR91] are the watersheds defining these stages. The first stage was primarily concerned with the graphical presentation of either one or two variate data. The second stage was dominated by Tukey's exploratory data analysis. Scientists started looking at graphical data with a different perspective. Although most of the graphics was still two dimensional, scientists were able to encode data with multiple parameters, i.e., multivariate, into meaningful two dimensional plots. The momentum of this work carried on through the next stage when NSF recognized the importance of mdmv data visualization. The involvement of computer scientists accelerated the growth of the research by computerizing many of the old ideas and developing many new ones. The mission was formally defined and many promising concepts were developed during the following few years. The final (current) stage is concerned with the elaboration and assessment of mdmv visualization techniques. It remains to be seen whether the existing mdmv visualization concepts can lead to better visualization of a problem and better understanding of the underlying science. This discussion of mdmv visualization is far from complete. There are other important topics including volume visualization and vector/tensor field visualization that are not covered. The principal concepts and research issues related to these subjects can be found in [Nie92, PvW92, KHK$^+$94, HPvW94].

## 2.1   Pre–1976 The Searching Stage

Scientists have studied multivariate visualization since 1782 when Crome used point symbols to show the geographical distribution in Europe of 56 commodities [Col93]. In 1950, Gibson [Gib50] started the research on visual texture perception. Later, Pickett and White [PW66] proposed mapping data sets onto artificial graphical objects composed of lines. This texture mapping work was further investigated by Pickett [Pic70], and was eventually computerized [PG88]. Chernoff [Che73] presented his arrays of cartoon faces for multivariate data in 1973. In this well-known technique, variates are mapped to the shape of the cartoon faces and their facial features including nose, mouth, and eyes. These faces are then displayed in a two dimensional graph.

The searching stage can be characterized by relatively small sized data, and tools for data visualization that usually consisted of color pencils and graph paper. The graphical output was mostly two dimensional $xy$-displays. Statisticians were the dominant research force during this period. Graphics was used to bring out the key features of the data, suggest statistical analysis methods that are applied to the data, and present the conclusions [Fis70].

## 2.2   1977–1985 The Awakening Stage

Tukey's exploratory data analysis signified a new era of scientific data visualization. Exploratory data analysis is more than a tool; it is a way of thinking. It teaches people how to visually decode information from the data. When the personal computer arrived, it became the scientist's most powerful tool ever. Now scientists could visualize data beyond two dimensions interactively. The painfully long calculations suddenly became available in real time. Statisticians could visualize data during each stage of the analysis instead of waiting until the final results were available. The availability of other computer hardware such as high resolution color displays also gave the study of mdmv visualization new opportunities.

During this stage, two and three dimensional spatial data were the most common data types being studied,

although multivariate data started gaining more attention. Asimov [Asi85] presented the grand tour technique for viewing projections of multivariate data on two dimensional planes. Earth resource satellites sent out decades ago are still transmitting data continuously. Gigabyte sized multivariate data had arrived.

## 2.3  1986–1991 The Discovery Stage

The 1987 NSF workshop formally declared the need for two and three dimensional spatial object visualization. The two dimensional projections of multivariate data sets is also included as one of the short-term potential targets for scientific visualization research. Once the mission was defined, scientists started pushing hard on the representation and visualization of mdmv data. The limited availability of high speed graphics hardware during the previous stage was gradually conquered. A majority of research was directed away from the development of exploratory data analysis tools, which lay heavily on statistical measures, towards colorful high dimensional graphics that required high speed computations. Some of the mdmv visualization concepts developed during this stage include: grand tour methods [BA86], parallel coordinates [IRC87, ID87, ID90], iconography [PG88, BG89b, Bed90, Lev91], worlds within worlds [FB90a, FB90b], dimension stacking [LWW90], hierarchical axis [MGTS90, MTS91a, MTS91b], hyperbox [AC91], and various ideas collected in [Cle93, CMM93]. Some of these techniques attempt to show all dimensions and all variates visually as one display, whereas others aim at *direct manipulation graphics*, in which the user interactively selects subsets for display by using an input device such as a mouse. Virtual reality [FB90a, FB90b] began to appear in the mdmv visualization literature.

## 2.4  1992–present The Elaboration and Assessment Stage

In 1990 and 1991, there were at least fourteen mdmv related papers published in the IEEE Visualization conferences. A total of four have been published in the three visualization conferences since then. This stage so far has been a period of retrenchment in the development of new mdmv visualization techniques. Some of the most recently developed tools are, each in a different way, elaborations of work done in previous stages. For example, HyperSlice [vWvL93] is an attempt to combine the panel matrix of scatterplot matrix with direct manipulation of brushing [BC87]. AutoVisual [BF92, BF93] is an extended version of worlds within worlds with a new rule-based interfaces. XmdvTool [War94] integrates four existing mdmv visualization tools: dimension stacking, scatterplot matrix, glyphs, and parallel coordinates into one system with enhanced $n$-dimensional brushing.

The research in mdmv visualization has also been diversified into multidisciplinary collaborations. Attempts to combine sound with graphics [SPW92, SBG92] are currently being made. The concept of a rule-based queue [BF92, BF93] was also introduced. One of the latest research issues of mdmv visualization is the need to evaluate the correctness, effectiveness, and usefulness of mdmv visualization techniques. Similar concerns also appear in the other fields of visualization research [RET[+]94, HPvW94].

# 3 Terminology

Unfortunately, the mdmv literature suffers from ill-defined and inconsistent terminology. The term *dimensionality* is especially overloaded. Mathematicians consider dimension as the number of independent variables in an algebraic equation. Engineers take dimension as measurements of any sort (breadth, length, height, and thickness). Even the prefix *multi* is frequently interchanged with another prefix *hyper*. In statistics literatures, the prefix *multi* means two or more, indicating a natural breakpoint between one and two dimension in probabilistic methods. For the breakpoint between three and four (or beyond), the prefix *hyper* is used [Cle93]. We use the prefix multi to refer to dimensionality of two or more.

Beddow [Bed92] points out the difference between multidimensional *objects* and multidimensional *data*. Multidimensional *objects* are *spatial* objects, and our goal is to understand their geometry. The most common form are two dimensional images and three dimensional volumes. They can best be described as *n*-dimensional Euclidean spaces $\mathbb{R}^n$. Multidimensional *data*, on the other hand, refers to the study of relationships among multiple parameters. Mathematically these parameters can be classified into two categories: *dependent* and *independent* [KK93]. Some statisticians prefer the terms *factor* and *response* [Cle93]. A variable is said to be dependent if it is a function of another variable, the independent variable. The relationship of an independent variable $x$ and a dependent variable $y$ can best be described by the mathematical equation $y = f(x)$. We adopt the convention that the term multidimensional refers to the dimensionality of the independent variables, while the term multivariate refers to the dimensionality of the dependent variables [BCH$^+$94]. This is by far the most popular way to describe the dimensionality of mdmv data sets in scientific visualization literature. For example, a three dimensional volume space in which both temperature and pressure are observed and recorded in various locations produces 3d2v data.

Beddow [Bed92] argues that analytic methods used to explore *n*-dimensional Euclidean spaces $\mathbb{R}^n$ are not appropriate for general multivariate analysis. In mdmv visualization research, the emphasis shifts away from the strong mathematical definition of dependent and independent variates towards the broader definition of multiple variables or factors. This happens not only in mdmv scientific visualization research but also in statistical studies. The tools are different, but the goal is the same: to find the hidden relationships between the variables (also known as *fitting* in statistics).

In general, raw *scientific data* can be categorized into a hierarchy of data types. The most general and the lowest of the hierarchy is the *nominal* data, whose values have no inherent ordering. For example, the names of the fifty states are nominal data. The next higher type of the hierarchy is *ordinal* data, whose values are ordered, but for which no meaningful distance metric exists. The seven rainbow colors (i.e., red, orange, $\cdots$) belong to this category. The highest of the hierarchy is *metric* data, which has a meaningful distance metric between any two values. Times, distances, and temperatures are examples. If we bin metric data into ranges, it becomes ordinal data. If we further remove the ordering constraints, the data is nominal. Some of the visualization techniques included in this survey are specially designed to handle metric data (see Sections 5.2.2 and 5.2.9.)

The above 3d2v temperature/pressure example more or less implies that each 3 dimensional coordinates contain *simple* (i.e., neither a set nor an interval) and *atomic* (i.e., not composite) values of pressure and temperatures. This is different from the case when we measure, for example, the chemical contents of a volume. Each coordinates

now has a set (instead of a simple value) of composite data (i.e., chemical elements.) The varying numbers of values of a variate plotted in any single dimensional point is known as the *density* of that coordinate.

# 4   Fundamental Objective and Approach

The main objectives of mdmv visualization are to visually summarize an mdmv data set, and find key trends and relationships among the variates. Different properties and characteristics of the data may changes the way we carry out visualization, but not its goals.

The traditional two dimensional point and line plots are among the most commonly used visualization techniques for data with lower number of variates. This technique can be enhanced by putting an array of plots into one display, so as to add another variate to the visual presentation. This approach is discussed in Sections 5.1.5 and 5.2.2.

We can also map the variates of the data into graphical primitives of differnt colors, sizes, shapes, and locations (see Sections 5.2.4, 5.2.5, and 5.2.6.) The display of all dimensions and all variates creates some kind of texture patterns, and provide critical insights needed for scientific discovery.

For large (larger than the number of pixels of a display) scientific data, we can display a certain portion of data and allow the user to navigate the rest interactively. This is described in Sections 5.2.2, 5.2.7, 5.2.8, and 5.2.9,

Most of the visualization techniques assume a Euclidean space environment. Orthogonal axes, however, are not always the best choice to plot data. Sections 5.2.7, 5.2.10, and 5.2.11 give some alternatives.

A powerful visualization technique is to display the data frame by frame according to a time variate. This animation approach is discussed in Sections 5.3.1, 5.3.2, and 5.3.3.

# 5   Multidimensional Multivariate Visualization and Concepts

The body of this paper covers the principal concepts and brief history of some of the popular mdmv visualization techniques. During the last decade, hundreds of so-called *new* mdmv visualization techniques have been invented. (Refer to [KK93] for more details in this regard.) A majority of them are designed for special purposes such as volume visualization and vector/tensor field visualization, which are not covered in our discussion. Some of the rest are merely *ad hoc* tools that produce pretty pictures. They are difficult to create and their results are hard to interpret. We are interested in techniques that are founded on a solid basis and that have potential for practical value.

Categorizing mdmv visualization techniques is a difficult task. Possible criteria for such a categorization include the goal of the visualization, the type and/or dimensionality of the data, the dimensionality of the visualization technique, etc. We have not found a convincing set of criteria that cleanly differentiate the visualization techniques we wish to describe. We have chosen to group the techniques into those based on 2-variate displays, those based on multivariate displays, and those using time as an animation parameter:

- *Techniques based on 2-variate displays* include the fundamental 2-variate displays and simultaneous views

of 2-variate displays. Most of the them developed in the statistics world. Both visual perception and statistical fitting of the data are of major concern. The data size is relatively small, usually in the order of hundreds of items. The graphics are mostly variations on two dimensional point and line plots.

- *Multivariate display* are the basis for many recently developed mdmv techniques, most of which use colorful graphics created by high-speed graphics computation. The data is usually larger and more complicated. A majority of the techniques were developed within the period of 1987–1991.

- *Animation* is a powerful tool for visualizing mdmv scientific data. Various movie animation techniques on mdmv data, and a scalar visualization animation model are presented. In principle, any single frame visualization technique can be extended to animation if the data can be represented as a time series showing two-way correlations.

## 5.1 Techniques Based on 2-variate Displays

This section highlights some of the tools and summarizes the general approach developed based on 2-variate displays. The discussion is based upon the book by Cleveland [Cle93], which has a good collection of elegant visualization techniques developed by Cleveland, Tukey, and others throughout the 80's. Tukey's exploratory data analysis [Tuk77] is an important milestone of data visualization; most of the techniques were developed with pencil and paper during the early 70's. Cleveland's work emphasizes the structure of data and the validity of statistical models fitted to data. A majority of the visualization techniques are two dimensional, with the exception of isosurface plotting. Color is rarely used. Most of the tools show correlations between two variates. Our discussion skips the formulas, algorithms, and theories; only the concepts and techniques are presented.

### 5.1.1 Data Types

The basic data types for statistical data analysis are *univariate*, *bivariate*, *trivariate*, and *hypervariate* which represent data with one dimension and one, two, three, and four or more variates. Cleveland also describes the *multiway* data type for data with higher dimensionality.

### 5.1.2 Reference Grids

The most common display unit in statistics visualization is a two dimensional scatterplot, as depicted in the left panel of Figure 1. In the middle panel, simple grid lines are drawn for enhancement of pattern perception, not for plotting accuracy. Grids are drawn in equal intervals instead of numerical values. These reference lines are particularly powerful when we need to do scanning and matching of a matrix of scatterplots.

### 5.1.3 Fitted Curve

In statistics, fitting means finding a description of a data set. For example, if a data set fits into a normal distribution, the whole data set can then be described by two numbers: its mean and standard deviation. In
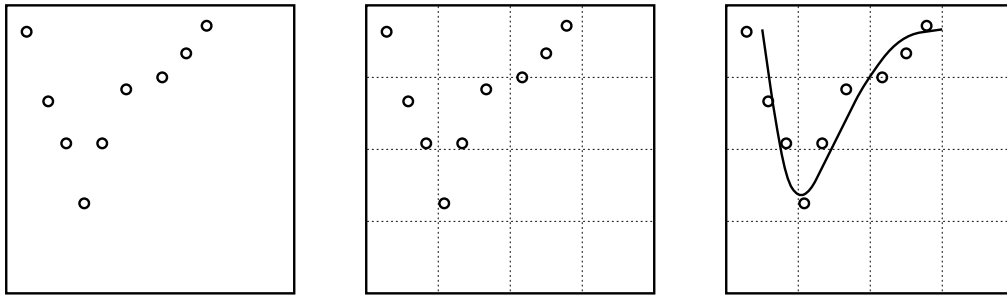
Figure 1: Left: A simple 2D scatterplot. Middle: A scatterplot with visual reference grids. Right: A fitted curve is included in the plot.

statistics visualization, fitting means finding a smooth curve that describes the underlying pattern. In the right panel of Figure 1, a curve fit to the data is plotted; a pattern not apparent from the scatterplot before may suddenly emerge. Fitting formulas are not discussed in this paper; [Tay90, Cle93] are good references for this matter.

### 5.1.4 Banking

The perception of the orientations of line segments can be enhanced by adjusting the *aspect* ratio of the graph. The aspect ratio of a graph is defined as the height of the data rectangle divided by the width. A line segment with an orientation of 45° or −45° is the best to convey linear properties of the curve. This technique is known as the *banking to* 45° *principle* [CMM93]. In Figure 2, the same curve is plotted in three different aspect ratios. Only
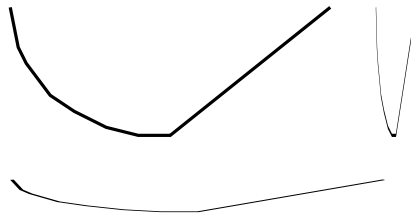


Figure 2: The same curve is plotted in three different aspect ratio. The upper left one conveys more information than the other two.

the upper left panel shows both a curve on the left and a straight line on the right. The banking method is covered in [Cle93].

### 5.1.5 Scatterplot Matrix

One of the more popular statistics mdmv visualization techniques is the scatterplot matrix which presents multiple adjacent scatterplots. Each display panel in a scatterplot matrix is identified by its row and column numbers in the matrix. For example, the identity of the upper left panel of the matrix in Figure 3 is (1,3), and the lower right panel is (3,1). The empty diagonal panels denote the variable names. Panel (2,1) is a scatterplot of parameter $X$ against $Y$ while panel (1,2) is the reverse, i.e., $Y$ versus $X$. In a scatterplot matrix, every variate is treated identically. The
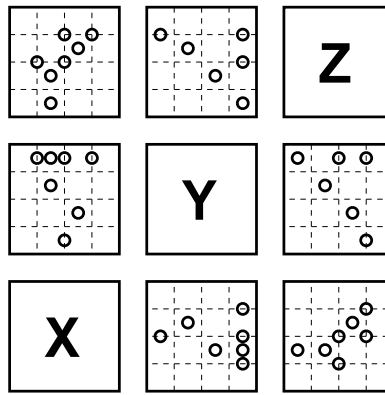
7

Figure 3: A scatterplot matrix displays of data with three variates $X$, $Y$, and $Z$.

basic idea is to visually *link* features in one panel with features in others. The redundancy is designed to improve the effect of visual linking. The technique is further enhanced with the help of reference grids. The pattern can be detected in both horizontal and vertical directions. The concept of linking is also discussed in [BMMS91].

The idea of pairwise adjacencies of variables is also a basis for the hyperbox [AC91], hierarchical axis [MGTS90, MTS91a, MTS91b], and HyperSlice [vWvL93]. Despite its popularity in mdmv visualization applications, nobody knows the identity of the original inventor [Cle93]. The technique was first presented in [CCKT83]. A variety of powerful tools using this kind of multi-panel display are presented in [Cle93]. The scatterplot matrix is also implemented in XmdvTool [War94].

### 5.1.6 Other Two Dimensional Analytical Techniques

Cleveland's book also includes other powerful graphical techniques such as *medium-difference plot*, *quantile-quantile plot*, *spread-location plot*, *given plot*, and *conditional plot*; fitting tools such as *loess* and *bisquare*; and visual perception techniques such as *jittering* and *outlier deletion*.

## 5.2   Multivariate Visualization Techniques

The scatterplot matrix uses multiple 2-way displays in an effort to provide correlation information among many variates simultaneously. The techniques described in this section are, however, aimed at extending the possibilities of multivariate correlation. All the techniques, with the exception of brushing and parallel coordinates, were developed after the 1987 NSF workshop. All of them claim positive results with real life mdmv scientific data. These techniques are also aimed at presenting much larger data sets than those appropriate for the statistical visualization techniques. Today's scientific data is huge; terabyte sized data will soon be common. A static scatterplot is just not big enough to display more than a few hundred data items. These techniques are broadly categorized into five sub-groups:

- *Brushing* allows direct manipulation of a mdmv visualization display. Only brushing a scatterplot matrix is described.

- *Panel matrix* involves pairwise two dimensional plots of adjacent variates. Techniques included are Hyper-Slice and hyperbox. Both of them are elaborations of the scatterplot matrix.

- *Iconography* uses variates to determine values of parameters of small graphical objects, called *icons* or *glyphs*. Thousands of data points are represented by thousands of these icons which create a visual display characterized by varying texture patterns determined by the data. The mappings of data values to graphical parameters are usually chosen to generate texture patterns that hopefully bring insight into the data. Three iconographic techniques are described: stick figure icon, autoglyph, and color icons.

- *Hierarchical displays* map a subset of variates into different hierarchical levels of the display. Hierarchical axis, dimension stacking, and worlds within worlds belong to this group. These techniques support, or at least enable, dynamic interactive analysis.

- *Non-Cartesian displays* map data into non-Cartesian axes. They include parallel coordinates and VisDB. Parallel coordinates is the only technique that is capable of studying both multidimensional objects and multidimensional data.

### 5.2.1 Brushing

Brushing was first presented in [BC87]. It is included as one of the many direct manipulation techniques in [Cle93]. There are two kinds of brushing a scatterplot matrix: labeling and enhanced linking. Labeling involves an interactive brush (e.g., a mouse pointer) that causes information label(s) to pop-up for particular display item(s). In enhanced linking, the brush is an adjustable rectangle. It is used to cover a set of points in one of the panels. Figure 4 shows a rectangle brush in panel (3,2). Data inside the rectangle is displayed with a "+" instead of a "○."
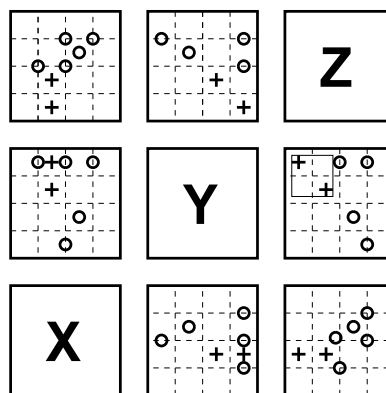


Figure 4: Enhanced brushing with the square brush located on panel (3,2).

The same changes are applied to the corresponding data points in the other panels. By looking at different panels and comparing the vertical and horizontal extent of the brush, this enhanced linking technique provides a powerful direct manipulation tool for visual conditioning analysis. It is shown that the effect of brushing is more intense in a dynamic interactive display. In general, brushing can be added to many other mdmv visualization techniques [War94]. More applications can be found in [Cle93].

9

### 5.2.2 HyperSlice

HyperSlice [vWvL93] is one of the techniques invented during the elaboration and assessment stage. Like the scatterplot matrix, it has a matrix of panels, although each individual scatterplot is replaced with color or grey shaded graphics representing a scalar function of the variates. Furthermore, panels along the diagonal show the scalar function in terms of a single variate.

HyperSlice defines a focal point of interest $c = (c_1, \ c_2, \ \cdots, \ c_n)$ and a set of scalar widths $w_i$, where $i = 1, \ \cdots, \ n$. Only data within the range $R = [c_i - w_i/2, \ c_i + w_i/2]$ are displayed in the panel matrix. The rest of the data only appears if the user steers the focal point near it. Color Plate 1 shows the display of a HyperSlice of four variates. Like the coordinate system used in the scatterplot matrix, a HyperSlice panel is identified by a
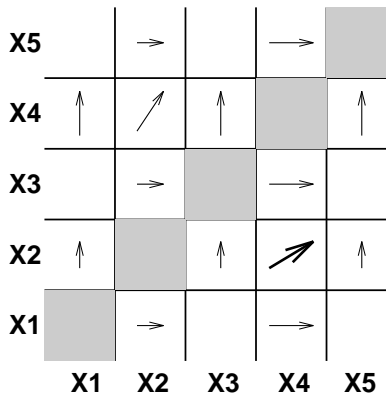


Figure 5: Navigate a five variate HyperSlice by dragging panel (4,2).

horizontal and a vertical coordinate. For an off-diagonal panel $i, j$ such that $i \neq j$, the color shows the value of the scalar function that results from fixing the values of all variates except $i$ and $j$ to the values of the focal point, while varying $i$ and $j$ over their ranges in $R$. The diagonal panels show a graph of the scalar function versus one variate which changes over its range in $R$.

The most important improvement of HyperSlice over the traditional scatterplot matrix is the idea of interactively *navigating* in the data around a user defined focal point. The user changes the focal point by interacting with any of the panels, as shown in Figure 5. The user moves the mouse into any panel and defines a direction by button down, move, and up. For example, the boldface arrow in panel (4,2) represents such an interaction. The direction of each arrow shows the motion of the focal point when the focal point is dragged in panel (4,2). Notice that the length (magnitude) of the vertical arrows across the $X_2$ row, is the same as the vertical component of the arrow in (4,2). Similarly, each horizontal arrow in column $X_4$ has the same length as the horizontal component of the arrow in panel (4,2). Panels solely related to $X_1$, $X_3$, and $X_5$ move perpendicular to the image plan. Since the matrix is somewhat similar to an orthogonal matrix (along the grey diagonal panel), the motion on the upper left half is the mirror projection of the lower right.

Interactive data navigation is a welcome addition to direct manipulation graphics. The use of the width scalar supports the notion of multiresolution analysis, and begins to address more than two-way correlations. Changing the focal point in one panel affects two variates which in turn results in simultaneous visual changes in displays of

these variates with others. HyperSlice is an example of a successful elaboration which builds on another successful tool.

### 5.2.3  Hyperbox

We place the discussion of hyperbox [AC91] here because it works very much like the scatterplot matrix and HyperSlice. It too involves pairwise two dimensional plots of adjacent variates, yet hyperbox is very different from the other two.

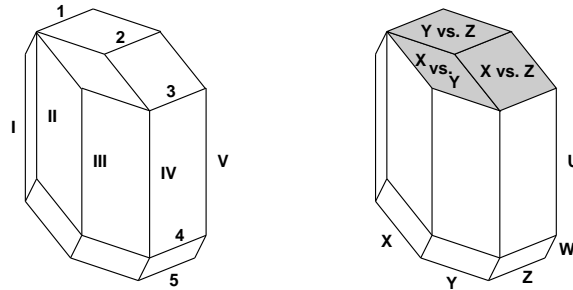A hyperbox is a two dimensional depiction of an $n$-dimensional box. Figure 6 shows a simple hyperbox of



Figure 6: Left: Hyperbox of dimension 5. Right: Variates $x, y, z, w$, and $u$ are mapped to different dimensions. The two dimensional plots of $x$ versus $y$, $y$ versus $z$, and $x$ versus $z$ are shown in grey.

dimension five. An $n$-dimensional hyperbox is made up of $n^2$ *lines* and $n(n-1)/2$ *faces*. There are $5^2 = 25$ lines and $5(5-1)/2 = 10$ faces in a hyperbox of dimension five. For each line in a hyperbox, there are $n-1$ other lines with the same length and orientation. The length and slope of the lines are arbitrary. Both of them can be mapped to the data variates for visualization. Lines with the same length and orientation form a *direction set*. In Figure 6, lines 1, 2, 3, 4, and 5 form one direction set while lines I, II, III, IV, and V form another. Given a five variate data set, $x, y, z, w$, and $u$, each variate is mapped to one direction, as shown in the hyperbox on the right hand side of Figure 6. Each face of the hyperbox can now be used to plot data of two variates such as a scatterplot or a line plot.

To support data analysis, variates can be selected by $cutting$ the hyperbox along each direction set. A hyperbox cut is similar to a *discrete table selection* in relational databases, except that the variates are discarded instead of selected. Suppose the data we have is time-series related and $u$ is the time variate. Then the time variate can be cut and grouped into periods analogous to a histogram. Figure 7 depicts an array of hyperboxes with variate $u$
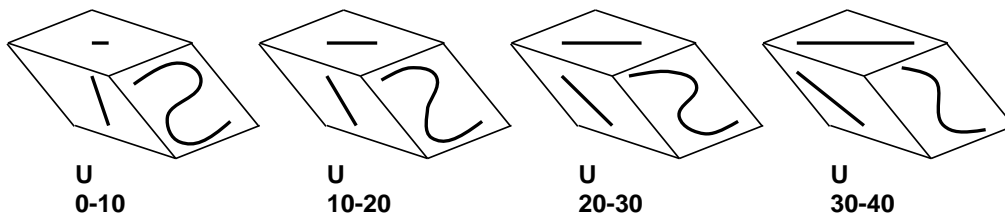


Figure 7: An array of hyperboxes excluding dimension $u$ and $w$ is shown in different time period.

11

and $w$ being cut. Each hyperbox depicts all values of variates $x$, $y$, and $z$ that occur for all $u$ in the range defined for that hyperbox. With a hyperbox, and any other panel matrix type mdmv technique, occlusion occurs when the most recent value replaces the earlier displayed data value at the same spot. This is one of the reasons why an array of hyperboxes is shown instead. An example hyperbox visualization is shown in Color Plate 2.

The design of the hyperbox requires a little practice to understand. It is a more powerful tool compared to the scatterplot matrix in the sense that it is possible to map variates to both the size and shape of each facet. It also gives scientists the option to emphasize some of the more important variates and de-emphasize others. On the other hand, it violates Cleveland's banking to 45° principle. The arbitrary setting of length and orientation make it impossible to do the right banking on all facets. This means that some of the plots may not be able to convey the *right* information.

The idea behind Cleveland's reference grid in the previous discussion is to maximize the visual perception of spatial locality during data scanning and matching. Once again, the arbitrary assignment of length and orientation of each facet makes it impossible for the hyperbox display to take full advantage of this technique.

### 5.2.4  Stick Figure Icon

Pickett and Grinstein [PG88] presented the stick figure icon visualization technique. It is rich in concepts, and practical development [GPW89, SGB91]. The original stick icon is a five stick figure with controllable limb angle. Figure 8 depicts a family of twelve of them. This icon family is designed to display data with up to five variates.
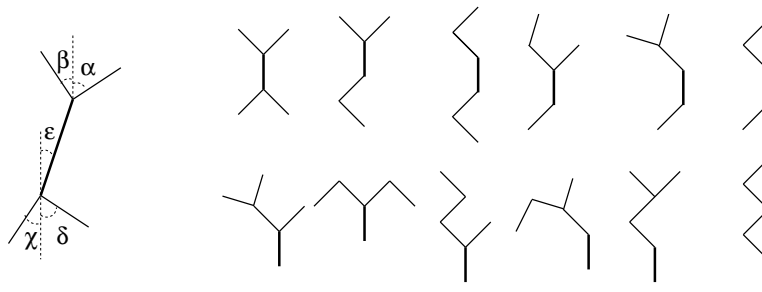


Figure 8: Left: A five stick figure icon with orientation plotted according to some values. Right: A stick figure icon family. Each one has a body and four limbs.

Four of them can be mapped to the orientation of each limb and the fifth can be mapped to the inclination of the body. Other variates can map to the length, thickness, or color of the limbs. An example stick figure icon visualization showing 7 variates is shown in Color Plate 3. A two stick icon is also presented in [SGB91] to display bivariate MRI data. The notion of icon construction is to allow humans to exploit the capacity to sense and discriminate the texture in a complex image. In the bivariate data example, the stick icon successfully locates a hot spot not seen in the original MR images. The authors report that these multiline icons have been used to display as many as thirty variates. However, the flexibility of this technique can also can be a weakness. The visual discernment of an important pattern can be highly dependent upon the selection of an appropriate mapping of the data parameters to the visual parameters. Since the number of potential mappings grows as the factorial of the number of variates, the selection process can become a bottleneck.

### 5.2.5 Autoglyph

Beddow [Bed90] describes what he calls an *autoglyph*. The first generation of his glyph, known as Datapix, was comprised of two rectangles, a filled arc, and a circle. A few examples are discussed in [Bed92]. Based on the results of Datapix, a second generation of autoglyph, which contains only a circle and a box, was developed. Suppose we have twelve variate data. Each variate is first normalized and then divided into three groups based on the standard deviation function, and a group representing missing values. The high and low groups are assigned to black and grey, and the rest are white. Each variate can then be mapped into one square tile inside the rectangle glyph, as shown in Figure 9. The paper also presents examples of using primary colors instead of
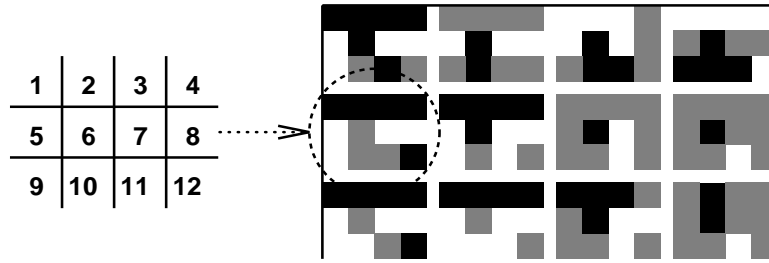


Figure 9: Autoglyph designed for twelve variate data.

black/grey/white. Beddow [Bed90, Bed92] gives a brief analysis of the use of color in such glyphs. Autoglyph was originally designed to study the *correlation* among large numbers of variates. It has a very compact display and can be extended to using threshold functions other than standard deviation. Keller and Keller [KK93] do not recommend this technique for group presentation because it requires time to study.

### 5.2.6 Color Icon

Levkowitz [Lev91] describes a *color icon* that merges color, shape, and texture perception for iconographic integration. Color and shape are perceptually orthogonal, and are particularly good at *unmixing* parameters of a mdmv display. A color icon is an area on the display to which color, shape, size, orientation, boundaries, and area subdividers can be mapped by multivariate data. Figure 10 shows a square shaped color icon which maps up to
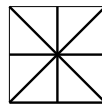


Figure 10: A square color icon.

six variates. Each variate is mapped to one of the six thick lines. The thin lines serve as boundaries to separate neighboring icons.

There are two different ways to paint a color icon. The first approach requires color shading. A color is assigned to each thick line according to the value of the mapping variate. The color of the color icon is then computed by interpolating the colors assigned to all thick lines. A second way to paint a color icon is to assign

13

color to each pie-shaped sub-area according to the values of the mapping variates. The first approach provides better parameter blending, while the second one gives better parameter separability. Figure 11 depicts the mapping
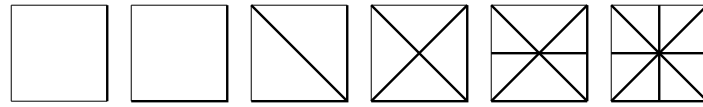


Figure 11: Mapping mdmv data with one to six variates to color icons. The value is mapped to the thick line only.

of color icons from data with one to six variates. The number of variate mapping can be tripled by having each variate control one of the HSV values. Icons with different shapes (such as hexagonal) can be used in place the square shaped icon. An example of color icon visualization is shown in Color Plate 4.

### 5.2.7    Hierarchical Axis

The conventional way to describe a three dimensional Euclidean space is by using three orthogonal axes. In hierarchical axis [MGTS90, MTS91a, MTS91b], axes are laid out horizontally in a hierarchical fashion. For example, assume we have three variable $x$, $y$, and $z$, and the domain of each variable is $\{0,1,2\}$. The Euclidean 3D space and the hierarchy axes are depicted in Figure 12. This technique depends upon a relatively coarse
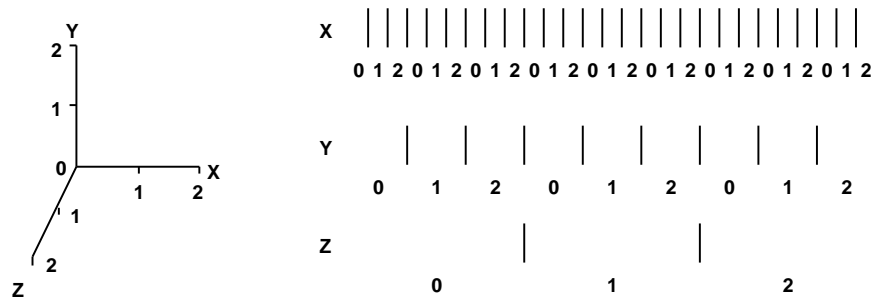


Figure 12: Left: Orthogonal axes. Right: Hierarchy axes.

discretization of the data. A new term *speed* is coined to describe the hierarchical axes, with $x$ being the *fastest* axis and $z$ being the slowest [MGTS90, LWW90, MTS91a]. Once these axes are defined, a variety of potential display options can be used.

One simple example is the histogram (or histograms within histograms) plot. Given a set of data $\{(x,y,z)\}$ = {(0,0,1), (1,0,0), (2,0,1)}, we have a histogram plotted on the left hand side of Figure 13. The height of the dark grey rectangle shows the value of the dependent variate $z$. The independent variates $x$ and $y$ are described by the horizontal hierarchical axes located at the bottom. In the middle plot, the height of the light grey rectangle is defined as the *sum* of all $z$ variables, i.e., the dark grey rectangles, enclosed inside. In this example, the sum equals to two. The use of this histogram can be clarified in the last plot on the right, which includes additional data. The histogram in the middle becomes embedded inside another histogram, which shows the sum of the values of each inner histogram. The sum function is only one of the many possible options used in this system. Other possible functions are *min/max* and *mean/standard deviation*.
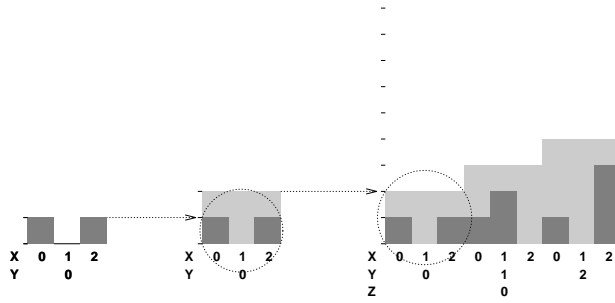
14

Figure 13: Left: A simple histogram plot. Middle and Right: Histograms within histograms plots.

The hierarchical axis technique can plot as many as twenty variates in one screen. For data with a larger number of records, i.e., larger than the number of columns of pixels on the display screen, a technique called *subspace zooming* is introduced. A display of multivariate data involves a series of panels (the number of panels is equal to the number of variates). Each panel displays data from two hierarchical axes, ordered from the slowest to the fastest. This can be considered as a *tree* structure with the panel showing the slowest axis as the root. The other panels are nodes. A subspace is a sub-tree of the root. A series of panels is a *path* of the tree. Only the root panel is static, and only one path is shown at a time. The panels of the other paths are hidden until the user interactively clicks the specific data of any non-terminal panel to select another subspace (sub-tree).

A panel matrix of histograms is also presented in [MGTS90, MTS91a, MTS91b] to gain visual perception power similar to the scatterplot matrix. It is suggested that brushing (i.e., direct manipulation graphics) can also be implemented by mapping all or some of the variates to axes with different speeds according to selection conditions. An example visualization is shown in Color Plate 5.

### 5.2.8  Dimension Stacking

Dimension Stacking [LWW90] is a variant of hierarchical axis. In the hierarchical axis technique, each element of the fastest axis is a one dimensional histogram. In dimension stacking, each element is a two dimensional $xy$-plot. If the data has an odd number of variates, a dummy variate is added. Dimension stacking also has the flavor of iconographics [PG88]. Each two dimensional $xy$-plot element forms a texture pattern for data visualization and analysis. A major advantage of dimension stacking over hierarchical axis is that no extra function or rule, such as the use of sum in our previous example, is needed to plot the data.

Assume we have a data set with four independent variates $x$, $y$, $z$, and $w$, such that the corresponding sizes of the variates are 2, 3, 5, and 6. Figure 14 shows two different variate mapping schemes. In the first example (left), $w$ and $z$ are the fastest axes while $x$ and $y$ are slower. The second rectangle ($x$ versus $y$) contains six (two by three) smaller rectangles, each of which is a two dimensional plot of $z$ versus $w$, like the leftmost rectangle. If there is a dependent variate available, its values can be plotted as color/grey intensity in each of the squares. Otherwise, each of the two dimensional plots can be a simple scatterplot. The second example shows a different mapping with the $x$ and $y$ axes being the fastest ones. This hierarchical display technique is also implemented in XmdvTool [War94]. An example of dimension stacking visualization is shown in Color Plate 6.
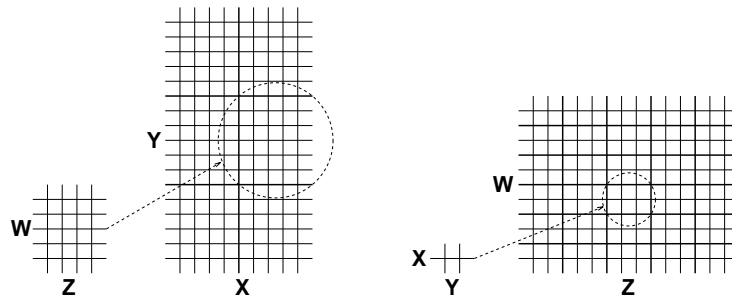
15

Figure 14: Two different variate mapping of the same data in dimension stacking.

### 5.2.9 Worlds Within Worlds

All mdmv visualization techniques we have discussed so far, with the exception of one of the hierarchical axis techniques, involve the generation of *static* objects such as texture maps. The idea behind worlds within worlds [FB90a, FB90b] is somewhat different. Dimensions are nested together with at most three variates being shown at each level, so as to generate an interactive hierarchy of displays. The slowest three axes are represented only by a display of three orthogonal axes, as shown in Figure 15. A three dimensional power glove is used interactively
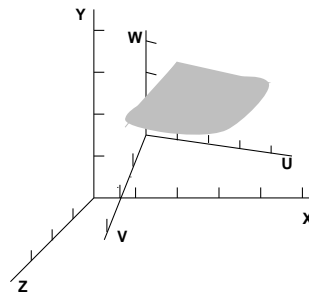


Figure 15: Worlds within worlds. Variate $x$, $y$, and $z$ are plotted initially. Variate $u$, $v$, and $w$ are plotted after all previous variates are defined.

to define a position in the space defined by these three axes. A new set of these axes appears at this point. The glove can then be used to pick a point in this space. This continues until all variates are defined. At the lowest level the final variate can be displayed as a surface in the innermost *world* as shown in Figure 15. The output is a three dimensional stereo display of virtual worlds within which users can explore their data. This dynamic data visualization technique gives a new meaning to *direct manipulation* graphics. Different variate mappings give different views of the data. The direct manipulation process is in fact a data retrieval query. Users have to know what they are looking for as most of the information is not visible in the initial display. This interactive process tends to be difficult and tedious because there are too many possible combinations of variate mappings. A new generation of worlds within worlds, AutoVisual [BF92, BF93], adds a rule-based user interface. Now users can specify the task through the interface, and the system generates virtual worlds accordingly. An example of worlds within worlds visualization is shown in Color Plate 7.

### 5.2.10 Parallel Coordinates

All techniques we have discussed so far are designed to do data analysis on multidimensional *data*. They are not really aimed at studying the *geometry* of multidimensional *objects*. Parallel coordinates [IRC87, ID87, ID90], on the other hand, can do both.

In a parallel coordinate system, the axes of a multidimensional space are defined as parallel vertical lines separated by a distance $d$. A *point* in Cartesian coordinates corresponds to a *polyline* in parallel coordinates. To avoid confusion, we use lower case letters for lines, and upper case letters for points in Cartesian spaces. In parallel coordinates, we use similar conventions except we put a bar superscript on all letters.

To see how a multidimensional object is represented in parallel coordinates, consider a simple two dimensional straight line

$$l : x_2 = mx_1 + b$$

where $m < \infty$. In Figure 16, the continuous straight line in Cartesian coordinates is sampled, and the values are
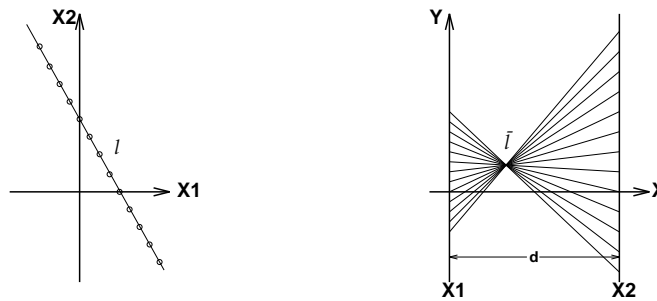


Figure 16: Left: Two dimensional Cartesian coordinates. Right: Parallel coordinates.

plotted on corresponding axes in parallel coordinates. A collection of points, $A$, sampled from a straight line in Cartesian coordinates, corresponds to a set of lines $\overline{A}$ in parallel coordinates that intersect at the point $\overline{l}$,

$$\overline{l} : \left( \frac{d}{1-m}, \frac{b}{1-m} \right)$$

for $m \neq 1$. For example, given a straight line $x_1/2 - x_2 + 1 = 0$, and $d = 5$, we get $m = 1/2$ and $b = 1$. When the sampled points are plotted in parallel coordinates, all lines intersect at $(\frac{5}{1-1/2}, \frac{1}{1-1/2}) = (10, 2)$, as shown in Figure 17. Notice that the location of the intersection point shows an important property of the data. In Figure 16 where two variates $x_1$ and $x_2$ are inversely proportional to each other, i.e., $x_1 \propto 1/x_2$, the intersection point is in between the two parallel axes. In Figure 17, where variate $x_1$ is directly proportional to $x_2$, i.e. $x_1 \propto x_2$, the intersection point is located outside the two parallel axes.

Parallel coordinates allow humans to visualize three dimensional time series data better than Cartesian coordinates. A simple application is aircraft collision checking [ID90]. In Figure 18, the locations of two aircraft are displayed in Cartesian coordinates and parallel coordinates. It is almost impossible to confirm a collision solely by judging the locations of two aircraft in any one single view of a three dimensional Cartesian plot. For example, suppose we have an isometric projection [FvDFH90] with three aircraft located at coordinates (0,0,0), (1,1,1), and
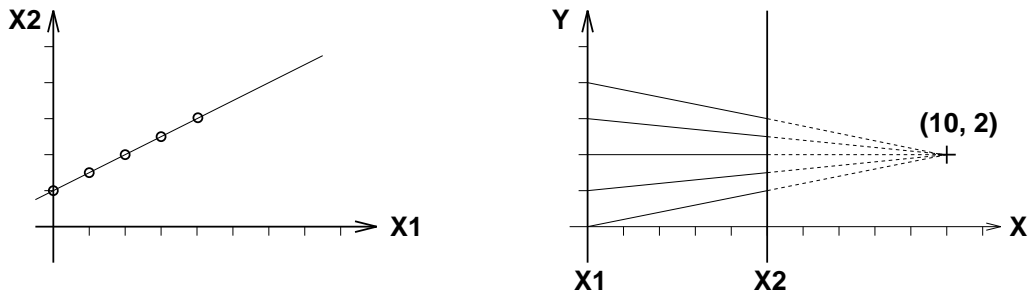
17

Figure 17: Left: The straight line, $x_1/2 - x_2 + 1 = 0$, is plotted in Cartesian coordinates. Right: the same line is plotted in parallel coordinates. The intersection point, $(10, 2)$, is located outside the two axes.
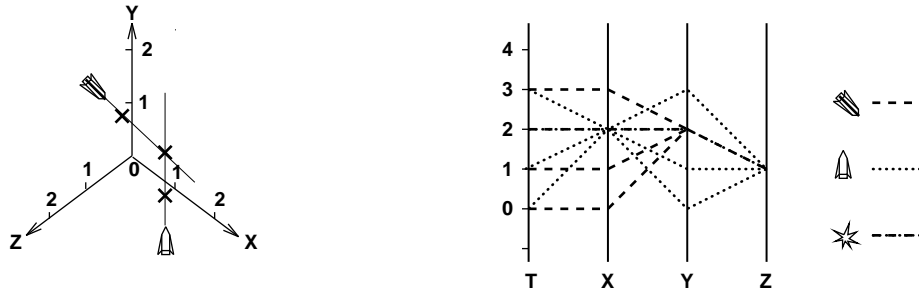


Figure 18: Left: Locations of two aircraft in three dimensional Cartesian coordinates at time 0. Right: The trajectory is plotted with time axis, T, in parallel coordinates.

(2,2,2) at time $t$. They are all displayed at the same spot, yet no collision occurs. On the other hand, a parallel coordinate plot including time $t$, and the coordinates $x, y, z$ is shown on the right hand side of Figure 18. Two aircraft collide if and only if they are in the same location at the same time. That means there will be a collision in location (2,2,1) at T = 2. A four dimensional intersection can be detected by searching for any overlapping dashed lines. In our example, an overlap is detected at (2,2,2,1) of the parallel coordinate plot.

To help avoid collision, *parallelograms* can be defined along with the trajectory of the aircraft. The number, size, and shape of the parallelogram are computed according to its relative velocity and locations with respect to others, as shown in Figure 19. If the two aircraft are flying at the same velocity, the lower right hand aircraft must
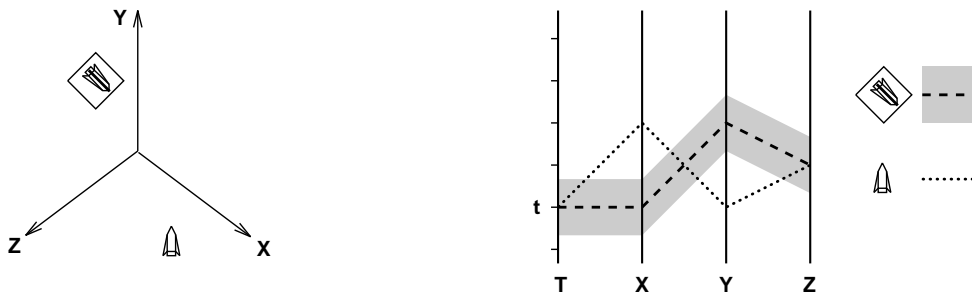


Figure 19: Left: Conflict parallelogram. Right: At time $t$, the graph shows the location of one aircraft is not entirely inside the grey area of the others, so no collision occurs.

avoid any contact with the parallelogram of the upper left hand aircraft. Like our previous example, it is almost impossible to spot the conflict in the three dimensional Cartesian plot. In Figure 19, the safety zone is indicated in grey in the parallel coordinate plot. There is a conflict at any time if the plotting of location/time of one aircraft is entirely inside the grey area of another.

Parallel coordinates can also be used to study correlations among variates in mdmv data analysis. By spotting the locations of the intersection points, see Figures 16 and 17, we can have a rough idea about the relationships between each pair of variates. This is one of the more promising applications of parallel coordinates in mdmv visualization. The problem with this technique is the limited space available for each parallel axis. The display can rapidly darken with even a modest amount of data. An implementation of parallel coordinates is also available in XmdvTool [War94] with brushing.

### 5.2.11   The VisDB System

Keim et al. [KKS93, KKS94, KK94] describe a multivariate visualization technique that is motivated by the desire to visualize information from a database. They use a combination of *distance functions* and *weighting factors* to visualize the *relevance factor* of very large multivariate data. The user issues a database query which identifies a *focal point* in multidimensional space. The data is arranged using a function that represents the relevance factor of that item with respect to the focal point defined by the query. Each display item is represented by a single pixel whose color is also determined by the relevance factor.

Once the relevance factors of the extracted data are computed, several display methods are possible. In a normal arrangement, data is sorted and arranged such that data with the highest relevance factors (smallest distance) are centered in the middle and the rest are placed in a rectangular spiral around this region as shown in Figure 20.
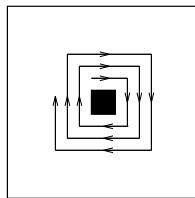


Figure 20: Normal arrangement.

In this basic form the information is somewhat limited since the position and color of each data item is determined by the same value – the distance of the item from the focal point. The upper left display in color panel 9 shows this display. The overall pattern of color variation gives a general sense of how all the items in the database (or at least all that fit in the display) relate to the focal point, but it does not provide any visual insight regarding the relationships among the multiple variates. Such a comparison can be accomplished by generating additional displays that are viewed simultaneously. There is one display for each variate that contributes to the query; the position of each data item is the same as in the first display, but the color is based on only the distance of that variate from its value at the focal point. By comparing the color patterns of two or more variate displays, it may be possible to infer correlations of these variates with respect to the query that defines the focal point. See Color Plate 8.

19

The system has a heavy database orientation. The definition of the distance function becomes somewhat arbitrary for non-metric variates and more complex for nested queries with Boolean operators. This approach is different from all the other techniques described in that it does not display the data directly, but instead presents the data based on the computed relevance factors.

### 5.2.12 XmdvTool

Cleveland begins and ends his book [Cle93] with the same quote, "Tools matter." The idea is that you have to pick the right technique to visualize mdmv data. The implication is that no technique alone is powerful and flexible enough to handle all mdmv scientific data. Ward [War94] integrates four popular static mdmv visualization techniques, one from each of the four static sub-groups defined earlier, into a single analysis system, XmdvTool. The four tools included are: scatterplot (panel matrix), dimension stacking (hierarchical display), star glyph (iconography), and parallel coordinates. The original brushing [BC87], which relies heavily on mouse clicking interaction, is modified into a more complicated tool with user-controlled parameters. A user can control the shape, size, boundary, position, motion, and orientation of the $n$-dimensional brush. Some of these parameters are customized while the others can be controlled with slider widgets. The original brushing was implemented on the scatterplot matrix only. Brushing in XmdvTool is implemented on all four options. One of the major differences compared to the original version is that the brush itself is also displayed with the brushed data. In the case of parallel coordinates, the brush is just like the grey region of Figure 19, and the brushed data are the polylines inside the brush.

## 5.3 Animation of Multidimensional Multivariate Data

Data animation is no longer considered merely a means to present results. It can also be used as an exploratory tool to look for known phenomena of the data, as well as to investigate unpredicted and significant effects hidden behind the data [BG89a, EKdM$^+$93].

The development of scientific data animation has not been as intensive as static techniques. However, researchers have tried to use animation throughout the course of mdmv visualization development. Any of the techniques we have described can be applied to each step of times series data and then displayed as an animated sequence. Below we discuss three approaches that in some way go beyond just presenting time series of static displays – grand tour methods [BA86], Exvis on a supercomputer [SGB91], and a scalar visualization animation model [BMP91].

### 5.3.1 Grand Tour Methods

Asimov [Asi85] created the grand tour technique to project multidimensional data onto two dimensional planes. Buja and Asimov [BA86] later developed grand tour animation methods. To quote from this paper, "Our methods are based on the simple idea of moving projection planes in high (4–10) dimensional data spaces. That is, we design 1-parameter families of 2-planes in $p$-space, with the parameter being thought of as time." That is to say, suppose we have data with $p + 1$ variates, we take one variate out as a time parameter for animation, and project

the rest of the $p$ variates onto two dimensional planes in rapid succession according to the changes of the time parameter. This technique encodes data in the smooth motion of scatter plots, and provides a multitude of aspects among variates simultaneously.

Grand tour methods were tested with modeled data, as well as real life data from a government housing project and particle physics research. A few observations presented in the paper include:

- Smoothness of motion is a basic requirement for scientific data animation. Jittering causes fatigue for human eyes.

- Excessive within-screen-spin (i.e., rotation that takes place within the projection plane surface instead of the projection spaces) is uninformative and may disturb human eyes. Therefore it should be avoided.

- The two dimensional plane which encodes the velocity, i.e., the speed vector of the data point, needs to be orthogonal to the two dimensional projection plane to avoid confusing location (projection plane) and speed (velocity plane) of the dynamic scatter plot point.

### 5.3.2   Exvis on a Supercomputer

The implementation of the stick figure icon was extended from workstations to a data parallel supercomputer [SGB91], so as to gain sufficient computational power for real-time icon generation. For data with $n$ variates, there are $n!$ possible variate mappings to display parameters in iconography. In this version of Exvis, a set of display controls are included to support real-time interactive analysis. These include the size of the bounding box of each icon, the orientation of each icon with respect to a reference axis, and the maximum amount of random displacement of each icon in the $xy$-directions. Scientists can adjust these control parameters until interesting spots are detected visually from the texture patterns. This kind of continuous parametric interaction can be considered as fine-tuning the physical representation of the data.

### 5.3.3   A Scalar Visualization Animation Model

Bragatto et al. [BMP91] describe a scientific animation model designed and implemented to animate scalar fields of mdmv data. The data supported are multidimensional volumetric, i.e., three dimensional multivariate data defined on a uniform grid. For example, given a three dimensional space, internal pressure is measured regularly over a period of time to create time-dependent scalar data. Three of the possible animation techniques are: a sequence of three dimensional volumes animated according to the time variate, a single static volume if time is not included, or a series of two dimensional planes if one of the three spatial dimensions is dropped. Data can be selected spatially (i.e., a subset of three spatial dimensions) or temporally (i.e., a particular time interval). Data is mapped to either three dimensional shaded blocks with pseudocolor, three dimensional iso-surface blocks, or individual blocks characterized by color and textures for each grid point. An animation is comprised of one or more *scenes*. Each scene contains *actors* (including objects from the data, light, and camera classes), and *actions* (group of events or actions). There is also a system *clock* to control the speed of the animation. Animation is described by a linear-list type language. The system can be extended to multiple scalar data sets and vector fields.

# 6 Conclusions

Most of the mdmv visualization techniques described in this paper were developed and implemented before 1992. Although there are newer tools such as AutoVisual [BF92, BF93], HyperSlice [vWvL93], and XmdvTool [War94], the intensity and variety of the recent activities are not comparable to the period from 1987 to 1991. An interesting fact that emerges from this survey is the remarkable stability of some of the two-way correlation techniques such as the scatterplot matrix, as compared to the generally unverified success of the multiway correlation methods.

We believe that the emphasis of mdmv visualization research must shift away from the design of yet more visual displays towards the rigorous evaluation of experimental visualization techniques. We must learn what approaches actually lead to more accurate results, enhanced productivity, and better understanding of the underlying science.

Finally, we believe that future mdmv visualization research must be integrated with developments in many other areas. Terabyte mdmv data is here. Scientists have to deal with data that is many thousands of times bigger than the number of pixels on a display. Visualization techniques initially motivated solely by visual perceptions are diversified. There is a great deal of current work on the design and implementation of scientific databases, multimedia systems, virtual worlds, and multiresolution analysis supporting mdmv visualization.

## Acknowledgements

## References

[AC91]     Bowen Alpern and Larry Carter. Hyperbox. In Gregory M. Nielson and Larry Rosenblum, editors, *Proceedings of IEEE Visualization '91*, pages 133–139, San Diego, California, October 1991.

[Asi85]    Daniel Asimov. The grand tour: A tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing*, 6(1):128–143, January 1985.

[BA86]     Andreas Buja and Daniel Asimov. Grand tour methods: An outline. In *Proceedings of the 18th Symposium on the Interface*, pages 63–67. American Statistical Association, 1986.

[BC87]     Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29:127–142, 1987.

[BCH+94]   R. Daniel Bergeron, William Cody, William Hibbard, David T. Kao, Kristina Miceli, Lloyd Treinish, and Sandra Walther. Database issues for data visualization: developing a data model. *Computers and Graphics*, 1994. To be published.

[Bed90]     Jeff Beddow. Shape coding of multidimensional data on a microcomputer display. In Arie Kaufman, editor, *Proceedings of IEEE Visualization '90*, pages 238–246, San Francisco, California, October 1990.

[Bed92]     Jeff Beddow. An overview of multidimensional visualization: Elements and methods. In Jeff Beddow and Cliff Beshers, editors, *Designing a Visualization Interface for Multidimensional Multivariate Data*. IEEE Visualization '92 Tutorial 8, Boston, Massachusetts, October 1992.

[BF92]      Clifford Beshers and Steven Feiner. Automated design of virtual worlds for visualizing multivariate relations. In Arie E. Kaufman and Gregory M. Nielson, editors, *Proceedings of IEEE Visualization '92*, pages 283–290, Boston, Massachusetts, October 1992.

[BF93]      Clifford Beshers and Steven Feiner. Autovisual: rule-based design of interactive multivariate visualizations. *IEEE Computer Graphics and Applications*, 13(4):41–49, 1993.

[BG89a]     R. Daniel Bergeron and Georges G. Grinstein. The impact of scientific visualization on workstation development. In J. L. Encarnacao and Geroges G. Grinstein, editors, *Workstations for Experiments*, pages 3–11. Springer–Verlag, 1989.

[BG89b]     R. Daniel Bergeron and Georges G. Grinstein. A reference model for scientific visualization. In *Proceedings Eurographics '89*, pages 393–399, 1989.

[BMMS91]    Andreas Buja, John A. McDonald, John Michalak, and Werner Stuetzle. Interactive data visualization using focusing and linking. In Gregory M. Nielson and Larry Rosenblum, editors, *Proceedings of IEEE Visualization '91*, pages 156–163, San Diego, California, October 1991.

[BMP91]     Paolo Bragatto, Naida Mazzino, and Patrizia Palamidese. Animated visualization of scalar fields. In *Conference Proceedings of Second Eurographics Workshop on Animation and Simulation*, pages 115–127, Vienna, Austria, September 1991.

[CCKT83]    J. M. Chambers, William S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis*. Chapman and Hall, New York, 1983.

[Che73]     H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of American Statistical Association*, 68:361–368, 1973.

[Cle93]     William S. Cleveland. *Visualizing Data*. Hobart Press, Summit, New Jersey, 1993.

[CMM93]     William S. Cleveland, M. E. McGill, and R. McGill. The shape parameter of a two-variable graph. *Journal of American Statistical Association*, 38:289–300, 1993.

[Col93]     Brian M. Collins. Data visualization – Has it all been seen before? In Rae A. Earnshaw and David Watson, editors, *Animation and Scientific Visualization – Tools and Applications*, chapter 1, pages 3–28. Academic Press, 1993.

[EKdM+93]   Jose L. Encarnacao, Detlef Kromker, Jose Mario de Martino, Gabriele Englert, Stefan Haas, Edwin Klement, Fritz Loseries, Wolfgang Muller, Georgios Sakas, and Ralf Rainer Vohsbeck Petermann. Advanced research and development topics in animation and scientific visualization. In Rae A. Earnshaw and David Watson, editors, *Animation and Scientific Visualization – Tools and Applications*, chapter 3, pages 37–73. Academic Press, 1993.

[FB90a]   Steven Feiner and Clifford Beschers. Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. In *Proc. UIST '90, ACM Symposium on User Interface Software and Technology*, pages 76–83, Snowbird, UT, October 1990.

[FB90b]   Steven Feiner and Clifford Beshers. Visualizing n-dimensional virtual worlds with n-Vision. *Computer Graphics*, 24(2):37–38, March 1990.

[Fis70]   Sir Ronald A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, 14 edition, 1970.

[FvDFH90]   James Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics – Principles and Practice*. The Systems Programming Series. Addision-Wesley, second edition, 1990.

[Gib50]   J. J. Gibson. *The Perception of the Visual World*. Houghton Mifflin Co., Boston, Mass, 1950.

[GPW89]   Georges G. Grinstein, Ronald M. Pickett, and Marian G. William. EXVIS: An exploratory visualization environment. In *Proceedings of Graphics Interface '89*, pages 254–261, 1989.

[HPvW94]   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, March 1994.

[ID87]   Alfred Inselberg and Bernard Dimsdale. Parallel coordinates for visualizing multi-dimensional geometry. In T. L. Kunii, editor, *Proceedings of Computer Graphics International '87*, Tokyo, 1987. Springer-Verlag.

[ID90]   Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In Arie Kaufman, editor, *Proceedings of IEEE Visualization '90*, pages 361–375, San Francisco, California, October 1990.

[IRC87]   A. Inselberg, M. Reif, and T. Chomut. Convexity algorithms in parallel coordinates. *Journal of ACM*, 34(4):765–801, October 1987.

[KHK+94]   Arie Kaufman, Karl Heinz Hohne, Wolfgang Kruger, Lawrence Rosenblum, and Peter Schroder. Research issues in volume visualization. *IEEE Computer Graphics and Applications*, 14(2):63–67, March 1994.

[KK93]   Peter R. Keller and Mary M. Keller. *Visual Cues, Practical Data Visualization*. IEEE Computer Society Press, 1993.

[KK94]      Daniel A. Keim and Hans-Peter Kriegel. VisDB: Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, 14(5):40–49, September 1994.

[KKS93]     Daniel A. Keim, Hans-Peter Kriegel, and Thomas Seidl. Visual feedback in querying large database. In Gregory M. Nielson and R. Daniel Bergeron, editors, *Proceedings IEEE Visualization '93*, pages 158–165, October 1993.

[KKS94]     Daniel A. Keim, Hans-Peter Kriegel, and Thomas Seidl. Supporting data mining of large databases by visual feedback queries. In *Proceedings Tenth International Conference on Data Engineering*, pages 302–313, February 1994.

[Lev91]     Haim Levkowitz. Color icons: Merging color and texture perception for integrated visualization of multiple parameters. In Gregory M. Nielson and Larry Rosenblum, editors, *Proceedings of IEEE Visualization '91*, pages 164–170, San Diego, California, October 1991.

[LWW90]     Jeffrey LeBlanc, Matthew O. Ward, and Norman Wittels. Exploring n-dimensional databases. In Arie Kaufman, editor, *Proceedings of IEEE Visualization '90*, pages 230–237, San Francisco, California, October 1990.

[MDB87]     Bruce H. McCormick, Thomas A. DeFanti, and Maxine D. Brown. Visualization in scientific computing. *Computer Graphics*, 21(6):1–14, November 1987.

[MGTS90]    Ted Mihalisin, E. Gawlinski, John Timlin, and John Schwegler. Visualizing a scalar field on an n-dimensional lattice. In Arie Kaufman, editor, *Proceedings of IEEE Visualization '90*, pages 255–262, San Francisco, California, October 1990.

[MTS91a]    Ted Mihalisin, John Timlin, and John Schwegler. Visualization and analysis of multi-variate data: A technique for all fields. In Gregory M. Nielson and Larry Rosenblum, editors, *Proceedings of IEEE Visualization '91*, pages 171–178, San Diego, California, October 1991.

[MTS91b]    Ted Mihalisin, John Timlin, and John Schwegler. Visualizing multivariate functions, data, and distributions. *IEEE Computer Graphics and Applications*, 11(3):28–35, May 1991.

[Nie92]     Gregory M. Nielson. Modeling and visualizing volumetric and surface-on-surface data. In Hans Hagen, Heinrich Muller, and Gregory M. Nielson, editors, *Focus on Scientific Visualization*, pages 191–242. Springer–Verlag, 1992.

[NR91]      Gregory M. Nielson and Larry Rosenblum, editors. *Proceedings IEEE Visualization '91*, San Diego, California, October 1991. IEEE Computer Society Press.

[PG88]      Ronald M. Pickett and Georges G. Grinstein. Iconographics displays for visualizing multidimensional data. In *Proceedings IEEE Conference on Systems, Man, and Cybernetics*, pages 514–519, Beijing and Shenyang, PRC, May 1988.

[Pic70]      Ronald M. Pickett. Visual analyses of texture in the detection and recognition of objects. In B. S. Lipkin and A. Rosenfeld, editors, *Picture Processing and Psycho-Pictorics*. Academic Press, New York, 1970.

[PvW92]      Frits H. Post and Theo van Walsum. Fluid flow visualization. In Hans Hagen, Heinrich Muller, and Gregory M. Nielson, editors, *Focus on Scientific Visualization*, pages 1–40. Springer–Verlag, 1992.

[PW66]       R. M. Pickett and B. W. White. Constructing data pictures. In *Proceedings of Society for Information Display Seventh National Symposium*, pages 75–81, 1966.

[RET$^+$94]  Philip K. Roberston, Rae A. Earnshaw, Daniel Thalmann, Michel Grave, Julian Gallop, and Eric M. De Jong. Research issues in the foundations of visualization. *IEEE Computer Graphics and Applications*, 14(2):73–76, March 1994.

[SBG92]      Stuart Smith, R. Daniel Bergeron, and Georges G. Grinstein. Stereophonic and surface sound generation for exploratory data analysis. In M. Blattner and R. Dannenberg, editors, *Multimedia and Multimodal Interface Design*. ACM Press, New York, 1992.

[SGB91]      Stuart Smith, Georges G. Grinstein, and R. Daniel Bergeron. Interactive data exploration with a supercomputer. In Gregory M. Nielson and Larry Rosenblum, editors, *Proceedings IEEE Visualization '91*, San Diego, California, October 1991.

[SPW92]      Stuart Smith, Ronald M. Pickett, and Marian G. Williams. Environments for exploring auditory representations of multidimensional data. In *Proceedings of the International Conference on Auditory Display*, Santa Fe Institute, Santa Fe, NM, 1992.

[Tay90]      John Keenan Taylor. *Statistical Techniques for Data Analysis*. Lewis Publishers, Boca Raton, Florida, 1990.

[Tuk77]      John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

[vWvL93]     Jarke J. van Wijk and Robert van Liere. HyperSlice. In Gregory M. Nielson and R. Daniel Bergeron, editors, *Proceedings IEEE Visualization '93*, pages 119–125, San Jose, California, October 1993.

[War94]      Matthew O. Ward. XmdvTool: Integrating multiple methods for visualizing multivariate data. In R. Daniel Bergeron and Arie E. Kaufman, editors, *Proceeding IEEE Visualization '94*, pages 326–336, Washington, DC, October 1994.
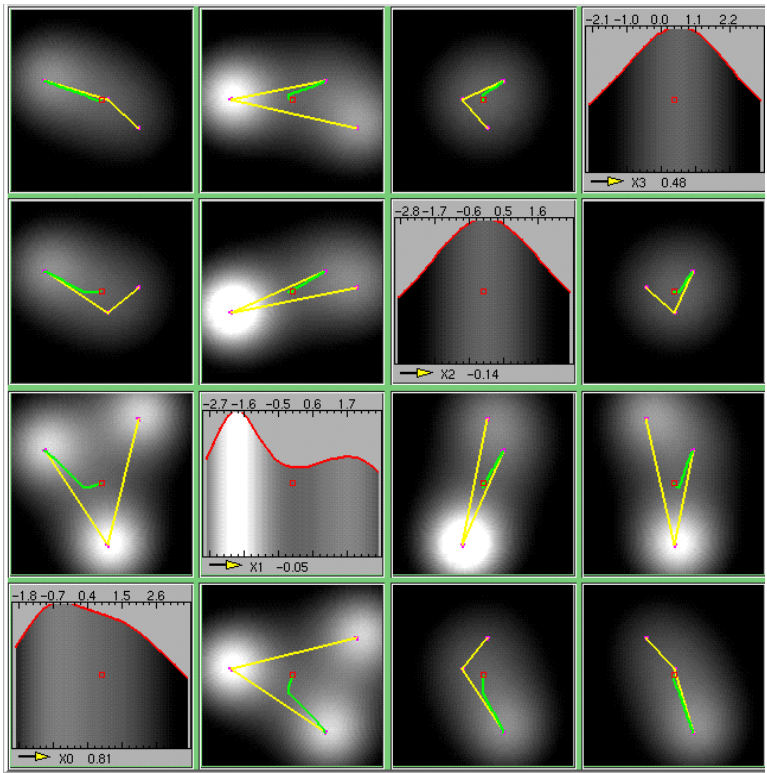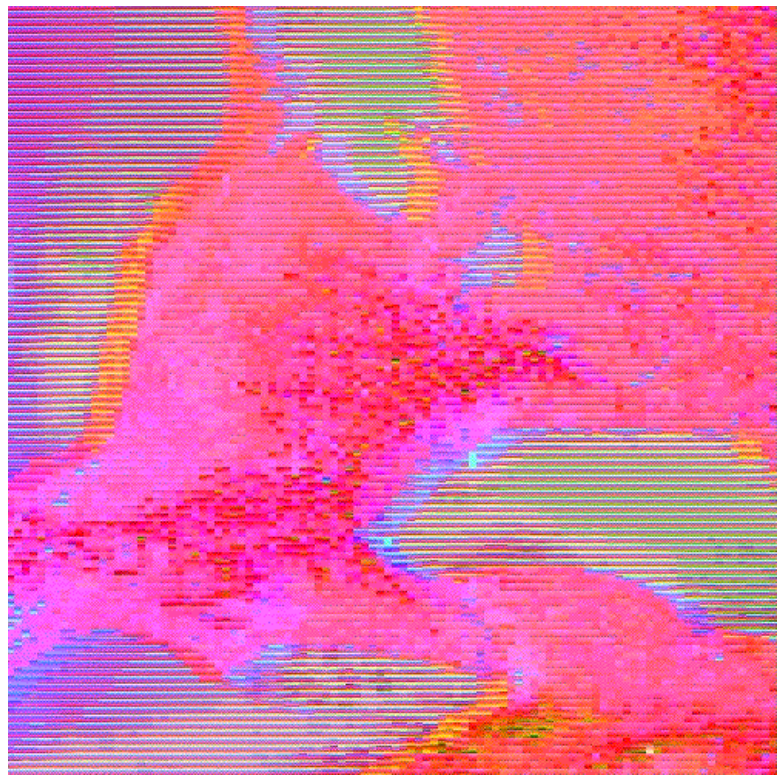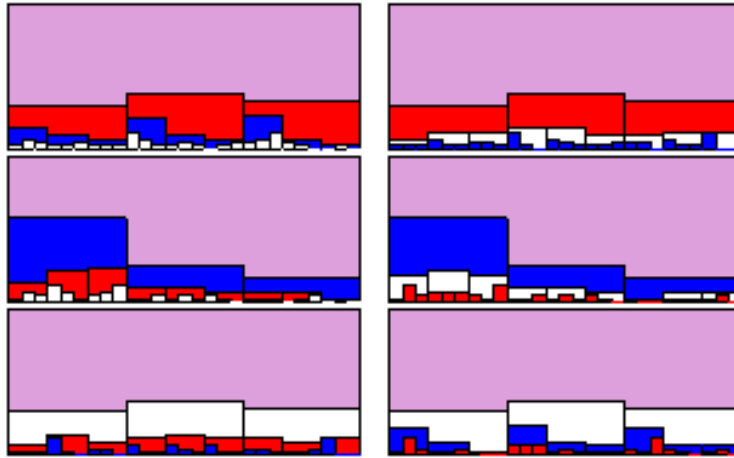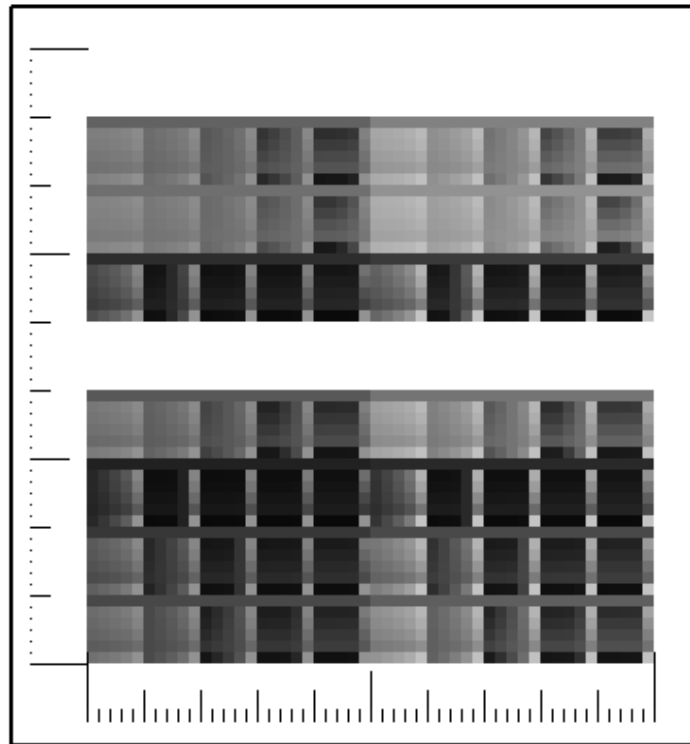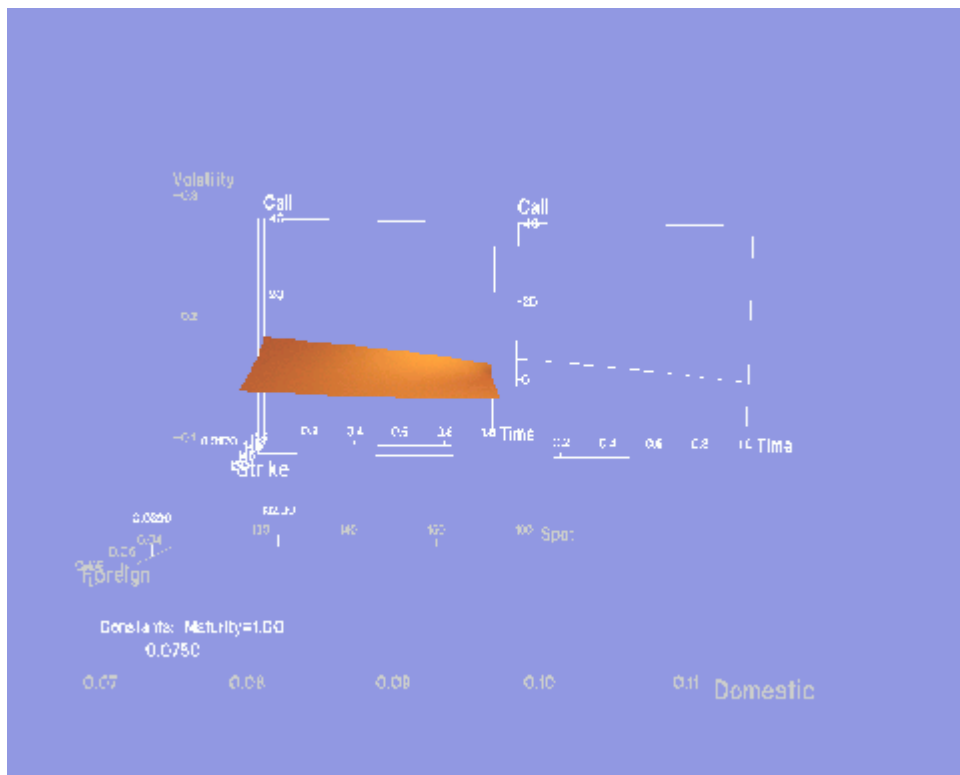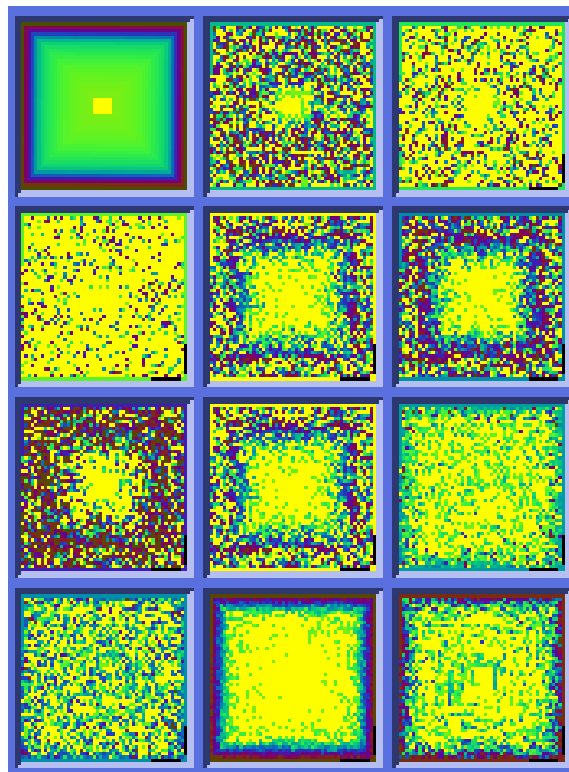
Color Plate 1.
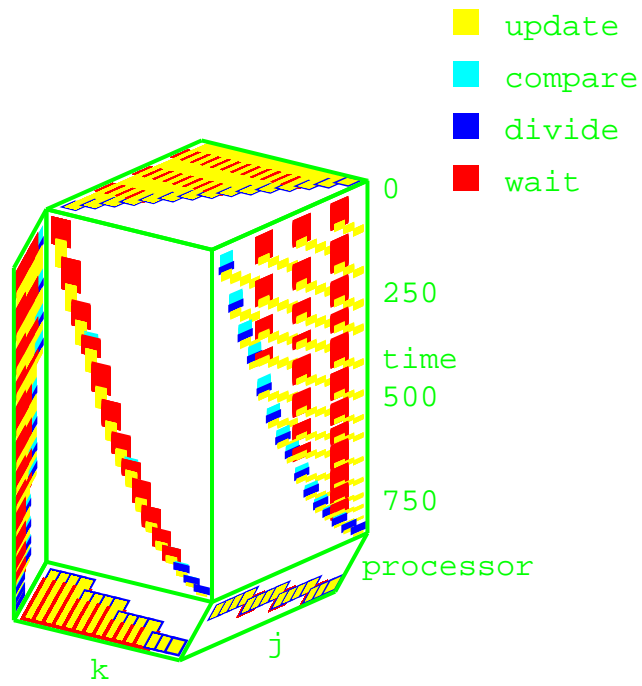


Color Plate 2.
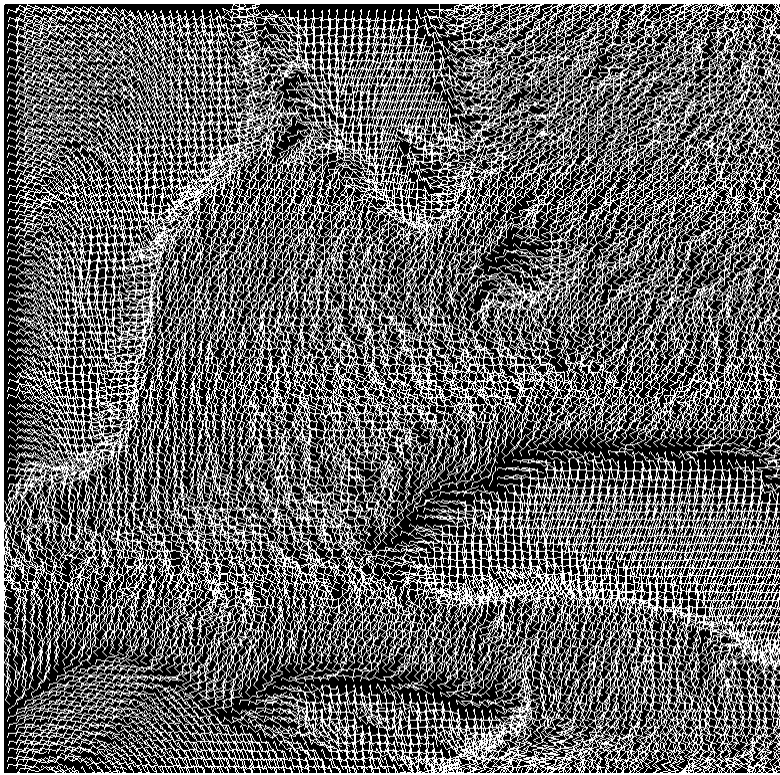
Color Plate 3.



Color Plate 4.

Color Plate 5.



Color Plate 6.

Color Plate 7.



Color Plate 8.