VisBubbles: A Workflow-Driven Framework for Scientific Data Analysis of Time-Varying Biological Datasets

Guangxia Li — Zhejiang University
Andrew C. Bragdon — Microsoft Research
Zhigeng Pan[*] Mingmin Zhang — Zhejiang University
Sharon M. Swartz David H. Laidlaw — Brown University
Chaoyang Zhang Hanyu Liu Jian Chen[†] — University of Southern Mississippi

1 Introduction

Our long-term collaborations with bat flight biologists have revealed that several workflow issues continue to be persistent barriers in scientists' data analysis tasks. Only limited attention has been given to user interfaces that bridge programming and visualization in environments involving multiple information resources, feature extraction by programming, and analytic study. We present VisBubbles (Fig. 1), a unified environment supporting programming, visualization, and interaction concurrently for data analysis workflow. The framework was developed in a participatory design process and discussion with biologists continues to provide new insights into general-purpose support for data-analysis workflow.

2 Design Analysis

Our long-term collaboration with bat biologists has revealed two workflow design issues. The first is that **visual interfaces have not provided enough support in the dynamic data analysis process**. The biologists' analytic process is dynamic, not static. That the path of their exploration is often unpredictable imposes several design requirements. They need easy access to a tremendous amount of data to reason through their analysis, and they need an interface that supports their multiple diverse and simultaneous tasks.

The second issue is that **program implementation and data analysis are conducted in multiple separate working environments**. Evolutionary biologists use Matlab to conduct analysis and then program the results into visualizations for confirmation or generation of new hypotheses. If more analysis is needed, they switch back to Matlab to make changes. Switching back and forth between Matlab and the visualization process introduces significant context switching consts.

3 Our Methods and Preliminary Results

We addressed these issues in the design of VisBubbles, that has two key design apsects. One key aspect of VisBubbles is its extension of static multiple views to a metaphorical interface of bubbles that becomes a flexible layout to support analysis, motivated by the recent success of integrated development environment in Code Bubbles [Bragdon et al. 2010]. Unlike windows, bubbles do not overlap but instead push each other out of the way, thus supporting simultaneous side-by-side comparison and freeing users from the windows arrangement, so they can focus on their analytical tasks. Bubbles can be moved freely in space to leverage the action sequence of the data analysis process (considering humans' limited mnemonic abilities of 7 ± 2 items). Bubbles can also be grouped to form a flexible multiple view environments. When grouped, data are automatically linked to support interactive queries in such a way that interacting in one view activates the same interactions in all other grouped views.

[*]e-mail: zgpan@cad.zju.edu.cn
[†]e-mail: jian.chen@usm.edu

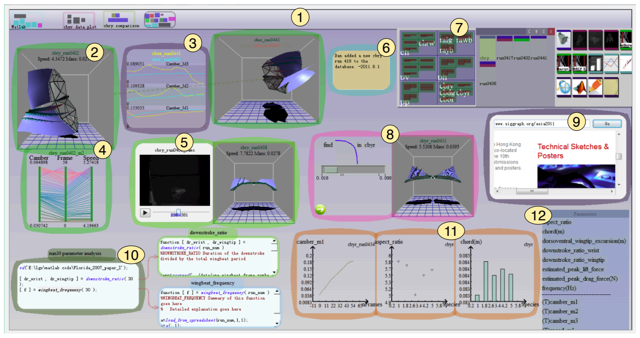

Figure 1: *VisBubbles interface. An example visual interface that a biologist can build using VisBubbles, which contains (1) a virtual work space and the following bubbles: (2) geometry, (3) comparison plot, (4) parallel coordinates, (5) video, (6) note-taking, (7) data source manager, (8) sketch bubbles for shape fitting, (9) web bubbles, (10) Matlab programming bubbles, (11) 2D plots, and (12) authorized data.*

In a nutshell, biologists would look at all information to determine which hypothesis is worth consideration. VisBubbles lets the users load datasets of interest by composing the data (item 7 left) and the good default visualizations (item 7 right) by direct dragging. A new bubble pops up to show a visualization, e.g., as a mesh view (Figure 1 (2)) or plots. By doing so, the interface supports presentation of information of heterogeneous types. VisBubbles extends the metaphor of bubbles [Bragdon et al. 2010] in that each bubble represents a function unit that is valid for programming and can be interpreted to create a visualization. Here each visualization is created by a certain data type, e.g., spatial wing data are shown as a three-dimensional (3D) mesh.

Another key aspect is the pipeline approach that combines programming and interaction in such a way that newly authorized data can be visualized without leaving the data analysis environment. Any parameters derived from the Matlab can be placed in the authorized data bubble (Figure 1(12)), which can also be composed with the good default visualizations (Figure 1(7) right) to see newly authorized data (Figure 1(11)). Biologists who used our interface reported that our interface externalized the data analysis sequence by presenting presentation, rather than hiding the query process, which could aid more effective data analysis and student training.

References

BRAGDON, A., REISS, S., ZELEZNIK, R., KARUMURI, S., CHEUNG, W., KAPLAN, J., COLEMAN, C., ADEPUTRA, F., AND LAVIOLA JR, J. 2010. Code bubbles: rethinking the user interface paradigm of integrated development environments. *ACM/IEEE International Conference on Software Engineering*, 455–464.

Copyright is held by the author / owner(s).
SIGGRAPH Asia 2011, Hong Kong, China, December 12 – 15, 2011.
ISBN 978-1-4503-0807-6/11/0012