



POSTERS COMPENDIUM

October 27 – November 1, 2002

Vis2002 Poster Co-Chairs:

Georges-Pierre Bonneau

Rob Erbacher

Kwan-Liu Ma

Infovis2002 Poster Co-Chairs:

Tamara Munzner

Alan Keahey



POSTERS COMPENDIUM

Preface for the InfoVis 2002 Interactive Posters Compendium

InfoVis 2002, the IEEE Symposium on Information Visualization 2002, is now in its eighth year. We continue to be held in conjunction with the IEEE Visualization 2002 conference, and this year we are distributing a combined InfoVis/Vis Posters Compendium at the conference.

We are delighted that the Interactive Poster category, which was first introduced last year, has elicited a strong response from the research community. This year we received 42 poster submissions, of which 28 were accepted. All submissions were reviewed by both of the posters co-chairs.

The Interactive Posters category includes both traditional posters and demonstrations of interactive systems, either live on a laptop or through video. We encouraged both submissions of original unpublished work and submissions showcasing systems of interest to the information visualization community that have been presented in other venues.

A major change for this year is that we are adding a new slot to the main session. This will be a rapid-fire Posters Preview, where each poster author will have two minutes to pique the interest of audience members, who can later see the full poster and discuss the work with the authors at length during the poster session. The poster session is co-located with Monday night's symposium reception.

We gratefully acknowledge the support of Microsoft Research, whose Conference Management Toolkit was a wonderful help in managing the submissions. We thank the other organizers of both InfoVis and Vis, including the Vis Conference Chair Hanspeter Pfister, the Vis Local Arrangements co-chairs JP Lee and Karen Dickie, and especially the InfoVis Publication Chair Matt Ward for exceeding the call of duty in shepherding the creation of this compendium. We also thank the InfoVis Steering Committee, and most importantly the contributors for their support of the symposium.

InfoVis 2002 Posters Co-chairs:

Tamara Munzner, University of British Columbia
Alan Keahey, Visintuit

InfoVis 2002 Posters Program

Interactive Poster: GRUVY: An Annotated Sequence Viewer	6
Kevin Audleman (Brown University)	
Samuel Heath (Brown University)	
kforbes@cs.brown.edu	
Interactive Poster: Towards an XML Toolkit for a Software Repository Supporting Information Visualization Education and Research	8
Jason Baumgartner (Indiana University)	
Katy Borner (Indiana University)	
jlbaumga@indiana.edu	
Interactive Poster: Interactive Space-Time Maps for Document Visualization ..	10
Patrick Chiu (FX Palo Alto Laboratory)	
Khai Truong (Georgia Institute of Technology)	
chiu@fxpal.com	
Interactive Poster: ASRA - Analytic Stimulus Response Animation	12
Ralf Doerner (Fraunhofer)	
Colin Ware (University of New Hampshire)	
colinw@cisunix.unh.edu	
Interactive Poster: Treemap Visualizations of Newsgroups	14
Andrew T. Fiore (MIT)	
Marc A. Smith (Microsoft Research)	
atf@media.mit.edu	
Interactive Poster: A Zooming Model for Geographic Maps	16
Daniele Frigioni (University of L'Aquila)	
Laura Tarantino (University of L'Aquila)	
laura@ing.univaq.it	
Interactive Poster: Visualizing Egocentric Social Networks from Document Collections	18
Douglas Gordin (IBM T.J. Watson Research Center)	
dgordin@us.ibm.com	
Interactive Poster: Visualizing E-Business and Web Services Using Pixel Bar Charts	20
Ming Hao (Hewlett Packard Laboratories)	
Daniel Keim (University of Constance, Germany)	
Umeshwar Dayal (Hewlett Packard Laboratories)	
Fabio Casati (Hewlett Packard Laboratories)	
Akhil Sahai (Hewlett Packard Laboratories)	
Vijay Machiraju (Hewlett Packard Laboratories)	
Gerry Wade (Hewlett Packard Laboratories)	
ming_hao@hp.com	
Interactive Poster: Visual Queries for Finding Patterns in Time Series Data	22
Harry Hochheiser (University of Maryland)	
Ben Shneiderman (University of Maryland)	
hsh@cs.umd.edu	

Interactive Poster: Visual Data Mining with the Interactive Dendogram	24
Peter Imrich (Stony Brook University)	
Klaus Mueller (SUNY Stony Brook)	
Ray Mugno (Stony Brook University)	
Dan Imre (Brookhaven National Lab)	
Alla Zelenyuk (Brookhaven National Lab)	
Wei Zhu (Stony Brook University)	
muellerk@acm.org	
Interactive Poster: Addressing Scale and Context in Source Code Visualization	26
Stoney Jackson (UC Davis)	
Premkumar Devanbu (UC Davis)	
Kwan-Liu Ma (UC Davis)	
jacksoni@cs.ucdavis.edu	
Interactive Poster: Visual Information Retrieval with Bubble World	28
Timothy Jacobs (Air Force Institute of Technology)	
Christopher Van Berendonck (Royal Australian Air Force)	
timothy.jacobs@afit.edu	
Interactive Poster: Baseball Decision Coach	30
Scott Listfield (Visual i/o)	
Doug Marttila (Visual i/o)	
Mark Schindler (Visual i/o)	
Angela Shen-Hsieh (Visual i/o)	
John Wynne (Visual i/o)	
slistfield@visual-io.com	
Interactive Poster: Immersive Information Modeling using Particles	32
Andrew Vande Moere (ETH-Zurich)	
moere@arch.ethz.ch	
Interactive Poster: Streaming Graphics with Dancer	34
Andrew Norton (SPSS Inc)	
Leland Wilkinson (SPSS Inc.)	
leland@spss.com	
Interactive Poster: Ringtuples: Visualization of Transitions Between Two States	36
Rodrigo Obando (Fairfield University)	
robando@fair1.fairfield.edu	
Interactive Poster: Illuminated Diagrams: Using Light and Print to Comparative Advantage	38
W. Bradford Paley (Digital Image Design)	
brad@didi.com	
Interactive Poster: TextArc: Showing Word Frequency and Distribution in Text	40
W. Bradford Paley (Digital Image Design)	
brad@didi.com	
Interactive Poster: Visualizing Protein-Protein Interactions on a Genomic Scale	42
Hilary Spencer (Stanford University)	
Steven Bennett (Stanford University)	

hspencer@stanford.edu

Interactive Poster: Visualisation of Neurophysiological Data	44
Liz Stuart (University of Plymouth)	
Martin Walter (University of Plymouth)	
Roman Borisyuk (University of Plymouth)	
lizs@soc.plym.ac.uk	
Interactive Poster: Interactive Data Visualization using Mondrian	46
Martin Theus (University of Augsburg)	
martin.theus@math.uni-augsburg.de	
Interactive Poster: glorps - charting multi-dimensional data	48
Glenn Thorpe (home)	
glenn@thorpies.com	
Interactive Poster: Illuminating Data Analysis	50
Antony Unwin (University of Augsburg)	
Ulrich Fahrner (University of Augsburg)	
Robert Schmied (University of Augsburg)	
mac@math.uni-augsburg.de	
Interactive Poster: XmdvTool: Interactive Visual Data Exploration System For High-dimensional Data Sets	52
Matthew Ward (WPI)	
Elke Rundensteiner (WPI)	
Jing Yang (WPI)	
Punit Doshi (WPI)	
Geraldine Rosario (WPI)	
yangjing@cs.wpi.edu	
Interactive Poster: Metavisualization of Dynamic Queries	54
Christopher Weaver (U Wisconsin-Madison)	
Miron Livny (University of Wisconsin-Madison)	
weaver@cs.wisc.edu	
Interactive Poster: Variable Density Scroll Areas for Rapid Image Retrieval	56
Louis Weitzman (IBM)	
Jessica Wu (IBM)	
louisw@us.ibm.com	
Interactive Poster: Web Site Visualization Using a Hierarchical Rectangle ... Packing Technique	58
Yumi Yamaguchi (IBM Japan)	
Takayuki Itoh (IBM Japan)	
Yuko Ikehata (IBM Japan)	
Yasumasa Kajinaga (IBM Japan)	
itot@computer.org	
Interactive Poster: eTree: A Browse and Query Interface for Online Communities	60
Jun Zhang (University of Michigan)	
Alison Lee (IBM T.J. Watson Research Center)	
junzh@umich.edu	

GRUVY: An Annotated Sequence Viewer

Kevin F. Audleman

Samuel A. Heath

{kforbes, sah}@cs.brown.edu

Department of Computer Science, Brown University, Providence, RI

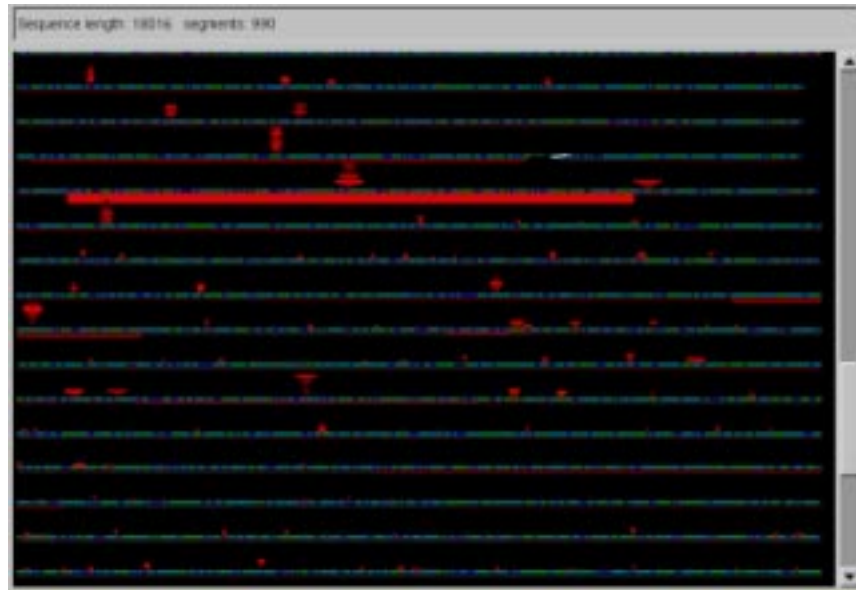


Figure 1: *GRUVY* visualizing output from the SBH algorithm. Six pieces of data are overlaid at each point that describe the effectiveness of the algorithm. *GRUVY* provides an easy way to find interesting events in the reconstruction process.

1 Introduction

In this paper we present GRUVY, a visualization tool to aid in the examination of large sets of DNA sequence data. It provides a graphical representation of the data which enables users to easily extract important information. It also enhances the process of finding correlations between multiple pieces of data by spatially grouping related items. We demonstrate GRUVY's effectiveness on data produced by the Sequencing By Hybridization (SBH) algorithm, and discuss how it may be extended to work with other data sets.

2 SBH Algorithm

SBH is a technique for reconstructing DNA sequences. It recreates sequences of up to 20,000 bases from a complete set of short subsequences, using a family of sub-algorithms which add segments of 1 to 100 bases at a time. The final result of SBH is a string of characters representing the reconstructed sequence, but the interesting data is contained in a log file describing the process of reconstruction. It is possible to characterize the behavior and improve the performance of the SBH method by examining this log data.

However, it is difficult to extract relevant information from the log files. They are megabytes in size, and contain thousands of numerical entries which require close examination to differentiate. It is often important to correlate several different pieces of data, and

the text format of the files makes this difficult. A more effective means of visualizing the data would dramatically speed this process.

3 Data Visualization

GRUVY provides an improved method for examining SBH log data, by deriving a single unified graphical view. The basic structure of this view is a linear sequence. Additional data associated with each part of the sequence is overlaid at the relevant point, allowing regions of interest to be identified at a glance. For instance, a segment that took an unusually long amount of time to reconstruct is indicated in Figure 1 by the strong red bar.

Once such a segment has been identified, other data describing the segment must also be examined to understand how it was produced. All of these parameters are overlaid in the same space in GRUVY, making their examination an easy task. In this example, the relevant parameters are the height, width, and max width of the tree; and the work performed.

There are seven elements of data from SBH that GRUVY displays simultaneously. The SBH algorithm reconstructs the sequence in segments, which are the basic elements of the view. For each segment, the following data is presented:

- The **sub-algorithm** which produced the segment, a discrete variable which takes one of six possible values.

- The **work** required to produce the segment, a continuous variable in the range $[1, \infty)$.
- The estimated **entropy** of the surrounding sequence (measured in bits per nucleotide), another continuous variable in the range $(0, 2]$.
- The **direction** of the segment, a binary value set to either *forward* or *reverse*.
- The **tree** of potential paths explored to produce the segment, including the **depth**, **width** and **maximum width** parameters.

The SBH algorithm can be better understood by examining relationships between these parameters. For instance, the specific nature of the branching process which produces the path trees is not well understood, but a close correlation between the size and shape of the trees and work performed is expected. GRUVY provides a intuitive way to make these connections.

GRUVY also makes it easy to view higher-dimensional correlations which are extremely difficult to examine without a visual tool. For instance, we expect a three-way correlation between the entropy of the sequence, work performed, and the size of the tree. The total size of the tree determines the amount of work required to produce the segment, and regions of the sequence with low entropy should require more work to reconstruct. However, the overall shape of the tree provides invaluable insight into the actual behavior of the algorithm when difficult segments are encountered. This complex correlation would have been nearly impossible to examine without GRUVY, as it is difficult to present a three-dimensional table containing trees in text form.

In general, the algorithm's input parameters and the properties of the reconstructed sequence affect the method's behavior in many different ways. GRUVY provides an integrated way of observing all of these effects, and has proven invaluable in identifying such relationships.

4 Conclusion

GRUVY has proven effective at aiding in the examination of the SBH algorithm. Its unified view of SBH output logs greatly speeds the process of their examination, and provides an intuitive means for identifying relationships between different data elements.

GRUVY's flexible framework can be extended to work with any data based on an underlying genomic sequence—an extremely common format in bioinformatics. We believe it will be easy to apply GRUVY to other sequence-based data sets, such as annotated genomic data from GenBank.

References

- [1] S.A. Heath and F.P. Preparata, Enhanced sequence reconstruction with DNA microarray application. COCOON 2001, (2001), 64-74
- [2] F.P. Preparata, A.M. Frieze, E. Upfal. On the Power of Universal Bases in Sequencing by Hybridization. Third Annual International Conference on Computational Molecular Biology. April 11 - 14, 1999, Lyon, France, pp. 295-301.
- [3] A. Kozik, E. Kochetkova, and R. Michelmore. GenomePixelizer. 2001 <http://www.atgc.org/GenomePixelizer/GenomePixelizer/>.
- [4] O. Bents, A. Dieckmann, U. Nerlich, M. Reinsch, and T. Will. GeneView. 1997. <http://bibiserv.techfak.uni-bielefeld.de/GeneView/>.

- [5] C. Mungall and M. McKeen. Anubis. 1997. <http://www.ri.bbsrc.ac.uk/anubis/userdocs.html>.
- [6] D.L. Van Laar. Psychological and cartographic principles for the production of visual layering effects in computer displays. In *Displays vol. 22 (2001)* 125-135.
- [7] Thomas J. Ball and Stephen G. Eick. Software Visualization in the Large. In *IEEE Computer*, Vol 29, No. 4, April 1996, pages 33-43.

Interactive Poster: Towards an XML Toolkit for a Software Repository Supporting Information Visualization Education and Research

Jason Baumgartner and Katy Börner
 Indiana University, SLIS
 Bloomington IN 47405, USA
jlbaumga@indiana.edu, katy@indiana.edu

Motivation

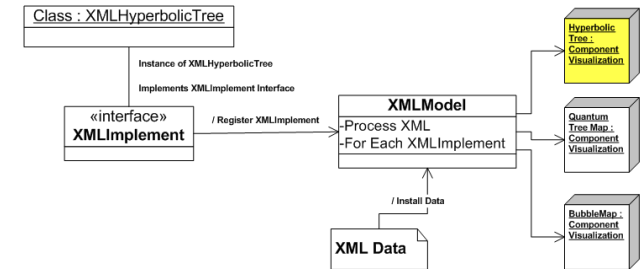
Many information visualizations (IVs) are highly interactive and dynamic. In order to design effective information visualizations, theoretical knowledge on human perception as well as available approaches, techniques and their evaluation is essential. In addition, practical skills are needed to select appropriate algorithms, interaction metaphors, and visual designs that best fit a user group, their tasks, and their data. A number of excellent IV textbooks have been published recently by Card et al. (1999), Chen (1999), Ware (2000), Spence (2000), and Dodge & Kitchin (2000). However, 2-dimensional printouts on paper often cannot convey the true visual appearance and interactive performance of IVs. While several textbooks come with accompanying web sites containing screen-sized snapshots of user interfaces as well as animations and movies, none of them facilitates the exploration, application, evaluation, and comparison of algorithms.

XML Toolkit

Since 2000, our research group is working on the design of an *Information Visualization Data and Software Repository* (Börner & Zhou, 2001) for teaching and research purposes. Currently, the repository¹ provides access to diverse data mining and dimensionality reduction techniques such as Latent Semantic Analysis, Multidimensional Scaling, and Pathfinder Networks, as well as to spatial layout algorithms such as the GRIDL GGraphical Interface for Digital Libraries, diverse Treemap algorithms, Force Directed Placement, and Hyperbolic Tree algorithm.

This poster introduces an XML toolkit that extends the repository to create a unified architecture in which multiple data analysis and information visualization algorithms can be incorporated and combined. The toolkit provides an XML-based interchange format to unify data input, interchange, and output formats. Factory and interface classes allow all software packages to implement and to use the standard XML

format and ensure that packages can be easily interchanged, compared, and combined. Simple configurations of the XML input format suffice to use different algorithms in a wide variety of applications. All Java-based algorithms can be run in stand-alone



mode as an applet or application.

Figure 1: General Architecture of the XML Toolkit

The general structure of the IV repository XML toolkit, depicted in Figure 1, relies on the use of factory and interface classes to instantiate and populate the various visualization algorithms. The algorithm classes must implement the XMLImplement class to be registered with the toolkit. The XML data and the interfaced objects are managed by a model that controls access of the data and its population into the objects. Thus a Hyperbolic Tree, a Quantum Treemap, and a Bubblemap can all implement this interface, register with the model, and then concurrently visualize and interact with the same dataset.

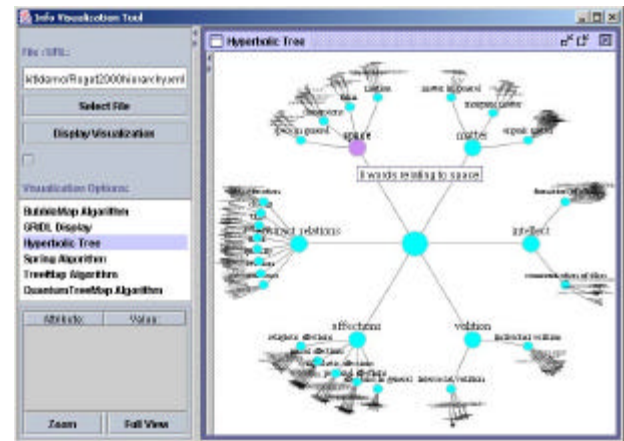


Figure 2: Graphical User Interface to the Current Software Repository

Non-programmers with a basic understanding of the included algorithms can use a graphical user interface to visualize their data sets. Figure 2 shows a snapshot of the interface with a visualization of Roget's Thesaurus using a hyperbolic tree layout (Baumgartner & Waugh, 2002). The left control panel can be used to load XML

¹ <http://ella.slis.indiana.edu/~katy/L697/code/>

formatted data files, e.g., “Roget2000hierarchy.xml”, and to select the preferred “Visualization Option”, e.g., “Hyperbolic Tree”. Upon clicking “Display Visualization” the data set is displayed for interactive exploration in the right window. The control panel can be minimized and a control panel specific to the Hyperbolic Tree can be opened to choose among diverse display options. Alternatively, a different layout algorithm suited to a hierarchical data set, e.g., Treemaps, can be selected. Multiple visualizations of one data set can be generated simultaneously for comparison.

The repository was used in the Information Visualization course taught at Indiana University to complement the theoretical study of information visualization techniques and to augment the critique and evaluation of existing applications. It facilitates the design, comparison, and customization of IVs for different user groups, visualization tasks, and data sets. Students can concentrate on interface and interactivity aspects of visualizations. In addition, the repository enabled students to take on class projects of considerable complexity. Final projects completed in 2001 resulted in multiple conference and workshop papers (Baumgartner & Waugh, 2002; Feng & Börner, 2002; Mongin, Mostafa, & Fieber, 2001; Paolillo & Heald, 2002). Paper write-ups based on the 2002 final project results are currently under review.

The toolkit is freely available for non-commercial purposes. Our hope is that it will facilitate easy sharing, comparison, and evaluation of existing and new IV algorithms and help to collectively understand the underlying issues of differing visualizations, and to pool together existing and future IV efforts.

Outlook

In the near future, users will be able to use the XML toolkit to save interaction data such as manipulation changes in the data, the state of the visualization, etc. that could be advantageous to compare visualizations of different data sets among others.

Metadata schemas, like the Dublin Core and the Resource Description Framework (RDF), will be employed to provide an interoperable way to represent meaning with data. There will be two schema sets, one for retaining information about a data resource and another one to provide information for a particular service. The Dublin Core tag set will be used as the framework for the data schema, which will store information about a given resource. There will also be a schema set that extends the Dublin Core and RDF to represent information about displaying content in a given IV algorithm. The schema set will be registered with the Open Archives Initiative (OAI) protocol (Lagoze & Sompel, 2001) to allow the greatest interoperability of the data.

Furthermore, the existing schemas will all be centered on exclusive document description, vector graphic markup, geographical layout, etc. The full schema set, that is currently under review, defines simple format objects, encapsulates the more complex format structures through importing the Scalable Vector Graphics 1.0 (<http://www.w3.org/TR/SVG/>) specification and uses it within a given structure that subclasses the top-level container class for that specification. A description of the full schema set is available for review and comments at <http://ella.slis.indiana.edu/~jlbaumga/infovis/infovis.rdf>.

Acknowledgements

We are grateful to the students taking the IV class at Indiana University in Spring 2001 and 2002. They have been and still are a great motivation for the on-going design and expansion of the software repository. Ben Shneiderman and his research group generously contributed source code to the repository.

References

- Baumgartner, J., & Waugh, T. (2002, January 20-25). *Roget2000: A 2D hyperbolic tree visualization of Roget's Thesaurus*. Visualization and Data Analysis. Proceedings of SPIE, San Jose, CA, USA.
- Börner, K., & Zhou, Y. (2001, July 25-27). *A Software Repository for Education and Research in Information Visualization*. Information Visualisation Conference, London, England: IEEE Press, pp. 257-262.
- Card, S., Mackinlay, J., & Shneiderman, B. (Eds.). (1999). *Readings in Information Visualization: Using Vision to Think*: Morgan Kaufmann.
- Chen, C. (1999). *Information Visualisation and Virtual Environments*. London: Springer Verlag.
- Dodge, M., & Kitchin, R. (2000). *Mapping Cyberspace*: Routledge.
- Feng, Y., & Börner, K. (2002, January 20-25). *Using Semantic Treemaps to Categorize and Visualize Bookmark Files*. Visualization and Data Analysis. Proceedings of SPIE, San Jose, CA, pp. 218-227.
- Lagoze, C., & Sompel, H. V. (2001). *The Open Archives Initiative: Building a low-barrier interoperability framework*. First ACM+IEEE Joint Conference on Digital Libraries, Portland, Oregon, USA: ACM Press.
- Mongin, L., Mostafa, J., & Fieber, J. (2001). *Using a Spring Embedding Algorithm to Display Term Relationships from a Medical Concept Discovery System*. Annual Conference of the American Medical Informatics Association, pp. 661-662.
- Paolillo, J., & Heald, D. (2002). *Demographic Participation in the Discursive Management of Usenet*. 35th Hawaii International Conference on System Sciences, Wailea, Maui, Hawaii: IEEE Press.
- Spence, B. (2000). *Information Visualization*: Addison-Wesley.
- Ware, C. (2000). *Information Visualization: Perception for Design*: Morgan Kaufmann.

Interactive Poster: Interactive Space-Time Maps for Document Visualization

Patrick Chiu
FX Palo Alto Laboratory
3400 Hillview Ave., Bldg. 4
Palo Alto, CA 94304 USA
chiu@fxpal.com

Khai N. Truong
College of Computing & GVU Center
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
khai@cc.gatech.edu

Abstract

This work presents constructs called interactive space-time maps along with an application called the SpaceTime Browser for visualizing and retrieving documents. A 3D visualization with 2D planar maps and a time line is employed. Users can select regions on the maps and choose precise time intervals by sliding the maps along the telescopic time line. Regions are highlighted to indicate the presence of documents with matching space-time attributes, and documents are retrieved and displayed in an adjoining workspace. We provide two examples: (1) organizing travel photos, (2) managing documents created by room location-aware devices in a building.

1. Introduction

Document collections can be visualized and browsed through space and time attributes associated with the documents. By making use of these natural attributes, users are freed from having to deal with filenames and directories that are more suitable for computers than people. The space or time attribute may be associated with a document's content or context. Examples of each type are: the address of a house for sale in a real estate listing (space-content), the date of an event in a news report (time-content), the room where notes were taken (space-context), and the revision time of a paper draft (time-context).

Graphical user interfaces have been designed for handling documents that focus either on location (see [7]) or time (see [4], [6]). A unified visualization of both the space and time dimensions can be achieved with "space-time diagrams", which are commonly used to picture physical phenomena and examples of these may be found in popular science books (e.g. [5]).

In this paper, we present a design of interactive and dynamic space-time diagrams for information visualization on computers, which we call *interactive space-time maps*. These are integrated into an application called the *SpaceTime Browser*. We illustrate these concepts with two concrete examples: organizing travel

photos (Fig. 1), and managing documents created by room location-aware devices in a building (Fig. 2).

2. Interactive Space-Time Maps

Our design of interactive space-time maps employs a 3D visualization with one dimension (the vertical z-axis) for time and the other two dimensions for space (see left part of Fig. 1). Spatial information is represented by a rectangular map that contains an image with active regions; e.g. a map of a geographical area (Fig. 1) or a floor plan of a building (Fig. 2). Regions are labeled with tooltips that pop up when the cursor hovers over the region. A user can select a region (e.g. an island in Fig. 1 or a room in Fig. 2) by clicking on it with the mouse. Zooming in or out of a region is performed through a context menu brought up by a right-click on the region.

Abstract geometric regions may also be used in the image of a space-time map to represent virtual places or to consolidate remote sites. Data spatialized and reduced to 2D, from simple scatterplots to results of techniques such as multidimensional scaling, provides another source of maps.

The timeline is telescopic: a segment can be expanded/contracted by right-clicking on it and dragging down/up. Granularity levels are given in familiar units of years, months, weeks, days, etc. A handle allows the user to slide a map up and down along the time axis to choose a time point for browsing. Two sliders are provided for selecting how much time to include before and after a chosen time point (Fig. 2).

If there are documents whose time attributes fall into the selected time interval, the regions on the space-time map that match the space attributes of these documents are highlighted in color (Fig. 1). Selecting highlighted regions retrieves the documents satisfying the space-time constraints; these documents are displayed in a separate workspace (see right part of Fig. 1).

3. Space-Time Browser

The SpaceTime Browser is an application that has space-time maps integrated with a document workspace

(Fig. 1). It pulls data from information repositories containing documents tagged with key space-time attributes. Documents are retrieved dynamically when the user selects time intervals and regions.

A region that has been selected has a line connecting it to the workspace's left edge. Displayed in the workspace are the documents whose space and time attributes belong inside the selected regions and time intervals. A document's border is colored to reflect information about its time and space attributes. Vertical border edges, parallel to the time axis, are colored to indicate whether the document's time attribute is at (black), before (shade of red), or after (shade of blue) the time coordinate of the space-time map. Horizontal border edges are colored with the highlight color of the matching region. See Fig. 1.

Documents are faded to indicate how relevant their space and time attributes are to the selected regions and time intervals. One factor affecting relevance comes from sensor uncertainty when automatic location detection systems are used (see [2]). A relevance score incorporates location uncertainty together with how far the time attribute is from the time coordinate of the map. Normalizing the relevance score between 0.0 and 1.0 yields the alpha value for fading. As shown in Fig. 1, the upper-left photo is noticeably faded.

Multiple maps can represent several floors of a building (Fig. 2), or two copies of a space-time map can show the same regions at different times. Adding a map is performed by activating a context menu at the desired position and choosing the Add Map command. A user can move documents to different maps. A document can be dragged from the workspace to the desired region on a map. An example of this is moving a meeting agenda to a conference room for next week's meeting.

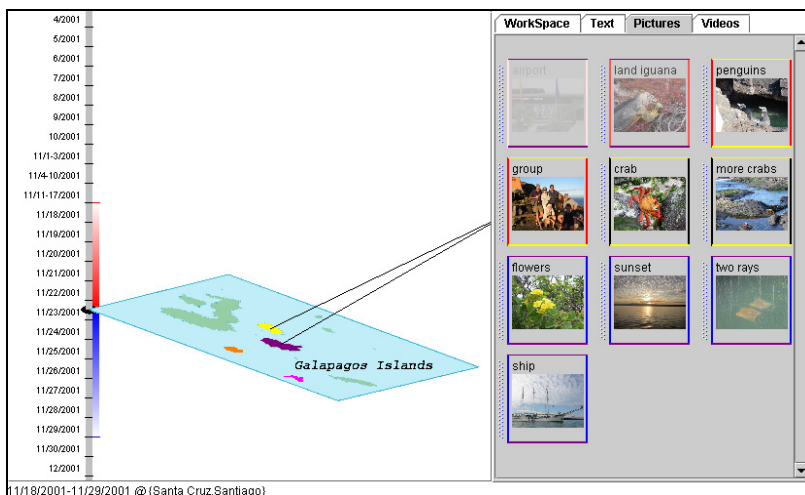


Figure 1. Screenshot of SpaceTime Browser. A space-time map of the Galapagos Islands is integrated with a workspace on the right for viewing photos and other types of documents.

4. Conclusion and Future Work

We presented interactive space-time maps and the SpaceTime Browser for visualizing and retrieving documents. We believe that people tend to find it easier to identify documents with times and places than with made-up filenames and hierarchical directories. Our plan is to deploy the SpaceTime Browser and learn more about the effectiveness of this technique, beginning with multimedia content created by digital cameras and documents captured in different rooms and times ([1], [2], [3]). As information appliances become ubiquitous and location-aware [7], new techniques such as interactive space-time maps will be helpful to manage the mass of data.

5. References

1. Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R., and Pinkerton, M. Cyberguide: A mobile context-aware tour guide. *ACM Wireless Networks*, 3 (1997), pp. 421-4333.
2. Castro, P., Chiu, P., Kremenek, T., Muntz, R. A probabilistic room location service for wireless networked environments. *Proceedings of Ubicomp '01*, pp. 18-34.
3. Chiu, P., Kapuskar, A., Reitmeier, S., and Wilcox, L. Room with a rear view: Meeting capture in a multimedia conference room. *IEEE MultiMedia*, 7, 4 (Oct-Dec 2000), 48-54.
4. Freeman, E., Gelernter, D. LifeStreams: A storage model for personal data. *ACM SIGMOD Bulletin*, (Mar 1996), 80-86.
5. Hawking, S. *A Brief History of Time*. 10th anniversary edition (1998). Bantam Doubleday.
6. Rekimoto, J. Time-machine computing: A time-centric approach for the information environment. *Proceedings of UIST '99*, pp.45-54.
7. Want, R. and Schilit, B. (Eds.). Special Issue: Location-Based Computing. *IEEE Computer*, 34, 8 (Aug 2001).

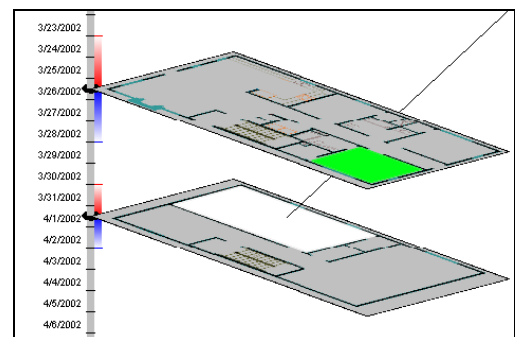


Figure 2. Detail of multiple space-time maps and time sliders. Two floors of Georgia Tech's Aware Home building are shown with rooms dynamically highlighted.

Interactive Poster: ASRA - Analytic Stimulus Response Animation

Ralf Dörner
Fraunhofer AGC
doerner@agc.fhg.de

and

Colin Ware
University of New Hampshire
cware@ccom.unh.edu

1. What is ASRA?

ASRA (Analytic Stimulus Response Animation) is a new animation-based technique for interactive multi-dimensional data analysis. The technique affords the tight coupling of visualization and user interaction and it is especially meant to support the user in exploring the data visually in the early stages of data analysis, e.g. for identifying clusters or outliers. Since this type of exploration is not quantitative or automated, we want to give the user new means of exploiting her knowledge and visual capabilities when dealing and interacting with large or complex data sets. ASRA postulates that data is represented as data avatars in a two- or three dimensional arena. We map data attributes to behavioral response functions and thereby create a simple stimulus response animation system that specifies how the data avatars are animated in an arena. A user can infer from the resulting behavior of the data avatars how the corresponding data is related. In addition, the user can interact with the visualization by creating stimuli (e.g. dragging of objects or change of parameters in the behavior model). ASRA relies on the abilities of the human user to analyze patterns, recognize clusters and draw conclusions.

This extended abstract is organized as follows. In the next section we sketch the basic elements of ASRA with a focus on the behavior model that translates properties of the data to animation. In section 3 we deal with facilities for user interaction that are offered by ASRA. Following this we briefly discuss advantages and disadvantages of ASRA and we conclude with an example and a few points regarding our implementation.

2. ASRA Basics

In ASRA we distinguish two different spaces: the feature space defined by the attributes of the data and the arena space. The data sets we wish to analyze can be thought of points in feature space. Each point in the feature space is associated one-to-one with a data avatar in arena space. The values the point has in each dimension of the feature space are considered to be attributes of the data avatar. A data avatar can be thought of as a graphical representation of the point in feature space, e.g. it can be a geometrical object, a text string or a picture or icon. Since the feature space can have an arbitrary number of dimensions, our data avatars may have an arbitrary number of

corresponding attributes. Arena space, however, can only have up to 3 dimensions. The general idea is that the movement of the data avatars and their spatial relation to each other tells us something about relationships of the data in feature space. Therefore, we define stimuli based on the attributes of the data avatars and the response is determined by both the external stimuli and the internally stored attributes. For instance, if a data avatar discovers another data avatar that has similar attributes the resulting behavior is a movement towards this avatar. Stimuli can not only occur between data avatars but they can also be introduced by a user (e.g. by mouse clicks). Thus, data avatars exhibit behavior as a result of interaction with other avatars in their environment, static features of the environment (such as the walls) and user input.

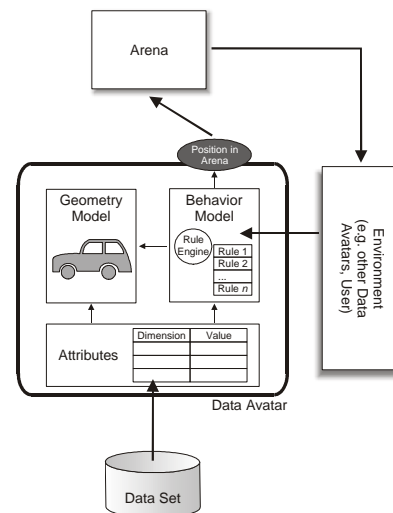


Figure 1: Architecture of a Data Avatar

Note, that our technique is not a projection of a multidimensional feature space to a two- or three dimensional space – the absolute position of a data avatar has no meaning. Only the spatial relationships and the movements of avatars are of interest. Therefore the behavior model of the data avatar is critical. We use a simple rule based model. Each rule describes the relationship between a stimulus and a tendency to move in the arena space. In the above example, a rule could state that if there is another data avatar with similar attributes in a certain radius (stimulus) that we move towards this data avatar and change the color (response). Typical inter-

avatar behavior rules are based on similarity metrics. These govern the strength of attraction or repulsion forces causing movements within the arena. We can also write rules how a data avatar reacts on user interactions. This is one of the main ways to implement our proposed interaction techniques (see section 3). As a summary, the architecture of a data avatar is shown in Fig. 1.

3. Interaction in ASRA

There are three main ways for a user to interact with the animation of the data avatars in the arena. First, the user can manipulate the behavior model, e.g. by enabling or disabling rules, changing rule parameters or changing the attributes of the data avatars. Second, the user can directly explore the arena. For example, she can click on individual data avatars, mark or unmark them, drag them to other data avatars to see whether they are repelled or attracted, or stir the data avatars to investigate whether a structure they formed is stable or not. Third, there are supporting interaction techniques, e.g. for controlling the time parameter of the animation, or to reveal and retrieve information about data avatars.

4. Pros and Cons

Our experiences with ASRA suggests that it is a promising analytic visualization tool having a number of advantages. ASRA encourages interaction and exploration which may bring new ideas in the user concerning the interpretation of the data set. The visualization of the data set is easy to understand and to manipulate even if the data set is higher dimensional and contains a large number of data points. ASRA requires no pre-knowledge about the data (e.g. the number of clusters expected), normalization and similarity metrics do not have to be chosen in advance. The metrics can be exchanged and the transition will be animated. ASRA is able to visualize fuzzy structures in the data and is able to deal with noise.

ASRA has a number of properties that are less attractive. Since position in the arena space has no semantics it is difficult to get a quantitative interpretation of the data.

Interactive changes of the parameters are not reversible in the sense that if one changes a parameter to a new value and then back to its former value, data avatars will not have the same positions they had before the changes. Also, each data avatar may effect all other data avatars, thus resulting in a complexity of $O(n^2)$.

The technique presented is no substitute for more sophisticated data analysis methods (e.g. discriminant analysis or machine learning based methods). However, these methods could be integrated, e.g. as rules leading to a more sophisticated behavior of the data avatars.

5. An Example

The example data reflects 11 properties of different European countries. The user manipulates attraction and repulsion thresholds (using sliders). As a result, the data avatars form clusters in the arena, e.g. the Scandinavian countries cluster in the lower left corner (Fig 2a). After selecting one data avatar (“Denmark”) with the mouse, dragging it in the center and releasing it (Fig 2b), the data avatar moves back to its original cluster (Fig 2c). This indicates that the “Denmark dataset” is similar to the data sets of all other Scandinavian countries.

6. Interactive Demonstration

We implemented the ASRA technique in the visualization tool *AnimVis* using Java2D. We give a live demonstration of the system and invite users to interactively analyze data sets hands on with *AnimVis*. Moreover, we show videos how different data sets were analyzed with ASRA and present a poster that summarizes basic ideas and algorithms of the ASRA technique.

Acknowledgments

This work was supported by a fellowship to the first author within the Postdoc-Programme of the German Academic Exchange Service (DAAD). Support for the second author was provided by NOAA.

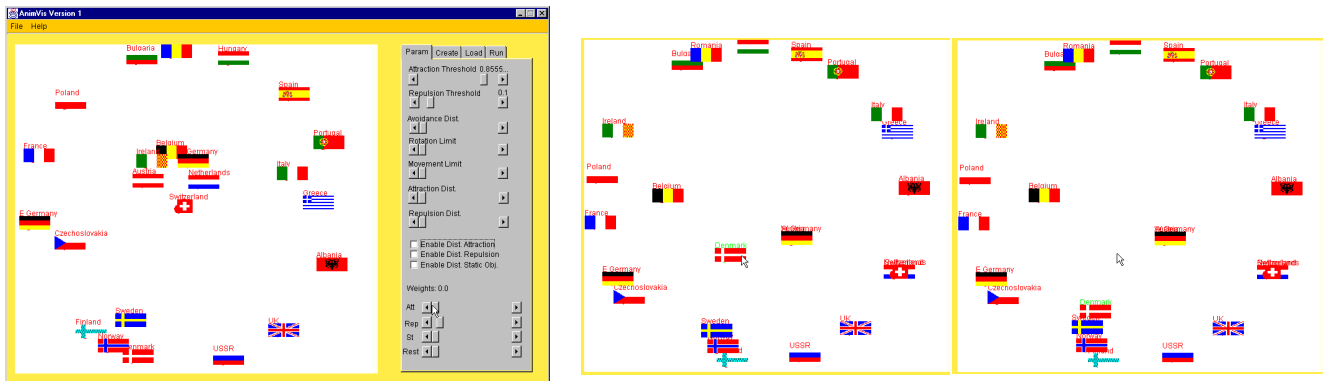


Figure 2 a–c: Screenshots of *AnimVis*

Interactive Poster: Treemap Visualizations of Newsgroups

Andrew T. Fiore
MIT Media Laboratory
E15-468
Cambridge, MA 02139 USA
+1 617 253 4576
atf@media.mit.edu

Marc A. Smith
Microsoft Research
One Microsoft Way
Redmond, WA 98052 USA
+1 425 706-6896
masmith@microsoft.com

Abstract

In this paper, we describe treemap visualizations of Usenet newsgroup. These images illuminate patterns of behavior and suggest interfaces for easier exploration of large-scale social cyberspaces.

1. Introduction

In our treemaps [1, 2, 3] of Usenet, the size of a box reflects the number of messages that have been posted in the newsgroup that it represents. If a newsgroup has children (e.g., *rec.motorcycles* contains *rec.motorcycles.dirt*), they appear inside its bounds, and the parent's size reflects the cumulative number of messages in the newsgroup itself and in all of its children. The number of posts made in the parent group is reflected in the area of the parent that is not occupied by any children.

Assigning colors to the rectangles that represent newsgroups allows us to encode another dimension of information about the groups. Our first treemaps of Usenet indicated growing groups with green and shrinking groups with red; the intensity of the color revealed the extent of the growth or decline. But color can represent other metrics as well. We have also examined treemaps colored by number of posters, number of replies, average message length, and number of.

The metric that determines the size of each newsgroup's rectangle — in this case, number of messages — must aggregate up the hierarchy. (That is, if *comp.sys* has 1,000 messages, then *comp* must have at least 1,000 messages, because it encompasses *comp.sys* and, very likely, other groups as well.) For color metrics, this constraint does not hold, so we are free to use descriptors, like the average message length, that do not add up from child to parent.

To avoid jarring discontinuities and to capture the flavor of subtrees of newsgroups, we determine the color of a newsgroup that contains other groups by taking the weighted average of its children's colors.

2. Learning from treemaps of Usenet

When seen printed in a small space at a low resolution, treemaps lose much of their detail, but high-level patterns

in the structure of Usenet remain visible even when tree maps are presented at limited resolutions (cf. micro/macro design [5]). The relative activity levels of the various hierarchies, for example, become immediately apparent.

Hierarchies vary also in terms of the extent to which they are subdivided. Sub-hierarchies like *alt.music* and *soc.culture* have fairly flat structures, with few further sub-divisions below the third level. In contrast, hierarchies like *microsoft.public* and *comp* have grown deeper structures, with more levels but fewer nodes at each level, so they appear as busy, overlapping spaces.

In the following sections, we present some of our observations gleaned from studying treemaps and then looking to tables of data from the Netscan project to quantify our visual observations.

Top-level hierarchies

The top-level hierarchies, which correspond to the first part of each newsgroup's name, provide the crudest topical organization of the groups. Figure 1 shows a treemap of the activity, by number of posts, of all newsgroups in December 2001. The largest separate regions represent the top-level hierarchies. In this treemap, green groups have grown since November 2001, while red groups have shrunk. More intense color indicates greater change. The activity levels of many newsgroups and sub-hierarchies vary dramatically over time, frequently growing or declining by 30 to 40 percent in a month.

In Figure 1, the *alt* hierarchy looms over the rest, occupying more than half of the image, a massive continent of loosely related newsgroups making up 36 percent of all newsgroups and receiving 47 percent of all messages from 44 percent of all posters. *Alt* has grown so large in part because its newsgroup creation process operates less restrictively than that of the other hierarchies, which follow formally established procedures to create new groups. This means that the most active area of the Usenet is not governed by the same political system that rules the others. Activity does not necessarily equal quality, value, or user satisfaction, but it does demonstrate a way in which different patterns of social regulation affect the growth and structure of social cyberspaces [4].

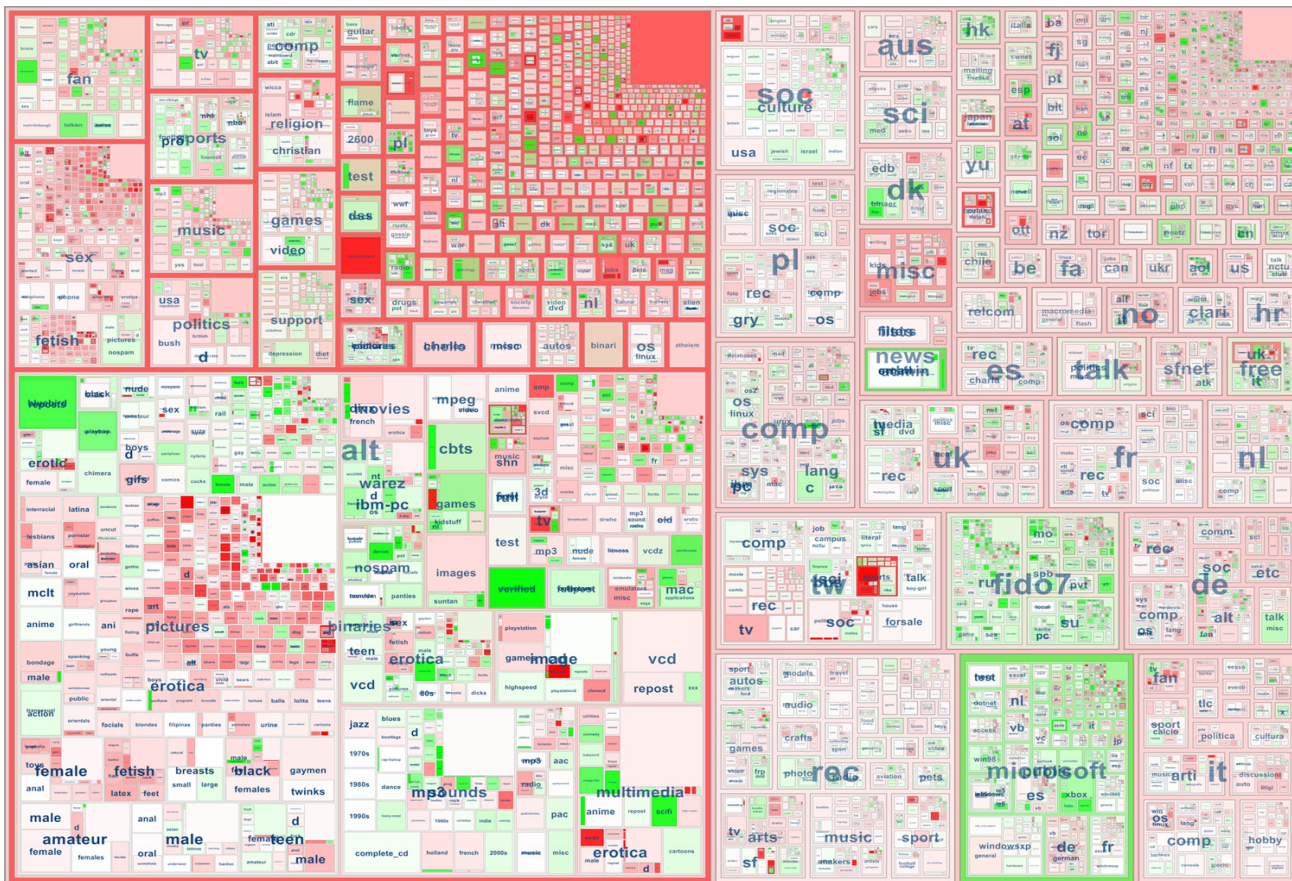


Figure 1. Tree map of all of Usenet newsgroups by number of posts in December 2001. Green groups have grown since November 2001; red groups have shrunk. The intensity of the color indicates the degree of growth or decline.

3. Conclusions

Treemaps hold promise for the study and exploration of large-scale social cyberspaces. These maps capture high-level patterns in such spaces while preserving the ability to examine fine detail.

As an on-screen interface, treemaps suffer from the low resolution of typical display devices. The most rewarding use of treemaps of Usenet comes from up-close study of a high-resolution printed copy. We are presently developing an interactive panning and zooming interface that permits “drilling down” through the hierarchy of newsgroups, which should provide an effective alternative to high resolution for making detail accessible. Such an interface could make the process of finding a group to read quicker and more certain as to the quality of the conversation, especially if the maps were colored by some indicator of interactivity, like number of replies.

4. Acknowledgements

We thank the Netscan team, in particular Duncan Davenport; Helena Mentis of Cornell; and Fernanda Viegas of MIT.

5. References

1. Bruls, M., K. Huising, and J. J. van Wijk. Squarified treemaps. In *Proceedings of Joint Eurographics and IEEE TCVG Symposium on Visualization (TCVG 2000)* IEEE Press, pp. 33-42.
2. Shneiderman, B. and M. Wattenberg. Ordered Treemap Layouts. In *Proceedings IEEE Symposium on Information Visualization 2001*, October 2001.
3. Shneiderman, B. Tree visualization with treemaps: a 2-d space-filling approach. *ACM Transactions on Graphics*, vol. 11, 1 (January 1992), 92-99.
4. Smith, Marc, “Invisible Crowds in Cyberspace: Measuring and Mapping the Social Structure of USENET” in *Communities in Cyberspace*, M. Smith and P. Kollock (eds.), Routledge Press, London (1999).
5. Tufte, E. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire CT, 1983.

Interactive Poster: A Zooming Model for Geographic Maps *

DANIELE FRIGIONI

LAURA TARANTINO

Dipartimento di Ingegneria Elettrica
Università degli Studi dell'Aquila
I-67040 Monteluco di Roio, L'Aquila - Italy
frigioni@ing.univaq.it, laura@ing.univaq.it

Abstract

We discuss a zooming model for geographic maps based on a Level-Of-Detail (LOD) approach, aimed at visualizing sequences of gradually simplified representations of geographic areas. We studied a set of basic zooming primitives as transitions between maps at different LOD, along with properties of the corresponding map space; we introduce the notion of multiple zooming, and briefly recall some theoretical results with practical implications. The theoretical model is implemented in terms of PLA-structures, a kind of topological invariant particularly suitable for the inclusion in a visual interaction environment based on a focus+context approach.

Keywords: Geographic Information System, Focus+Context, Zooming Theory, Topological Invariant.

1 Introduction

We report here some results achieved in the framework of a project aimed at designing a visual interaction environment for Geographical Information Systems (GISs). In particular, we briefly discuss the formalization of a zooming model for GISs suitable for the inclusion in visual interfaces based on the focus+context approach, characterized by fish-eye view visualization, details-on-demand data presentation, and zoom-based navigation. The selective reduction of the information for the peripheral area (the context) is obtained by a combination of *filtering* and *distortion* techniques.

Detail filtering is particularly appropriate in GIS applications, where automated data capturing tools allow

*Partially supported by the Italian Ministry of Education, University and Research (MIUR) under the project *SPADA: Representation and Processing of Spatial Data in Geographic Information Systems* and by the University of L'Aquila under the project "*Representation and Interaction Techniques for Spatial Data*".

to collect huge volumes of geographic data with extreme accuracy. This makes manipulation and rendering of complex geographic datasets beyond the power of state-of-the-art computers. The basic idea is to dynamically adapt the resolution to the needs of specific interaction tasks, possibly taking into account user's interest in displayed data. To achieve this effect, the concept of Level-of-Detail (LOD) has been introduced, sometimes summarized as "*always use the best resolution you need - or you can afford - and never use more than that*". All LOD models aim at defining sequences of gradually simplified representations of the same area that get structured either into a multi-layer model or into a hierarchical model [5]. In other words, a LOD interaction model provides both a *map space* and *rules for traversing it*.

2 Main results of our work

We restrict our attention to databases in which the spatial component is a possibly infinite set of points in the 2-dimensional real space with no unbounded curves. According to our conceptual model, we deal with spatially related collections of *regions* with *features*. As basic abstraction mechanisms for modelling spatially related collections of objects, we consider the *nested partition* and the *aggregation*. Informally, the former is a recursive subdivision of the plane into pairwise disjoint regions of dimension 2 (denoted *blocks*), quite common in geographic maps (consider, e.g., countries partitioned into states, partitioned into counties, etc). The latter defines associations among regions, to model real applications where isolated points, lines, and areas appear within regions (to represent cities, roads, lakes, etc).

Many researchers have proposed partition-based spatial analysis functions, with operations like overlay, generalization, and reclassification, all producing new par-

titions as a result (extended discussion on such topic can be found in [2]). Differently from these works, we propose partition-based primitives to navigate sequences of gradually simplified representations of a given area.

According to a LOD interaction model, we consider a map space and rules for traversing it as transitions between maps at different LOD (i.e., between partitions at different levels of refinement). In a previous paper we proposed a set of basic interaction primitives allowing to require/hide details regarding sub-blocks and features of a given focus region [1]. We found that the complete map space is a sort of generalization of a reduced space including basic transitions deriving only from the primitives concerning blocks (called *zooming* primitives). We hence restrict here our conceptual model to regions participating in the nested partition only, to extend the basic model with new primitives in a scenario without features.

We achieved new results concerning (1) the conceptual model, (2) the map space, (3) a new *multiple zoom-in* primitive, and (4) the interaction model.

The conceptual model. We base the definition of well-formed spatial dataset on the notion of partition refinement, which constitutes the basis of elementary zooming primitives. It can be shown that a well-formed spatial dataset constitutes a poset. Discussions on the use of posets to represent spatial relations appeared in the literature since many years (see, e.g., [3, 4]). In all these papers posets are used to model the topological relations among regions, differently from our work in which posets are used to model the nested partition, and consequently the map space.

The basic map space. Our model is based on a “minimal” (yet powerful) set of basic zooming primitives, denoted \mathcal{ZP}_{base} , which includes two elementary primitives: *zoom-in* adds details regarding sub-blocks of the focus region; *zoom-out* subtracts details related to the focus region. We have shown that \mathcal{ZP}_{base} is *complete* (i.e., we can reach any node from the least refined map), and that it guarantees full *traversability* of the map space (there is a path between any two nodes in the space). These properties allow us to use \mathcal{ZP}_{base} as a kernel for extended set of primitives defined in terms of elementary ones. Extensions are appropriate because the full traversability of the map space associated to \mathcal{ZP}_{base} is only a minimal requirement that does not guarantee *efficient* traversability of the graph (many steps are required to connect two nodes in the map space), and additional transitions should provide shortcuts for selected paths.

Multiple zooming. A new *multiple zooming* primitive is proposed to offer shortcuts for selected paths in the map space. This new primitive is defined as a generalization of *zoom-in* and *zoom-out*, by considering a *set*

of focus regions to zoom-in and/or zoom-out in place of a single focus region at a time. Let P be a partition, and X_{in} and X_{out} two disjoint subsets of P that contain the blocks of P that we want to zoom-in and zoom-out, respectively. Roughly speaking, the execution of $multiple\text{-}zoom(P, X_{in}, X_{out})$ gives rise to the execution of a sequence of zoom-in and zoom-out operations having as focuses blocks of X_{in} and X_{out} , respectively. It is possible to show that the result of $multiple\text{-}zoom(P, X_{in}, X_{out})$ is independent of the order in which blocks of X_{in} and X_{out} are considered.

The interaction model. The interaction model is an intermediate level bridging the gap between the data model and the visual model, providing the structure for the user interaction. Since in a focus+context approach regions get deformed while preserving topology, the interaction model comprises: *PLA-structures* (a kind of topological invariant) to represent topology of maps, and *primitives* as transitions between PLA-structures. To achieve good performances in terms of memory requirements we conform to a true multi-resolution approach by maintaining and storing an implicit representation of the interaction space whose size is polynomial in the number of blocks of the database. To achieve good performances in terms of time requirements the interaction structures are generated during the interaction process following an incremental approach aimed at enriching/reducing the information associated to the current structure.

Future work will proceed along the following research lines: (1) the map space will be further analyzed to characterize types of paths as additional primitives; (2) the navigational model will be extended to include the treatment of features.

References

- [1] S. Cicerone, D. Frigioni, and L. Tarantino. Exploration of geographic databases: Supporting a focus+context interaction style. *J. of Applied System Studies*, 3(2), 2002.
- [2] M. Erwig and M. Schneider. Partition and conquer. In *Proceedings of 3rd International Conference on Spatial Information Theory (COSIT'97)*, volume 1329 of *Lecture Notes in Computer Science*, pages 389–408. Springer, 1997.
- [3] L. Forlizzi and E. Nardelli. Characterization results for the poset based representation of topological relations – i: Introduction and models. *Informatica*, 23(2):223–237, 1999.
- [4] W. Kainz, M. J. Egenhofer, and I. Greasley. Modeling spatial relations and operations with partially ordered sets. *Intl. J. of Geographical Inf. Systems*, 7(3):215–229, 1993.
- [5] E. Puppo and S. Scopigno. Simplification, lod and multiresolution - principles and applications. In *EURO-GRAPHICS'97*, vol. 16. Balckwell Publishers, 1997.

Interactive Poster: Visualizing Egocentric Social Networks from Document Collections

Douglas N. Gordin
IBM T.J. Watson Research Center
dgordin@us.ibm.com

Abstract

Social relations provide primary indices for document collections. Yet, the analysis and visualization of these relations are not well supported by document collection systems. Data extraction and visualization techniques are presented that powerfully illustrate an individual's social network, thereby enabling search, navigation, and discovery in document collections.

1. Introduction

Social relations provide primary indices for document collections. Yet, the analysis and visualization of these relations are not well supported by document collection systems. The utility of social relations for categorizing documents is well documented by social scientists that have identified the crucial role that communities play in the diffusion and creation of ideas. For example, if a specialist is given the title of a document, she might seek to gain further understanding of its perspective by finding out who the authors have studied with or what organization sponsored the work. In general, the formal elaborations of linkages like these are called social networks.

The goal here is to show how a set of three visualizations can powerfully illustrate an individual's social network, based purely on extracting data from a document collection. The individual that will be illustrated in this way is Jock Mackinlay, a pioneer in the field of information visualization. This choice was motivated by his compelling keynote presentation at SmartGraphics 2002 where he gave a two-decade retrospective of his work in information visualization along with his Xerox Parc colleagues. I wondered what I could learn about Mackinlay's social network by analyzing his publications. Using the Raisin system [1], I analyzed the US patents of Mackinlay and his colleagues and produced a suite of three visualizations that I argue provide a compelling index into the work of an influential information visualization creator and his immediate colleagues. Before going through these figures, it is useful to establish some social network terminology.

There are different sorts of social networks with varying utility. Egocentric social networks focus on an individual's network. Typically, they are drawn as a circular pattern with the focal person in the center and

related nodes connected like spokes on a wheel (see Figure 1). In contrast, a relational social network shows aims to show the connections between groups of people (e.g., see Figures 2 and 3). It is also useful to distinguish between *one-mode* and *two-mode* social networks. One-mode social networks are made up solely of people and their relationships (e.g., see Figure 2 and 3). Two-mode social networks contain people and some other type such as events or documents and show the relationship between these two types (e.g., see Figures 1). In graph terms, two-mode networks are bipartite since arcs only occur between nodes of different types. I focus here on egocentric one-mode social networks, that is, understanding one author and his colleagues. However, a two mode network is used to begin the investigation (i.e., a social network of authors and their patents).

2. Extended Example

A suite of three visualizations illustrate Mackinlay's egocentric social network. Figure 1 shows a synoptic visualization of Mackinlay (blue), his patents (green), his co-authors (yellow), and his co-authors' patents (green). Visualizations that trace out a person's influence in this way are called "degrees of separation" networks (as in the popular Six Degrees of Separation drama). Each level of indirection is a degree of separation. This terminology is usually applied to one-mode social networks (i.e., including only people). In contrast, Figure 1 is a two-mode graph, as it includes both people and patents. Hence, I call it a one-and-a-half degree of separation social network where the half a degree denotes the inclusion of a second mode (i.e., patents in this case).

The layout is based on a multi-dimensional scaling algorithm that positions the nodes so that the Euclidean distance is similar to the graph distance (i.e., number of hops between nodes). This layout provides visual clusters of nodes that are meaningful. For example, contrast the group of nodes labeled 'A' with the nodes labeled 'B'. Both sets show documents authored by Mackinlay. However, the former (i.e., 'A') are centrally placed since their co-authors are people with whom Mackinlay collaborated extensively. In contrast, the later (i.e., 'B') are placed more to the periphery since their co-authors were ones he collaborated with much less.

Mackinlay's one -mode social network is shown in Figure 2, though excluding Mackinlay. This graph was produced by modifying Figure 1 as follows: (A) Arcs are

added that connect the co-authors of a patent; (B) All patent nodes are removed; (C) the node for Mackinlay is removed. Removing Mackinlay simplifies the graph, since otherwise he would be connected to every other node. Thus, Figure 2 reveals how Mackinlay's coauthors collaborated with each other. Note the linkages take into account all the patents written by Mackinlay's coauthors, whether or not Mackinlay was an author on them. Again the multi-dimensional scaling layout creates visual clusters. In particular, the cluster of authors at the center are Mackinlay's primary collaborators and, readily identifiable as core contributors to the field of information visualization.

A partial ordering of the authors is shown in Figure 3. This partial ordering (or lattice) is based on co-authorship, so that where author A is considered greater than author B, if author A is a co-author on everything that author B wrote, but author A also wrote other documents as well. In the figure the "greatest" authors are at the top and the authors they "dominate" or "subsume" shown below them. This is not to imply a pecking order, rather it suggests the timeline of people becoming involved in a program of work, and how collaborations both linked and partitioned this group of researchers. For example, this network could be used to track a diffusion of ideas and characterizing the authors' spheres of expertise.

3. Summary and Conclusion

Document collections can improve their search, navigation, and discovery capabilities by providing the means to visualize and analyze egocentric social networks. This point is demonstrated by analyzing the Raisin system on the US Patents of Mackinlay and his colleagues. Other systems, notably Referral Web [2], have shown social networks based on co-authorship. However, Referral Web differs in that only one-mode social networks are shown, no specialized facilities for egocentric networks are provided, and few social network analyses are provided (e.g., no partial orders). In future work I would like to provide the means to flexibly focus on single authors or groups of authors, including finding those individuals that serve as primary linkages between differing communities of practice.

4. References

[1] Gordin, D.N. & Farrell, R.G. Visualizing the Patent Document Collection as a Graph of Inventors and Their Inventions. IEEE Symposium on Information Visualization 2001, October, San Diego, CA.
 [2] H. Kautz, B. Selman, M. Shah, "Referral Web: Combining Social Networks and Collaborative Filtering." *Communications of the ACM*, Vol 40, No 3. March 1997.

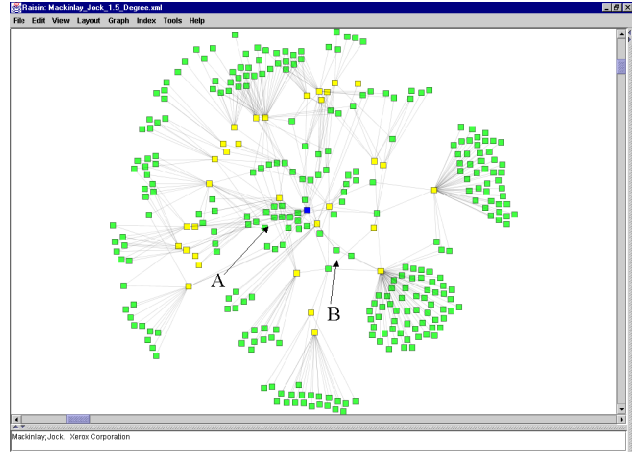


Figure 1: Two-mode social network with 1.5° degrees of separation.

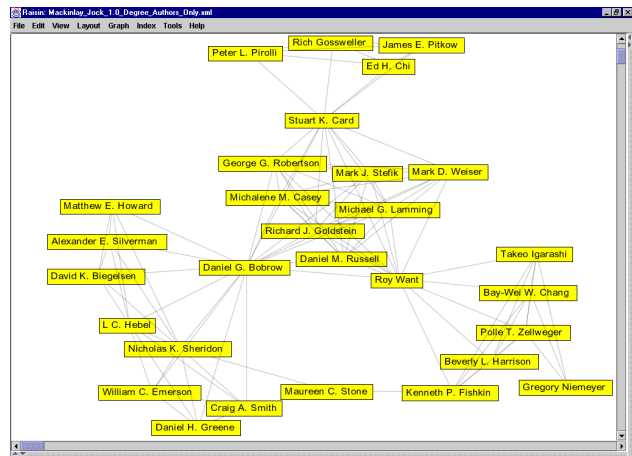


Figure 2: One-mode social network without focal author.

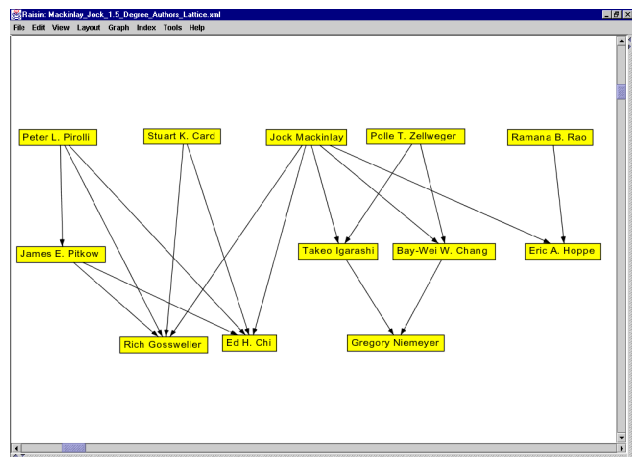


Figure 3: One-mode social network showing partial order between authors of documents.

Interactive Poster: Visualizing E-Business and Web Services Using Pixel Bar Charts

Ming C. Hao, Daniel Keim*, Umeshwar Dayal, Vijay Machiraju, Fabio Casati, Akhil Sahai, Gerry Wade
 Hewlett Packard Research Laboratories, Palo Alto, CA
 (ming_hao, umeshwar_dayal, vijay_machiraju)@hp.com

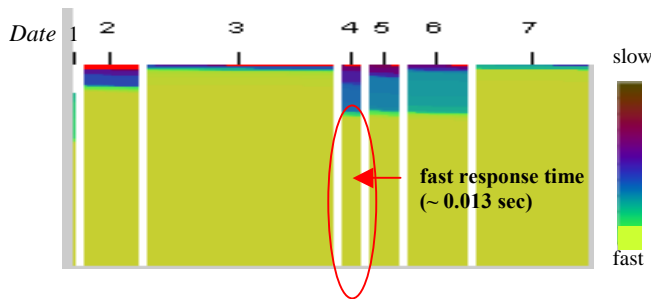
1. Motivation

The rapid growth of electronic business and services has led to the availability of large volumes of data. Business research has been focusing on how to turn raw data into actionable knowledge. In order to find and retain customers, business analysts need to improve their business process quality based on prior information. In particular, it is crucial for companies to analyze their business contracts and to reduce the risk of violation.

Basic bar charts are widely used, but only show highly aggregated data. Finding the valuable information hidden in the data is essential to the success of business. For example, in a weekly business performance report illustrated in Figure 1A, the analyst can easily find that day 4 has the highest response time. However, for analyzing large volumes of business transactions, the visualization of aggregated data is not sufficient. What is needed is not only an overview of the data but also detail information for each data item.



(A) Bar Chart (x-axis = day, y-axis = response time)



(B) Pixel Bar Chart
 (x-axis = day, y-axis = response time, color = response time)

Figure 1: Business Service Response Time Analysis

2. Pixel Bar Charts

In this interactive poster, we describe a number of applications of a new visualization technique - pixel bar charts [1], which have been integrated with a visual data mining system [2]. Pixel bar charts are derived from bar charts. The basic idea of a pixel bar chart is to present the data values directly instead of aggregating them (see Figure 1B). The approach shows each data item (such as a business transaction) by a single pixel in the bar chart. The detail information of one attribute of each data item is encoded into the pixel color and can be accessed and displayed as needed.

2.1. Construction of Pixel Bar Charts

Pixel bar charts integrate the idea of bar charts and X-Y diagrams. As illustrated in Figure 2, to construct a pixel bar chart, we need to specify the following attributes:

- dividing attribute (for between-bar partitioning, D_x, D_y)
- ordering attributes (for within-bar ordering, O_x, O_y)
- coloring attributes (for pixel coloring, C).

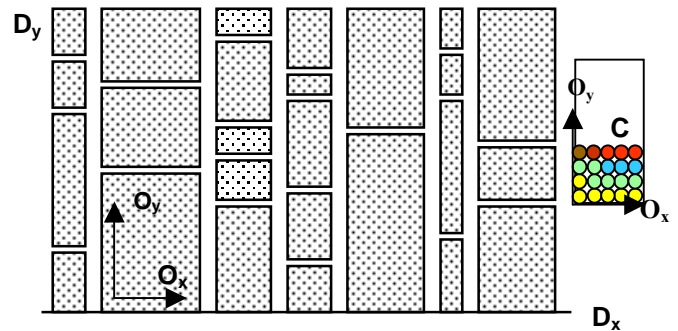


Figure 2: Definition of a Pixel Bar Chart
 (e.g., D_x = day, D_y = process name, O_x = response time, O_y = quality value, C = I/O rate)

The pixel bar chart shown in Figure 1B displays the same data set as Figure 1A. Each pixel represents a service transaction record. It uses *day* as the dividing attribute and *response time* as the y-ordering attributes, i.e. pixels are placed from bottom to top. The color represents the response time *value* of a business transaction. Fast response times correspond to bright colors (yellow, green) and the slow response times correspond to dark colors (blue, red). From the data distribution, we can discover that a large number of fast business services occurred on day 4 (large yellow area). In particular, it is crucial for analysts to realize that the number of transactions processed on day 4 is very small (narrow width bar). These valuable facts get lost in a traditional bar chart. As a result, based on Figure 1A the business analyst may derive a wrong decision.

2.2. Multi-Pixel Bar Charts

In many cases, the data to be analyzed consists of multiple attributes. Different color mappings are used to represent multiple attributes. The dividing and ordering attributes remain the same. In Figure 5, we show an example of two pixel bar charts with *date* as the partition attribute and *CPU utilization* as the ordering attribute. The attributes which are mapped to colors are *I/O rate* and *CPU utilization*.

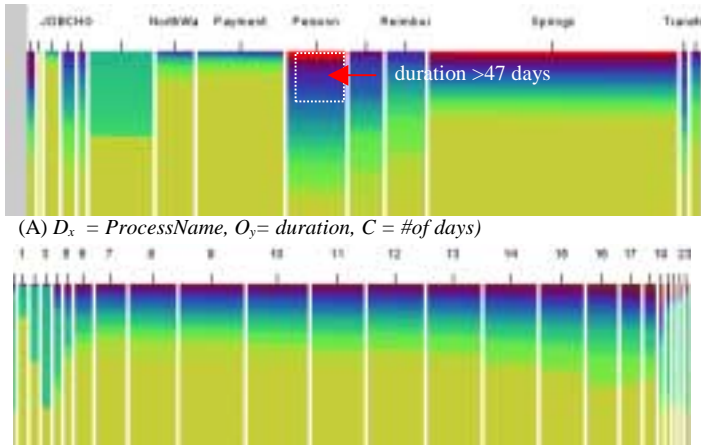
3. Applications

The pixel bar chart technique has been prototyped in several business process and web service applications at Hewlett Packard Laboratories. It has been used to visually mine large volumes of transaction data collected from the HP web sites.

*Presently with University of Constance, Germany, keim@informatik.uni-konstanz.de.

3.1. Business Process Distribution Analysis

Figure 3 illustrates 63,544 business process instances from HP web sites. Each pixel corresponds to a process instance.



(A) $D_x = ProcessName, O_y = duration, C = \#of\ days$
 (B) $D_x = hour, O_y = duration, C = \#of\ days$
 Figure 3: Analysis of Business Process Duration by Date

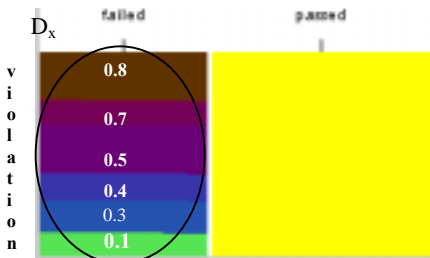
Many important facts may be discovered in Figure 3:

- 1) There are 14 different process names. The process *Springs* has the highest number of business instances and corresponds to the widest bar width. The duration of most processes is low (smaller than 1 day, color yellow) except for the process *Personnel* (more than 47 days, colored dark blue and red).
- 2) The busiest time is between the hours 9 and 14 (see wider width bars in Figure 3B). Evenings have long running processes.
- 3) Large number of business process instances achieves good response times (colors yellow and green).

3.2 Service Level Agreement Analysis

One of the common questions e-business managers ask is how their service agreements perform. Most important is the Service Level Agreements' (SLA) violation level. Figure 4 shows a sample pixel bar chart of 390,000 web service transactions with 986 SLAs. A pixel represents an SLA

(A) Analysis of SLA Status (failed, passed)



drilldown from above "failed" bar

(B) Drill down on failed SLAs

SLA 1000 has the highest violation levels (color dark brown and burgundy)

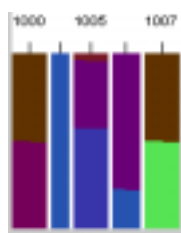


Figure 4 Business Service Level Agreement Analysis

transaction record. The ordering attributes are the SLA violation level from bottom to top. The color is the value of violation level.

3.3. Web Service Correlation Analysis

In Figure 5A, the pixel bar chart shows a month worth of web service CPU utilization numbers (23,731 transactions). Each pixel represents a web service transaction record. The x-axis dividing attribute is date (30 days) and the y-axis ordering attribute is the CPU utilization percentage. The colors in the different bar charts represent CPU utilization and I/O rate. The color red represents web services which exceed a threshold of 99% CPU utilization. Among others, the analysts can obtain the following information from the pixel bar chart:

- a) The highest CPU utilization occurred on 3/18 & 3/19 (Monday & Tuesday) with most red and blue colors.
- b) Web services have different CPU utilization levels during the month.
- c) Web services have large CPU utilization during the weekdays and small CPU utilization during the weekends.
- d) On all days, there are some web services which have a high CPU utilization, i.e. all bars have red areas.

Figure 5B illustrates the correlations between the CPU utilization and the I/O rate. The x-axis dividing attribute is the date and the y-axis ordering attribute is the CPU utilization. The colors in the different bar charts represent the I/O rates. The color red represents transactions of which the I/O rate exceeds 12%. The analysts can make the conclusion that there is a close correlation between CPU utilization and I/O rate. Figure 5 shows that the high I/O rates are corresponding to the high CPU utilization across all 30 days (with the same patterns of yellow, green, blue, and red colors).

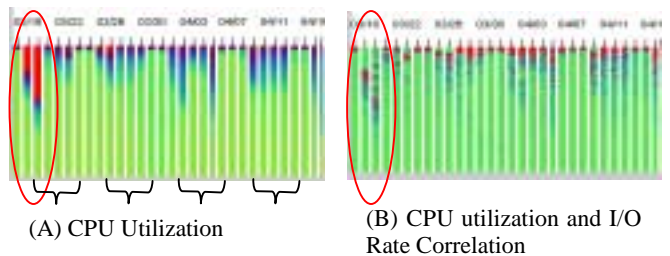


Figure 5: Web Service Multi-Attribute Correlations

5. Conclusion

This poster presents an interactive pixel bar chart technique enabling analysts to visualize large multi-attribute business process and web services. By clicking on a specific pixel or a bar, the user can drill down detail information of the pixel or bar. The interactive poster presentation will show a live demonstration of applications described in this document.

6. References

[1] D. A. Keim, M. C. Hao, U. Dayal, M. Hsu, J. Ladisch "Pixel Bar Ch Charts: A New Technique for Visualizing Multi-Attribute Data Sets without Aggregation", IEEE InfoVis2001, San Diego, CA, 2001.
 [2] M. Hao, U. Dayal, M. Hsu, B. Deletto, J. Becker 'A Java-based Visual Mining Infrastructure and Applications', IEEE InfoVis99, San Francisco, CA. 1999.

Visual Queries for Finding Patterns in Time Series Data

Harry Hochheiser
Department of Computer Science
Human-Computer Interaction Lab
University of Maryland
College Park, MD 20742 USA
hsh@cs.umd.edu

Ben Shneiderman
Department of Computer Science
Human-Computer Interaction Lab,
Institute for Advanced Computer Studies, and
Institute for Systems Research
University of Maryland
College Park, MD 20742 USA
ben@cs.umd.edu

Abstract

This paper introduces Timeboxes: rectangular, direct-manipulation queries for studying time-series datasets. Timeboxes are the primary query tool in our TimeSearcher application, which supports interactive exploration via dynamic queries.

Keywords

Information Visualization, Time Series Data, Dynamic Query, Visual Query

1. Introduction

A time series data set is defined as a set of n items, each containing m real-valued measurements taken at equal time intervals. Timeboxes are rectangular visual query operators that are placed and directly manipulated on a timeline, with the boundaries of the region providing the relevant query parameters.

Timeboxes are rectangular query regions drawn directly on a two-dimensional graph. The extent of the Timebox on the time (x) axis specifies the time period of interest, while the extent on the value (y) axis specifies a constraint on the range of values of interest in the given time period. More specifically if an item in a data set is to satisfy timebox that goes between (x_1, y_1) and (x_2, y_2) , for every point in the time range $x_1 \leq x \leq x_2$, the value of that item must be in the range $y_1 \leq y \leq y_2$ (assuming $y_2 \geq y_1$ and $x_2 \geq x_1$). Multiple timeboxes can be drawn to specify conjunctive queries.

To create a timebox, users click on the desired starting point of the timebox and drag the pointer to the de-

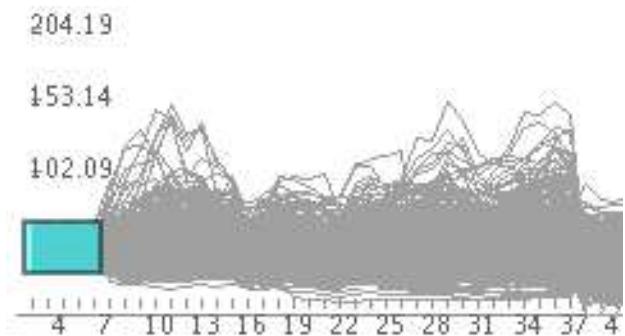


Figure 1. A single timebox query

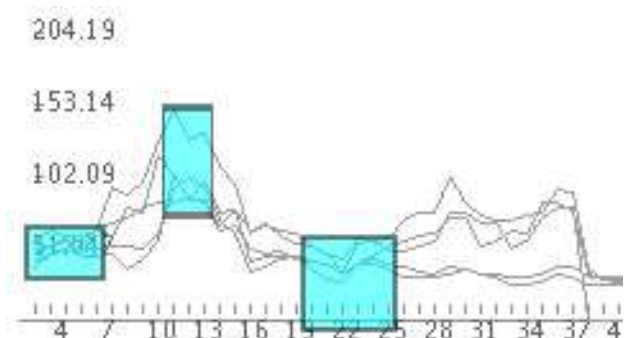


Figure 2. A complex query containing three timeboxes.

sired location of the opposite corner. Queries can be modified by dragging the box or resizing it via appropriate resize handles. Queries are reprocessed on each mouse event, and rapid (<100ms) processing provides the virtually instantaneous response necessary for dynamic queries. An overview “graph envelope” is constructed by overlaying each of the items in the data set on the query space.

Timeboxes can be useful for identifying items in a data set that match given patterns. For example, an analyst interested in finding stocks with weekly prices that rose and fell during a four-month period might start by drawing a single timebox (Figure 1), and then adding two additional conjunctive clauses at higher and then lower values (Figure 2). As timeboxes are added to the query, items that do not match the query are removed from the graph envelope.

TimeSearcher (<http://www.cs.umd.edu/hcil/timesearcher>) [2] is an interactive tool that uses timeboxes to pose queries over a time series data set attributes. TimeSearcher provides a linear list of all of the items in the display list, along with a details-on-demand window and other feedback. TimeSearcher also provides users with a pair of range sliders that can be used to change the height or width of a timebox without affecting the other parameter, and a simple drag-and-drop tool for query-by-example.

2. Gene Expression Levels in DNA Microarray Experiments

The advent of DNA microarray technology has led to the possibility of experiments that examine the response of an entire genome to some event or stimulus. In order to study the shift between anaerobic to aerobic metabolism, DeRisi, Iyer, and Brown used DNA microarrays to examine gene expression changes in yeast (*Saccharomyces cerevisiae*) cells at several points in time after their placement in fresh medium [1]. Measurements were made every 2 hours between 9 and 21 hours after initial placement, for a total of 7 time points. Figure 3 shows a sample query, identifying genes with expression levels that peaked at the sixth measurement (19 hours).

3. Conclusion

MIMSY [4] provided an early example of searches for temporal patterns in stock market data, using traditional GUI widgets to specify temporal constraints. Patterns uses boolean combinations of graphical primitives indicating transitions of interest to support time series queries [3]. QuerySketch is an innovative query-by-example tool that uses an easily drawn sketch of a time-series profile to retrieve similar profiles, with similarity defined by Euclidean distance [6]. Spotfire's Array Explorer 3 [5] supports

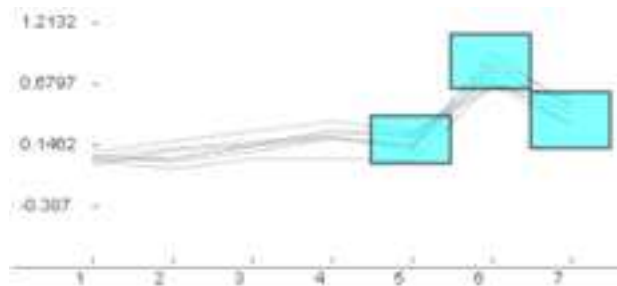


Figure 3. Yeast genes with peak expression levels at the 6th measurement

graphically editable queries of temporal patterns, but the result set is generated by complex metrics in a multidimensional space.

The TimeSearcher prototype has provided preliminary validation of the timebox query model. Extensions to the tool have been informed by discussions with molecular biologists, environmental scientists, financial analysts, and other motivated users. Further assessments of the tool, in the form of usability evaluations and empirical studies are planned. Future work will involve extensions of the timebox model to handle more expressive queries.

4. Acknowledgments

Thanks to Eric Baehrecke for his assistance. HH was supported by a fellowship from America Online.

References

- [1] J. DeRisi, V. Iyer, and P. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278:680–686, 24 October 1997.
- [2] H. Hochheiser and B. Shneiderman. Visual specification of queries for finding patterns in time-series data. In K. P. Jantke and A. Shinohara, editors, *Proceedings of Discovery Science 2001*, number 2226 in Lecture Notes in Artificial Intelligence, pages 441–446, Berlin, 2001. Springer.
- [3] J. P. Morrill. Distributed recognition of patterns in time series data. *Communications of the ACM*, 45(5):45–51, May 1998.
- [4] W. G. Roth. MIMSY: A system for analyzing time series data in the stock market domain. Master's thesis, University of Wisconsin, Department of Computer Science, 1993.
- [5] Spotfire. <http://www.spotfire.com>.
- [6] M. Wattenberg. Sketching a graph to query a time series database. In *Proceedings of the 2001 Conference Human Factors in Computing Systems, Extended Abstracts*, pages 381–382, Seattle WA, March 31–April 5 2001. ACM Press.

Interactive Poster: Visual Data Mining with the Interactive Dendrogram

Peter Imrich¹ Klaus Mueller¹ Raymond Mugno² Dan Imre³ Alla Zelenyuk³ Wei Zhu²

¹ Center for Visual Computing, Computer Science, Stony Brook University

² Applied Mathematics and Statistics, Stony Brook University

³ Environmental Sciences, Brookhaven National Laboratory

Abstract

We describe the interactive dendrogram, a visual framework for the mining, survey, and classification of large, high-dimensional data collections. The interactive dendrogram is the visual front-end of a classification algorithm providing a hierarchical data decomposition and model refinement capabilities. Hierarchy nodes are placed on concentric circles whose radii are determined by the dissimilarity of the node's sub-tree. A number of features are provided to enable focused viewing, such as collapsible sub-trees, non-linear radial distortion, and fish-eye magnification.

1 Introduction

Classification hierarchies or decision trees are often visualized as static dendograms. We have extended this paradigm into, what we call, interactive dendograms, which offer a variety of user controls to aid in the task of data mining and model development in the scenario of large and time-varying data. Our particular application deals with the survey and analysis of a large collection of millions of digitized aerosol particle spectra. Such a 250-bin mass spectrum is shown in Fig. 1g.

The task of classification of these acquired aerosols based on their spectra is difficult since no comprehensive mass spectrum-indexed library of aerosols exists to date. Given the overwhelming magnitude of the database, we have taken a two-tier approach to enable an interactive approach. To skim the data, we first run an off-line process based on k-means clustering that partitions the data into a few thousand representative groups, each having a classification error of less than the error of the acquisition machine. This procedure calculates the mean spectrum and the covariance matrix for each group and saves them to a file, along with the magnitude, mass and time stamp distribution, and others. The resulting list of "particles" is then utilized in the interactive visual data mining.

The verification of a clustering result will most likely require human expertise, provided by a domain expert, who then either agrees with the clustering or initiates another round of classification with some of the parameter settings changed. Two basic tasks make up this process: (i) evaluation of the present classification results, which is closely related to the task of surveying the information embodied by the data, and (ii) refinement of the model, which basically involves the adjustment of parameters.

2 Related Work

Dendograms are a popular visualization method to illustrate the outcome of decision tree-type clustering in statistics. Usually, these dendograms have a cartesian layout and are drawn as an up-right tree. However, this layout does not make good use of space, it is sparse towards the root and crowded towards the leaf nodes. The spacing between nodes at different levels in the hierarchy is not uniform, which is due to the shrinking number of nodes from bottom to top. For this reason, long, wide-spanning connecting lines are needed to merge nodes at higher levels. A better layout in this respect seems to be the polar or radial layout, where leaf nodes are located on the outer ring and the root is located in the center. A more uniform node spacing results, leading to a better utilization of space and resulting in a better illustration of the class relationships.

3 Our Approach

The radius r of the concentric circle onto which a non-leaf node with dissimilarity $nodeDissim$, obtained from the clustering algorithm [2], is placed is:

$$r = \left(\frac{maxDissim - nodeDissim}{maxDissim - minDissim} \right) \cdot maxRad \quad (1)$$

where $minDissim$ is the partitioning threshold of the k-means pre-processing algorithm, $maxDissim$ the dissimilarity of the root node, and $maxRad$ the radius of the outer-most circle. We arrange all leaf nodes on the outer-most circle, no matter where the parent node of a sub-tree is located. This is an appropriate placement since each leaf node stands for a partition of the original (large, preprocessed) dataset, and all partitions (or clusters) have been chosen in such a way that their dissimilarities are less or equal $minDissim$.

We have chosen the first layout method of [3], namely the equi-spaced arrangement of leaves on the outer-most circle. We chose this arrangement because it lends itself best for the (future) attachment of node labels. In this configuration, however, when the number of nodes in a sub-tree is large or the tree is highly imbalanced (as is the case in our application), sub-tree edges that connect a node on circle r_1 to a parent node $r_2 < r_1$ may pass through the circle with $r_3 < r_2$, which leads to a cluttered display (Fig. 1a). To avoid this, the angular range of sub-tree edges may not exceed a fan of 180° [1]. By using curved arcs instead of edges this constraint may be relaxed (Fig. 1b). However, this may give rise to curved edges that wrap along a large angular distance, destroying the locality of the graph. To reduce

{imrich, mueller}@cs.sunysb.edu, {imre, alla}@bnl.gov
{zhu, mugno}@ams.sunysb.edu

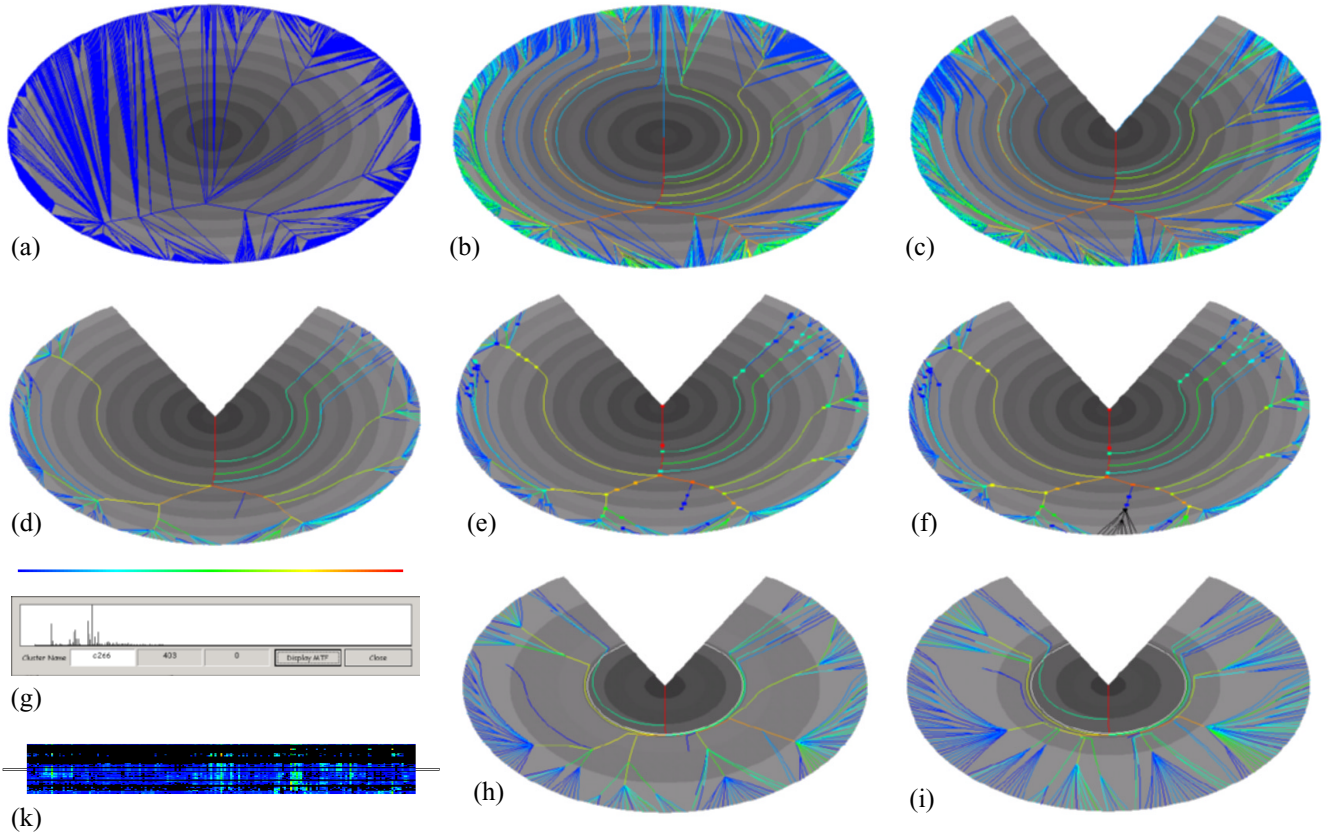


Figure 1: Various features of the interactive dendrogram (see Section 3 for a detailed description).

these effects, the user may specify an maximum wrap angle (Fig. 1c).

Edges are colored using a rainbow colormap to indicate the number of particles they carry. In our application, the user may interactively select the minimum number of particles that nodes need to have in order to be drawn. Modifying this threshold automatically redraws the graph with the full rainbow color spectrum mapped to the active node magnitude interval. This pruning of the graph also leads to less curving of the edge arcs (Fig. 1d). The fact that a sub-tree was pruned is indicated by a small knot at the pruned inner node (which is left at the original location in the graph, Fig. 1e). By clicking on the node the user has the ability to expand the node into its full sub-tree (Fig. 1f). A repeat click on the knot will re-collapse the sub-tree.

We also have generalized the mapping of dissimilarity to radius as follows:

$$r = \text{transFunc}\left(\frac{\text{maxDissim} - \text{nodeDissim}}{\text{maxDissim} - \text{minDissim}}\right) \cdot \text{maxRad} \quad (2)$$

where *transFunc* can be any increasing monotonic mapping, established using an interactive curve editor. The mapping can be used to emphasize certain dissimilarity ranges and compress others. We also provide an interactive, piecewise linear stretch function in the dendrogram itself where the user first specifies two arbitrary circles in the dendrogram. These circles then can be moved apart or brought together which compresses some circular sections in the dendrograms and spreads others (Fig. 1h and Fig. 1i). Finally, whenever a

node is selected, a window pops up that shows the node's average spectrum, calculated from all nodes in its sub-tree (Fig. 1g).

Aerosol researchers are also very interested in the distribution of particles over time. For this purpose we model the data as a cylindrical stack of dendrograms which can be sliced perpendicular to the time axis. The surface of the cylinder is given by the particle distributions in the leaf nodes, i.e., the outer-most circles in the dendrogram stack (Fig. 1k).

Using the various viewers in conjunction with the interactive dendrogram the scientist can perform the following tasks: (i) label the nodes with aerosol class descriptions, such as "aromatics" or "sulfates", (ii) discard nodes as erroneous measurements, (iii) move nodes along with their sub-trees to other parts of the hierarchy, which triggers a refinement of the classification rules.

4 Conclusions

We have described the interactive dendrogram, a visual paradigm and application for the mining, survey, and classification of large, high-dimensional data collections.

References

- [1] G. Battista, P. Eades, R. Tamassia, and I. Tollis, *Graph Drawing*, Prentice-Hall, 1999.
- [2] L. Kaufman and P.J. Rousseeuw, *Finding Groups in Data - An Introduction to Cluster Analysis*, John Wiley & Sons, 1990.
- [3] R. Wilson, R. Bergeron, "Dynamic hierarchy specification and visualization," *Information Visualization 1999*, pp. 65-72.

Interactive Poster: Addressing Scale and Context in Source Code Visualization

Stoney Jackson, Premkumar Devanbu, Kwan-Liu Ma
Department of Computer Science,
University of California, Davis
{jacksoni,devanbu,ma}@cs.ucdavis.edu

Abstract

Program comprehension is key to software maintenance, taking 50% or more of programmer time. Understanding programs is complicated by scale and contextuality. We introduce a visualization system for source code, based on a “source code listings on the walls” metaphor. The entire source code of large systems is displayed on the walls of a navigable virtual space. Within this visual metaphor, we introduce enhanced versions of traditional techniques such as “semantic-zooming” and “fisheye” views, which are specifically tuned for dealing with scale and context issues applicable to source code. This poster describes and illustrates these techniques.

1. Introduction

Software systems are notoriously hard to develop and maintain. *Program Understanding* is central; studies show that it consumes nearly 50% of maintenance time [3, 9].

Scale is one issue. Systems comprising millions of lines of code are common; but humans can only deal with a few screens full of text at a time. While modularization helps, it does not address the orthogonal problem of *contextuality*. Even a small software module, must be understood in context; but, the required context may not be nearby. This *delocalization* effect was noted early on [14], and has led to language innovations [8, 15]. Several contexts may be relevant to the function of a code fragment.

This project aims to use a 3D environment, with a *source code on the walls* metaphor, to render source code at scale, with visual attributes and spatial location capturing context. Techniques such as semantic-zooming and fisheye views are also adapted to this context.

We describe 3 distinct, but related approaches that provide a controlled tradeoff between scale and context. We begin with *SourceQuake*, our intuitive 3D approach to scale, and then describe two complementary techniques, semantic fisheyes and semantic zooming.

3. SourceQuake: scale vs. detail

Figure 1 shows our *SourceQuake Arena*, a 3D programming environment. The *source-code on the walls* metaphor leads to a smoother, and more intuitive way to tradeoff detail for scale. User can zoom in and out for more or less detail, and navigate by simply moving or

glancing. We believe such an environment *affords*¹ interaction and thus makes it easy to learn and use.



Figure 1. The SourceQuake Arena.

The geometry and position of the virtual walls are configurable, as is the layout and the visual attributes of the code on the walls. Programmer can thus customize to suit a given task. In Fig. 1, a configuration similar to the Perspective Wall [10] has been used to provide a nice balance between details and scale. Here, color classifies code fragments by function; green is one task, and blue another. Thus, the Arena’s geometry and the visual attributes of the code together reveal *context, at scale*.

There are several current approaches for managing the level of detail. We now describe how we adapt them for the 3D environment.

4. Semantic-zooming

Figure 2 illustrates one semantic zooming approach. Here, we only use the code’s syntax tree to construct a new representation. As a programmer logically zooms away from the code, code fragments lowest in the syntax tree are collapsed into a single keyword, thus increasing scale. In Figure 2, the *try*, *catch*, and *if* blocks in the code segment on the left have been collapsed into **TRY**, **CATCH**, and **IF** tokens. Upon zooming closer, the blocks explode back into their original detailed form.

¹ The term *afford* is used here in the relaxed sense of Gibson’s notion of *affordance*, as described in [16] C. Ware, *Information Visualization: Perception for Design*. San Francisco: Morgan Kaufmann, 2000.

```

while (!stack.isEmpty() && (depth >= 0)) {
    String currentPath = (String) stack.pop();
    parseProperties(req, resources, generatedXML,
        currentPath, type, properties);

    try {
        object = resources.lookup(currentPath);
    } catch (NamingException e) {
        continue;
    } //catch

    if (object instanceof DirContext) {
        try {
            NamingEnumeration enum =
                resources.list(currentPath);
            WHILE
        } catch (NamingException e) {
            resp.sendError (HttpServletResponse.
                SC_INTERNAL_SERVER_ERROR, path);
            return;
        } //catch

        if (depth > 0) {
            //Displaying the lock-null resources present in that
            // collection
            String lockPath = currentPath;
            IF
            Vector currentLockNullResources =
                (Vector) lockNullResources.get(lockPath);
            IF
        } //if

        if (stack.isEmpty()) {
            depth--;
            stack = stackBelow;
            stackBelow = new Stack();
        } //if

        generatedXML.sendData();
    } //while
}

while (!stack.isEmpty() && (depth >= 0)) {
    String currentPath = (String) stack.pop();
    parseProperties(req, resources, generatedXML,
        currentPath, type, properties);

    try {
        object = resources.lookup(currentPath);
    } catch (NamingException e) {
        continue;
    } //catch

    if (object instanceof DirContext) {
        TRY
        CATCH

        if (stack.isEmpty()) {
            depth--;
            stack = stackBelow;
            stackBelow = new Stack();
        } //if

        generatedXML.sendData();
    } //while
}

```

Figure 2. Semantic zooming for source code.

Programs are rich in semantics and derivable data. We can zoom using any such semantic information: abstract syntax trees, control-dependence, data-dependence, heuristic analysis products [6, 11], can all be used for zooming.

5. Semantic fisheye views

Semantic fisheyes can help cope with scale and context by showing less irrelevant code and more relevant code. Code's relevance is based on current editing position in the code and other program related data (e.g., control flow or profile data). In semantic fisheyes, as opposed to classic fisheyes [4, 5], this function is runtime configurable, is definable on the data other than just the syntactic structure of code, and allows mappings to visual attributes other than elision. This flexibility allows a programmer to customize the fisheye to yield views suited to his task.

6. Related work

Seesoft type views address scale by representing each line of code as a single pixel [1]. On a 1600x1200 resolution display, nearly 2 million lines of code can be visualized. Typically, such techniques display code at a few, coarse-grained levels of detail. We exploit current graphics hardware to support a continuous spectrum of levels of detail. Perspective Wall displays large numbers of files on a screen using a perspective to address scale [10]. Files are dealt with as abstract units, sans contents.

Semantic Zooming [8] represents objects differently at different levels of zoom. This approach has not been applied to source code. We have defined semantic zooming

for source code in terms of semantic fisheye, allowing a multitude of semantically rich views to be derived.

Holoprasting [13] techniques help deal with context and scale by allowing a programmer to elide pieces of code that are currently unimportant, thus revealing more code that is relevant. Holoprasting can be performed manually, automatically, or semi-automatically; Desert [12], Furnas's fisheye [4, 5], and Jaba [2] are examples of each respectively. These techniques are based entirely on the syntactic structure of the code. The Semantic Fisheye Framework [7] allows fisheyes to be defined in terms of many sets of data. We have adapted this technique to the domain of source-code visualization.

7. Conclusion

In the SourceQuake project, we are using a "code on the wall" metaphor for source code visualization. This metaphor exploits the natural human intuitions of the physical world exploration to allow smooth trade-offs between scale and detail. We enhance the purely geometric zooming and panning in two ways: with semantic zooming based on code structure, and fish-eye views based on code semantics. We have experimented with these approaches using code from the Apache web server.

8. References

- [1] T. Ball and S. Eick, "Software Visualization in the Large," *Computer*, vol. 29, 1996.
- [2] A. Cockburn, "Supporting Tailorable Program Visualization Through Literate Programming and Fisheye Views," *IST*, vol. 43, 2001.
- [3] T. Corbi, "Program Understanding: Challenge for the 1990s," *IBM Systems Journal*, vol. 28, pp. 294-306, 1989.
- [4] G. Furnas, "The FISHEYE View: A New Look at Structured Files," Bell Labs, Murray Hill, New Jersey 1981.
- [5] G. Furnas, "Generalized Fisheye Views," *CHI*, 1986.
- [6] W. Griswold, "Coping with Crosscutting Software Changes Using Information Transparency," in *Reflection 2001*, 2001.
- [7] P. Janeczek and P. Pu, "A Framework for Designing Fisheye Views to Support Multiple Semantic Contexts," *AVI*, 2002.
- [8] G. Kiczales, et al. "An Overview of AspectJ," *ECOOP*, 2001.
- [9] B. Lientz, E. Swanson, and G. Tompkins, "Characteristics of Application Software Maintenance," *CACM*, vol. 21, 1978.
- [10] J. Mackinlay, G. Robertson, and S. Card, "The Perspective Wall: Detail and Context Smoothly Integrated," *CHI*, 1991.
- [11] G. Murphy, et al. "Separating Features in Source Code: An Exploratory Study," *ICSE*, 2001.
- [12] S. Reiss, "The Desert Environment," *TOSEM*, vol. 8, 1999.
- [13] S. Smith, D. Barnard, and I. Macleod, "Holoprasted Displays in an Interactive Environment," *International Journal of Man-Machine Studies*, vol. 20, pp. 343-55, 1984.
- [14] E. Soloway, et al. "Designing Documentation to Compensate for Delocalized Plans," *CACM*, vol. 31, 1988.
- [15] P. Tarr, et al. "N Degrees of Separation: Multi-Dimensional Separation of Concerns," *ICSE*, 1999.
- [16] C. Ware, *Information Visualization: Perception for Design*. San Francisco: Morgan Kaufmann, 2000.

Interactive Poster: Visual Information Retrieval with Bubble World

Timothy Jacobs
Air Force Institute of Technology
Timothy.Jacobs@afit.edu

Christopher Van Berendonck
Royal Australian Air Force
Chris.VanBerendonck@defence.gov.au

Abstract

Visualization has significant advantages over traditional textual lists for improving cognition in information retrieval. To realize these advantages, we identify a set of cognitive principles and usage patterns for information retrieval. We apply these principles and patterns to the design of a prototype visual information retrieval system, Bubble World, in which we apply a variety of visual techniques to transform the internal mental representations of the information retrieval problem to an efficient external view. We provide query refinement and interaction techniques that facilitate the specification of complex search schemas to enhance the knowledge acquisition process.

1. Introduction

Traditional information retrieval techniques return a textual list of documents ranked from most relevant to least relevant, based on the determination of a search algorithm (or model). Not only are large lists of irrelevant documents frequently presented to the user, but also only limited clues are presented for determining how best to adjust the query to improve the result.

It is widely accepted that visualization of information can enhance cognition. A number of visual information retrieval systems have been developed to validate this belief (for example, see [2, 3, 5]), however, these systems have not found widespread use.

In this paper, we introduce a visual information retrieval system in which we apply a variety of visual techniques to successfully transform the internal mental representations of the information retrieval problem to an efficient external view. We provide query refinement and interaction techniques that facilitate specification of complex search schemas to enhance the knowledge acquisition process.

2. Cognitive Principles

To maximize the cognitive benefits of our graphical representations, we based our design on a set of cognitive strategies for information retrieval [6]. These strategies, based on Scaife and Rogers [4], are summarized here.

- Restrict the interpretations of the external model to enforce a tight coupling between it and the information retrieval domain space.

- Provide computational offloading through the careful selection of graphical representations and the associated interaction and manipulation models.
- Provide a graphical representation of the problem that maps closely to the internal mental models of most users.
- Constrain the temporal and spatial domains to enhance the correct inferences about the external representations.

3. Usage Patterns

Focusing on the information retrieval domain, we identified typical usage patterns to further guide our design. These usage patterns are derived from experiments conducted by Cunningham and others [1].

- Users rarely amend default settings.
- Queries tend to be short and simple.
- Clusters of common query terms are apparent.
- Users have difficulty selecting terms with the appropriate specificity.
- Users submit a high percentage of malformed or erroneous queries.
- Users submit few sequences of queries, spend short periods searching and are unlikely to repeat the vis it.
- Users refine their query.
- Few, if any, documents are viewed from the returned list and those that are can be found at the top of the list.

From these usage patterns we derived a set of design corollaries for visual information retrieval [6].

4. Query Visualization

In our prototype, a query is specified in a natural language as a sentence or phrase that is then broken into keyword nodes. These nodes are placed symmetrically around a circle to represent the keywords of the query.

Bubble World implements the vector model to retrieve documents relative to the keyword nodes. These documents are represented as icons that are positioned according to their relevance to each of the keyword nodes.

With this spatial positioning technique, only relative strength relationships are visible. Therefore, this technique is supplemented with size coding to represent absolute strength of the keyword relevance.

Although the shape of an icon indicates the number of nodes with which a document is associated, it is not always clear which specific nodes are involved. To handle this, the icon in focus develops a shaded polygon with each vertex represented by a node having a positive influence on the position of the icon. In this way, the user can not only discriminate which nodes have influence on the icon, but also gains clarity about the relative positioning of the icon to each of these nodes.

5. Query Refinement

Bubble World provides two mechanisms for query refinement – 1) filtering with nested queries and 2) query expansion or contraction by modifying keyword nodes. Nested queries constrain the set of documents matching specific keyword nodes. A child bubble (circle) is created from a keyword node (parent) with specialization terms that further refine the parent node. Those documents containing the parent node are filtered into the child bubble and represent the documents relevant to the parent node and the specialization terms. Figure 1 shows the representation of nested query results where “President of the United States” is the original query. This query is first filtered by “Bush or Reagan” and then “Bush” is filtered further by the query “Gulf War”.

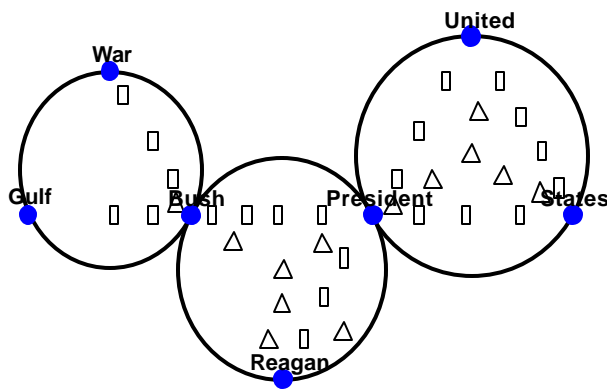


Figure 1. Nested query results

Users may also refine queries by adding or removing keyword nodes from the original query. Bubble World adjusts the location of keyword nodes on the circle and reapplies the vector model to retrieve relevant documents.

6. Interaction Model

Since the set of relevant documents may be quite large, Bubble World provides an interaction model for filtering and drilling down into the most relevant documents. This interaction model includes three distinct environments: the

exploratory view, the icon view, and the document view. The exploratory view provides an overview of the document collection relevant to the specified query terms.

Due to the manner in which documents are positioned in the exploratory view, many documents may be mapped to the same icon. In this case, the user selects an icon to move to the icon view for further exploration of the icon’s repository collection. The icon view includes graphical representations showing the relative occurrence of keyword terms in each document represented by the icon.

When a single document of interest is found, the user selects the icon for that document and moves to the document view depicting the actual text of the document. Additional graphical tools support browsing and locating keywords in the document.

7. Summary and Conclusion

Bubble World integrates a variety of visual techniques for exploring large document collections to locate relevant information. Visual techniques are selected and implemented based on a set of cognitive principles and usage characteristics tailored to information retrieval. Preliminary experimental results indicate that these techniques lead to minor improvements in retrieval time and significant improvements in retrieval accuracy over textual document retrieval lists.

8. Acknowledgements

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

9. References

- [1] Cunningham, J.S.S. et al, “A transaction log analysis of a digital library,” *International Journal on Digital Libraries*, Vol 3, no. 2, pp 152–169, 2000.
- [2] Hemmje, M. et al, “LyberWorld – a visualization user interface supporting fulltext retrieval,” in *Proceedings of the 17th International ACM SIGIR Conference*, pp 249–259, 1994.
- [3] Korfhage, R. R., “To see or not to see – is that the query?” in *Proceedings of the 4th International ACM SIGIR Conference*, pp. 134–141, 1991.
- [4] Scaife, M. and Y. Rogers, “External Cognition: how do graphical representations work?” *International Journal of Human-Computer Studies*, vol. 45. pp. 185-213, 1996.
- [5] Spoerri, A. “InfoCrystal: A visual tool for information retrieval,” in *Proceedings of Visualization ’93*, pp. 150–157, IEEE Computer Society Press, 1993.
- [6] Van Berendonck, C., *Bubble World: A Novel Visual Information Retrieval Technique*, Master’s thesis, Air Force Institute of Technology, 2002.

Interactive Poster - Baseball Decision Coach

Scott Listfield, Doug Marttila, Mark Schindler, Angela Shen-Hsieh, John Wynne
 info@visual-io.com

Abstract

This system is designed around the fundamental concept of making an informed go/no-go decision. In this case, the example used was a baseball decision regarding whether to pull a pitcher from the game at a crucial moment.

1. Introduction

Many factors can go into making this decision: how the pitcher has fared against this batter in the past, how he has fared against the on-deck batter, the status of the game, the relievers in the bullpen. This system balances quantitative and qualitative aspects of the decision by compiling the hard data around each factor (sample data loaded here from xml files), and allowing users to provide their own expert, subjective input. Each criteria is then given a weighting factor indicating its importance to the overall decision. This weighting factor can be derived from a statistical analysis of historical outcomes (the default) or can be adjusted by the user. All information from all factors is then aggregated into a simple data visualization indicating pull or leave the pitcher.

	Leave Pitcher in			Pull Pitcher		Current Situation	weighting prob of success	
	1	2	3	4	5			
Pitch Count	41-60	21-40	61-80	1-20	81-100	47	1	0.10
Confidence	81-100	61-80	41-60	21-40	1-20	72	2	0.09
P/B History	0-400	401-600	601-1000	1001-1500	1500+	1209	4	0.10
Men on base	0	0	1	2	3	1	3	0.08
Score	< -5	< -2	(-1) - (+1)	> +2	> +5	1	3	0.10
Balls	0	0	1	2	3	3	5	0.06
Strikes	0	0	1	2	2	2	4.5	0.07
Outs	0	0	1	2	2	0	1.5	0.08
Inning	5-6	3-4	7-8	1-2	9+	8	3	0.10
Importance	5-6	3-4	7-8	1-2	9-10	7	3	0.10
Next Bat Hist.	0-400	401-600	601-1000	1001-1500	1500+	1028	4	0.09
Bulpen	1-2	3-4	5-6	7-8	9-10	6	3	0.03

3.01 1.00

Fig. 1

The Opinion Gathering tool can then be used to bring together the input of multiple users (i.e. field managers, coaches, senior management, other experts) and support a collaborative decision making process.

2. The Interface

2.1 Individual/Group Toggle (upper right)

There are two modes to this interface: *Statistical Analysis* (where each individual user aggregates his/her opinion) and *Opinion Gathering* (where the group opinion is compiled.) For the purposes of this demo, only the Statistics side is loaded with dynamic data (Red Sox vs. Yankees.) Toggling to the Opinions side, you will see only an example group opinion.

2.2 Weighting Window (click small black square on the handle of the pie chart)

Open this window to see a list of the criteria being used for this decision. Weights default to predetermined values but can be manually changed (i.e., if you believe the number of pitches the pitcher has thrown should be a more heavily weighted factor in this situation, increase the number beside Pitch Count.) The sum of the weighting factors must equal 100.

The bars beside each criteria indicate the 'score' of that factor based on the current data and the user's input. The longer the bar, the more that factor indicates the pitcher should be pulled.

2.3 Aggregate view (the pie chart)

The pie chart sums all the factors to provide an at-a-glance understanding of the decision. The more filled the pie, the more likely the pitcher should be pulled. There are three variables:

- > Length of the arc (size of the wedge) indicates the weighting of that criteria
- > Radius indicates the 'score' of that factor. Again, the longer the radius, the more that factor indicates the pitcher should be pulled.
- > Shading of the wedges simply groups criteria into more digestible 'chunks': here, all criteria pertaining to the pitcher is the darkest shade, game status is the medium shade, and future factors the lightest.

Click on any tethers' end point to close that window. The handle allows you to move the pie and re-organize the data windows in a way that makes sense to you.

2.4 Criteria Views (click yellow box to expand/minimize and gray box to hide)

Data for each criteria is displayed in a view tethered to its pie wedge. Data in related views are linked and will change results dynamically; for example, if you use the pulldown to change the pitcher in the Pitcher:Pitch Count view, you will see that data in the minimized Current Batter and Next Batter views will change and will, in turn, change the pie chart.

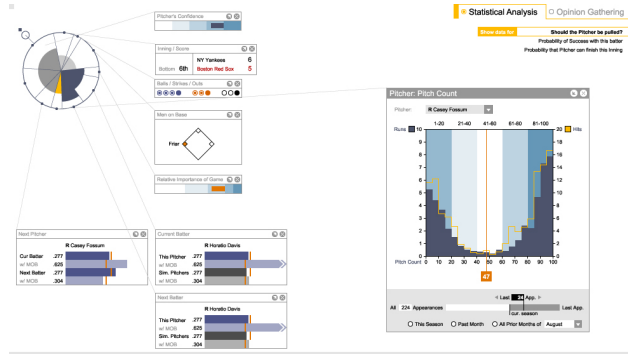


Fig. 2

2.5 Score (segments of blue shading)

Each criteria window displays historical or current data. Each user can factor in his or her own expertise, experience and intuition about the situation by adjusting settings for the amount of history considered or ranges of confidence or importance. Each factor is then scored within a range. Each range has been assigned a shade of blue: the darker the blue, the more that factor indicates the pitcher should be pulled.

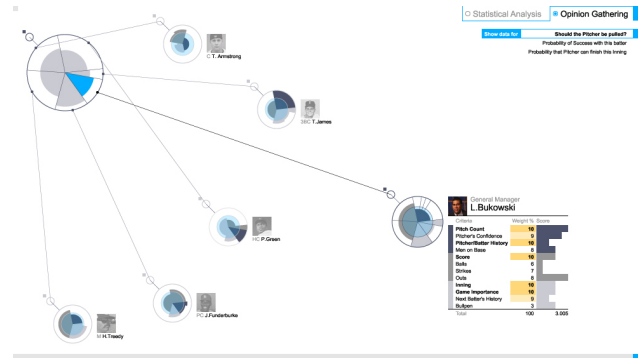


Fig. 3

Interactive Poster: Immersive Information Modeling using Particles

Andrew Vande Moere

Chair for CAAD, Swiss Federal University Zurich, Switzerland

moere@arch.ethz.ch

Abstract

This paper introduces an information visualization metaphor that allows users to interactively explore the contents of database systems inside immersive virtual reality environments. Data objects are represented by particles, coined 'infoticles', which stream inside a three-dimensional world. Users are able to interact with the information representation by manipulating tools, such as forces and surfaces, which are placed inside the virtual world itself. These tools, representing respectively user's interests and data filters, influence the collection of infoticles according to the rules of Newtonian mechanics. Both the static and dynamic characteristics of the resulting flows of infoticles help users to filter and analyze the database contents. These described principles are demonstrated through a prototype that uses our university's financial budget data.

1. Introduction

This paper describes a merged visualization and interaction metaphor for data exploration inside immersive virtual reality environments. We are currently using our university's financial data to explore the relationships between budgets, departments and amounts of students over time. The use of spatially distributed points [1,4] and spring-embedded techniques [3] is a widely investigated topic, and several information visualization applications for high-end immersive systems [2,5] have already been implemented. Mostly, such applications need a certain amount of dedicated pre-computation of a Cartesian mapped, static world and offer limited user interaction. In contrast, our concept is based upon the translation of informational values into dynamic behavior, such as motion, directionality and global form, of a collection of information loaded particles, coined 'infoticles', that users can influence through a set of tools.

2. Infoticles

Each infoticle emitter corresponds with a unique database table, while each single infoticle corresponds, conceptually as well as programmatically, with a certain data attribute retrieved from the database rows (which can be interpreted: \$1000 can be translated into 100 infoticles of value \$10). These particles are coined 'infoticles' as

they behave depending on the informational values they contain. Thousands of infoticles inherit certain data attributes such as speed, direction and lifespan, and they stream inside the virtual world in a time-ordered way. Consequently, possible data sources can be not only static databases, but also flows of time-varying data values, such as real-time stock market quotes or live news agency data streams. Next to the efficient and non-occluding directional particle lines [Fig.2], infoticles can also be represented by texture blending rendering, resulting in detailed density maps or cloudlike overviews [Fig.3].

3. Tools

Users are able to interact with the application through the manipulation of a set of *modeling tools* that influence all infoticles according to the rules of Newtonian mechanics. Infoticles are attracted by those forces, and are let through by those surface filters that carry an equal data value. All other infoticles are unaffected by the forces, and are bounced back by the filters [Fig.1]. In practice, forces, representing user's interests, cause infoticles to direct or cluster into specific directions of attention. Properly positioned surfaces, depicting data-filtering queries, divide the workspace into user-defined spatial areas with common data values.

4. Interface

Special attention has been paid to the development of an intuitive interface that does not break the three-dimensional illusion nor occludes the visualization with menus or widget elements. Therefore, the interaction tools are depicted by simple icons and are placed inside the three-dimensional scene itself. As the only true static elements of the virtual world, they thus function simultaneously as spatial orientation and information interpretation elements. The virtual world is built up in human proportions, providing users with intuitive ways of orientation. Moreover, infoticles are not mapped onto coordinate axes, but instead partly expose their meaning through their relative distances and proximity to the tools. We thus predict that users show less interpretation problems, as there is no need to detect the fixed Cartesian graph directions. A six-dimensional mouse is used as the primary input device, so that users are able to navigate around, freeze and animate the application, as well as

select, move and rotate all elements in the virtual world. The mouse steers a 3D cursor, which functions like a virtual flashlight that lights up the text labels exposing the data values of the infoticles in view.

5. Analysis

Infoticles expose their meaning through the proximity to tools, the proportionality of the clusters and colors, the self-organizing solution paths, and the sudden behavioral changes of direction and speed as users adapt the spatial setting of the environment. These characteristics are more easily perceived due to the unique properties of spatial awareness and stereoscopic depth of immersive systems. Users can enforce data filtering and clustering by combining filters and forces in spatial constellations, so that infoticle streams will divide and different flocks will emerge unveiling the proportionality and amounts of certain data attributes. Ultimately, users are able to ‘model’ their personalized spatial configuration, and respond immediately upon the visual results to refine or refocus their goals. The concept of *motion* makes the system react in real-time, reinforces the depth perception and exposes the aspect of history in ways of directional paths. By continuously streaming the equal data sets in a time-ordered way, users can learn from their previous modeling actions, and observe the time dimension of the data.

6. Conclusion & Future

Early experiences have shown that users appreciate the intuitive interaction and absolute creative power over the application. We consider these early experiments as very promising and are exploring other applications for the infoticle metaphor. We are convinced that this technique is especially suitable for visualizing large, multi-dimensional, or real-time datasets. Other applications are planned that map informational values to specific particle variables (e.g. speed, lifespan, mass, etc.), so that meaning can be discovered through dynamic behavior of individual infoticles. Additional research has to prove if this technique can be used as a general way of three-dimensional data browsing, capable of dealing with many different sorts of data and complex user queries.

7. References

- [1] C. Ahlberg and B. Shneiderman, Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays, Proc. CHI '94, April 1994, pp.313-317.
- [2] S. Benford, D. Snowdon, C. Greenhalgh, and R. Ingram, “VR-VIBE: A Virtual Environment for Co-

operative Information Retrieval”, *Eurographics'95*, August 1995, pp. 349-360.

- [3] P. Eades, “A Heuristic for Graph Drawing”, *Congressus Numerantium*, No. 42, pp. 149-160.
- [4] J.M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, and P.J. Haas, “Interactive Data Analysis: The Control Project”, *IEEE Computer*, August 1999.
- [5] J. Leigh, A. Johnson, T. DeFanti, S. Bailey, and R. Grossman, “A Tele-Immersive Environment for Collaborative Exploratory Analysis of Massive Data Sets”, *ASCI 99*, June 1999, pp. 3-9.

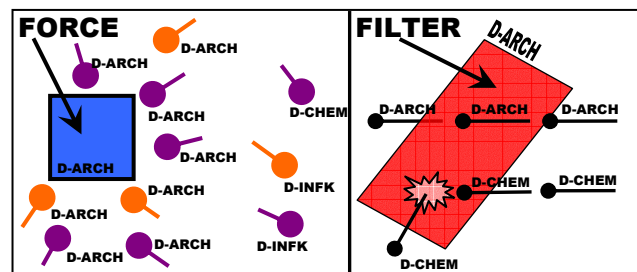


Figure 1. Infoticle Tools Concept Diagram.

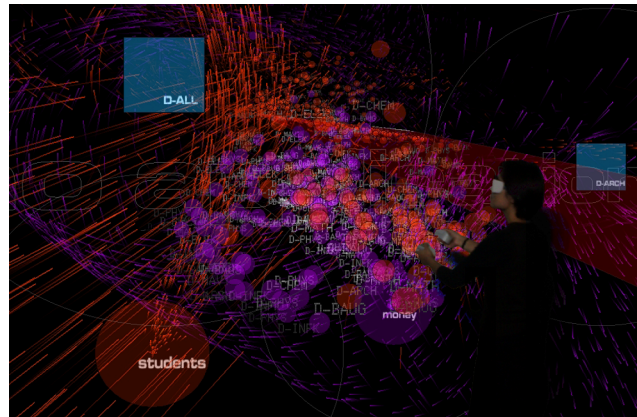


Figure 2. A user immersed in an infoticle system.

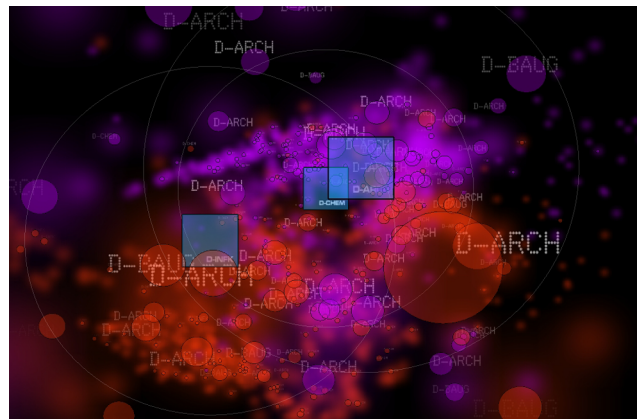


Figure 3. Density view.

Interactive Poster: Streaming Graphics with Dancer

Andrew A. Norton
SPSS, Inc.
233 South Wacker Drive
Chicago, IL 60606 USA
+1 312 651 3154
anorton@spss.com

Leland Wilkinson
SPSS, Inc.
233 South Wacker Drive
Chicago, IL 60606 USA
+1 312 651 3270
leland@spss.com

Abstract

Dancer is an object-oriented class library for processing changing tables of data and rendering them to a graphics and/or sound environment. We present an architecture designed to integrate and buffer streaming data in a way that allows real-time analytics and display of up to 20 frames per second in a 2D or 3D environment.

Keywords

Streaming data, streaming graphics, real time visualization

Introduction

The term *streaming graphics* evokes the terms *streaming media* [4], and *dynamic graphics* [2], [5], [6]. While similar to both in its outward appearance, streaming graphics is basically different from both. Streaming media systems are generally concerned with delivering sound and video information in real time. Dynamic graphics are concerned with using motion to reveal structure in static data. Streaming graphics systems are concerned with displaying analyses (summaries) of streaming data in real time. Applications of streaming graphics involve many different environments, including real-time monitoring of manufacturing processes, health indicators, financial statistics, and Web data..

The Grammar of Graphics

Dancer is based upon Wilkinson's *The Grammar of Graphics* [7]. It views statistical graphics as a chain of functions that display aesthetic representations derived from data according to user-specified rules. These rules are displayed to the user in the form of legends, axes, and other guide notations.

Theory

Dancer is based on components that define the attributes of a displayed graphic, using the graphics grammar theory. We illustrate these components through the example of graphing a simple error function $f(x) = e^{-x^2}$.

Frame

The graph $G = \{(x, f(x)) : x \in \mathbf{R} \text{ and } f(x) = e^{-x^2}\}$ is a subset of \mathbf{R}^2 . To display G , we choose a bounded region of \mathbf{R}^2 , $F = [x_{min}, x_{max}] \times [y_{min}, y_{max}]$, and we physically represent the set of points $P = F \cap G$ by choosing a coordinate system and making a graphic with ink or some other perceivable medium.

Graphics algebra provides a method for specifying F (which we call a *frame*) when we wish to construct a graphic based on some function of a set of data. Wilkinson [7] presents three algebraic operators called *cross* (*), *nest* (/), and *blend* (+), together with the rules for their use. They

are derived from the set operators *product* (\times), *discrete union* (\sqcup), and *union* (\cup), respectively.

Elements

An element is a geometric object that represents data values within a frame. An element can represent many different data values through multiple glyphs (for example, a Point element may contain thousands of individual point glyphs). Dancer provides a library of pre-written elements (such as Bar, Line, and Point), or users can write their own.

Aesthetics

An Aesthetic is a function mapping frame coordinates to display indices. If we wish to display the set of points $P = F \cap G$ with a plotted curve defined by (x, y) coordinates on a 2D display device, then we employ the position aesthetic function $A : x \rightarrow x_{\text{position}}, f(x) \rightarrow y_{\text{position}}$. The realized graphic would be $G_A = A(F \cap G)$. Aesthetics are based on the typologies originating in [1] and elaborated in [7].

Domains

Each data cell is declared to have a specific domain, which is a set (finite or infinite) of possible values. These domains may be declared by the user, or adopted from another table.

Each aesthetic, including position, is associated by the frame with a domain and a function. Functions are required to compute their range given an input domain, so we can easily find the actual domain of the aesthetic. This provides the source material for rendering legends.

Animation

Dancer is specifically focused on graphs that change, either by user action or process-driven events. Dancer graphs are intended to be long-lived and persist as the data or specifications are modified. Dancer allows the user to explore the graph through direct manipulation. For example, the user could change the clipping of axes, or make only certain panels visible [3].

Dancer works in real time as the data arrive. It can handle over 10,000 data updates per second, changing the element displays in response. The frame only changes if the domains are changed, which may be set to happened automatically as new values are encountered.

Architecture

Dancer is a framework. Its value is in the structure it provides and the management of user-extensible components. It allows the user to share reliable code and thereby have a smoother faster implementation of customized graphs.

Functions, elements, domains, and renderers are all user-extensible.

The current release of Dancer is implemented in Java 1.4. The compiled code is quite small (about 300 kilobytes) because of extensive use of the standard Java class libraries.

The implementation of Dancer is divided into three layers: *data*, *aesthetics*, and *renderer*. Each layer is dependent only upon the layers below.

The Data layer provides data access, data transformations, and event management.

The Aesthetics layer uses element and frame specifications to generate tables that marshal the data required for rendering the elements.

The Renderer layer generates perceivable representations of the aesthetics. There may be multiple renderers, each based upon the same frame and displaying differing elements. For example, we may have a 3-D visual renderer and also a sound renderer.

Data Layer

The Data Layer is comprised of tables of cells organized in a network structure of dependencies. A cell instance has the following responsibilities:

- Providing a value on request
- Handling event notifications
- Providing cell properties

Cells could be used on their own, but for practicality, cells are usually grouped into tables. Cells within tables share properties and algorithms. Tables organize cells into variables and rows, such that

- Each cell belongs to one variable and one row.
- Domains are assigned by variable.
- Metadata is typically assigned by variable.
- Listeners may subscribe to all cells satisfying a given filter (for example, all cells for a given variable).

The general strategy is for events to be fired immediately, but for data values to be computed on demand. This is based on an expectation that data updates may be much more frequent than data retrievals (Dancer can accommodate over 10,000 data updates per second, but typically only refreshes the screen 20 times per second).

This strategy is also motivated by a desire to minimize the load on the event notification (data updating) thread, so that the use of Dancer does not interfere with other activities that may be driven by the event notification thread.

Aesthetics Layer

The end result of the Aesthetics layer is an Aesthetics table for each element. This table contains a row for each glyph, with a variable for each aesthetic. The cell values are contained within the domain of the aesthetic, so they are ready for use by a Renderer.

The elements are based upon data values for each aesthetic, which are transformed by the aesthetic functions defined by the Frame.

Each aesthetic is defined as one of

- Default (a constant defined by the Aesthetic)
- UserConstant (a user-specified constant)
- SplitVariable (generates separate glyphs for cases that differ on the specified variable)
- ShadeVariable (generates a single glyph “shaded” by values on the specified variable)

Renderer Layer

A renderer converts the aesthetic table (plus the frame information) into a perceivable result. Elements are designed to work together with a particular renderer or family of renderers.

Future Plans

Recent developments in Dancer have focused on working with time as a key to be manipulated, rather than as a process to be played back in the renderer.

Instant Replay

For example, the user can be watching a live process unfold, and then choose to drag a control back and replay what just happened. The speed of these animations is fully adjustable.

Alerts

We have also developed an Alert capability. Up to 800,000 events per second can be scanned. When an alert is triggered (based on user-specified criteria such as threshold or traffic), the response may include saving a specified number of events before and after the alert occurred.

The events associated with these alerts can then be reviewed and compared.

Acknowledgements

We thank Jack Noonan and SPSS for supporting this research.

References

- [1] Bertin, J. *Semiologie Graphique*. Paris: Editions Gauthier-Villars, 1967. English translation by W.J. Berg as *Semiology of Graphics*, Madison, WI: University of Wisconsin Press, 1983.
- [2] Cleveland, W.S. and McGill, M.E., Eds. (1988). *Dynamic Graphics for Statistics*. Belmont, CA: Wadsworth Advanced Books.
- [3] Datar, M., Gionis, A., Indyk, P., and Motwani, R. (2002) Maintaining stream statistics over sliding windows. Annual ACM-SIAM Symposium on Discrete Algorithms, San Francisco.
- [4] Feamster, N.G. (2001). Adaptive delivery of real-time streaming video. Master's thesis, Department of Electrical Engineering and Computer Science, MIT. Online at <http://nms.lcs.mit.edu/papers/feamster-thesis.pdf>
- [5] Swayne, D.F., Cook, D., and Buja, A. XGobi: Interactive Dynamic Data Visualization in the X Window System. *Journal of Computational and Graphical Statistics*, 1998 (7), 113-130.
- [6] Velleman, P.F. *Data Desk*. Ithaca, NY: Data Description Inc.. 1988.
- [7] Wilkinson, L. *The Grammar of Graphics*. New York: Springer Verlag, 1999.

Interactive Poster: Ringtuples: Visualization of Transitions Between Two States

Rodrigo A. Obando
Charles F. Dolan School of Business
Fairfield University
Fairfield, CT, USA.
E-mail: RObando@fair1.fairfield.edu.

Keywords

Dynamic icons, visual data mining, clustering.

1. Introduction

In the area of Data Mining or Knowledge Discovery, the main purpose is to find patterns in collections of data instances. Whether these data instances are snapshots, sequences or time-series, we want to find similarities among thousands and, many times, millions of data samples. In the case of time-series, the purpose has been to find similar time sequences or portions of them within other sequences.

This paper concentrates in finding statistical profiles of behaviors of two-state systems for quick finding and comparison. Examples of these two-state systems include software bugs, invoices, and product shelf-life cycles. These systems show the behavior of going from one state two another and staying in each state for a finite amount of time. Instead of looking at the time profile, we are interested in recording the time the system stays in one state. This model is a statistical representation of the system and even though two systems might not have the same time-series behavior, they may share the same statistical representation. We present such a model and propose a visualization for Visual Data Mining.

2. Transition Between Two States

The systems of interest may be represented as Information Objects (IO) as defined in [1]. The systems used in this paper contain only two states. These states may be called the undesirable state U and the desirable state D . A way that these objects are characterized is by their state at a given time. In the case of a software bug we are interested in whether it is still present or it is fixed; in the case of an invoice whether it is open or paid. These systems may be modelled as the ordered set $S(k, M) = \{IO_i\}$ where

$IO_i = \{v_i, \tau_i\}$ for $i = 0 \dots k - 1$, $v_i > 0$, and $\tau_i = \lfloor \frac{t_i}{M} \rfloor$, where $k - 1$ is the number of times the system has changed from the undesirable state to the desirable state, IO_0 represents the present state, v_i is the value of the system during the i^{th} undesirable state, M is the value of each time interval, and t_i is the time the system spent in the undesirable state if $1 \leq i \leq k - 1$ and it is the time spent so far in the present state if $i = 0$ and the system has entered the undesirable state, it is 0 otherwise.

This expression of the system allows us to determine a total value for the system. The total value $V(k)$ of the system is the sum of all v_i as in $V(k) = \sum_{i=0}^{k-1} v_i$. If the individual objects are invoices and v_i their monetary value, $V(k)$ represents the total revenue.

We may combine all IO_i objects with the same τ_i by adding their v_i values. This process expresses the system with up to $2m$ elements, with m as the number of possible time intervals. We express this system as the ordered set $S(2m, M) = \{IO_j\}$ where $IO_j = \{v_j, \tau_j\}$ for $j = 0 \dots 2m - 1$.

The last m objects are the possible present states for m intervals and the first m objects are the m possible time intervals. The objects are ordered such that $IO_p < IO_q$ if $\tau_p < \tau_q$.

The vectorial representation of the value of the system is $\vec{V} = \{v_0, v_1, v_2, \dots, v_{2m-1}\}$. As an added benefit, we can represent two or more systems by adding their value vectors.

3. Ringtuple: Visualizing the vectorial value

Although $V(k)$ is a scalar, it may be visualized as a total area in a shape. We map the value of $V(k)$ to a circle with unit area. The various v_i should be represented by partitioning the area of the unit circle proportionally to their values. In a pie chart, this is done by defining wedges with area proportional to the values to be mapped. In our case, we map our values to areas in the circle that are defined by concentric circles.

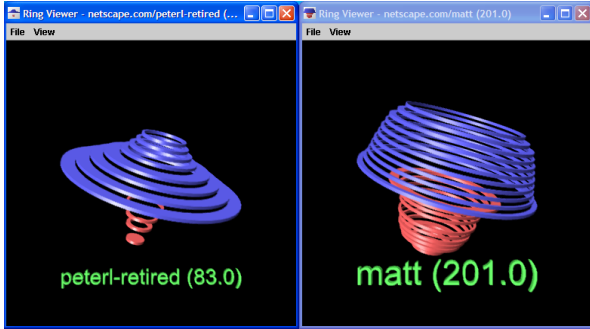


Figure 1. Two Ringtuples.

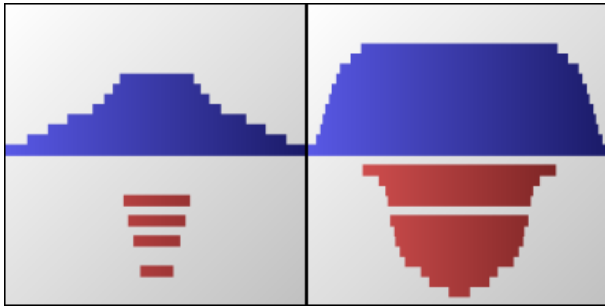


Figure 2. Icons of Ringtuples.

We may create a vector of radii \vec{R} as in $\vec{R} = \{r_0, r_1, r_2, \dots, r_{2m-2}, r_{2m-1}\}$ where $r_p = \sqrt{\frac{1}{\pi V(k)} \sum_{j=p}^{2m-1} v_j}$. The rings that are formed by these circles are also displaced along a third dimension according to their position in \vec{R} as it shown in Fig. 1.

This visualization of \vec{R} provides an intuitive view of the distribution of the value $V(k)$ among all present and past states of the system. We see that the lower the red rings are translated the worst the system *is* at present. Likewise, the higher the blue rings are translated the worst the system *was* when those objects transitioned from state U to state D . Therefore, we call the blue rings the past rings and the red rings the present rings. We call this representation of vector \vec{V} a *ringtuple*.

Once we learn how to interpret the positioning and the meaning of the rings, a ringtuple can be used to quickly assess the quality of a system or group of systems. We can see that by quickly scanning many ringtuples, one can discriminate sets of systems based solely on their ringtuples. The overall shape of the ringtuple becomes the indicator of the system status and the cognitive powers of the user come to play a decisive role for feature finding. A user can find culprits in a large set of systems by simply looking at the corresponding ringtuples.

If we take the profile of these ringtuples from a point of view perpendicular to the z -coordinate, we may obtain

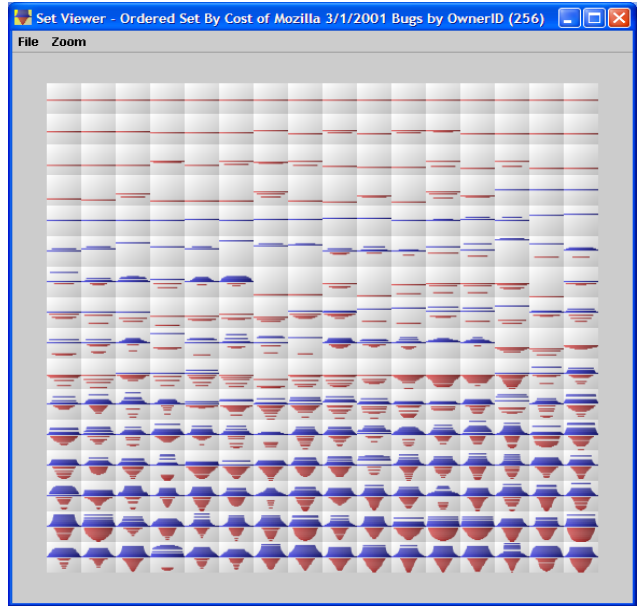


Figure 3. Icon Wall with ordered Ringtuples.

a 2-D mapping of the ringtuples. An example of the result of this procedure is shown in Fig. 2. These two icons are derived from the same ringtuples depicted in Fig. 1.

The idea of using icons or glyphs to represent multivariate data points or attributes has been suggested and used in the past. A good characteristic of the 2-D ringtuple is that it can be used as a *dynamic* icon [3]. Once you know how to interpret the rings, the ringtuple becomes a sensory symbol [2].

We may display many of these icons in an Icon Wall as shown in Fig. 3. These icons are ordered according to their shape and can be used for Visual Data Mining. The underlying ordering function can be used as a metric for partitioning the entire set into clusters based on the shape of the ringtuples.

References

- [1] Matthias Kreuzeler and Heidrun Schumann, A Flexible Approach for Visual Data Mining, *IEEE Transactions on Visualization and Computer Graphics*, 9(1):39-51, January-March, 2002.
- [2] Colin Ware, *Information Visualization Perception For Design*, Morgan Kaufmann Academic Press, 2000, pp. 7-14.
- [3] Mei C. Chuah and Stephen G. Eick, Managing Software with New Visual Representation, *Proceedings of the 1997 Symposium on Information Visualization*, 30-37, October, 1997.

Illuminated Diagrams: Using Light and Print to Comparative Advantage

W. Bradford Paley

Digital Image Design Incorporated (didi)

brad@didi.com

Abstract

A hybrid medium is presented; it exploits the best characteristics of contemporary print and projector capabilities. This large-scale display consists of a print carrying static data, and light projected onto the surface of the print. The projected light adds many capabilities: interactivity, attention direction, and transient detail, while the bulk of the information still comes from the print's ultra-high information density.

1. Introduction

Ultra-high resolution large-scale displays have been in use for centuries, e.g. prints, drawings, and paintings; but their use as an information visualization medium has been largely unexplored. This may result from the difficulty of designing a data representation that takes advantage of the high resolution (with small details or glyphs) but still has enough variation on the global level to help direct the viewer to interesting areas.

We have developed a hybrid display technique that may be easier to design for, since it uses two presentation technologies, feeding different channels of human perception, and begins to describe which information should be presented in which channel.

2. Related Work

Fine artists have projected onto sculpture, paintings, prints, and people for decades, but the intent has been self-expression rather than the representation of information.

Raskar's "Shader Lamps," e.g. [4] project onto dioramas to add textures and simulate physical motion, but do not assume that anything more than the three-dimensional shape of the diorama carries information. The "Virtual Showcase" [1] of a raptor skull superimposes 3D images of muscles on a model of a skull, but again doesn't assume information in the skull itself. Tangible interfaces to projections, e.g. [5] concentrate on manipulating the projected image, not exposing features in the physical projection substrate. The Focus+Context work of Baudisch [2] relates to the current approach but still has significant differences, including an opposite use of the projector: here we use it to direct focus, in that work it is used to provide context. RASA and related work by McGee et. al. [3] project onto paper as we do here, but the purpose is still primarily to augment the paper with

projected low-resolution symbolic information, rather than using the lower resolution as a "smart spotlight."

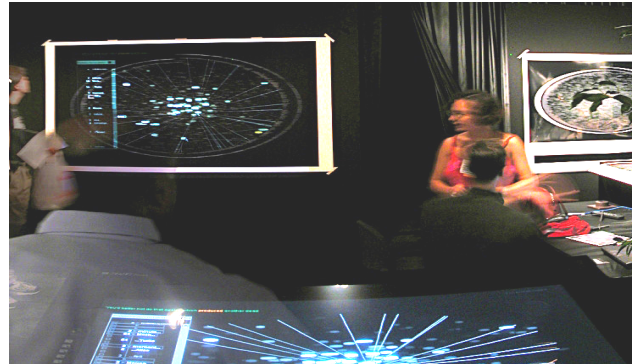


Figure 1: An illuminated Diagram at SIGGRAPH 02

3. Physical Setup

An information-dense print (in Fig. 1 roughly 6' by 4' print of a TextArc of Lewis Carroll's *Alice's Adventures in Wonderland*) is mounted on a wall. A projector is focused on the print, and the computer running it is calibrated so that the computer can locate any point on the print. A touch screen (seen at the bottom of Fig. 1) allows user control.

4. Two Channels, Two Kinds of Information

An Illuminated Diagram uses two technological channels, print and projection, to display two very different types of data: A representation of a vast amount of spatially arranged static data, and a relatively small amount of moving data.

4.1. Static printed data

Some kinds of static data can be well represented as tiny, self-contained glyphs. In TextArc, the individual words fit this description, as do the star-like "distribution glyphs" to the left of each word, as seen in Fig. 2, a detail of a print.

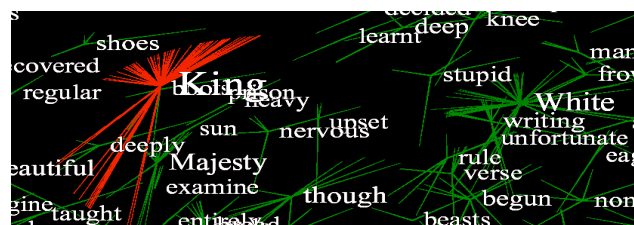


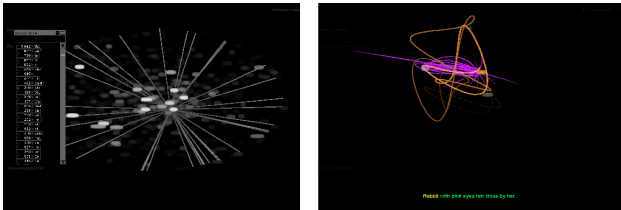
Figure 2: Printed words and distribution glyphs

Other static data might be represented as tiny charts, graphs or full networks, e.g. Connecting lines or enclosing contours that would become hopeless tangled in an ordinary static print this dense can become useful once we can indicate and group them with the light from the projector.

4.1. Dynamic projected data

The type of dynamic data that seems to work well in an Illuminated Diagram directly addresses or enhances the static image. Projected dynamic data can serve many distinct information visualization purposes. It can:

- Direct attention to a specific spot, static or moving
- Identify a specific shaped area, and how it may change
- Relate several elements to one another (with color, brightness, a network of lines, or motion similarities)
- Reveal a complicated diagram a section at a time, helping people learn how to use it
- Add annotation in the form of new lines, shapes, glyphs, an overlay window, or text
- Override a visual encoding in the print—e.g. a bright spot in the print left unlit will draw less attention



Figures 3a, 3b: Projected overlays, by themselves

5. Poster Example: an Illuminated TextArc

In a TextArc the static data is relative word frequency (controlling the brightness and size of a word) and distribution (each word is positioned at the centroid of its uses around the ellipse representing the entire text; its glyph points out each use). The dynamic data takes six forms: spotlights, glyph extensions, a “story line,” word association nets, an overlay concordance window, and a subtitle; all controlled by a touch screen on the projector’s computer.

Spotlights pick out each word as the viewer scans their fingers over the TextArc by using a touch screen. The touch screen displays only a very crude representation (Fig. 3a) encouraging the user to look at the print for details. Spotlights are matched in size to the word the viewer is touching, and “feathered:” gradually reduced in brightness around the edges. Feathering helps reduce distracting aliased edges—harder to overlook than they are in a normal projection because of their juxtaposition with the fine resolution of the print. The concordance window is readable despite the fact it is projected on black.

Glyph extensions extend the lines that make up the distribution glyphs. This helps find relationships between the

touched word and words in the ellipse, and also directs attention more firmly back towards the touched word.

The story line, word association nets, and subtitle get projected when the viewer tells the TextArc to show the original sequence of words—demonstrating the mapping from the linear text as originally written to the spatially hashed version produced by the rules that make a TextArc. Fig. 3b shows them as they are projected on the print.

6. Other Domains

Other knowledge work domains have a similar split between a vast amount of static details and a need for dynamic data overlays. We have identified such splits and done initial conceptual design for Illuminated Displays in shipping, trucking, rail and air traffic control; industrial and chemical process control; bioinformatics and financial statistical information; and science, art museum, and educational displays.

7. Effectively Engaging Human Perception

Illuminated Diagrams engage with at least two different capabilities of human perception more effectively than many other information visualization techniques. They provide the eye with as much information as can be distinguished (or more), at the borders of acuity, while retaining the ability to use animation, light and color changes and motion to direct attention. They may also tap into the extreme dynamic range of light levels that the eye can distinguish—more than either the best print or best projector can independently.

8. References

- [1] O. Bimber, B. Fröhlich, D. Schmalstieg, and L.M Encarnaçãõ. “Virtual Showcases” *SIGGRAPH ’01 Proceedings* ACM Press, July 2001, pp. 277.
- [2] P. Baudisch, N. Good, and P. Stewart. “Focus Plus Context Screens: Combining Display Technology with Visualization Techniques” *Proceedings of UIST ’01*, November 2001, pp.31-40.
- [3] D. McGee, P. Cohen, and L. Wu. “Something from nothing: Augmenting a paper-based work practice with multimodal interaction” *Proceedings, Designing Augmented Reality Environments*, ACM Press, 2000, pp. 71-80.
- [4] R. Raskar; G. Welch, K-L Low, D. Bandyopadhyay, “Shader Lamps: Animating Real Objects with Image Based Illumination” *Eurographics Workshop on Rendering*, June 2001
- [5] B. Ullmer, H. Ishii, “The metaDESK: Models and Prototypes for Tangible User Interfaces” *Proceedings of UIST’97*, ACM Press, 1997, pp. 223-232.

TextArc: Showing Word Frequency and Distribution in Text

W. Bradford Paley

Digital Image Design Incorporated (*didi.com*)

brad@didi.com

Abstract

TextArc is an alternate view of a text, tailored to expose the frequency and distribution of the words of an entire text on a single page or screen. In texts having no markup or meta-information, one of the quickest ways of getting a feeling for the content of a text is to scan through the words that are used most frequently. Knowing the distribution of those words in the text can support another level of understanding, e.g. helping to reveal chapters in a text that concentrate on a specific subject. A structure and method of displaying an entire text on a single page or screen is presented. It reveals both frequency and distribution, and provides a well-understood and organized space that works as a background for other tools.

1. Introduction

TextArc was developed to help people deal with the ever-increasing influx of data they are forced to accept and integrate into their knowledge base. Much of that data comes in the form of raw text—e-mails, news stories, academic papers, and even a significant amount of data that could theoretically be categorized or indexed still comes to us as plain ASCII. TextArc was developed as a way to get an overview of a medium-sized body of raw text, e.g. the amount one might receive in a single day or week, and provide pointers into that text to let people more easily get to the things meaningful to their goals.

2. Existing Text Overview Methods

There are already many tools directed getting an overview of texts. Simple indices, concordances, lexicons, and other structured lists of words have been serving well for centuries. Computational linguistics techniques have recently added tools that generate automatic summaries, identify key ideas, and do semantic analysis. Several graphical techniques have also been developed to address this need, in the hopes of tapping into the vast visual processing capabilities of the human brain. Self-organizing maps have been deployed, as have multidimensional scaling techniques, to help users group similar concepts.

These approaches generally factor out one key dimension that has great meaning in a text: its original linear order. Since authors spend so much effort in crafting that order we tried to develop a technique that would

respect and build on that order. This was done in the hopes that the new view to be complimentary to existing graphical and non-graphical text overview techniques, with the conviction that the expressive variety among views available to any knowledge worker is as important as the design of any one view.

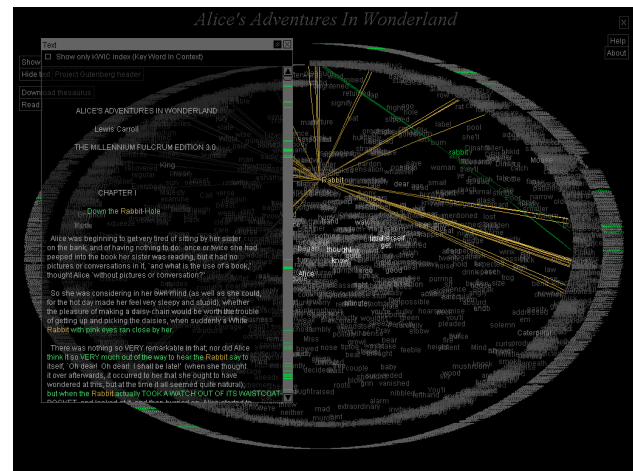


Figure 1: The interactive TextArc at textarc.org

3. TextArc Structure

A TextArc is a structure built entirely of the words in a text, generally placed in the same order that they appear in the text. Words that lose much of their meaning when taken out of context (“stop words” such as “and,” “if,” “the,” e.g.) are not initially drawn, though they may be turned back on in a control panel in the interactive version. Words are also “stemmed;” grouped together based on their word stem (e.g. “jump,” jumped,” and “jumping” are represented by one word), though they can be ungrouped in the same control panel.

3.1. Text line placement

To create a TextArc first the entire text is drawn in an ellipse around the outside of the page or screen, line by line, in a tiny—potentially even unreadable—font. Lines are positioned at even increments around the ellipse: starting at the top center, keeping their baseline horizontal, and stepping each line’s starting point clockwise around the ellipse. The steps around the ellipse are scaled to make the last line appear next to the first line: the angle of each step is roughly 360° divided by the number of lines in the text.

A radial baseline for lines would improve readability at the top and bottom, and is scheduled for future work.

Every line is drawn in order to retain the typographic structure of the document. Chapter breaks, headings, block quotes, poetry, references, and other typographically distinct features become visual landmarks to help users orient themselves at a global level.

3.2. Text word placement

The text is then repeated word by word along an inner ellipse. Words are as close as we can come to concepts in a raw text without making significant assumptions about language or using knowledge external to the corpus presented. TextArc was designed to work as well in Japanese as English, for example, or even in unknown languages or coding systems.

While lines strictly adhere to the rule positioning them around an ellipse, words do not. One additional rule causes the word scattering that is the key organizing principle of TextArc: if a word appears more than once in the text it is drawn only once, at the centroid of all of the points around the ellipse where it “should” appear. This fills the center of the ellipse with words that appear more than once. The averaging action of the centroid “pulls” words toward the center if they are distributed evenly throughout the text, or alternatively pulls them away from the center if they are not evenly distributed; placing them closer to chapters in which they appear more often.

3.3. Overall elliptical shape

The text is drawn in an ellipse to maximize the use of rectangular screen area, to minimize the amount of word overlap, and to ensure a relatively consistent “pull” on every word from each position it appears in the text.

The ellipse is “broken” at the top: we reduce the radius of the circle that is scaled to become the ellipse by just enough to cause an unmistakable visual gap at the top. This is done to make the beginning and ending points distinct, and to make it clear that the text is being written out following a clock metaphor. It is done at the expense of distorting the space that the word positions sample, but the tiny distortions introduced may be unnoticeable since user interpretation of word position is relatively coarse.

3.4. Word brightness and size

Words are drawn on a black background and get lighter as they are used more frequently. This is done on the assumption that a word used more frequently might be more important, so it should stand out from the background more distinctly. In the printed version, excerpted in Figure 2, type size also encodes frequency.

4. Interacting with the Prototype

TextArc has been implemented in the Java programming language, available for free public use at <http://textarc.org>. Interaction allows the relatively low resolution of current computer screens to express a good amount of the information TextArc extracts from a text. For example, when the user’s cursor is over a word lines are drawn to the position each word was used around the arc of the text. In Figure 1, the word “rabbit” is highlighted, showing lines to every position it is used in Lewis Carroll’s *Alice’s Adventures in Wonderland*.

Brushing and linking also help make sense of the display. When an overlay concordance window is brought up, mousing over a word “arms” (brightens) it in that window and in the arc itself. Likewise, when an overlay window showing the original text is visible (as in Figure 1), words clicked and highlighted in the arc are also highlighted in the text.

In an interesting tradeoff of spatial resolution for interaction time, the star-like glyphs next to each word in the printed version simultaneously show every word’s distribution for short texts. No mouse-scanning necessary.

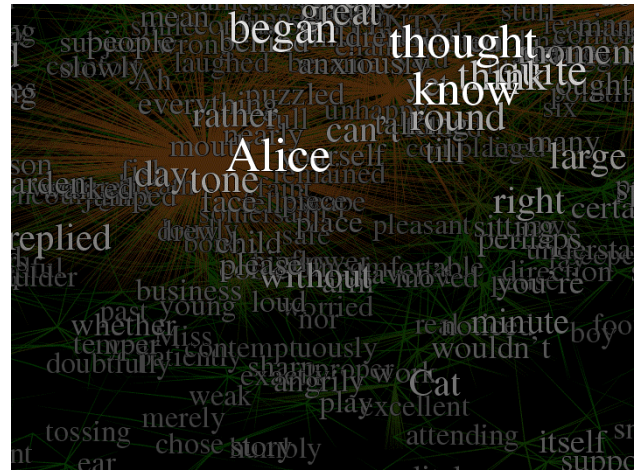


Figure 2: A printed TextArc (detail) with glyphs

5. Summary

TextArc’s unusual structure is well defined and carries some information about a text to the viewer through the generally underused visual channel. It seems to show the distribution of a word in a document very effectively, filling a void left by previous text analysis techniques, and therefore may contribute a new view to help people understand text documents, using their perceptual as well as their linguistic abilities.

Interactive Poster: Visualizing Protein-Protein Interactions on a Genomic Scale

Hilary Spencer and Steven P. Bennett
Stanford University, Stanford, CA 94305
hspencer@stanford.edu , encino@stanford.edu

Recently, high-throughput approaches for discovering protein-protein interactions have created large databases containing pairwise physical interactions in many organisms. Visualization of this data helps biologists understand the underlying cellular processes. Although there have been several automated attempts at visualizing this data, the resulting representations are difficult to visually decode. In this paper, we present ProInAct, a system for viewing large (2000+) numbers of proteins, and visualizing them in the context of their physical interactions. The system is novel in creating clusters of proteins based upon functional classification, subcellular localization, or expression profile data from DNA microarrays, and in drawing orthogonal graphs of the interactions. The system also supports exploration of the network with navigation aides and a tool that finds potential paths between two or more proteins.

1. Introduction

With the invention of techniques such as DNA microarray analysis, researchers can now ask genomic-scale questions about organisms by allowing them to investigate the gene expression profiles of a given organism in one experiment. However, each organism has thousands of genes. Each gene subsequently has a different expression profile by tissue type, which may also differ between experimental conditions and times. In addition to microarray analysis, high-throughput post-translational analysis now allows researchers to collect data on the proteins transcribed by an organism's genes. The collection of these experiments has led to protein-protein interaction databases containing pairwise interaction data for thousands of proteins in several organisms [1, 2]. This interaction data is crucial for interpreting the functions of and relationships between sequenced genes.

One consequence of these large datasets is that they require a well-designed visualization. As the number of known interactions increases, creating and maintaining a mental map of all the relationships becomes very difficult. An interaction network, or a visual representation of the interactions, allows researchers to immediately see how their individual research fits in with the information in the

databases and potentially facilitates the design of new experiments.

Most of the recent methods create straight line drawings of the interaction data, using a force-directed graph layout method [3]. Although these representations allow for the visualization of large number of interactions, they are difficult to process mentally, especially for beginners, and the discovery of potential pathways is inhibited by the wide range of edge angles and edge crossings. These graphs are also static images and therefore do not support interactive path finding.

The goal of ProInAct was to create a clear and useful protein interaction visualization that is similar to hand drawn models [see 4 for example], but fully automated in order to handle large datasets. The system was designed to facilitate tracing all the known interactions for any given protein and support basic levels of interactive exploration. ProInAct is also unique in integrating other sources of data into the visualization, such as gene expression profiles.

2. Methods and Implementation

To build the protein-protein interaction map, we limited our study to the budding yeast, or *Saccharomyces cerevisiae*. The interaction data was collected from several databases: DIP (Database of Interacting Proteins), MIPS (Munich Information Center for Protein Sequences), and YPD (Yeast Proteome Database), and from studies by Uetz et. al. and Ito et. al [1, 2, 5, 6], resulting in 15,264 interaction pairs. The interaction data was clustered in 3 different ways. The first and second methods used information about subcellular localization and functional classification (obtained from the Incyte Genomics BioKnowledge database), and resulted in 24 and 40 clusters, respectively. The third clustering method was based on experimental data from the first microarray experiment, contained in the Stanford Microarray Database [7]. This method used hierarchical clustering to determine the relationships between two genes based on their expression vectors. The clusters are then drawn as color-coded orthogonal subgraphs in a large graph. The graph drawing or visualization component was implemented using the yFiles graph drawing package [8].

The standard model for drawing orthogonal graphs is a network flow model, where each edge has an associated cost, and each node has a constant gain. This algorithm minimizes the number of edge crossings and number of bends in each edge, in addition to minimizing the total area of the graph. The minimum number of bends can be calculated by finding the minimum cost flow in the network, which takes $O(n^2 \log n)$ time. We use a variation of this algorithm (the Kandinsky algorithm) which is better suited to graphs with high connectivity [9].

The visualization component of the system includes a path finding tool that allows a user to input two proteins and visualize the most direct path between them. While not all interaction paths present in the graph occur in the cell, this feature allows researchers to potentially see how a particular pair of proteins may interact. Hiding all but the proteins in the pathway helps to reduce the visual complexity of the network. In order to explore the data, the system includes standard zooming and panning techniques. Additional data about specific proteins, such as other names, can be discovered by right-clicking on a given node.

3. Results

Analysis of the visualization provided interesting insights into the yeast cell. An examination of the functional cluster "protein degradation" shows that all of the nodes are peptide components for the 26S proteasome, a multi-protein complex in the cell responsible for seeking out and degrading other cellular proteins that no longer function properly. The functional clustering only shows that the proteins are all involved in the process of protein degradation, however when combined with the interaction data, one is able to begin to understand how each protein is involved.

We also tested the path finding benefits of the system by selecting two proteins known to be involved in maintaining cell wall integrity in yeast: PKC1 and MPK1 [10]. These proteins formed a clear pathway in the graph of proteins clustered by function, though the intermediate protein BCK1 was missing from the visualization.

4. Discussion

ProInAct offers a unique method of visualizing protein-protein interactions which is easy to interpret and takes advantage of the large amount of data available from both high-throughput post-translational analyses and DNA microarray experiments. It is fully automated, though by mimicking the hand drawn interaction maps, it takes advantage of drawing conventions that biologists are already familiar with. The use of orthogonal graph drawing techniques facilitates the tracing of pathways in

the map and allows biologists to easily make connections with existing literature on known interactions. At the moment, ProInAct does not allow edges between different clusters in the same graph, a limitation that we hope to address in future releases.

- [1] Mewes, H. W., et al., MIPS: a database for genomes and protein sequences. *Nucleic Acids Res.*, 2002. 30(1): pp.31-4.
- [2] Xenarios, I., et al., DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.*, 2002.30(1): pp. 303-5.
- [3] Basalaj, W. and K. Eilbeck. *Straight-Line Drawings of Protein Interactions*, in LNCS: Graph Drawing, J. Kratochvil, Editor. 1999, Springer-Verlag.
- [4] Kohn, K.W., Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Mol. Biol. Cell*, 1999.10(8): pp.2703-34.
- [5] Uetz, P., et al., A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature*, 2000.403(6770):pp.623-7.
- [6] Ito, T., et al., Toward a protein-protein interaction map of the budding yeast. *Proc. Natl. Acad. Sci. USA*, 2000. 97(3): pp. 1143-7.
- [7] Spellman, P.T., et al., Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Mol Biol Cell*, 1998. 9(12): pp. 3273-97.
- [8] Wiese, R., M.Eiglsperger, and P. Schabert. *yFiles Graph Drawing Package*. 2002, yWorks.
- [9] Di Battista, G., et al., *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, New Jersey. 1999.
- [10] Minden, A., and M. Karin. Regulation and function of the JNK subgroup of MAP kinases. *Biochimica et Biophysica Acta* 1007. pp.F85 -F104.

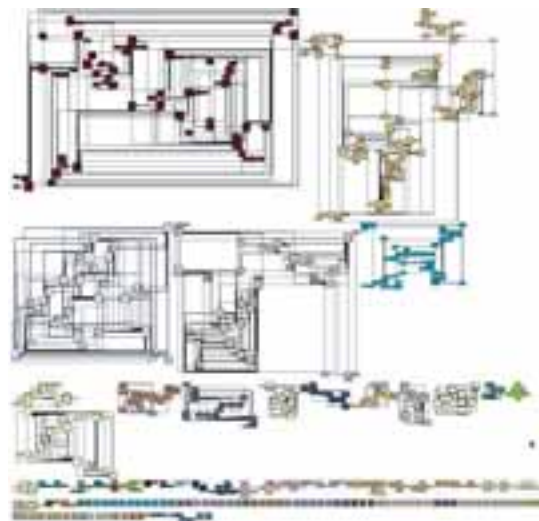


Figure 1: Proteins clustered by functional interaction.

Interactive Poster: Visualisation of Neurophysiological Data

E. Stuart, M. Walter and R. Borisyuk

*Centre for Neural and Adaptive Systems, School of Computing,
University of Plymouth, Plymouth, Devon, UK, PL4 8AA
lstuart@plymouth.ac.uk*

1. Introduction

It is generally understood that the human brain encodes and stores information. However, debate is ongoing about how this information is programmed into the individual components of the brain. A large number of Neuroscientists subscribe to the concept of temporal coding, which argues information is inherent in the patterns of spike trains, how often spikes occur and when spike occur relative to each other. Hence, the principle of synchronisation of neural activity [1] [2] is one of basic principles that underlie information processing in the brain.

Currently, large quantities of simultaneously recorded multidimensional spike train data sets are available. However, software tools to support the analysis of this data are incomplete. The quantity of data recorded is also due to significantly increase as hardware limitations are surmounted. As hardware constraints are reduced, it is essential to explore new approaches for the analysis of these vast data sets.

One of the common techniques used to analyse the synchrony between two simultaneously recorded spike trains, is the Cross-correlogram. This is a simple, yet very effective, technique where output is usually presented as a 2-dimensional histogram. Inevitably, as the quantity of data increases, this forms of analysis becomes infeasible. Hence, the requirement for improved software support is established and it is clear that new methods need to be defined to support the analysis of these increasing data sets. At the Multiple Unit Laboratory [3], support for the analysis of this data has been addressed for some time and several analysis tools have been developed. One of their most significant tools is the "Gravity Transformation" [4] [5]. The gravity transformation is credited as a significant contribution to the analysis of this type of data. However, the standard output facility, known as a distance graph, is not suitable for large numbers of particles.

2. Parallel Coordinates

In 1990, Inselberg [6] [7] was largely responsible for the regeneration of parallel coordinates, particularly in the field of Information

Visualisation. Since this time, parallel coordinates have been used in a variety of ways to visualise dense, multivariate data sets.

The novelty of parallel coordinates lies in the fact that axes are adjacent to each other, as opposed to orthogonal axes traditional in Cartesian coordinates. Data points are denoted as vertical axis coordinate values distributed along the horizontal axis. Hence, a specific point in n -dimensional Euclidean space is represented by n vertical axis values distributed along the horizontal axis.

Since 1990, researchers in the area of information visualization have experimented with a variety of derivations of parallel coordinates such as the concept of hierarchical parallel coordinates [9]. In this paper, the use of another derivation, animated parallel coordinates, is proposed and evaluated for data exploration.

3. Software support for data exploration

It is widely accepted that parallel coordinates may be used to identify and reveal the relationships in multivariate data sets. This technique of representing data is useful as a means of presentation and exploration of data sets. The use of parallel coordinates in the animation of large complex data sets is demonstrated.

For the data produced by the gravity transform algorithm, each parallel coordinate displayed represents the position of a particle in n -dimensional space, where n is the number of adjacent axes in the display. See Figure 1 where $n=10$. Note the ten vertical axes used to represent data points in 10-dimensional space. Recall, each individual (10-dimensional) coordinate is represented by a line that intersects each of the vertical axes.

The animation of the parallel coordinate display, that represents the position of all the particles at time t , has provided a greater understanding of the inherent synchrony of the particles over time. As a result of this, parallel coordinate displays have been used for relatively larger values of n than the standard output display of the gravity transformation [8]. It is also useful in depicting the aggregation of particles that occurs as the gravity transformation algorithm collapses.

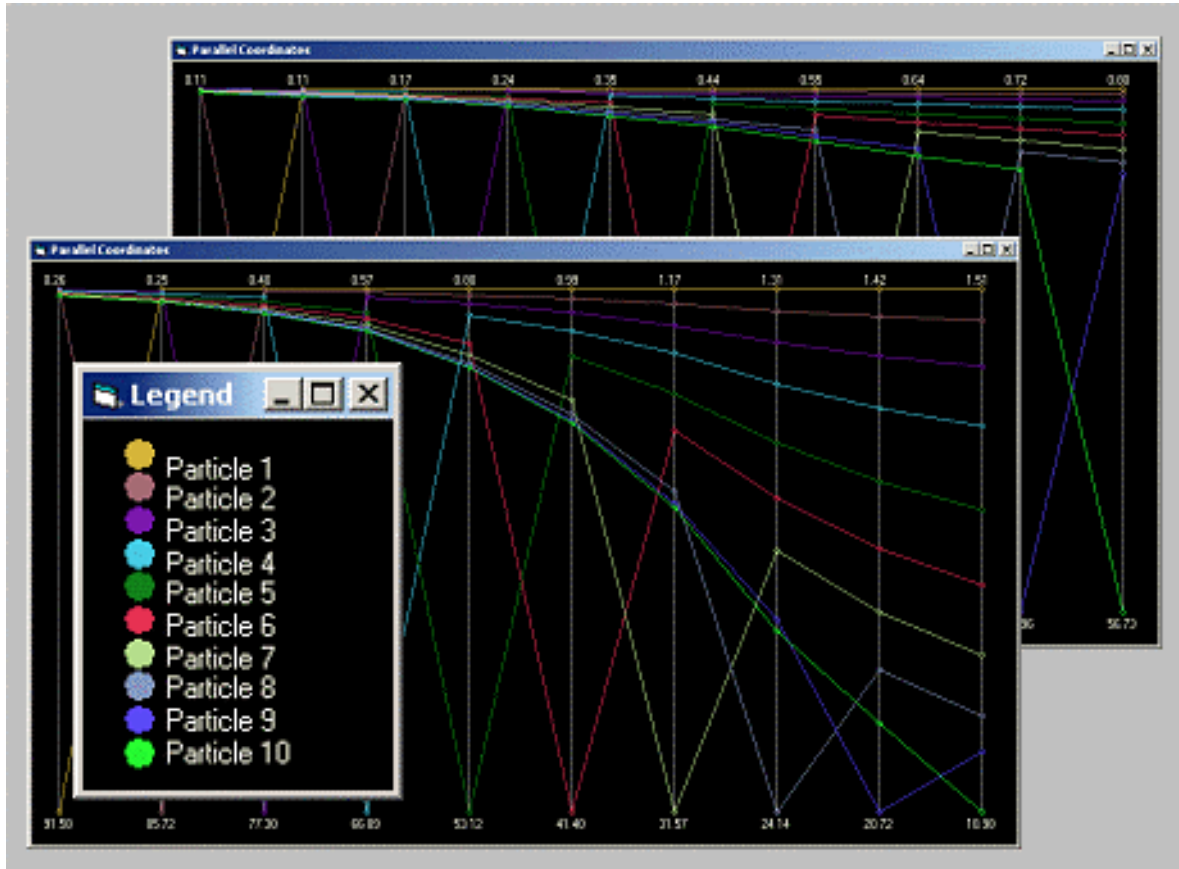


Figure 1: Snapshots from the animated parallel coordinate displays of VISA

This animation technique is part of a software system called VISA, Visualisation of Inter-Spike Associations, which supports the analysis of multidimensional spike train data. Additionally, the system supports the display of any subset of data for closer inspection, effectively zooming in upon selected data. VISA, also supports the representation of data in a static mode.

This research is supported by the Engineering and Physical Sciences Research Council grant number GR/N04904. Additionally, the research of Roman Borisyuk is supported in part by the Russian Foundation of Basic Research (grant 99-04-49112) and by EPSRC (grant GR/N63888/01).

References

- [1] Borisyuk RM & Borisyuk GN (1997), "Information coding on the basis of synchronisation of neuronal activity" *BioSystems*, 40, 3-10.
- [2] Fries P, Neuenschwander S, Engel AK, Goebel R & Singer W (2001) "Rapid feature selective neuronal synchronization through correlated latency shifting" *Nature Neuroscience*, 4(2), 194-200.
- [3] Gerstein GL & Aertsen AM (1985) "Representation of Cooperative Firing Activity Among Simultaneously Recorded Neurons" *Journal of Neurophysiology*, 54(6), 1513-1528.
- [4] Gerstein GL et al (1985), "Cooperative Firing Activity in Simultaneously Recorded Populations of Neurons: Detection and Measurement" *Journal of Neuroscience*, 5(4), 881-889.
- [5] <http://mulab.physiol.upenn.edu/index.html>
- [6] Inselberg A & Dimsdale B (1990), "Parallel Coordinates: A tool for visualising multidimensional geometry" In *Proceedings of Visualization'90*, 361-378.
- [7] Wegman EJ (1990), "Hyperdimensional Data Analysis Using Parallel Coordinates" *Journal of the American Statistical Association*, 85(411), 664-675.
- [8] Stuart, L., Walter, M. and R. Borisyuk, *Visualisation of Synchronous Firing in Multi-dimensional Spike Trains*, In *BioSystems* (journal paper in press), (2002).

Interactive Poster: Interactive Data Visualization using Mondrian

Martin Theus

University of Augsburg, Department of Computeroriented Statistics and Data Analysis
Universitätsstr. 14, 86135 Augsburg, Germany, martin.theus@math.uni-augsburg.de

1 Introduction

This poster presents the Mondrian data visualization software. In addition to standard plots like histograms, barcharts, scatterplots and maps, Mondrian offers mosaic plots for high dimensional categorical and parallel coordinates/boxplots for high dimensional continuous data. All plots support linked highlighting, interactive queries and zooming. All interactions follow a very strict set of conventions to flatten the learning curve.

Unique is Mondrian's special selection technique, which allows advanced selections in complex data sets.

Besides loading data from local (ASCII) files it can connect to databases, avoiding a local copy of the data on the client machine. Special plotting techniques have been implemented for plotting very large data sets.

Mondrian is written in 100% pure JAVA.

2 Special Plots for High Dimensional Data

Although linking and highlighting across different plots can already increase the number of dimensions to look at simultaneously, it is desirable to find visualizations which include many variables at a time. Mosaic plots for categorical data and parallel coordinate plots for continuous data are ideal for gaining insight into high dimensional data.

2.1. Mosaic Plots

Mosaic plots are a relatively new development. Recent implementations include a static version for S-Plus and R by Emmerson [1] and an interactive version by Hofmann [2] within the MANET software. To flexibly reorder the variables in the plot and to include and exclude variables the four arrow keys are used. Empty cells which occur very often if the number of crossed categories is very high, are not subdivided on lower levels. In situations with many crossed variables this usually reduces the number of cells to draw drastically.

Since Mondrian supports queries there are no labels printed around a mosaic plot. With only a few variables

and categories in a mosaic plot, labels usually fit around the plot. But more complex plots with, e.g. 8 binary variables would need twice as much space for the labels as for the plot itself. Additionally to queries, the name of the data set and the names of the variables in the plot are shown in the title bar of the plot window. Many data sets and most database queries present data in an already summarized form, i.e. a table. In Mondrian mosaic plots as well as barcharts can handle data which is summarized, specifying attribute variables and a count variable. Obviously any non-negative numeric variable can be used as a weight variable, which allows for very flexible plots, which might be hard to interpret. A simple look up of values can be performed with barcharts by weighting case names by their values.

2.2. Parallel Coordinates/Boxplots

Parallel Coordinates are a powerful tool to analyze high dimensional data sets graphically. Since static representations of parallel coordinates are usually not very revealing several interactive implementations arose very early. The probably most advanced implementations are the ones of Inselberg [3] and Wegman [6]. In addition to the standard selection, highlighting and interrogation methods parallel coordinates in Mondrian support the following features: Coordinates can be rearranged manually to look at the most interesting adjacencies. Usually only a few adjacencies are of interest. Zooming is implemented for each axis individually. Since parallel coordinates are cluttered very much with an increasing number of observations displayed, zooming can focus on a more detailed view of the variable.

Mondrian offers a special feature to plot categorical variables in parallel box/coordinate plots. Whereas most implementations only use the number coding of a categorical variable, Mondrian plots a stacked barchart, with left-to-right highlighting for each categorical variable. This display is consistent with all other plots representing counts. Additionally lines can be displayed for the highlighted points in boxplot mode. Wills [7] gives an alternate method of incorporating categorical variables into parallel coordinates based on circles-sizes, which is not compatible to the way

counts are displayed in barcharts.

3 Smart Selection

The main tasks in interactive data visualization is the identification of patterns and subgroups. Thus selecting and identifying data is of major importance.

3.1. The Progress in Selection Techniques

The *standard* way of selecting data is to select data and by doing so replace any other selection that might have been present. There is no way of refining a selection or selecting over different plots and/or variables. A more *advanced* way to handle selections is to allow to combine the current selection with a new selection with boolean functions like *and*, *or*, *Xor*, *not*. This allows to refine a selection step by step to drill down to a very specific subset of the data. When dealing with a whole *sequence* of selections, it is often desirable to change a selection at an earlier stage, without having to redefine all preceding and successive selection steps. By storing the sequence of selections it is possible to make changes to any step in the sequence. Selection Sequences have been first implemented in MANET [5].

The approach used by Mondrian combines the advanced selection as described in Wills [7] and Shneiderman's [4] dynamic queries. Mondrian keeps a list of any selection associated with a data set. For each entry in the list the selection area in screen coordinates and data coordinates, the selection step, the corresponding plot window and the selection mode (e.g. and, or, not) is stored. The currently selected subset of the data can then be determined by processing all elements of the list, no matter which kind of modification to the list was the reason for an update of the selection subset.

3.2. Selection Rectangles

Allowing multiple selections in a single window as well as across different windows makes a visual guide to the selections performed indispensable. Mondrian introduces *Selection Rectangles*. Selection rectangles indicate the area which was selected. An existing selection rectangle can be used as a brush by simply dragging the selection rectangle. The eight handles on the rectangle permit a flexible resizing of the rectangles. This enables various slicing techniques.

The selection mode can be changed via a pop-up menu. The deletion of a selection can be performed via this pop-up, too. The last selection which was performed can be deleted by simply pressing the backspace key. Only the last selection is plotted in black. Selections performed at an earlier stage are plotted in a lighter gray to make them less dominant in the plot. Since selections are stored in terms of the data coordinates they are invariant to any alterations of a

plot. Typical scenarios are things like interactive reordering of the axes in a parallel coordinate plot, flipping the axes in a scatterplot or zooming. These operations automatically update the selection rectangles. The new screen coordinates of the selection rectangles are calculated from the data coordinates. The ability to handle more than one selection in one window is indispensable when dealing with parallel coordinates.

4 Database Connections and very large Data Sets

Mondrian allows to connect directly to databases. When connected to a database, Mondrian does not keep a copy of the data outside the database. The data which is needed for the plots is retrieved via SQL queries. Obviously this only makes sense for plots which just need a summary of the data, which are, barcharts, mosaic plots, histograms and binned scatterplots. The amount of data which is needed to build these plots is a constant, which does not grow with the size of the data itself.

Special modifications to standard plots have been implemented in Mondrian, in order to make these plots usable for even very large data sets. Scatterplots can be displayed in a binned version, i.e. as the result of a 2-dim. breakdown over the x - and y -variable. Additionally α -channel transparency can be used for density estimations in scatterplots as well as to avoid overplotting in parallel coordinates.

References

- [1] J. Emerson. Mosaic displays in S-PLUS: A general implementation and case study. *Statistical Computing & Statistical Graphics Newsletter*, 9(1):17–23, 1998.
- [2] H. Hofmann. Simpson on board the titanic? interactive methods for dealing with multivariate categorical data. *Statistical Computing & Statistical Graphics Newsletter*, 9(2):16–19, 1998.
- [3] A. Inselberg. Visual data mining with parallel coordinates. *Computational Statistics*, 13(1):47–63, 1998.
- [4] B. Shneidermann. Dynamic queries for visual information seeking. *IEEE Software*, 11(6):70–77, 1994.
- [5] M. Theus, H. Hofmann, and W. A. Selection sequences — interactive analysis of massive data sets. In *Proceedings of the 29th Symposium on the Interface: Computing Science and Statistics*, 1998.
- [6] E. Wegman and Q. Luo. High dimensional clustering using parallel coordinates and the grand tour. In *Computing Science and Statistics 28*, pages 352–360, 1997.
- [7] G. Wills. Selections: 524,288 Ways to Say: "This is Interesting". In *Proceedings of InfoViz '96, IEEE Symposium on Information Visualization*, pages 54–60. IEEE Computer Society Press, 1996.

Interactive Poster: Glorps - charting multi-dimensional data

Glenn Thorpe
glenn@thorpies.com

INTRODUCTION

A glorp is a method of charting multi-dimensional data. Glorps provide a natural three dimensional grid for logically locating datapoints representing all the possible incremental combinations of input values. This grid allows the visual identification of trends resulting from changing input values. As with univariate charts and 3D bivariate surface charts, resulting features such as steady gradients, plateaus, curves and sudden breaks are visible.

SIX COMPONENT FORM

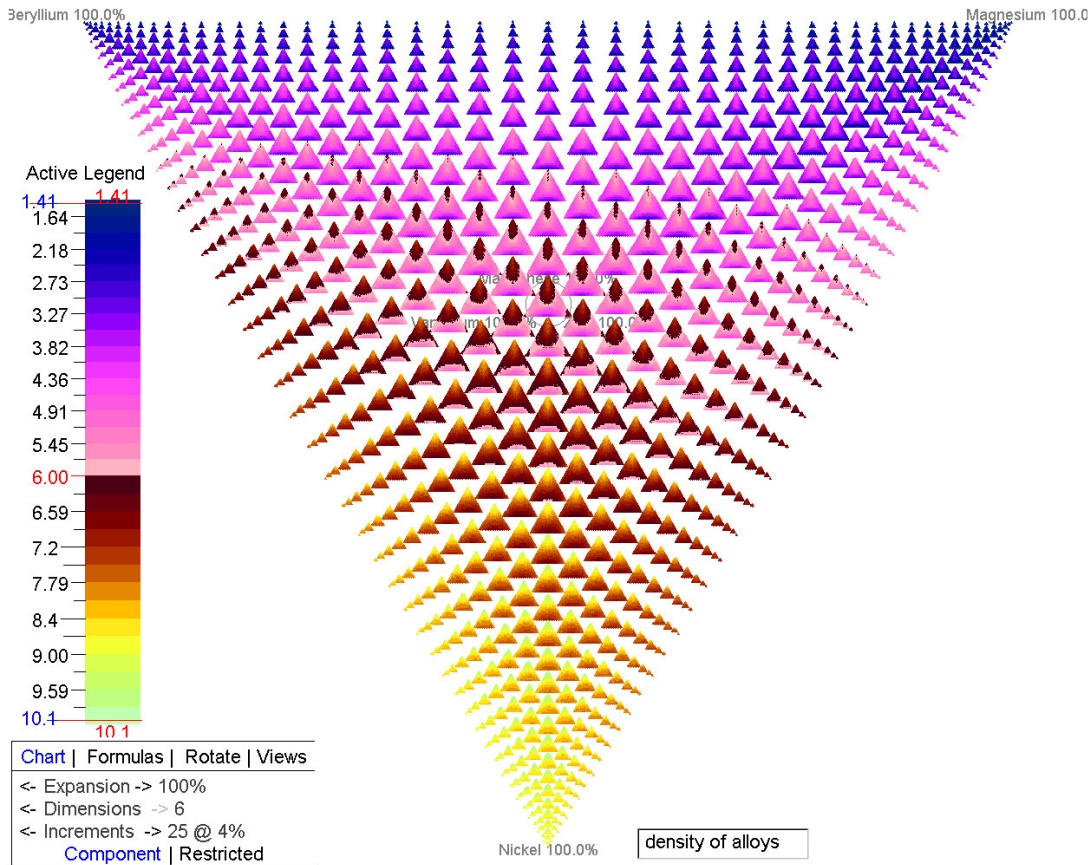
The glorp comes in two forms. The full component form is used where an object is composed wholly of

particular components. The glorp is a grid for locating all datapoints representing the possible combinations of the components that can create the whole. Examples of this type of glorp are:

- displaying an outcome, such as mass or electrical conductivity, for all the combinations of a number of elements that can create an alloy
- displaying the returns on an invested amount of money where the money is invested in any proportion between a number of different investment categories.

Figure 1 is a screenshot of a glorp showing the specific gravity of materials comprised of any combination of the elements beryllium, magnesium, nickel, copper, manganese and aluminium.

Figure 1 - Six component glorp showing the density of alloys



In 3D view Figure 1 includes a point representing each of the 142,506 possible 4% incremental combinations of six component elements. From the

glorp, trends resulting from changing the composition from any point in any direction can be visually identified quickly and simply.

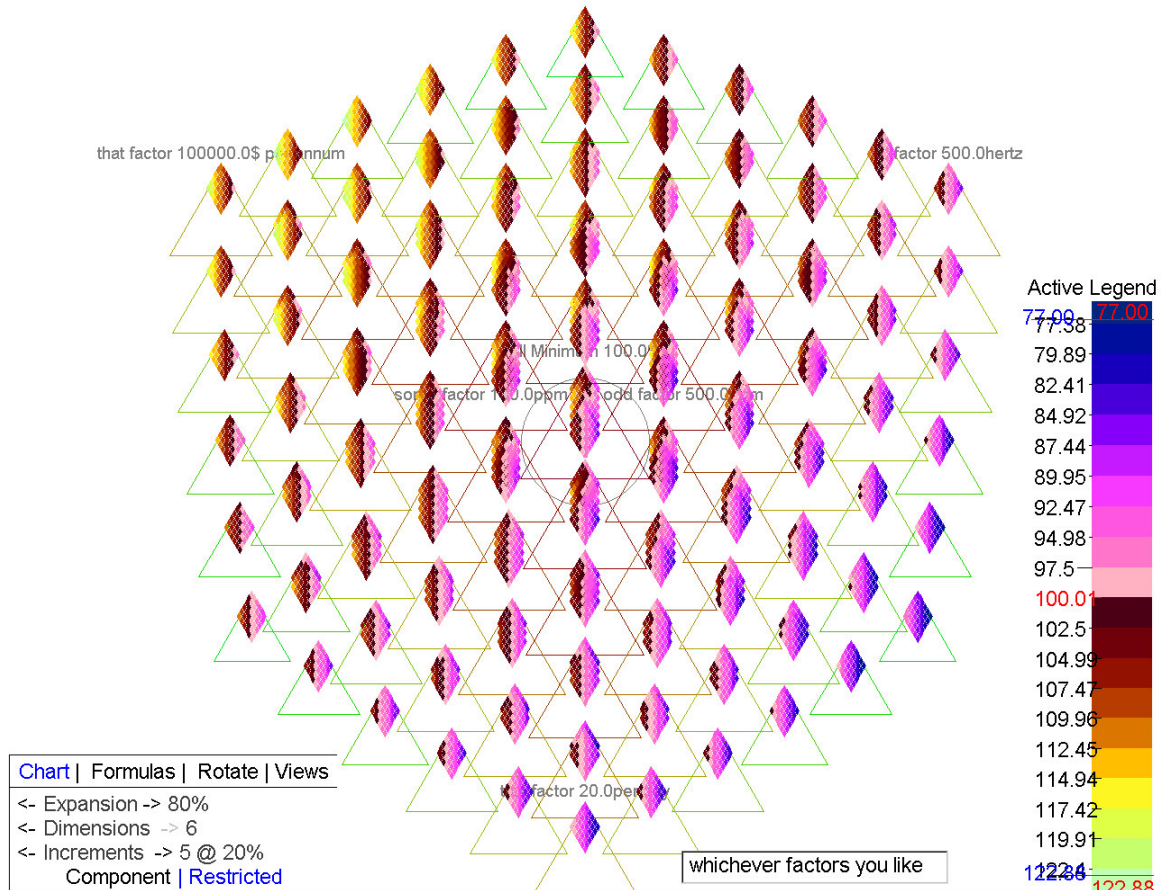
FIVE ORTHOGONAL FACTOR FORM

The second type of glorp is a grid for displaying any combination of fully independent factors. This second type of glorp is a restricted subset of the full component

glorp. One of the axes is replaced with an inverted combined total increment value, or a "lack of" axis. All other values are restricted to 1/numberOfFactors of the total combined number of increments.

Figure 2 displays the effect of simple formulated additive interactions between five independent factors.

Figure 2 - Five Factor glorp



In 3D view the above picture includes a point representing each of the 7,776 possible combinations of five factors each with six incremental values. Each factor is independently scaled.

The triangles are placeholders representing the equivalent locations in the unrestricted six component chart. The "Lack of" axis is located at the top apex of the inner triangle. This lack of axis can be "rotated" to an outer axis to provide a different view.

The five factor chart is suitable for applications such as showing the interaction effects of drugs on clinical measurements. An example would be the effect on blood pressure, where the five factors may be three drugs (each with different bloodstream concentration ranges) and two other clinical traits such as age and existing blood pressure.

CURRENT DEVELOPMENT

The glorp concept has been developed to a working 2D model. This model is available on the website www.thorpias.com. The model allows

- adjustment of the number of increments for each component or factor
- adjustment of the formulated additive effect of each input variable
- the rotation of any axis to any position
- scrolling through the 3D surfaces
- adjustment of color coded outcome values through the Active Legend

Illuminating Data Analysis

Antony Unwin, Ulrich Fahrner, Robert Schmied
Department of Computer-Oriented Statistics and Data Analysis
University of Augsburg

Abstract

In the 1960s John W. Tukey said "Exploratory data analysis is detective work". Our research group in Augsburg develops interactive graphical software for data analysis (e.g. MANET and MONDRIAN for EDA, CASSATT for parallel coordinates, and KLIMT for trees: www.rosuda.org). The level of interactivity is crucial and involves not just a menu-driven interface but the direct querying, selection and manipulation of the statistical objects displayed. All windows are linked so that whatever happens in one is reflected in all others. Although the software is easy to use after a short introduction and demonstration, it is difficult to describe in text alone how to work with these tools and we are looking for new ways to get across these ideas. We analyze video-based solutions and their multimedial combinations for this purpose.

1. Multimedia Support

Much that is hard to explain by a conventional teaching approach may be conveyed effectively with the aid of multimedia methods, including the use of sound, video or animation. Knowledge that is provided in a text or online handbook can be supported and extended in this way. But these



Figure 1. John Tukey

new approaches have to be developed and evaluated. Properly used they can be supportive for some goals but they are

not suitable for all situations. There are no quick short-cuts to multimedia success. The famous data analyst John Tukey can be seen in a fascinating early film (1973) explaining 3-d rotating plots. He calls himself a detective, who has to find interesting and specific structures in the data. What is the right medium and which combination has to be chosen to show the work of a data detective? A few statisticians have made use of video (e.g. Video Lending Library, The American Statistical Association¹) but not really as well as Tukey. We want to take advantage of the new tools available to see if we could do better. We use camcorders, direct recording of computer screens and the blue box technique for filming comments. All computing was done on a Mac G4 with dual processors using Final Cut Pro.

2. Concentration on Video

We are currently working on two complementary research projects to investigate whether video can be helpful in (a) explaining how to use interactive statistical software and in (b) illustrating the process of data analysis. We are particularly interested in determining for which tasks video is best suited and, conversely, for which tasks alternative media are better. Starting a programme and choosing a data set is hardly a problem for a student with computer experience and can easily be explained in text. Learning how to interpret and explore scatterplots is far more difficult.

There are several alternative ways of using video and we are considering which to employ. Is a monolog clear but perhaps too dull while a dialog may be more interesting but too complicated to follow? Which perspectives should be used when filming and how can what happens on the computer screen be rendered best on film? How should film of speakers be integrated with film of the screen? How long can film clips be while still holding the attention and interest of the viewer?

¹<http://www.bell-labs.com/topic/societies/asagraphics/library/index.html>

3. What is in the interactive poster for Info Vis?

To illustrate the process of data analysis we have taken a data set from this year's Soccer World Cup held in Japan and South Korea (www.fifaworldcup.com). We chose this data set because it is topical, interesting and readily understandable for all. We explored the following five questions to demonstrate different interactive features. A short film of the data analysis and a soccer expert's comments were made for each question.

Did France play badly?

The World Cup holders from 1998 went out without scoring a goal (Used selection and linking between different kinds of plots).



Figure 2. Soccer expert with the French star Zidane in the background

Were the host nations Japan and South Korea favoured by the referees?

Both Spain and Italy complained bitterly about their defeats by South Korea (Using a scatterplot interactively to see how two variables are related).

How did the USA play?

The USA team did better than expected and were tactically astute (Linking many dotplots to show how one team compares to all the others).

Do the data highlight other teams as having a distinctive playing style?

Individual teams are often described as playing differently from others (e.g. Ireland and the long ball game) (Selecting extremes in the dotplots to identify teams with unusual statistics).

Do South American teams have a different playing style from other teams?

In the past South American and European players really only met at World Cups. Now many play together in European leagues (Comparing a group of cases with the rest

graphically. Detecting local and global outliers).

How these questions were investigated using the data set is shown in our five sample videos. They give first insights into the possibilities of using video technology in illuminating statistical software and Exploratory Data Analysis.



Figure 3. Data analyst with graphics displays in the background

4. DVD, Internetpages and Videostreaming

We are still considering the best way to present this material to our target group. We could produce a DVD to accompany a handbook or textbook, but the interface would have to be carefully constructed. (A CD would have too little capacity for videos of sufficient quality, despite the new compression methods like mp4.) The DVDs would have to be playable on every common DVD-player, just like any commercial project. There would be an accompanying Internet site which would be complementary to other media utilised. The embedded videos would be stored on a streaming server, to prevent a lengthy and discouraging download time. The interface would have to permit direct selection, fast forwarding and immediate ending of the videos and must provide informative parallel identifications of each video (a self-explanatory title and a distinctive image from the film at the very least). Whatever the attractions of watching a video in detail the first time around, for further viewings you want to get to the particular chunk of interest. The analyst should be in control when using interactive graphical tools in data analysis and the viewer should have just as flexible a control in using illustrative videos.

5. Conclusions

For explaining EDA it seems to be very useful to implement multimedial strategies. Basically the advantage of multimedia is clear, but it's an open problem to manage the transfer from conventional teaching to teaching enhanced with multimedia. There is a lot to do and we are continuing our research.

Interactive Poster: XmdvTool: Interactive Visual Data Exploration System For High-dimensional Data Sets *

Matthew O. Ward, Elke A. Rundensteiner, Jing Yang, Punit R. Doshi and Geraldine Rosario
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609
{matt,rundenst,yangjing,punitd,ger}@cs.wpi.edu

Keywords: Dimension reduction framework, high-dimensional data visualization, interactive dimension reduction, radial space-filling hierarchy visualization, multi-focus distortion, visualization-driven data management.

1 Multi-Display Visual Data Exploration Basics

XmdvTool is a public-domain software developed at WPI for the interactive visual exploration of large-scale high-dimensional data sets [4, 1, 2, 5]. *XmdvTool* supports a user-driven interactive discovery process of data patterns and outliers. Our multi-disciplinary approach of coupling ideas from the fields of data visualization, data mining and database management has allowed us to overcome major hurdles, including those of display clutter due to high dimensions and display clutter due to large number of data items, as well as the intuitive navigation and efficient data access over such database sets to allow near real-time exploration.

XmdvTool incorporates several distinct display methods for multivariate data visualization that allow the users to view as well as manipulate data from different perspectives [5]. The displays are tightly linked, such that visual interactions via one display can be seamlessly refined via other displays. *XmdvTool* supports a variety of advanced visual interaction tools, including brushing in screen space, data space, and structure space, panning, zooming and distortion [2].

2 Visual Exploration Scale-Up

Conventional multivariate visualization techniques do not scale well with respect to the number of objects in the data set, nor to the number of dimensions, resulting in a display with an unacceptable level of clutter. To overcome this limitation, *XmdvTool* adopts a hierarchical approach and provides a suite of navigation and filtering tools that facilitate the viewing of objects and of dimensions at different levels of detail. This allows the systematic discovery of data trends.

Our past work on structure-based brushing [2] addressed the hierarchical display of a large number of objects. In this demo, we now instead present our latest work on tackling visual exploration when faced with data sets that consist of large numbers of dimensions. When visualizing data sets with a large number of dimensions, existing multi-dimensional visualization techniques (such as parallel coordinates, glyphs and scatterplot matrices) become

cluttered and thus ineffective. In particular, in this demonstration, we will showcase two key techniques towards overcoming this problem, namely, our visual hierarchical dimension reduction framework [7] and our visual exploration tool called the InterRing display [6] which addresses the hierarchical display and manipulation control of a large number of dimensions. To support efficient database access, our latest work on the database management side includes MinMax tree representation of queries, semantic caching, and prefetching [3].

3 Visual Hierarchical Dimension Reduction

To address this problem of high-dimensional clutter, we reduce the dimensionality of the data while maintaining the relationships between data points using a *visual hierarchical dimension reduction approach* [7]. We note that existing dimensionality reduction techniques usually generate lower dimensional spaces that have little intuitive meaning to users and allow little user interaction. In our demonstration, we will instead show a new approach to handling high dimensional data, named Visual Hierarchical Dimension Reduction (VHDR), which addresses these drawbacks.

In VHDR, we construct hierarchical dimension cluster trees based on clustering the dimensions instead of the data points. Once dimensions are grouped into a hierarchy, lower dimensional spaces are constructed using clusters of the hierarchy. Thereafter, we construct low dimensional data spaces guided by user interaction of the hierarchical dimension cluster tree. VHDR not only generates lower dimensional spaces that are meaningful to users, but also allows user interactions in most steps of the process.

Furthermore, we have extended several traditional multidimensional visualization methods, including parallel coordinates, glyphs and scatterplot matrices, to visually convey dimension cluster characteristics when visualizing the data set in lower dimensional spaces. Our case study of applying VHDR to real data sets (Figure 1) confirms that this approach is effective in supporting the exploration of high dimensional data sets.

4 InterRing: Visual Interaction Tool

To allow the user to view, navigate and reconfigure the hierarchy of dimensions, we have developed the InterRing display tool (Figure 2). InterRing is a radial, space-filling (RSF) hierarchy visualization display. As an RSF-based display, it has the ability to efficiently use the display space while effectively conveying the hierarchy structure. Several RSF systems and tools have been developed to date, each with varying degrees of support for interac-

*This work is supported under NSF grants IIS-9732897, IRIS-9729878, and IIS-0119276.

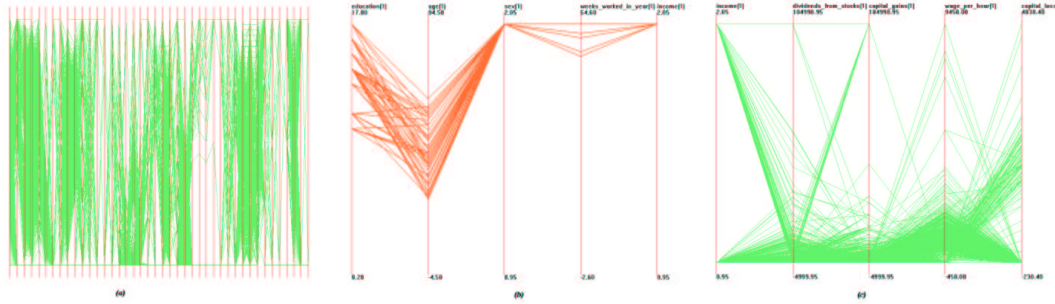


Figure 1: The Census dataset (42 dimensions, 20,000 data items) in parallel coordinates. Figure (a) shows the original high dimensional space. Figure (b) and (c) show two lower dimensional subspaces generated by VHDR.

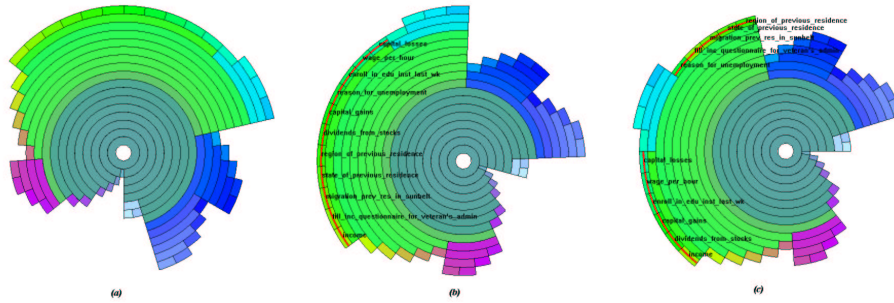


Figure 2: Dimension hierarchy of the Census dataset in InterRing. Figure (a) shows the automatically generated hierarchy. Figure (b) shows the detail of a cluster after brushing and rotation. Figures (c) shows the modified hierarchy after moving some dimensions from that cluster to elsewhere.

tive operations such as selection and navigation. In this demo, we show an extensive set of desirable operations on hierarchical structures, a set which we believe is significantly more complete than what is provided in prior RSF systems. Among other things, InterRing supports multi-focus distortions, interactive hierarchy re-configuration, and both semi-automated and manual selection. In our demonstration, we show the power and effectiveness of these and other operations.

5 Scalability of Data Access

To allow for the near real-time behavior required among interactive tools, we address how database queries can be internally represented, and also employ data caching and prefetching techniques. To support operations such as visual hierarchical drill-down and roll-up, we represent the hierarchical structure as a Min-Max tree [3], thereby reducing the recursive hierarchy navigation queries into range queries. MinMax has allowed us to achieve performance levels required for interactive visualization even when connecting to large persistent data sets on Oracle.

Exploration via visual interaction tools typically results in predictable user navigation patterns. To take advantage of this, we use customized caching and prefetching techniques which we show to be effective in our tool. In particular, we employ semantic caching principles to maintain relevant results of the previously executed queries indexed by their query specifications in the local client buffer. We also apply a variety of prefetching techniques based on both the analysis of current user navigation patterns as well as the archived user navigation history.

6 XmdvTool Source Code and Homepage

XmdvTool 5.0 is implemented in C/C++ with Tcl/Tk and OpenGL primitives. Interaction to Oracle8i is written in PRO*C/C++ embedded SQL primitives. The XmdvTool Home Page at <http://davis.wpi.edu/~xmdv> provides downloads of regular releases of our software.

References

- [1] Y. Fua, M. Ward, and E. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. *Proc. of Visualization '99*, p. 43-50, Oct. 1999.
- [2] Y. Fua, M. Ward, and E. Rundensteiner. Navigating hierarchies with structure-based brushes. *Proc. of Information Visualization '99*, p. 58-64, Oct. 1999.
- [3] I. Stroe, E. A. Rundensteiner, and M. O. Ward. Scalable visual hierarchy exploration. *Database and Expert Systems Applications*, p. 784-793, 2000.
- [4] M. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. *Proc. of Visualization '94*, p. 326-33, 1994.
- [5] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical displays: A general framework for visualization and exploration of large multivariate data sets. *Computer & Graphics*, to appear, 2002.
- [6] J. Yang, M. O. Ward, and E. A. Rundensteiner. Interring: An interactive tool for visually navigating and manipulating hierarchical structures. *InfoVis 2002*, accepted, 2002.
- [7] J. Yang, M. O. Ward, and E. A. Rundensteiner. Visual hierarchical dimension reduction for exploration of high dimensional datasets. *Technical Report WPI-CS-TR-02-22*, 2002.

Interactive Poster: Metavisualization of Dynamic Queries

Christopher E. Weaver and Miron Livny
University of Wisconsin–Madison
Computer Science Department
1210 West Dayton Street, Madison, WI 53706
{weaver, miron}@cs.wisc.edu

Abstract

A *metavisualization* is a visualization of data that represents the interactive structure of another visualization. The complex linking structure of dynamic query visualizations makes them particularly suitable for metavisualization. Here we describe metavisualization of DEVisé visualizations using the *Improvise* visualization system.

Keywords: Dynamic Query, Interactive Structure, Linked Views, Metavisualization

1. Introduction

In recent dynamic query visualization systems [1], users can create and link views interactively during data exploration. Consequently, the user interfaces of typical dynamic query visualizations have become more complex, in terms of both the number of views and the amount of linking. North and Shneiderman [3] pointed out how hard it can be for users to understand these visualizations. Unlike views, links appear only as a side-effect of interaction.

The goal of metavisualization is to reveal the hidden structure in visualizations that different kinds of users need to know to complete their tasks. A metavisualization that is useful to the average user might be different from one useful to a designer, or one used by a researcher to analyze the linking mechanism in a prototype visualization system.

In this poster we describe metavisualization of DEVisé visualizations. We build metavisualizations using our dynamic query visualization and metavisualization system, *Improvise*, which we describe briefly next.

2. Improvise

Improvise is an end-user application for dynamic query visualization. It combines a simple direct linking mechanism, *Live Properties*, with a more powerful indirect linking mechanism, *Dynamic Query Expressions*.

Live Properties is used to link *controls*, which include views, sliders, and other widgets. Each control defines *live properties*, which act as ports through which controls are connected to each other. Each live property can bind to a single *variable*, which contains a value. Each live property has a default value which is used when it is not bound to any variable. Controls are directly linked by changing and responding to changes to variable values.

Dynamic Query Expressions is based on the relational database model. *Improvise* views project records into graphical attributes, and show only those records that pass a filter. Filters and projections consist of expressions defined as trees of operators, including *function operators*, *attribute operators*, and *value operators*. Function operators perform arithmetic and other kinds of calculations. Attribute operators evaluate to the value of a record field. Value operators are user-editable constants.

Indirect links are created by using *variable operators* (which evaluate to the value of a variable) in expressions. Projections and filters can themselves be values of variables bound to live properties. Controls are notified whenever any of the variables they depend upon—directly or indirectly—change. This twofold linking mechanism combines easy creation of simple, common links with more involved creation of complicated, unusual links.

3. Metavisualization of DEVisé Visualizations

DEVisé [2] is an interactive editor and browser for dynamic query visualization of large datasets. In DEVisé visualizations, visual queries on data are presented as scatterplot views that map data records into graphical attributes of shapes. DEVisé has four kinds of links. A *visual link* constrains two views to render identical X and/or Y ranges. A *record link* renders in a destination view only those records which are visible in a source view. A *cursor* is a selection box in a view that has the same X and Y ranges as some other view. *Piles* are visual links in which the linked views happen to be stacked on the screen. DEVisé links are often

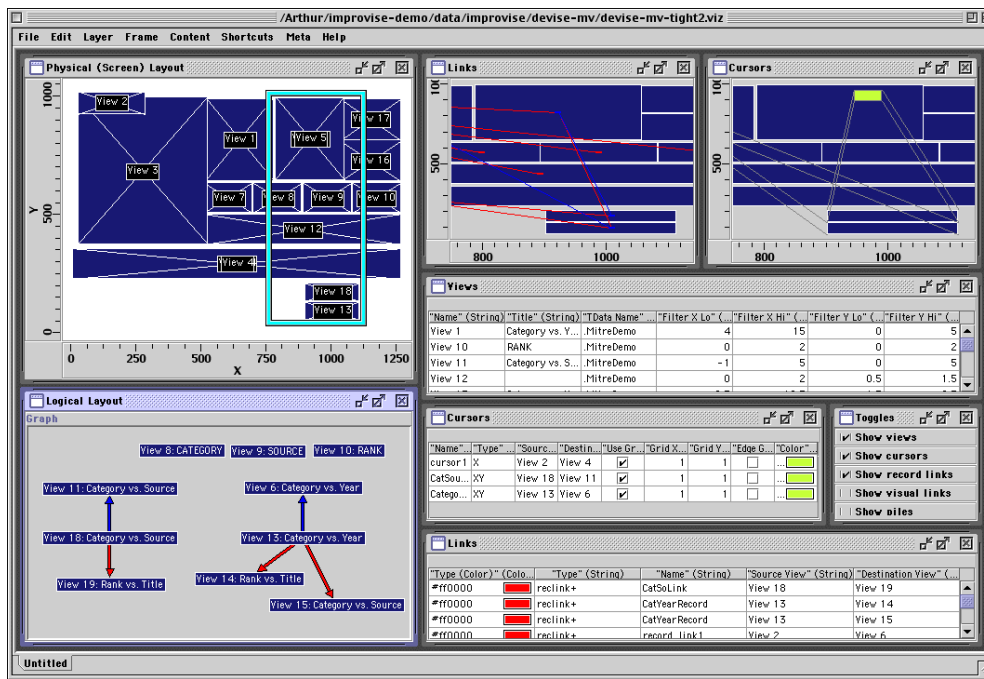


Figure 1. Metavisualization of a DEVise visualization. Scatterplots show screen layout. A graph shows linking structure. Tables describe views and links. Checkboxes toggle filtering of views/links.

hard to deduce from visualization appearance alone. Cyclic chains of links are possible, making this task even harder.

We built an Improvise metavisualization to explore linking in DEVise. The metavisualization reads three files (written by DEVise) that describe the views, links, and cursors of a visualization. In figure 1, the large scatterplot draws a scale model of the visualization. The link and cursor scatterplots draw two layers to show how views are connected. In both, the bottom layer is the same as the large scatterplot. In the links scatterplot, colored arrows connect linked views. In the cursors scatterplot, an inset box metaphor connects a view to a selection in another view. The selection box in the large scatterplot zooms the smaller scatterplots.

The graph and three tables show data filtered three different ways. The first filter calculates if views are visible in the large scatterplot. The second calculates the logical OR of boolean variables that toggle visibility of record links, visual links, and piles. The third adds an OR clause for cursors. The “Views” table uses the first filter. The “Links” table uses the second. The “Cursors” table is unfiltered. The graph uses the first filter to determine node visibility, and the third filter to determine link visibility.

The user can toggle the checkboxes to explore relationships between links and views in the graph, while seeing detailed information in the tables. By dragging and stretching the selection box, the user can see how linking structure corresponds with screen location in the visualization. Adding

the visibility toggles, editing the filters, and laying out the checkboxes took only a few minutes. The visualization being metavisualized can be switched rapidly by modifying the metavisualization’s data variables.

4. Conclusions

Metavisualization can be achieved using existing dynamic query visualization techniques. We believe metavisualization can help users at various skill levels to analyze, design, and use dynamic query visualizations.

References

- [1] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic queries for information exploration: An implementation and evaluation. In *Proceedings of CHI '92: Human Factors in Computing Systems*, pages 619–626, Monterey, CA, May 1992. ACM.
- [2] M. Livny, R. Ramakrishnan, K. Beyer, G. Chen, D. Donjerkovic, S. Lawande, J. Myllymaki, and K. Wenger. DEVise: Integrated querying and visualization of large datasets. In *Proceedings of SIGMOD '97*, pages 301–312, Tucson, AZ, 1997. ACM.
- [3] C. North and B. Shneiderman. Snap-together visualization: A user interface for coordinating visualizations via relational schemata. In *Proceedings of Advanced Visual Interfaces 2000*, May 2000.

Interactive Poster: Variable Density Scroll Areas for Rapid Image Retrieval

Louis Weitzman

Internet Technology Group, IBM
1 Rogers Street
Cambridge, Ma 02142
louisw@us.ibm.com

Jessica Wu

Internet Technology Group, IBM
150 Kettletown Road
Southbury, Ct 06488
jessicaw@us.ibm.com

1. Abstract

This poster describes an interaction technique called *variable density scroll area* (VDSA) to assist in the browsing of large datasets. This technique actively supports the user in the task of browsing by allowing the user to scroll through a dataset with the ability to change the resolution of the scrolling action. It employs two major principles. First, we provide a two-dimensional scroll area in which scrolling occurs. The complete dataset is represented as a column displayed on the left edge of this area. By providing this context, the user is always able to determine the location of the current results within the entire dataset. The remainder of the scrolling area is used to refine the resolution of the scrolling action. The second principle involves the use of *rapid serial visual presentation* (RSVP) for displaying the chosen data in a fixed location. In the interface, the user can focus on this fixed location to see the data rapidly presented as they scroll through the large set of possibilities.

2. Introduction

In many applications, datasets grow to be very large and difficult to access using the standard interface elements, such as a scroll bar. Searching provides one way to restrict the size of a dataset, but is only applicable when the user has some notion of what they are looking for and the data has been properly meta-tagged. QBIC, Query-By-Image-Content, is another technique to restrict the size of a visual dataset by constructing visual queries into the image data itself. But this technique is not suitable for all applications. New techniques are needed to enable users to browse large collections of information. While interacting with these large datasets, users can easily lose the context of the source of the information. They “get lost” in the data. Especially when the data is visual, one would like the ability to “scroll” through that data, and quickly view the elements. Current techniques are not well suited for large datasets. The new technique described here addresses this problem, allowing the user to browse over a large set of information while maintaining the user’s context within the collection.

There are a number of interaction techniques that helped to motivate the design of this prototype. These techniques include: 1) presenting all the data, all the time, 2) animating between states to maintain context and ease the transitions, and 3) displaying sequential data using rapid serial visual presentation (RSVP).

Presenting all of the data, all of the time, is easy if the dataset is small. Once the dataset grows in size, issues of layout become critical. Additional visual techniques such as modification of size, transparency, or representation need to be used to ensure legibility. In combination with user interaction, the data that is of interest can then be revealed.

Animation is a way to allow users to maintain focus while experiencing state changes within the interface. It has been used effectively in many applications and has even made its way into today’s operating systems. These techniques were successfully demonstrated in two applications, *Elastic Catalog* and *ActiveNews*. RSVP has been shown to be an effective technique to rapidly view data in a serial fashion. RSVP was successfully demonstrated in *Temporal Typography* to present information and tell stories on interactive displays.

3. Variable Density Scroll Area

Variable density scrolling uses the three techniques described above to allow users to quickly browse through large collections of visual data. It displays a representation of the complete set of data as a graduated column on the left edge of the scrolling area. The middle area is used to refine the scrolling action and the resulting data from the scrolling action is presented on the right side in a fixed location.

As the user moves up and down the left column, the corresponding image is displayed to the right utilizing the RSVP technique. This approach raises a problem when the density of the data exceeds the resolution of the display. If the size of the dataset is more than the resolution of the left column, the column can not provide access to the whole collection. Variable density scrolling uses the real estate between the left column and the displayed data to further refine the user’s selection. There are two variations of this technique, a continuous and a discrete method, each which use the additional scrolling area differently.

Continuous Variable Density Scrolling

In the first technique, the scroll area is a continuous area. It shows how the user can change the scroll resolution on a continuous scale. In this example, Figure 1, the graduated column on the left represents the complete dataset. The user can move through this representation of the data but may miss data if the density of the data exceeds that of the display. As the mouse moves to the right, the user fine-tunes the scrolling action. Moving left scrolls at a lower resolution (moving through more of the data) while moving right scrolls at a higher resolution (moving through less data). In addition, the amount of data that scrolls is proportional to its vertical movement. On the right edge of this scroll area, a discrete subset of possibilities is displayed. The item selected is highlighted and displayed to the right of the scroll area. The user can quickly jump to different areas of the data (on the left edge) and focus in on the images by moving to the right to better refine the scrolling increment.

The algorithm for determining scroll area behavior may vary but here is a simple example. When the cursor is dragged within the light gray highlighted cone area, no scrolling occurs allowing the user to select one of the items presented in the cone. When the cursor is dragged in the dark gray area, outside the cone, the focus on the left edge moves towards the cursor at a speed proportional to the Y distance away from the current focus (farther away the faster the movement).

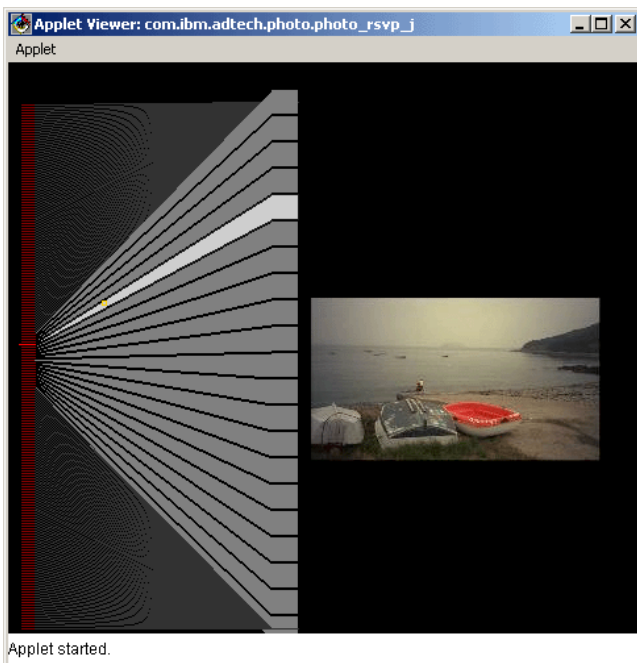


Figure 1. Continuous variable density scroll area. The graduated column on the left represents the complete dataset. The chosen image is displayed on the right in a fixed location.

In addition, the movement is inversely proportional to the x distance from the left edge (faster to the left and slower to the right). These two dimensions allow fine tuning of the scrolling rate based on the density of the information. A click in the dark gray area animates the focus on the left edge to that absolute position vertically.

Discrete Variable Density Scrolling

The second technique uses a discrete set of vertical scrolling columns. These scrolling columns work in concert to fine-tune the access to the data through interaction. A discrete variable density scroll area is similar to the continuous implementation but breaks up the scroll area into a discrete set of scroll columns. The data can be separated into any number of discrete scroll columns. The example illustrated in Figure 2 uses a set of four vertical scroll columns to provide access to the data. Similar to the continuous implementation, the left scroll column represents the complete set of data, thus, providing a fixed context for the complete dataset. The scroll columns to the right provide finer grain of resolution as the user moves from left to right. The red highlighted elements in any scrolling column show the portion of the data that is represented in the column to its immediate right. As before, the image is displayed to the right of the variable density scroll area in a fixed location.

4. Acknowledgments

We would like to thank Phillip Tiongson who helped in the design and implementation of this interaction technique and for the Internet Technology Group that has fostered an environment in which to explore new ideas.

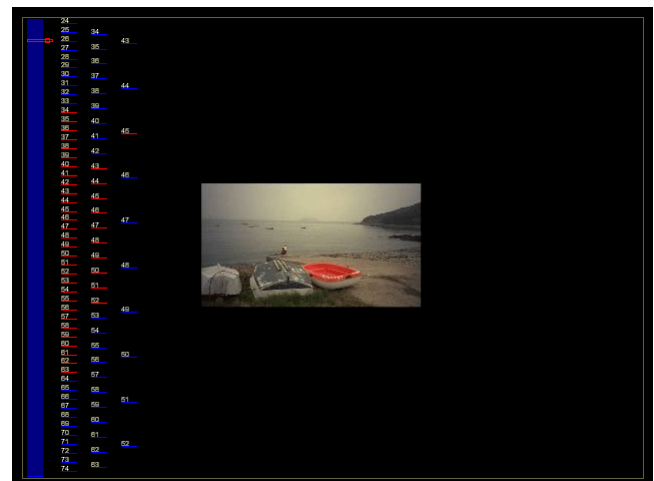


Figure 2. Discrete variable density scroll area. Multiple columns provide finer grain of scrolling control.

Web Site Visualization

Using a Hierarchical Rectangle Packing Technique

Yumi Yamaguchi Takayuki Itoh Yuko Ikehata Yasumasa Kajinaga

IBM Research, Tokyo Research Laboratory

1623-14 Shimotsuruma, Yamato-shi, Kanagawa 242-8502 JAPAN

{yyumi, itot, ikehata, kajinaga}@jp.ibm.com

1. INTRODUCTION

Visualization of Web sites is interesting not only for the navigation for end-users, but also for their monitoring and analysis. Many of existing Web site analysis software provides simple representations, such as bar charts, pie charts, and ranking tables, to provide access statistics of the Web sites. It is often difficult to quickly find subconscious but interesting access trends by only watching such simple representations.

This poster introduces our hierarchical data visualization algorithm [1,2] and presents the application of the algorithm for the Web site visualization. Our visualization algorithm represents hierarchical data as a group of nested rectangles. The algorithm places thousands of rectangles in nearly minimized display spaces in several seconds. Our application imports access log files of Web servers, and represents Web sitemaps by constructing hierarchical data according to the directory structures of Web pages, and access statistics by mapping them onto the sitemaps. This poster also shows an example of interesting access trend discovered by using our application.

2. PROPOSED HIERARCHICAL DATA VISUALIZATION

This section briefly describes our hierarchical data visualization algorithm. Figure 2 shows an illustration of the order of data layout in our technique. Our algorithm first packs icons (painted square dots in Figure 2) that denote leaf nodes, and then generates rectangles that enclose the icons to denote non-leaf nodes. Similarly, it packs a set of rectangles that belong to higher levels, and generates the larger rectangles that enclose them. Repeating the process from the lowest level toward the highest level, the algorithm places all of the data onto the layout area.

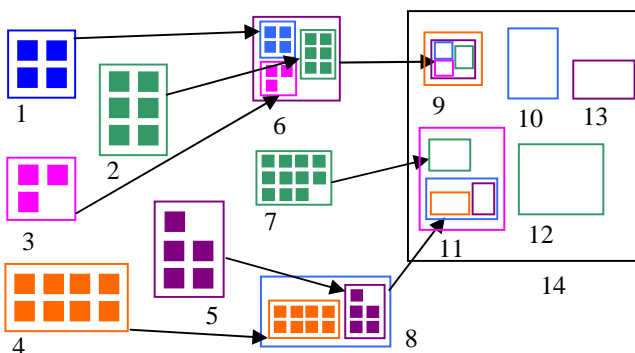


Figure 2. Algorithmic overview of the layout process for hierarchical data. Numbers in this figure denote the order of the process.

Our algorithm places a set of rectangle one-by-one, while it satisfies the following two conditions:

[Condition 1] Rectangles must be placed without overlapping each other.

[Condition 2] Rectangles should be placed where the layout area occupied by the rectangles is minimized.

To promptly pack rectangles, the algorithm sorts the rectangles according to their areas, and places them on a display space in the sorted order, as shown in Figure 3. Here, our algorithm favors accelerating the rectangle packing process rather than entirely minimizing the layout space. We therefore did not apply optimization schemes to find the configuration of rectangles, but used a heuristic to quickly find gaps and place the remaining rectangles in the gaps. The heuristic uses a triangular mesh connecting centers of previously placed rectangles. Our algorithm refers triangular elements in the order of their sizes to quickly find gaps to promptly place rectangles. References [1,2] describe the detailed algorithm.

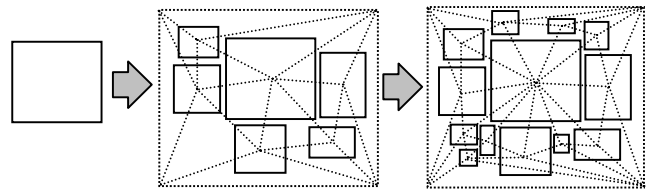


Figure 3. Illustration of rectangles packing algorithm. The algorithm first places the largest rectangle, and then places the others in the order of their areas. A triangular mesh that connects centers of rectangles are referred to quickly find gaps to promptly place the rectangles.

3. EXPERIMENTS WITH WEB SITE VISUALIZATION

This section describes our Web site visualization tool and an example of interesting access trend discovered by using the tool. Our Web site visualization tool imports access log files of Web servers. Our tool first extracts lines whose resources denote pages (HTML, CGI, etc.) and eliminates the others from the input access log file. It then collects the URLs of the resources from these lines, and builds the hierarchical Web site data according to the directory structure of the URLs. It then calculates the positions of the resources by using the visualization algorithm [1,2], and displays the data as a sitemap. Simultaneously, it groups the extracted lines according to user-specified attributes and represents the access statistics by a bar chart. Figure 4 shows an overview of the described procedure.

The tool supports the following two operations for the exploration of access distributions:

[Operation 1] When a user clicks one of the groups on the bar chart, the tool counts the number of accesses in the group for each resource, and maps the counts to heights of the icons of the sitemap.

[Operation 2] When a user clicks one of the icons on the sitemap, the tool groups the accesses for the clicked resource according to user-specified attribute, and represents the access statistics for the resource by another bar chart.

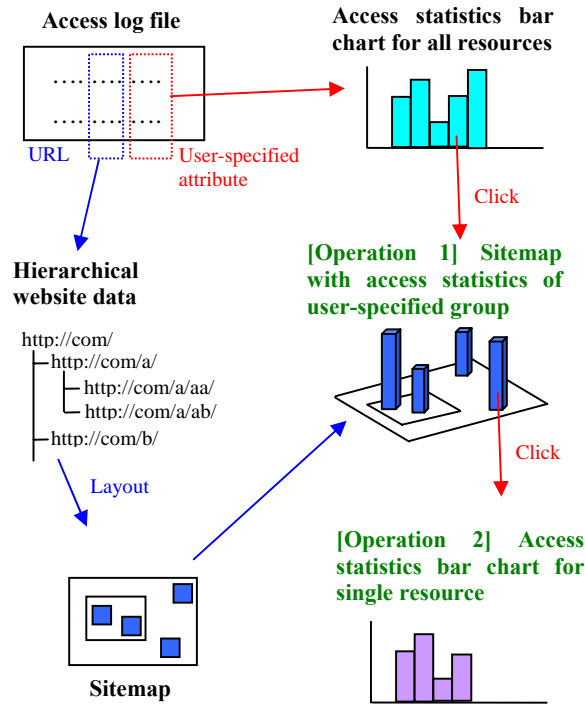


Figure 4. Overview of our website visualization tool.

In our experiment we combined the access log files of a research laboratory for one week, and visualized it using our tool. Figure 5 represents two kinds of the statistics of accesses as bar charts. The heights of the bars denote the number of accesses, and each bar is divided into 24 parts that denote each hour of the day. Figure 6 represents the sitemap generated by our hierarchical data visualization algorithm.

Figure 5(Left) shows that the number of accesses increased a great deal on the 7th day. When a user clicks one of the parts in the bar for that day as [Operation 1], the tool maps the access distribution in the clicked hour onto the sitemap. Figure 6 denotes the number of accesses in that hour for each Web page. Red icons denote the Web pages that had accesses in the hour, and their heights denote the number of accesses. Here we found two access characteristics in the figure. The upper green circle in the figure shows that most of Web pages in a particular API references directory were accessed during the hour. It is not easy to see such access patterns from traditional access ranking tools, but our system visually shows such characteristics. The lower green circle

shows that a particular Web page was accessed very frequently in this hour. It was not the top page of the Web site, but just a Web page that describes a research project. By applying [Operation 2] and grouping accesses for the page according to their referrers, we found that the page was mentioned by a business newspaper Web site that day, and therefore many users accessed the URL directly. It was also interesting that we found that the Web page was active only in the daytime on the 7th day.

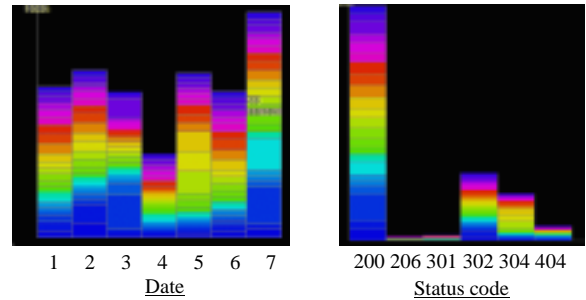


Figure 5. Examples of statistics of Web accesses represented as bar charts. (Left) 7 bars denote dates, and 24 colors denote hours. (Right) 6 bars denote status codes, and 24 colors denote hours.



Figure 6. (Upper) A sitemap with access distribution in one hour on the 7th day in Figure 5(Left). (Lower) A bar chart that represents the access statistics for the clicked resource grouped by their referrers.

REFERENCES

[1] Itoh T., Kajinaga Y., Ikehata Y., and Yamaguchi Y., Data Jewelry-Box: A Graphics Showcase for Large-Scale Hierarchical Data Visualization, IBM Research, TRL Research Report, RT0427.
 [2] Itoh T., Yamaguchi Y., Ikehata Y., and Kajinaga Y., Hierarchical Data Visualization Using a Fast Rectangle Packing Algorithm, IEEE Transactions on Visualization and Computer Graphics, in process.

eTree: A Browse and Query Interface for Online Communities

Jun Zhang
School of Information
University of Michigan
junzh@umich.edu

Alison Lee
IBM T.J. Watson Research Center
alisonl@us.ibm.com

Abstract

We use a life-like tree ecosystem metaphor to represent the people, activities and artifacts created by members of an online community. The resulting system, eTree, provides a social visualization and query interface to help end users browse, navigate, participate and learn about the participants in an online community. eTree was implemented for an online community. It provides a browsing tool to various information located on the community's online space and a tool that supported real-time social interaction. The system provides easy access to social and content information that can help to foster online social interaction and enable users to use the information to develop their social network in the online community.

1. Introduction

As new online spaces for information sharing and social interaction abound, Web-based online communities have become widely popular. The resulting large number of participants and messages makes it increasingly difficult for newcomers to comprehend the interaction context of the community, to track user participation and the social connections within the community.

Visualization technology is one solution because of its power to render large volumes of information in a limited space for exploration and comprehensive. By analyzing the key information overload problems of online communities and identifying the nature of online social interaction, we designed a life-like tree visualization interface to help users to browse, navigate, participate and learn about their online communities.

2. Problems and Related Work

Information overload is a key problem most users have in online spaces like discussion forums and newsgroups. This makes it very time-consuming for individual users to browse the large numbers of new posts to find the interesting topics. Many spaces provide functions like sorting threads by time or putting hot topics on top. However, users still need to switch their focus among different pages to seek the interesting topics.

The forums-threads-posts structure grows organically as a result of participant contributions. This results in a highly dynamic structure where the breadth, depth and

density of new activities and the degree and segment of involvement by the community are not easy to determine. Representing this growth and enabling community members to browse and query the evolving and emergent character of the community allows participants to learn about the community's members, goals, and activities.

Traditional Web-based communities focus mainly on sharing information and their interfaces are usually designed for accessing the content of the discussions rather than social information. However, an online community is distinguished by the people, their activities, and the artifacts (posts) they create. Integrating the social and content information and providing users with efficient access to such information is invaluable for fostering online social interactions [2].

These problems and their solutions are not individually new to the information visualization domain. For instance, context-plus-focus is widely used to help people navigate information in a large information space without losing the overall context. By presenting information visually and allowing dynamic user control through direct manipulation, visual information systems facilitate navigation of large information spaces and comprehension [1]. In social computing, PeopleGarden uses a botanical metaphor to provide static portraits of users' participation in online communities [5]. ConversationMap uses node-link graphs to display discussion threads and peoples' interaction network [4]. Our key contribution is the integration of these techniques to address a problem in the social computing domain. Specifically, eTree uses a visual ecosystem metaphor to provide a browse and query tool that enriches a community space with mechanisms for awareness, communication, interaction and the evolution and development of common ground [2].

3. eTree Design

eTree uses an ecosystem as a visual metaphor for mapping discussions messages into a tree with animals around. Online communities share many characteristics with such ecosystems. That is, an ecosystem evolves under the influence of its participants. Different animals congregate around different parts of the tree and form different botanical networks much like the way people take on different community roles, participate in varying degrees, and form different interaction networks.

Based on this mimesis, each community and the structure of their discussion forums are mapped onto a tree trunk and its branches, respectively tree (see Figure 1). Threads in a forum are mapped as leaves with the hot topics highlighted in yellow like flowers on the New posts in a thread appear as a lighter green. The links between the dots indicate the social sub-groups that exist based on daily co-posting interactions. Active participants (posters) are mapped as bees and placed in different locations on the perimeter of a ring centered around the tree. A poster's position in the circle represents the logarithm of the number of his posts. Non-active participants (lurkers) are mapped as ants and placed at the base of the tree. Links among the animals reveal the emergent online social interaction network aggregated from co-posting activities.

eTree is a dynamic visualization. New posts are added as new leaves in lighter green. Online users and their positions are highlighted as they move about the community space. This awareness function enables users to be aware of co-present visitors and to start synchronized communication with them. Newcomers can learn about the evolution of the discussions from an animation of the tree growth.

eTree uses a focus-plus-context mechanism that enables users to identify new or hot posts and to learn about the activities and interaction network of community members. Clicking on the visual elements results in details that are displayed as pages in a Web browser. eTree also provides different dynamic query filters that tightly couple queries with presentation to give users the flexibility to visually query community information [1]. For instance, users can easily click on particular people or leaves to query particular discussion threads or people which are highlighted in a different color.

4. eTree for Online Community

We implemented the eTree browser as a Java applet for Portkey — a summer intern Web site [2]. We rendered the structure of Portkey discussion forums as a 2D life-like tree and used different colored dots to represent the posters of the two groups of users (researchers and co-ops). Lurkers are presently not displayed but we plan to put them at the foot of the tree. Coupling the people and posts to Portkey Web site results in a visual and interactive query interface to the site (see Figure 1). Right-clicking on a visual element brings up a popup menu to query, for example, the co-posters for a selected thread.

Figure 1 shows that there are 21 forums on Portkey and the popularity of certain forums and threads. Hovering on the shorter branch reveals the less popular forum topic. We see that most of the posters are the co-ops (lighter grey dots) with only a handful from researchers. Limitations with displaying the large numbers of threads are overcome in part using a magnifying lens.

Techniques from [4] to represent large volumes of data like in 3D trees can be also applied



Figure 1: eTree in Portkey.

5. Conclusion

We presented a number of limitations with existing online community spaces that impacts participation in online communities. We use a botanical ecosystem as a visual metaphor for displaying social and content information for an online community space. The system, eTree, was implemented for a Web-based community site for summer interns and researchers to enable them to learn and interact on the site.

6. References

- [1] C. Ahlberg and B. Shneiderman, "Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays", *Conference Proceeding on Human Factors in Computing Systems*, ACM, New York, 1994, pp. 450-456.
- [2] A. Girgensohn and A. Lee, "Making Web Sites be Places for Social Interaction", To appear in *Proceeding of the ACM 2002 Conference on Computer Supported Cooperative Work*, ACM, New York, 2002.
- [3] E. Kleiberg, H. van de Wetering, and J. J. van Wijk, "Botanical Visualization of Huge Hierarchies", *Proceeding of InfoVis 2001*, IEEE, New York, 2001, pp. 87-94.
- [4] W. Sack, "Conversation Map: A Content-based Usenet Newsgroup Browser", *Proceeding of the 5th International Conference on Intelligent User Interfaces*, ACM, New York, 2000, pp. 233-240.
- [5] R. Xiong and J. Donath, "PeopleGarden: Creating Data Portraits for Users", *Proceeding of the 12th Annual ACM Symposium on User Interface Software and Technology*, ACM, New York, 1999, pp. 37-44.



POSTERS COMPENDIUM

Preface for the Vis2002 Posters Compendium

To foster greater interaction among attendees and to provide a forum for discussing exciting ongoing visualization research, we have added a poster session to this year's main conference technical program. Such a poster program offers a unique opportunity to showcase work-in-progress, student projects, or non-traditional visualization research. Especially, we have scheduled the session during the conference reception. This casual setting will allow the presenters to have one-on-one dialogue with attendees and also to better control the pace and level of the presentations.

We have put together a program consists of 20 posters selected from a total of 61 submissions. These posters focus on work that has produced new or exciting ideas or findings; some are case studies in all areas of science, engineering, and medicine. We would like to express our sincere thanks to those people who have submitted abstracts, and those who have assisted our selection process.

Vis2002 Poster Co-Chairs:

Georges-Pierre Bonneau, GRAVIR/IMAG/INRIA
Rob Erbacher, University at Albany - SUNY
Kwan-Liu Ma, University of California at Davis

Vis2002 Posters Program

Interactive Poster: Approximate Visual Hulls as Bounding Volumes for Occlusion Culling	68
Marc Alexa (Darmstadt University of Technology)	
Dirk Bartz (University of Tübingen)	
James T. Klosowski (IBM T.J. Watson Research Center)	
alex@gris.informatik.tu-darmstadt.de	
Interactive Poster: TIV: A Thread Interaction Viewer	70
Kevin F. Audleman (Brown University)	
David H. Laidlaw (Brown University)	
Steven P. Reiss (Brown University)	
kforbes@cs.brown.edu	
Interactive Poster: Applying Geometry and Color Correction to Tiled Display Walls	72
Justin Binns (Argonne National Laboratory)	
Gennette Gill (Argonne National Laboratory)	
Mark Hereld (Argonne National Laboratory)	
David Jones (Argonne National Laboratory)	
Ivan Judson (Argonne National Laboratory)	
Ti Leggett (Argonne National Laboratory)	
Aditi Majumder (Argonne National Laboratory)	
Matthew McCrory (Argonne National Laboratory)	
Michael E. Papka (Argonne National Laboratory)	
Rick Stevens (Argonne National Laboratory)	
binns@mcs.anl.gov, papka@mcs.anl.gov	
Interactive Poster: Smooth Curves on Polygonal Meshes	74
Georges-Pierre Bonneau (INRIA)	
Stefanie Hahmann (Laboratoire LMC-IMAG)	
Georges-Pierre.Bonneau@imag.fr	
Interactive Poster: Realtime Uncertainty Visualization of Mobile Objects in VGIS	76
Amin P. Charaniya (University of California at Santa Cruz)	
Srikumar Ramalingam (University of California at Santa Cruz)	
Suresh Lodha (University of California at Santa Cruz)	
William Ribarsky (Georgia Institute of Technology)	
Nick Faust (Georgia Institute of Technology)	
Zachary Wartell (Georgia Institute of Technology)	
Tony Wasilewski (Georgia Institute of Technology)	
amin@cse.ucsc.edu, lodha@cse.ucsc.edu	
Interactive Poster: Seat Belt Routings over Virtual Humans - The Belt Design Service	78
Andreas Divivier (tecmath AG)	
Achim Ebert (DFKI GmbH)	
Michael Bender (University of Applied Sciences)	
Hans Hagen (DFKI GmbH)	
divivier@hs.tecmath.com, bender@informatik.fh-kl.de	
Interactive Poster: A 4D Deformable Model to Visualize Time-Varying Volumetric Datasets	80
Johnny De Freitas-Caires (Universite de Reims)	
Aassif Benassarou (Universite de Reims)	

Eric Bittar (Universite de Reims)
Laurent Lucas (Universite de Reims)
laurent.lucas@univ-reims.fr

Interactive Poster: A Multi-Attribute Visualization for Analyzing Large Web Transactions	82
Ming C. Hao (Hewlett Packard Research Laboratories)	
Daniel Cotting (Hewlett Packard Research Laboratories)	
Thomas Holenstein (Hewlett Packard Research Laboratories)	
Vijay Machiraju (Hewlett Packard Research Laboratories)	
Pankaj Garg (Hewlett Packard Research Laboratories)	
ming_hao@hp.com	
Interactive Poster: Visualization of Vector Fields Using LIC and Volume Rendering	84
Anders Helgeland (Norwegian Defence Research Establishment)	
Oyvind Andreassen (Norwegian Defence Research Establishment)	
andershe@ifi.uio.no	
Interactive Poster: geeViz (Grid-Enabled Environment Visualization): Realtime Visualization of Distributed Systems	86
Joseph A. Insley (Argonne National Laboratory)	
Michael E. Papka (Argonne National Laboratory)	
insley@mcs.anl.gov	
Interactive Poster: The Great Potato Search: The Effects of Visual Context on Users' Feature Search and Recognition Abilities in an IVR Scene	88
Cullen D. Jackson (Brown University)	
David B. Karelitz (Brown University)	
Sean A. Cannella (Brown University)	
David H. Laidlaw (Brown University)	
cj@cs.brown.edu	
Interactive Poster: VITA: Visual Interface for Text Analysis	90
Zachary Jacobson (Health Canada)	
jzj@rogers.com	
Interactive Poster: Remote Exploration of Isosurfaces with Point-Based Non-PhotoRealistic Rendering	92
Guanfeng Ji (The Ohio State University)	
Han-Wei Shen (The Ohio State University)	
Jinzhu Gao (The Ohio State University)	
jig@cis.ohio-state.edu	
Interactive Poster: Showing Shape with Texture	94
Sunghee Kim (University of Minnesota)	
Haleh Hagh-Shenas (University of Minnesota)	
Victoria Interrante (University of Minnesota)	
skim@cs.umn.edu, haleh@cs.umn.edu	
Interactive Poster: comp-i: 3D Visualization of MIDI Dataset	96
Reiko Miyazaki (Ochanomizu University)	
Issei Fujishiro (Ochanomizu University)	
reiko@imv.is.ocha.ac.jp	
Interactive Poster: A Powerful Communication Framework for Distributed Virtual Reality	98

Bernd Reuther (University of Kaiserslautern)
Achim Ebert (DFKI GmbH)
Paul Muller (University of Kaiserslautern)
Hans Hagen (DFKI GmbH)
reuther@informatik.uni-kl.de

Interactive Poster: A Software Framework for Medical Visualization 100
Bartosz von Rymon-Lipinski (Research Center Caesar)
Thomas Jansen (Research Center Caesar)
Nils Hanssen (Research Center Caesar)
Marc Lievin (Research Center Caesar)
Erwin Keeve (Research Center Caesar)
lipinski@caesar.de

Interactive Poster: Effects of Relative Particle Sizing and Bin Consolidation
Enhancements to 5D Volume Visualization Algorithm 102
Rhonda J. Vickery (Mississippi State University)
Timothy R. Keen (Naval Research Laboratory)
Robert J. Moorhead (Mississippi State University)
Joerg Meyer (University of California, Irvine)
Randy J. Brou (Mississippi State University)
Ashley M. Noble (Mississippi State University)
Joel P. Martin (Mississippi State University)
Stephanie M. Doane (Mississippi State University)
rvickery@erc.msstate.edu

Interactive Poster: Application of DTI Visualization and Analysis on a Data Set
from a Brain Tumor Patient 104
Song Zhang (Brown University)
David H. Laidlaw (Brown University)
Mark E. Bastin (University of Edinburgh)
Saurabh Sinha (University of Edinburgh)
Thomas S. Deisboeck (MIT and Harvard Medical School)
sz@cs.brown.edu

Interactive Poster: Virtual Preoperative Laparoscopic Camera: Visualization and
Modeling System 106
Leonid Zhukov (California Institute of Technology)
Igor Guskov (University of Michigan)
Sergey Klibanov (University of Michigan)
Dmitry Oleynikov (University of Nebraska Medical Center)
zhukov@gg.caltech.edu

Interactive Poster: Approximate Visual Hulls as Bounding Volumes for Occlusion Culling

Marc Alexa
Darmstadt University of Technology
alexam@gris.informatik.tu-darmstadt.de

Dirk Bartz
WSI/GRIS, University of Tübingen
bartz@gris.uni-tuebingen.de

James T. Klosowski
IBM T.J. Watson Research Center
jklosow@us.ibm.com

Abstract

Bounding volumes are widely used in computer graphics to accelerate geometric computations such as collision detection and visibility culling. The most commonly used bounding volumes have always been convex in nature. Here, we propose the use of approximate visual hulls as bounding volumes for the task of visibility culling. These hulls are produced by simplifying the underlying geometry being approximated, and have the advantage in that they can be concave or even contain holes. The visual hulls can therefore approximate an object more closely using fewer bounding volumes.

1 Introduction

Geometric models are continuing to grow at an astonishing rate. This explosion in model size is the result of advanced data acquisition techniques as well as the tremendous power of commodity graphics hardware from vendors such as nVidia and ATI. To accommodate such large models, graphics and visualization algorithms utilize hierarchies of bounding volumes to accelerate geometric computations. Bounding volumes provide approximations of the underlying geometry that are less complex and less expensive to operate on than the geometry itself.

Selecting the most appropriate bounding volume for a given problem is challenging due to the tradeoffs involved. Typically, as a bounding volume increases in complexity, the resulting approximation also improves, but so does the cost of the computations on that volume. The most popular bounding volumes have traditionally been the bounding sphere [16, 11, 7] and the axis-aligned bounding box (AABB) [4, 6, 15, 2], due to their simplistic nature. This simplicity comes at the expense of relatively poor approximations to the original geometry, which can result in unnecessary computations being done. Another popular volume is the oriented bounding box (OBB) [12, 5], whose spanning axes are oriented according to the actual geometry. OBBs will typically provide better approximations than AABBs and bounding spheres, but they are also more complex to construct, store, and use.

Klosowski et al. [8] used discrete orientation polytopes (k -dops) as bounding volumes within their collision detection system for more efficient collision tests. Bartz et al. used these k -dops as bounding volumes for occlusion culling and compared them to other convex bounding volumes, such as AABBs, OBBs, bounding spheres, and even convex hulls [1].

All of these bounding volumes share the characteristic of being convex, which restricts their ability to accurately represent non-convex objects. However, image-based occlusion culling does not rely on the convexity of the bounding volumes; it only requires a bounding volume which completely encloses the actual object geometry. Such non-convex bounding volumes could be based on a simplified version of the original geometry, such as simplification envelopes [3] or progressive hulls [13]. Here, we use conservative approximations of the geometric visual hull of the given shape. These hulls are constructed to minimize the volumetric difference

between the visual hull and the approximation with a given geometry primitive budget.

In the following, we introduce the construction of the approximate visual hulls (AVH) in Section 2 which are used as geometric approximations for occlusion culling, as described in Section 3.

2 Approximate Visual Hulls

Let the object to be approximated be described by a piecewise linear mesh \mathcal{M} . The intersection of all generalized cones defined by the silhouettes of \mathcal{M} when projected onto a plane is its geometric visual hull $\mathcal{V} \supseteq \mathcal{M}$ [9]. The visual hull effectively represents the properties of \mathcal{M} as an occluder.

Our goal is to find an approximation of \mathcal{V} given a number of geometric primitives, which ideally minimizes the integral over the area differences to the set of real silhouettes. However, to avoid difficulties arising from perspective transformations, we instead try to minimize the volumetric difference.

Matusik et al. [10] show how to quickly compute a polygonal approximation from a fixed sets of discrete silhouettes. In our work, however, we are interested in a strictly conservative approximation for the visual hull arising from all possible silhouettes. Our approach is, thus, more inspired by the progressive hulls of Sander et al. [13], adapted to the goals described above. Specifically, we perform the following steps to construct an approximation of \mathcal{V} :

1. **Compute initial representation of \mathcal{V} : \mathcal{V}_0 .** The rationale to compute the visual hull is to remove strictly concave regions as they cannot contribute to any silhouette. Specifically, all strictly concave vertices are removed iteratively. Concavity is determined based on the angular excess and the face normals.
2. **Generate a sequence $\mathcal{V}_1 \subset \mathcal{V}_2 \dots$ of coarse approximations.** Edges are removed from \mathcal{V}_i so that $\mathcal{V}_{i+1} \supset \mathcal{V}_i$ in a geometric sense, i.e. \mathcal{V}_{i+1} encloses \mathcal{V}_i . Edges chosen to minimize the volumetric difference of the meshes. This is a small linear programming problem however, which makes the procedure an offline process. Edges are removed until the number of primitives matches a predefined bound.
3. **Optimize the vertex positions in \mathcal{V}_n .** Vertices are visited in turn and optimized to minimize the volumetric difference to the *best* approximation of the visual hull (i.e. \mathcal{V}_0) while enclosing it. This problem is made feasible by using the faces in \mathcal{V}_n as linear constraints for the vertices in \mathcal{V}_0 . Note that in contrast to [13], we do not require that approximations with a different number of primitives be nested.

Figure 1 shows the v-belt and an elbow of an engine exhaust system together with their AVHs consisting of 100 triangles each. In general, AVHs may be concave or have holes, which allows them to be tighter than convex bounding volumes while using roughly the same number of primitives.

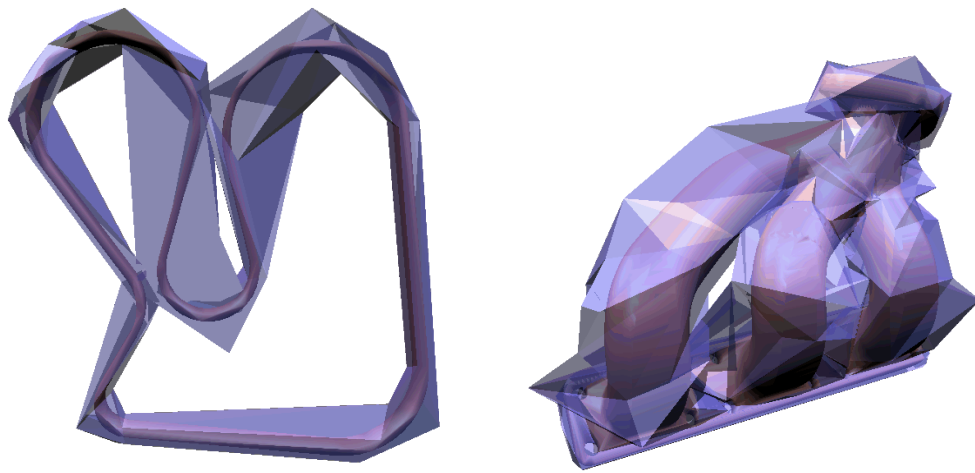


Figure 1: Approximate visual hulls of components in an engine exhaust system. The AVHs consist of 100 triangles each while the original meshes have 1K and 10K faces, respectively. Note that the hulls contain significantly less volume as compared to convex bounding volumes.

3 Culling algorithm

For each object in the scene, an AVH is computed in a preprocessing step. During interactive rendering, we perform two stages of hierarchical culling. In the first stage, we perform view frustum culling to determine which of the leaf nodes of the model hierarchy are within the user's field of view. For these nodes, we use the implicitly calculated near and far depth values to arrange the potentially visible nodes in a front-to-back list L . In the second culling stage, we process L in an interleaved fashion. After disabling the writing of the depth and color buffers, we test if rendering the bounding volume of the front-most leaf node in L would have modified the depth buffer, if it had been enabled. If no change would have resulted, the bounding volume is determined to be occluded, and its geometry does not need to be rendered. Otherwise, we enable the depth and color buffers for writing and render the associated actual geometry of the leaf node and proceed with the next node in the list. For the detection of the potential change in the depth buffer, we use the Hewlett-Packard occlusion culling flag (HP flag) [14], however, any other occlusion culling technique could be used (e.g. the virtual occlusion buffer [2]).

References

- [1] D. Bartz, J. Klosowski, and D. Staneker. Tighter Bounding Volumes for Better Occlusion Performance. In *Visual Proc. of ACM SIGGRAPH*, page 213, 2001.
- [2] D. Bartz, M. Meißner, and T. Hüttner. OpenGL-assisted Occlusion Culling of Large Polygonal Models. *Computers and Graphics - Special Issue on Visibility - Techniques and Applications*, 23(5):667–679, 1999.
- [3] J. Cohen, A. Varshney, D. Manocha, G. Turk, H. Weber, P. Agarwal, F. Brooks, and W. Wright. Simplification envelopes. In *Proc. of ACM SIGGRAPH*, pages 119–128, 1996.
- [4] J. Goldsmith and J. Salmon. Automatic Creation of Object Hierarchies for Ray Tracing. *IEEE Computer Graphics and Applications*, 7:14–20, 1987.
- [5] S. Gottschalk, M. Lin, and D. Manocha. OBBTree: A Hierarchical Structure for Rapid Interference Detection. In *Proc. of ACM SIGGRAPH*, pages 171–180, 1996.
- [6] N. Greene, M. Kass, and G. Miller. Hierarchical Z-Buffer Visibility. In *Proc. of ACM SIGGRAPH*, pages 231–238, 1993.
- [7] P. Hubbard. Approximating Polyhedra with Spheres for Time-Critical Collision Detection. *ACM Transactions on Graphics*, 15(3):179–210, 1996.
- [8] J. Klosowski, M. Held, J. Mitchell, H. Sowizral, and K. Zikan. Efficient Collision Detection Using Bounding Volume Hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):21–36, 1998.
- [9] A. Laurentini. The visual hull concept for silhouette bases image understanding. *IEEE PAMI*, 16(2):150–162, 1994.
- [10] W. Matusik, C. Buehler, R. Raskar, S. Gortler, and L. McMillan. Image-based visual hulls. In *Proc. of ACM SIGGRAPH*, pages 369–374, 2000.
- [11] I. J. Palmer and R. L. Grimsdale. Collision detection for animation using sphere-trees. *Computer Graphics Forum*, 14(2):105–116, jun 1995.
- [12] S. Rubin and T. Whitted. A 3-Dimensional Representation for Fast Rendering of Complex Scenes. In *Proc. of ACM SIGGRAPH*, pages 11–116, 1980.
- [13] P. Sander, X. Gu, S. Gortler, H. Hoppe, and J. Snyder. Silhouette clipping. In *Proc. of ACM SIGGRAPH*, pages 327–334, 2000.
- [14] N. Scott, D. Olsen, and E. Gannett. An Overview of the VISUALIZE fx Graphics Accelerator Hardware. *The Hewlett-Packard Journal*, pages 28–34, May 1998.
- [15] G. van den Bergen. Efficient Collision Detection of Complex Deformable Models using AABB Trees. *Journal of Graphics Tools*, 3(5):1–13, 1997.
- [16] H. Weghorst, G. Hooper, and D. Greenberg. Improved Computational Methods for Ray Tracing. *ACM Transactions on Graphics*, 3(1):52–69, 1984.

Interactive Poster of TIV: A Thread Interaction Viewer

Kevin Audleman

David H. Laidlaw

Steven P. Reiss

{kforbes, dhl, spr}@cs.brown.edu

Department of Computer Science, Brown University, Providence, RI

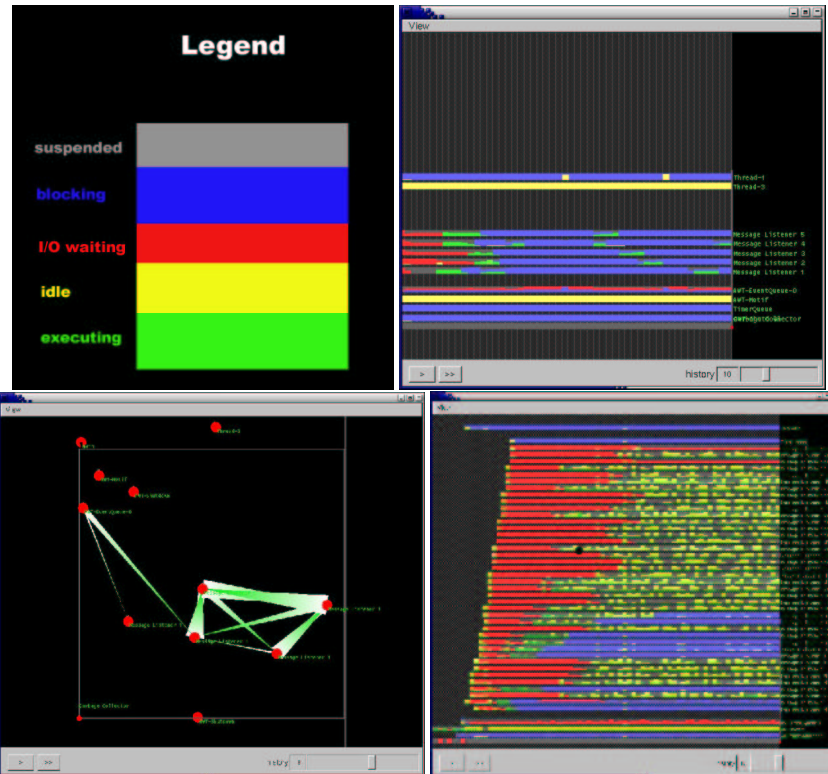


Figure 1: Screenshots of TIV. Clockwise, from top left: a) Thread states are represented as colored bars. b) The timeline view shows the states of execution of the threads, describing how they are acting. c) The interaction shows group behavior on mutual locks, describing how the threads are interacting. d) A larger view showed at the mid level of detail.

This paper presents Thread Interaction Viewer (TIV), a visualization tool for examining the thread behavior of large, long-running Java programs. It provides a novel view of thread scheduling information, I/O behavior, and blocking behavior in a manner that helps in the understanding and debugging of some thread related problems. This is facilitated by a flexible visualization framework that uses levels of detail to best reflect the relevant information for different numbers of threads. TIV uses a 3-D space to represent the threads, but relies primarily on two specific views in this space: the Timeline View and the Interaction View. These views together paint a picture of behavior: the timeline view shows how each thread is acting, and the interaction view how they're interacting. We have found TIV to be good at identifying problems that are the result of threads interacting improperly, such as erroneous synchronization behavior, errors in thread scheduling, and more complex user defined thread communication errors.

Introduction

Threaded programs have multiple threads of execution working in the same address space. They work in shared memory, communicating through the use of locking primitives. Properly coding the synchronization of these locks is necessary to prevent threads from interfering with each other, however doing so is a delicate and error prone process. Small flaws in logic can have a large impact on program performance. For instance, introducing a lock in the wrong place, such as on a heavily accessed class or object, can create so much blocking that it decreases performance. Identifying the cause of this sudden slowdown is not easy without the right tool.

TIV is designed to help the user identify thread communication based problems such as these. It uses a data visualization that uses two distinct views, the timeline view and the interaction view, to make problems of this type stand out. TIV's data representation allows for an efficient means of examining large and long running programs. We describe the details of the visualization below.

Visualization

TIV's data representation is a combination of thread scheduling states, I/O behavior, and shared blocking statistics. For each thread, TIV reports the time spent in each of the following states:

Running: Active on a processor

Runnable: Ready to run, but not currently on a processor.

I/O Waiting: Waiting on a function call to memory or a socket

Sleeping: Waiting on the wait queue of synchronization variable.

Suspended: Inactive until another threads wakes it up.

In the examples in this paper, each state is represented over a one second time window. However, this length of time is user controllable to allow for a coarse or fine examination of the data. An example will illustrate the use of TIV.

Example

Figure 1 illustrates a message server. It has five listener threads waiting on sockets to receive messages, which it then processes and displays in a GUI. However, it is running slowly. As we will demonstrate below, TIV makes it easy to identify the problem.

Figure 1b shows the **timeline view**, which is used to examine states. The five threads in the middle are the listener threads. In a proper execution, we would expect these threads to be primarily in the I/O waiting state, ready to accept incoming messages. However the timeline view shows us that they're mostly in the sleeping state. This explains the slowdown – the listeners are not available to accept incoming messages because they're spending all of their time blocking. The interaction view gives more detail.

Figure 1c shows the **interaction view**, which is used to examine blocking behavior. A connection between two threads T1 and T2 indicate that one of them is waiting on a lock the other holds. In this example, it can be seen that all of the listeners are waiting on each other. This indicates that there is a lock they are all trying to gain access to, which is unexpected. Examination of the code reveals that each thread is accessing a Java implemented `hashCode` that requires synchronous reads. Switching to a `HashMap` removes the synchronization and eliminates the slowdown.

Levels of Detail

TIV also gives a means for examining programs with a large number of threads through a flexible visualization framework. We have tested its effectiveness on programs up to 50 threads, and expect it will scale to about 200. The levels change the visualization based on how many threads are on the screen. There are three levels: **near**, **mid**, and **far**. Near is as described above, and mid and far successively compact the information displayed. This serves the purpose of keeping the visualization readable at large scales – five states for 100 threads produces an incomprehensible display, but two or three colors is much more understandable. It also serves the purpose of abstracting away detail that is not as relevant as the number of threads increase. For instance, examining the scheduling of threads onto processors (the proportion of *running* to *runnable*) is useful with a small number of threads, as it provides a means of finding problems with priorities, such as threads being starved. However with twenty or more threads, it is often useful to assume the thread library is doing a decent job. Mid level handles this issue by compressing *running* to *runnable* into *executing*. The levels are describes as:

- **Near:** This is the same as described above. It describes the states: running, runnable, I/O waiting, sleeping, and blocking. It is useful for up to 20 threads.

- **Mid:** This level shows: executing, I/O waiting, sleeping, and blocking. It is useful for up to 50 threads.
- **Far:** This level combines *blocking* and *I/O Waiting* into the single category *Non-runnable* to further reduce visual clutter. This level is designed to point out potential bad behavior that can then be examined in greater detail with a closer view. The states it describes are: executing, non-runnable, suspended. It is designed for up to 200 threads.

Figure 1d gives an example of TIV running at mid level. This is a message server with forty message listeners. The listeners initially start in the waiting on I/O state, but then are deluged by a massive number of messages. This can be seen by the large amount of time they spend in the *executing* state. The specifics of when each thread transitions from runnable to running aren't of any concern with this program, thus the mid level of detail is the right level to look at this data.

Conclusion

TIV is a visualization tool that aids in the process of debugging and understanding large threaded programs. It uses two main views, the timeline view and the interaction view, to communicate how threads are acting and interacting. It utilizes a level of detail system to focus the visualization for different numbers of threads. TIV is useful for identifying problems that occur through improper thread communications, and is effective for programs with large numbers of threads.

References

- [1] Qiang A. Zhao and John T. Stasko, Visualizing the execution of three ds-based parallel programs. Technical Report GIT-GVU-95/01, Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA, January 1995.
- [2] John T. Stasko, The PARADE Environment for Visualizing Parallel Program Executions: A Progress Report. Technical Report GIT-GVU-95-03, Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA, 1995.
- [3] B. Stein and Chassin de Kergommeaux, Interactive Visualization Environment of Multi-threaded Parallel Programs.
- [4] J. Guitart, J. Torres, E. Ayguad and J. Labarta. Java Instrumentation Suite: Accurate Analysis of Java Threaded Applications. *2nd Workshop on Java for High Performance Computing, Santa Fe, New Mexico (USA)*, pp. 15-25. May 2000.
- [5] <http://www.research.ibm.com/jinsight/docs/index.htm>
- [6] Michael Golm, Christian Wawersich, J. Baumann, M. Felser, and J. Kleinoder. Understanding the Performance of the Java Operating System JX using Visualization Techniques. *Workshop on Software Visualization, OOPSLA 2001, Tampa, FL*, October 15, 2001.
- [7] S. Hiranandani, K. Kennedy, C.-W. Tseng, and S. Warren. The D Editor: A New Interactive Parallel Programming Tool. *Proceedings of Supercomputing '94*, Washington, DC, November 1994.
- [8] Jack Dongarra, Orly Brewer, James Arthur Kohl, and Samuel Fineberg. A tool to aid in the design, implementation, and understanding of matrix algorithms for parallel processors. *Journal of Parallel and Distributed Computing*, 9(2):198-202 June 1990.

Interactive Poster: Applying Geometry and Color Correction to Tiled Display Walls

Justin Binns, Gennette Gill, Mark Hereld, David Jones, Ivan Judson, Ti Leggett,

Aditi Majumder, Matthew McCrory, Michael E. Papka, and Rick Stevens

Argonne National Laboratory

{hereld,majumder,papka,stevens}@mcs.anl.gov

Introduction

Tiled displays systems built by combining the images from arrays of projectors can provide a huge number of pixel elements to applications needing to visually represent a large amount of information, including a range of scientific visualization applications and high-fidelity collaborative environments. With commodity projectors and COTS rendering clusters, these large-area displays have the potential to provide very high performance and high resolution at low per pixel cost. But it is difficult to create the illusion of a unified seamless display for a variety of reasons. These include small misalignments of the projectors, optical distortion of the individual projector images because of imperfections in the lenses, intensity variation from projector to projector and across the field of each single projector, and color differences between projectors.

Argonne has been researching methods for addressing these limitations. In particular, we have developed two methods, one which measures and corrects the geometrical problems (alignment and distortion), and another which measures and corrects the color and luminance imperfections. These methods are briefly discussed below. Both methods produce correction values that can be applied to the projected images to create a single seamless image. For these techniques to become truly useful, however, they must be integrated into existing infrastructure so that all applications and users can benefit from them. A real time implementation of these correction methods is the subject of the rest of this paper.

Image Correction Techniques

Hereld et al. have developed a technique for quickly and accurately assessing and correcting the alignment and distortion of the tiled array using a commodity digital camera [1]. In our approach we take a few pictures of test patterns covering of the entire display. Automatic analysis of these extracts a detailed geometrical mapping to subpixel accuracy of projector pixels to the display

surface coordinates. We use the mapping to remove the distortions from the image by applying a piecewise bilinear warp conveniently handled by conventional texture mapping hardware.

Majumder et al. have solved the second piece of the calibration puzzle by introducing a camera-based method for correcting the luminance field of the entire display [2-4]. Camera images of the display projecting constant color values are analyzed to produce two quantitative measures of the system. First, for each projector of the color output is remapped as a function of intended input value, to correct the projector to projector color differences. Second, a smooth attenuation map is generated in the form of an alpha mask. This mask balances the brightness of the display, and it provides smooth blending across overlapping projected subimages.

Real-Time Geometry & Color Correction

Using the correction tables provided by the two systems described above as input, we have developed a set of functions for applications that use OpenGL. One solution is for application developers working directly with tiled displays. This library of functions for developers is currently implemented to work only with OpenGL and NVIDIA extensions [5] but could easily be extended to work with other extensions and DirectX. The second solution uses the functions just described to construct a Chromium SPU. Chromium is a system for controlling and modifying streams of OpenGL API commands [6] and is commonly used for the displaying of OpenGL applications in a tiled display environment.

To implement a solution that does not adversely affect the frame rate of the application, we use the latest in OpenGL extensions. The process is as follows:

1. The scene is rendered as it normally would be, including the use of vertex programs, texture shaders, and register combiners.
2. The framebuffer is then copied by using `glCopyTexSubImage2D` to texture memory for the color correction step.

Table 1 Results for three different applications, showing the unmodified frame rate, render to texture (RT), render to texture with color correction (RTCC), render to texture with geometry correction (RTGC), and complete system with render to texture with color correction and geometry correction turned on (RTCCGC). These tests all have the alpha mask step turned on.

Application	Unmodified	RT	RTCC	RTGC	RTCCGC
Solar System	89.3 fps	55.5 fps	48 fps	54.5 fps	45 fps
Atlantis	78.5 fps	63.2 fps	56.0 fps	61.5 fps	54 fps
Teapot	153 fps	91 fps	72 fps	89 fps	70 fps

3. Color correction is then applied by using texture shaders and register combiners, with the color correction represented as a texture map and alpha mask applied for overlapped projection systems (the alpha mask step does not need to be applied for nonoverlapped tiled displays).
4. The geometry correction is then applied by texture mapping the resulting image to a quad strip covering the display surface.
5. The final image appears on an individual tile of the tiled display.

This approach has been applied to a number of test cases; Table 1 shows the results of the various stages of the process, as well as the overall impact on the system.

From these measurements we estimate that the total cost in time to apply these corrections is between 6 – 11 milliseconds. This breaks down into a render to texture expense of approximately 3 – 7 milliseconds, a color correction and alpha mask cost of 2 – 3 milliseconds and a geometrical correction costing 0.2 – 0.5 milliseconds. Each of these are a content independent constant cost for applying these corrections.

Conclusion

This work has demonstrated that it is possible to correct both geometry and color errors in a tiled display environment without severe impact on the applications performance. This work has also demonstrated that it is possible to construct a Chromium SPU that allows users of the Chromium toolkit to benefit from color and geometry correction without having to modify existing applications. Future work will include testing the system with graphics adaptors that have similar capabilities to those of the NVIDIA adaptors, as well as extending it in

the lower level library to support DirectX in addition to OpenGL.

Acknowledgment

This work was supported in part by Department of Energy, under Contract W-31-109-Eng-38.

References

- [1] M. Hereld, I. Judson, and R. Stevens, “DottyToto: A Measurement Engine for Aligning Multi-Projector Display Systems,” Argonne National Laboratory, Argonne, ANL MCS Preprint ANL/MCS-P958-0502, May 29 2002.
- [2] A. Majumder, Z. He, H. Towles, and G. Welch, “Achieving Color Uniformity across Multi-Projector Displays,” presented at IEEE Visualization, Salt Lake, Utah, 2000.
- [3] A. Majumder, “Properties of Color Variation across Multi-Projector Displays” presented at Eurodisplay, Nice, France, 2002.
- [4] A. Majumder and R. Stevens, “LAM: Luminance Attenuation Map for Photometric Uniformity in Projection Based Displays,” presented at Virtual Reality and Software Technology, Hong Kong, China, 2002.
- [5] M. J. Kilgard, “NVIDIA OpenGL Extension Specifications,” 2002.
- [6] G. Humphreys, M. Houston, Y.-R. Ng, R. Frank, S. Ahern, P. Kirchner, and J. T. Klosowski, “Chromium: A Stream Processing Framework for Interactive Graphics on Clusters,” presented at SIGGRAPH, San Antonio, Texas, 2002.

Interactive Poster: Smooth Curves on Polygonal Meshes

Georges-Pierre Bonneau*
iMAGIS-GRAVIR, Grenoble

Stefanie Hahmann†
LMC-IMAG, Grenoble

Abstract

Curves on surfaces can be very useful to visualize surface features at low graphical and memory cost. Curves on surfaces are also used for surface segmentation with possible applications to visualization, reconstruction and parameterization of complex surfaces. In this paper an algorithm for building smooth polygonal lines on triangulated 2D-manifold polygonal meshes is introduced. The algorithm combines geometrical optimization with topological operations on PLs in order to iteratively smooth an initial crude PL interpolating a user-defined set of triangles. One key feature of this algorithm is that it relies solely on the geometry of the surface and the polygonal line. Another key feature is that during the smoothing the polygonal lines always stay on the surface.

Keywords: polygonal meshes, polygonal line, curves on meshes, visualization, curve smoothing, Dijkstra algorithm, discrete geodesic lines.

1 Introduction

Curves on surfaces have applications in many different areas. In visualization, their ability to reveal global surface features or local details at a very low graphical and memory cost has been used to help visualizing large data sets. Curves on surfaces are also important in surface segmentation. The applications of surface segmentation span a large field of research from visualization, to reconstruction and parameterization. Irregular polygonal meshes can be reparameterized on a coarse mesh by first constructing a set of smooth PL on a fine mesh that yield a segmentation of the mesh into three sided patches. The reconstruction of parametric surfaces from dense polygonal data can be first based on the interactive design of a mesh of smooth polygonal lines on the polygonal data.

In this poster we present an algorithm for building smooth polygonal lines (PL) on 2D manifold triangular meshes. An initial PL is first computed from a set of user specified mesh triangles. This is a crude (non smooth) approximation of the desired PL. Then an it-

*iMAGIS-GRAVIR, INRIA Rhône-Alpes, 655 Avenue de l'Europe, F-38330 Montbonnot (France), e-mail: Georges-Pierre.Bonneau@imag.fr, <http://www-imagis.imag.fr/~Georges-Pierre.Bonneau>

†Laboratoire LMC-IMAG, BP. 53, F-38041 Grenoble (France), e-mail:Stefanie.Hahmann@imag.fr, <http://www-lmc.imag.fr/lmc-mga/Stefanie.Hahmann>

erative execution of geometrical optimization steps and topological operations enables to smooth this initial PL on the surface. During this algorithm the PL always lies exactly on the mesh. A key feature of the algorithm is that it relies solely on the surface geometry and the PL geometry. It is independent of any surface parameterization or any external smooth curves. The algorithm has the following features:

- It is a purely geometry driven algorithm. This means that the smoothing criteria as well as the error measurements only depend on the intrinsic geometric properties of the surface or the PL.
- Between two user-specified or automatically computed points on the polygonal surface a smooth PL can be computed by using different smoothing filters.
 - We first introduce a “linear” filter which yields a local discrete geodesic. The visual effect is a global straightening of the initial PL.
 - Then several extensions are possible, we choose to implement a “cubic” smoothing filter. Here the global shape of the initial PL is preserved while small details are smoothed out.
- In any case smooth PL are obtained by iteratively moving the vertices of the PL on the mesh edges (geometrical modification) and by passing through mesh vertices if necessary (topological modification).
- The algorithm is easy to implement. However, the topological operations need to be well understood.
- The whole procedure of building curves on a polygonal surface can be both fully automatic or semi-automatic. It depends on the application. In the first case for example a mesh simplification algorithm can be used in order to produce a set of pairs of mesh points (corresponding to end points of the edges of the coarse mesh) which is given as input to the algorithm presented in this poster. In a semi-automatic environment a user can lay out a set of curves by specifying the end points of the desired PL. For each PL he/she can specify one or more (ordered) points which are interpolated by the initial crude PL. The smooth curve always interpolates the end points.

2 Results

Most of the time spent for building smooth PL is due to the user's choice of the input triangles that the PL will interpolate. Once this is done, building the initial triangle strip takes theoretically $O(n \log m)$ where n is the number of mesh vertices and m the number of mesh edges. Practically it took less than 0.5s on a 1 Ghz PIII even for the ball joint mesh that has 137062 vertices. The time spent for the iterative optimization process generally depends on the shape of the initial PL. Practically, all optimizations took less

then 5s for all the examples of this poster.

Most of the meshes shown in this poster are courtesy Cyberware and can be downloaded at <http://www.cyberware.com>. Fig. 1 and fig. 2 show a typical example where discrete geodesics can not be used, since the PL is closed. The mesh (a ball joint) has 137062 vertices and 274120 faces. The initial crude PL (left in fig. 1 and 2) interpolates 30 user specified triangles. The images on the right in fig. 1 and 2 show the result of our “cubic” filter. Notice in fig. 2 how the global shape of the initial PL is preserved, while small details are smoothed out.

Fig. 3 (a,b,c) show three PL obtained using the “linear” filter. This example illustrates the fact that our iterative algorithm prevent the PL from jumping over obstacles. It is not able to find shortest paths. The user can precisely choose where the PL should lie: if he/she wants the PL to traverse between the ear and the horn, he/she only has to select on triangle in this area during the initial triangle strip construction, and the PL will remain in this area throughout the optimization process.

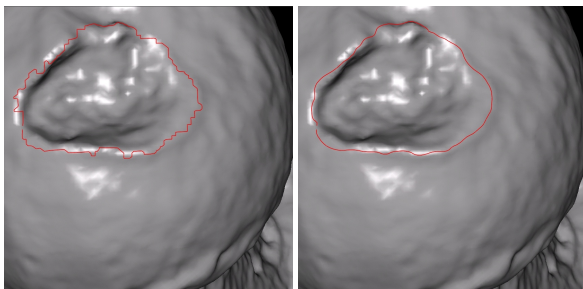


Figure 1: Ball joint mesh, 137062 vertices, 274120 faces. The initial PL shown left interpolates 30 user specified triangles. The smooth PL shown right is obtained using our cubic smoothing filter. This is a typical example where discrete geodesics can not be used.

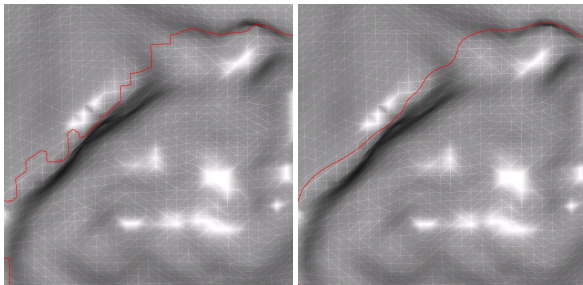


Figure 2: Local zoom on the top left corner of fig. 1. The mesh edges are shown in white.

3 Conclusion

In this poster a very simple and efficient algorithm for building smooth PL on 2D-manifold triangulations has been introduced. By contrast with prior works, this algorithm is based purely on intrinsic geometric properties of the mesh and the PL, and in particular it does not rely on any parameterization. Two different smoothing filters were proposed, one that tends to straighten the PL, and the other that removes small details while preserving the overall shape of the PL.

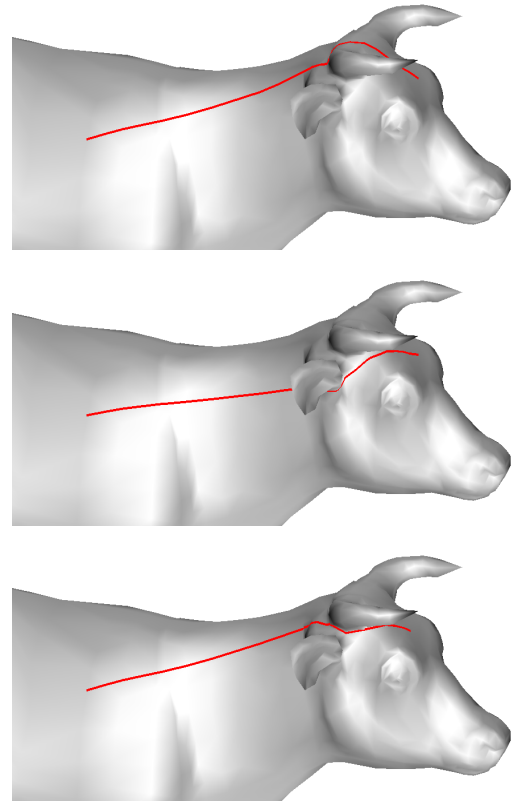


Figure 3: The three smooth PL are obtained using the linear filter with the initial PL shown in fig. ???. Our algorithms prevents the PL from jumping over obstacles.

In future works we will investigate other kind of filters, with larger support, and we will use the results of this poster as a tool for segmentation and visualization purposes.

Acknowledgements

This work was partially supported by the European Community 5-th framework program, with the Research Training Network MINGLE (Multiresolution IN Geometric modELing, HPRN-1999-00117). Special thanks to Alex Yvart for his Dijkstra code. Models are courtesy Cyberware.

Interactive Poster: Real-Time Uncertainty Visualization of Mobile Objects in VGIS

Amin P. Charaniya, Nikolai Faaland, Srikumar Ramalingam and Suresh K. Lodha
Department of Computer Science, University of California, Santa Cruz

William Ribarsky, Nick Faust, Zachary Wartell, and Tony Wasilewski
GVU Center, Georgia Institute of Technology

Abstract

In this paper we present the visualization of spatio-temporal, position and velocity uncertainty associated with mobile objects tracked using Real Time Kinematic DGPS within Virtual Geographic Information System (VGIS). This work demonstrates the need to integrate high resolution geo-spatial and time-dependent 3D data into an interactive environment.

1 Introduction and Overview

Uncertainty is associated with almost every engineering and science process. Estimation and visualization of this uncertainty is extremely crucial for decision making. In this work, we have focussed on uncertainty associated with mobile objects that are tracked using a field device such as a Global Positioning System (GPS) unit. These objects transmit their three-dimensional position and velocity estimates to a central server, that then visualizes these objects. These estimates can be highly sporadic and their accuracy can vary tremendously depending upon the environmental conditions and satellite availability [3]. Thus, we need a method for visualizing these objects along with the uncertainty that is a function of time and certain other parameters. The algorithm used for visualizing these objects is described by Lodha et al. [4]. In this work we have geo-spatially registered these objects within VGIS and visualized associated uncertainty. VGIS is a real-time 3D Geographic Information System developed by the GVU Center at the Georgia Institute of Technology [2]. One of the greatest strengths of VGIS is its ability to handle massive volumes of geographic data while maintaining interactive response.

2 Background

2.1 Uncertainty Computation

An algorithm for visualizing a particle with uncertain initial position and motion is described by Lodha et al. [4]. Using this method we represent the initial position, speed and direction of an object as probability density functions (PDF). Uncertainty in lo-

cation, speed, and direction is modeled by Gaussian probability distributions. We can then compute the probability of the object being present in a given region at any given instant of time using the position, speed and direction PDFs. We can propagate the object as a probability cloud that can be visualized as a galaxy of simple geometric glyphs with the size of each glyph representing the probability of the object being at that position at that instance of time. We can also use transparency and pseudo-coloring schemes to visualize the cloud with more opaque regions indicating higher probability.

2.2 VGIS Architecture and Overview

VGIS visualizes large amount of geographic data consisting of Digital Elevation Models (DEMs), Digital Ortho-photo Quads (DOQs), GIS raster layers (such as weather data, demographic data), 3D objects such as buildings, vehicles, trees and other objects. VGIS runs on a variety of UNIX, Windows, and Linux platforms. One of the driving factors behind VGIS is the ability to respond in real-time. Data is stored at various resolutions for real-time visualization. The run-time visualization component is multi-threaded so as to support real time interactive rates. VGIS can be used in a windows-based interface or virtual reality environment (HMD or large screen projection) with 3D head and hand tracking. VGIS supports visualization of large-scale collections of 3D objects [1]. This support has been implemented as part of VGIS in the form of an Object API. Objects geometry and texture can be loaded from a file. The primary object format is the 'VGIS Object Format (VOF)' which is native to VGIS. VGIS also supports external standards such as OBJ and FLT formats. Complex geometric models can be created using standard utilities such as Wavefront, Maya or 3DStudioMax and can be exported into these formats. These objects are then loaded into VGIS and placed in an appropriate scene graph. The object API supports traversal and modification of the object hierarchy. The position and orientation of individual objects in the hierarchy can be changed. Simple geometric transformations such as translation, rotation and scaling can be applied to these objects.

2.3 GPS-Tracked Mobile Objects

In this work we have employed a Real-Time Kinematic (RTK) DGPS using the Ashtech Z-12 sensor units. We use one base station, that is mounted on the roof of the Baskin School of Engineering that provides carrier-phase differential corrections in real time, and two rover units that are carried around the UCSC Campus. The differential correction signals are transmitted using low-bandwidth UHF band radios. The Z-12 units support 'Reverse Vector RTK' mode of communication where the base can track the rovers with extremely high accuracy (0.5cm as per the specifications).



(a) Ring and porkchop shapes positional uncertainty of mobile objects within VGIS.



(b) Banana and fan shapes also depict positional uncertainty of mobile objects within VGIS.

Figure 1: Visualization of uncertain objects in VGIS

3 Uncertainty Visualization in VGIS

We have inserted the Santa Cruz County elevation and imagery data in VGIS. 1-meter resolution DOQs and 10-meter resolution DEMs for the Santa Cruz County, 0.5-foot resolution DOQs for the Santa Cruz City and UCSC Campus can be visualized interactively within VGIS. These versions are currently working on SGI as well PC (Windows) platforms.

Visualizing the spatio-temporal uncertainty associated with GPS-tracked objects involves reading the instantaneous position and velocity as estimated by the GPS receiver, computing the uncertainty based on certain parameters and environmental settings and modifying the appearance of an objects that is rendered in VGIS. Since retrieving instantaneous position and velocity values from the receiver involves I/O we have implemented that in a separate thread. Once we have the values for position, velocity and certain other parameters we estimate the uncertainty associated with the object using simple models as described by Lodha et al. [3]. The visualization is done using the 'galaxy' representation [4] for the object. The uncertain object is represented as a galaxy of small 3D geometric glyphs (such as cubes or spheres) each of which is scaled differently depending upon the probability of the uncertain object being at that position. Depending upon the uncertainty parameters these objects can take various familiar shapes such as a banana, ring, fan, bow-tie and porkchop. Figure 1 shows uncertain objects visualized in VGIS.

We are submitting a concurrent proposal for demonstrating this system at the IEEE Visualization Conference as well. We are currently working on visualizing uncertainty using transparency within VGIS. Many additional enhancements such as addition of temporal data, volumetrically rendered data, weather data, LIDAR data, and building data are being worked upon. Addition of these heterogeneous spatio-temporal data are raising several interesting research directions that are being actively pursued.

References

- [1] D. Davis, W. Ribarsky, T. Jiang, N. Faust, and Sean Ho. Real-time visualization of scalably large collections of heterogenous objects. In *IEEE Visualization '99*, pages 437–440, 1999.
- [2] Nickolas Faust, William Ribarsky, T. Y. Jiang, and T. Wasilewski. Real-time global data model for the digital earth. In *International Conference on Discrete Global Grids*, 2000.
- [3] S. K. Lodha, A. P. Charaniya, N. Faaland, and S. Ramalingam. Visualization of GPS uncertainty in a 3D GIS environment. In *SPIE Aerospace/Defense Sensing '02*, 2002.
- [4] S. K. Lodha, N. Faaland, A. P. Charaniya, P. K. Varshney, K. Mehrotra, and C. Mohan. Visualization of uncertain particle movement. In *Computer Graphics and Imaging conference*, 2002.

Interactive Poster: Seat Belt Routings over Virtual Humans – The Belt Design Service

Andreas Divivier¹, Achim Ebert², Michael Bender³, Hans Hagen²

¹ tecmath AG, Kaiserslautern, Germany

² Intelligent Visualization and Simulation Systems, DFKI GmbH Kaiserslautern, Germany

³ Department of Computer Sciences, University of Applied Sciences, Zweibrücken, Germany

1. Introduction

The incorporation and simulation of virtual humans is becoming more and more important in a multiplicity of applications. By applying virtual human characters in the different stages of the product life cycle, the time consuming and expensive building of physical mock-ups and prototypes can be reduced enormously. Also the quality of the product itself can be improved.

Within the application of virtual human characters in the process of virtual car design the project *Belt Design Service* funded by the *Stiftung Rheinland-Pfalz für Innovation* had been initiated. The goal of the project was to realize a service for the simulation and evaluation of seat belt routings over virtual human occupants.

For a given vehicle geometry different seat belt routings are computed by incorporating user specific design guidelines, anthropometric data of passengers, kinematic data of the belt system, and possible anchor points of the belt system with the vehicle geometry. As a benefit, by comparing the simulation results for different locations of anchorage points as well as by varying anthropometric dimensions of the occupants, a given design might be improved considering a desired group of buyers. This is of great importance, since vehicle seat belt efficiency directly relates to the anthropometric dimensions and the seat belt geometric design.

2. Related Work

Up to now the calculation of seat belt routings has merely been based on geometric information neglecting anchor point kinematics as well as physical characteristics of the belt. Also differences with respect to anthropometric dimensions of occupants were not considered accurately. In the automobile area it is common to evaluate the comfort of CAD planned motor vehicles by the use of virtual humans before real prototypes exist. Thus whole groups of virtual drivers try to reach for virtual steering wheels, set their feet on fictitious pedals or try to achieve the control elements still in development. These tests lead to well adapted vehicles for genuine drivers, but the use of computer based human models for the simulation of seat belts in the automobile manufacture was neglected until now

(for example see [2], [3], [4]). Tools such as the two-dimensional 2D-SAE-H-Point template and the 3D-SAE-H-Point measurement machine Belt Fit Test Device (BTD) are still frequently used today. The main disadvantage of these methods is that the physique proportions with regard to the user population as well as individual features e.g. pregnancy are taken into account only insufficiently. Mobile parts of the belt system as well as collisions and contact points of the belt with the model or vehicle geometry generally remain unconsidered.

3. The Belt Design Service

The main goal of this paper lies particularly on the acquirement of solutions which allow a suitable modeling and computing of the seat belt. By comparing the simulation results for different locations of anchorage points as well as by varying anthropometric dimensions of the occupants, a given design might be improved. This is of great importance, since vehicle seat belt efficiency directly relates to the anthropometric dimensions and the seat belt geometric design.

In our simulation process, we incorporated principles of physical modeling, where the description of objects is supplemented by the laws of physics leading to a much more realistic representation and behavior. For the modeling of the belt we have chosen a mass / spring network. The actual simulation is based on computing the oscillatory motion of the mass points taking into account spring forces, friction and contact with the lap and torso surfaces. Hereby a more realistic twisting of the belt as well as a more realistic prognosis of the detachment points can be achieved.

3.1. Computation

The seat belt is defined by a network of mass particles, connected by damped springs in longitudinal, cross and diagonal direction. The number of belt particles is calculated from a user-defined belt resolution value defining the separation of two particles in the eased state. Based on the starting position of the belt anchor an initial routing for the chest and the pelvis belt are computed: the chest belt is evenly put around the

bounding box of the torso relevant parts of the body (torso, neck), and the pelvis belt is spread out in form of a semicircle whose center and orientation is computed from the bounding box of the seat relevant parts of the body (thigh, pelvis). The belt lines start at the respective points on the deflecting edge. In the first simulation step the forces acting on all mass particles are computed. Every mass particle is then moved under consideration of a predefined maximum moving length and a possible collision with the model. The calculation of the current forces and the movement of the mass particles is carried out until a maximum iteration bound is reached or the spring system is in equilibrium. For the computation of possible collisions the relevant belt data and vehicle geometry data is stored in an octree, reducing collision computations considerably.

For the implementation of anchor point kinematics different types of seat belt anchors, buckles and retractors commonly used in seat belt designs have been identified. For each type a simplified model forming a kinematic chain (i.e. a chain of links and joints) has been created. Forces induced by belt mass particles are used to calculate the moments of force for each joint as well as the corresponding movement of the anchor point.

Besides the real geometrical description of the belt routing, additional information is generated during the simulation: the length of the belt, the minimal and maximum distance of the belt lines to the human body and the positions of the contact and detachment points. Variations of belt anchorage points and comparison of computed belt routings lead therefore to an overall improved precision of register and an increased belt comfort.

3.2. Implementation and Visualization

In order to provide the Belt Design Service independent from the application software and the underlying human model, a CORBA-based client/server architecture has been realized. The client transfers the description of the occupants' geometry, the belt system parameters, as well as the geometry of the car to the server and then invokes the simulation process, waiting for possible solutions.

The server's object manager is responsible for the administration and conversion of incoming and outgoing data to and from an internal data format. Since the Service does not have a human representation, the skin surface transferred by the client serves as a rigid representation of the human model. Besides the geometrical description of the seat belt additional characteristic values (e.g. length of the belt, location of anchor points, slackness – factor) are computed to give information about potential quality and effectiveness of the belt. Under consideration of these values different

belt routings can be compared in order to rate modifications (e.g. different locations of the anchorage points).

Since hardware and software configuration and evolution is hardly predictable our second objective was the use of a software structure which can highly adapt to changing conditions. Therefore we used a component system to implement our simulation and visualization system and we used a software agent paradigm to tune the entire process. For this purpose we revert on our previously developed visualization system framework MacVis ([1]).

4. Results

Figure 1 illustrates the final state of the belt routings (light red stands for large forces and dark red for small forces).

By comparing the simulation results for different seat belt anchorage points and for different human occupants the belts precision of register can be optimized and thus a quality improvement of the entire product is already achieved during the design phase.

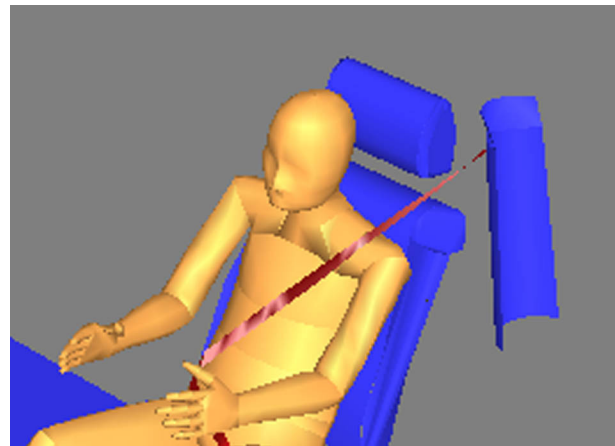


Figure 1: Final solution of the belt course.

5. References

- [1] Ebert, Divivier, Barthel, Bender, Hagen: Improving Development and Usage of Visualization Applications, IASTED, Marbella, Spain, 2001.
- [2] Lehto, Foley: „Physical Aspects of Car Design: Occupant Protection“, Automotive Ergonomics, Chapter 7, Peacock and Karwowski (eds.), Taylor and Francis, 1993.
- [3] Noy, Battista, Carrier: “Development of an Electronic Belt Fit Test Device”, Society of Automotive Engineers, 971137, 1997.
- [4] SAE (Society of Automotive Engineers, Inc.): “Surface Vehicle Recommended Practice – Seat Belt Hardware Test Procedures”, SAE J140, 1995.

Interactive poster: A 4D Deformable Model to Visualize Time-Varying Volumetric Datasets

Johnny De Freitas-Caires

Aassif Benassarou

Eric Bittar

Laurent Lucas

LERI (EA2618) - Université de Reims Champagne-Ardenne *
Modelization and Dynamic Animation for Simulation Group

Abstract

The aim of this work is to automatically extract and visualize quantitative parameters from time sequences of 3D images (also called 4D images). We propose an integrated framework based on a deformable surface model. We describe how the use of a deformable surface model which may change its topology, is relevant to solve the difficulties that arise in this field. Our method is illustrated by experimental results on real biomedical data obtained by confocal microscopy.

1 Introduction

Because they correspond to the dimensionality of what is the physical reality in contrast with 2D or/and 3D images which are 'only' reductions by projection or time fixation, 4D images provide an integral approach to the quantitative analysis of motion and deformation in a real tri-dimensional world over the time. They can then attempt to understand the underlying cause and try to build predictive models. Defined as 4-dimensional $\{x, y, z, t\}$ -recordings, time-varying volume datasets may be obtained from numerical simulations or remote sensing devices. Typical examples of time-varying dataset concerns CFD simulation for instance, or, in biology, the living cell imaging using chimeric fluorescent-tagged proteins to probe cellular models and study localization (structures) in relation with dynamics (functions). The main problem resulting from 3D continuum observations is the huge amount of data acquired to be processed over time. Instead of just one regular dataset of 256^3 voxels or more, there could be now a few tens to hundreds of timesteps.

If an obvious method simply consists in considering independently every 3D image of the time sequence to produce movies from a set of individual frames obtained by isosurfaces or volume rendering, it is clearly an inefficient approach to correctly analyze and interpret all the information contained in the series. In particular as mentioned in [1], questions may arise like, *How does a particular region of space vary over time ? How does its volume change ? Does it split or merge with other regions ?* These questions do not find satisfactory answers in a 3D basic framework. Indeed, they do not provide any analytical information on the evolving anamorphous regions such as their lifetimes, volumes mass and other morphometric or statistical features. In the same way, they do not allow users to characterize the evolutionary events present in continuum. So, analysis and visualization tools are crucial to understand time-varying datasets.

The objective we wish to achieve is to allow users to track and interactively visualize a set of multiple objects, while studying their topological and geometrical evolution (see Figure 1). The originality of our framework stands in the integration of specific ap-

*Laboratoire d'Etudes et de Recherches Informatiques, Rue des crayères. BP1035 51687 REIMS Cedex 2 (France), URL <http://www.univ-reims.fr/leri>, Contact : laurent.lucas@univ-reims.fr

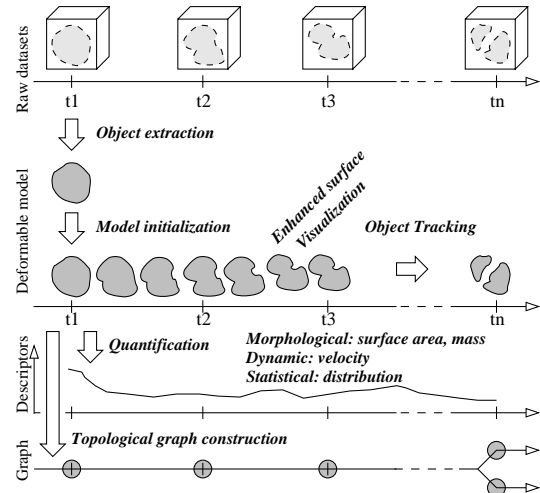


Figure 1: Data types and processes.

proaches: deformable models, tracking and visualization to study the evolution of multiple topology changing objects in a 4D image sequence. The advantage of deformable models is their continuous envelope, preserved by inner constraints, which interpolates the objects boundaries where they can be too noisy to be extracted automatically. Tracking is useful in time evolving data, but not sufficient for representing the objects. Visualization is necessary and useful to understand 4D data, but it is usually done by successively visualizing the 3D volumes at a specific time, thus missing the information on the evolution.

The paper is organized as follows: In section 2, we present the different processes that we use to achieve our goals, and the deformable model we chose. Experimental results are reported and discussed in section 3, before concluding and identifying topics for future research.

2 The deformable model and its evolution

The δ -snake (first introduced by J.O. Lachaud in [2]) is a three-dimensional deformable surface model made of triangular faces. It owes its name to the δ parameter that governs its structure. Indeed, the length of every edge of the mesh must be between δ and $\zeta * \delta$, with $\zeta = 2.5$. These constraints ensure the regularity of the mesh. If they are violated, Eulerian operations are applied until the regularity has been recovered. If an edge is too long, a new vertex is created in its middle. If two neighbor vertices are too close, they are merged. Another constraint permits to detect intersections of the surfaces: the distance between two non neighbor vertices must

be greater than $\zeta/\sqrt{3} * \delta$. Two non-Eulerian operators perform topological breaks. The first one, the “annular break”, is needed as soon as a connected part of the surface which is homotopic to a circle is shrinking to a point (see Figure 2). The second one, the “axial break”, is used when two distinct surface patches become too close and tend to collide. Our main contribution to this model is to have introduced a new energy term. In contrast to many earlier approaches, this term ensures the smoothness in the deformation process and it can be compared to a temporal regularization force.

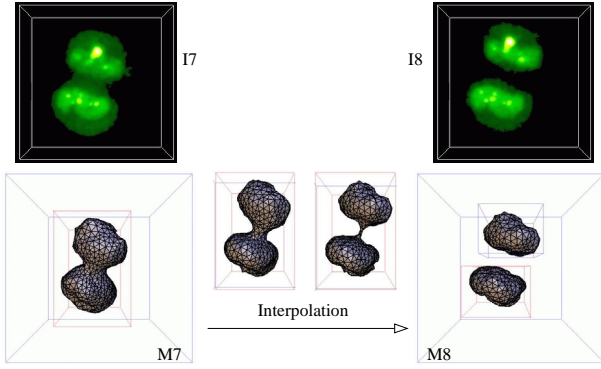


Figure 2: Splitting of objects

3 Graph representation of the evolution

The topological events, the surface mesh and the descriptors of the evolution (volume, position, velocity ...) are stored during the tracking process in an history database. The latter is then structured as a direct acyclic graph G , to organize the interpolation of every object between the images, and replay the evolution as needed. From the simplification of this graph G is computed another graph G' , which only keeps the information at the times of the acquisitions of the images. The edges of both graphs identify the topological events as presented in Figure 3 ii. One contribution of this work lies in this structuring stage because it ensures a coherent management of the continuum, to allow the easy development of a visualization toolkit.

An easy way to represent G simply consists to project it on a plane, as we can see on Figure 4 iii. Each object can also be colored in function of its history by assigning a constant color along its evolution, or by blending its descendants with a percentage of their parents colors (see Figure 5). This approach allows the user to visualize in a synthetic way the symbolic representation of the different topological events that occur during the evolution.

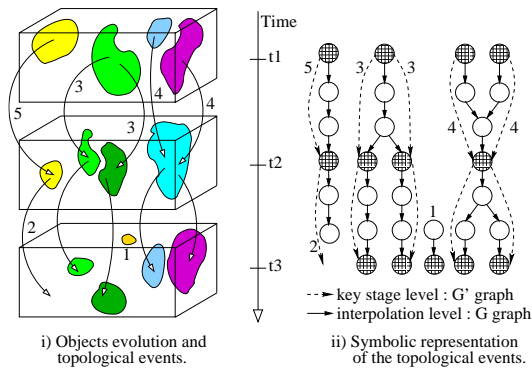


Figure 3: Objects evolution and topological events

Another way to represent the same graph consist to use a “trajectory” metaphor which consists to assign to each centroid of each object a position in a line path in order to visualize all the trajectories of all objects during the image volume sequence. Figure 4 shows the evolution of proteins inside of one nucleolus.

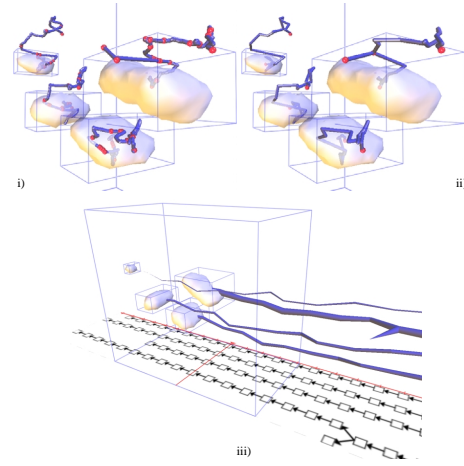


Figure 4: The trajectories shows the interpolation and key-stage levels

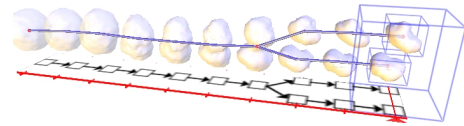


Figure 5: Integration of the time in the spatial representation and trajectories

4 Experimental results

The data are focused on the nucleolus which is, in the cell, the site of synthesis and maturation of ribosomal RNA (rRNA). Its study will make it possible to acquire knowledge which will be useful for the forecast of human cancers. Tagged proteins within the nucleolus throughout the cell cycle may be observed thanks to the Green Fluorescent Protein (GFP), an auto-fluorescent protein from the jellyfish *Aequorea victoria*. Thirty to fifty optical sections were recorded every three minutes for several hours within the whole volume of each cell.

5 Conclusion and Future Work

We have demonstrated the ability of our model to track 3D objects that change their topology in 4D images. Symbolic representations are also computed to help the structure understanding and visualize the evolution of the objects through different topological events. Later on, predictive aspects will have to be taken into account.

References

- [1] D. Silver and X. Wang. Volume tracking. In *Proc. of IEEE Visualization'96 Conference*, pages 157–164, San Francisco, CA, USA, 1996.
- [2] J.-O. Lachaud and A. Montanvert. Deformable meshes with automated topology changes for coarse-to-fine 3D surface extraction. *Medical Image Analysis*, 3(2):187–207, 1999.

Interactive Poster: A Multi-Attribute Visualization for Analyzing Large Web Transactions

Ming C. Hao, Dayal Umeshwar, Daniel Cotting*, Thomas Holenstein*, Markus Gross*
Hewlett Packard Research Laboratories, Palo Alto, CA
(ming_hao, dayal_umeshwar)@hp.com

1. Motivation

A transaction starts with a user (client) clicking on a web page (URL). The web browser sends the request through applications servers, to perform some service. For example, a user clicks on a web page to purchase a book. Recently, the rapid increase of transactions on the Internet has led to the availability of large volumes of transaction data. Business research efforts have focused on how to turn raw data into valuable information, which can be visually analyzed.

Web transactions are multidimensional. For example, clients, URLs, response time, and number of messages are attributes for a web transaction. Product type, country name, price, and quantity are attributes for a finance revenue transaction. Based on our practical experience, we have discovered that using one single type of relationship for all data items is too general. The relationships can vary significantly depending on the selected attributes. One of the key questions is how to lay out the multiple relationships simultaneously in a graph -- such as the relationships between the web client response time and URLs in a web access application, and the relationships between product types and countries in a sales analysis. Each pair of relationships plays an important role for overall visual analysis.

To date, many practical applications have shown that the physics-based visualization technique is one of the best methods for web browsing and information retrieval systems [1, 3]. However, the usefulness of this technique is limited if the volume of data objects grows large.

2. Our Approach

In this poster, we derive a freeze technique to enhance the physics-based visualization system. The idea is to set certain objects frozen (not to move) during the construction of a graph. As a result, we substantially reduce the force computation time and make already placed relationships unchanged before laying out the next one. This technique consists of an automated *classification*, a *freeze operation*, and a *graph layout*. These three components (steps) are iterated until the final graph is generated.

2.1. Automated Classification

A classifier automatically separates relationships among data items into primary and secondary classes based on data attributes. It determines which class of data items to

freeze and which class of data items will participate in the graph layout. There is only one primary relationship in a graph and many secondary relationships with respect to the primary relationships. For example, in a web access application, the relationship between pairs of the client response time is the primary class relationship.

2.2. A Freeze Operation

This component constructs a 3-dimensional array (x-axis, y-axis, z-axis) to record the position of the frozen data items. The frozen data items will not be moved during the layout of non-freeze data items. The order of freeze operation depends on the sequence of the relationships. To layout the primary relationship, the freeze operation needs to freeze the data items in the secondary class before placing the primary relationship. To layout the secondary relationship, the data items in the primary class need to be frozen and not participate in the graph generation.

2.3. A Graph Optimization and Layout

This component applies the “physics-based mass-spring” algorithms [2] through relaxation (many iterations). In order to compute the final layout for visualization, the total force (E) needs to be minimized. Our goal is to find a configuration of particle positions such that a local minimal of energy level reached.

$$E = \frac{1}{2} \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \epsilon(|X_i - X_j|, s_{ij})$$

Where items $i \in \{1, \dots, n\}$, and the position corresponding to item i in three-dimensional space is denoted as X_i . X represents all the points in a matrix. The distance between two points X_i and X_j is $|X_i - X_j|$. The relationship between two items is denoted as s_{ij} .

The above computation only applies on the non-freeze objects under certain boundary conditions. There are no force computations and position movements over time on frozen objects. As a result, this elimination shortens the graph layout time by at least a two orders of magnitude.

3. Applications

This iterated freeze technique has been prototyped in several e-service applications at Hewlett Packard Laboratories. It has been used to visually analyze large

*Presently with Department of Computer Science, Swiss Federal Institute of Technology, Zurich, Switzerland
dcotting@student.ethz.ch; holenst@inf.ethz.ch; grossm@inf.ethz.ch;

volumes of service and sales transactions at online web sites.

3.1 Web Access Analysis

Figure 1 (A, B) illustrates the construction of a web access graph. It contains 400,000 transaction records with 986 clients and over two thousand URLs. At initialization time, data items are classified into to classes; the clients (rectangles, primary class) and URLs (spheres, secondary class) are placed on a spherical surface.

- (A) Freeze URLs - layout client and client relationship:
Employ force-based algorithms to layout clients in such a way that clients with similar response times are arranged close together (automatically groups as low, medium, high).
- (B) Freeze clients - layout URL and client relationship:
URLs are placed close to the clients to show the access clients.

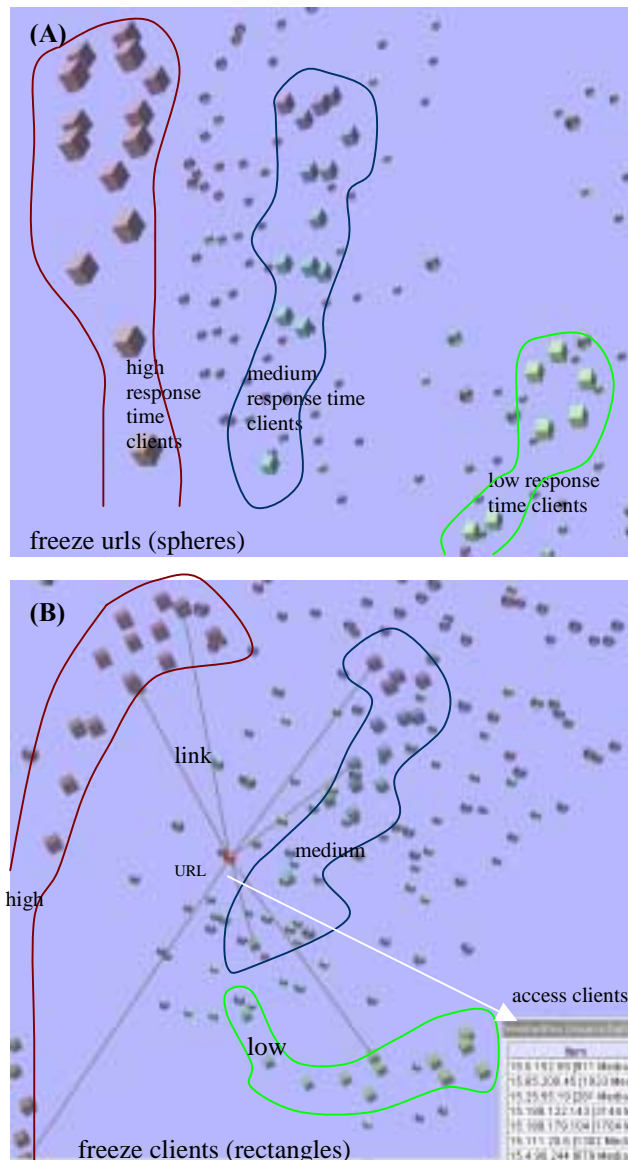


Figure 1: Web Access Analysis

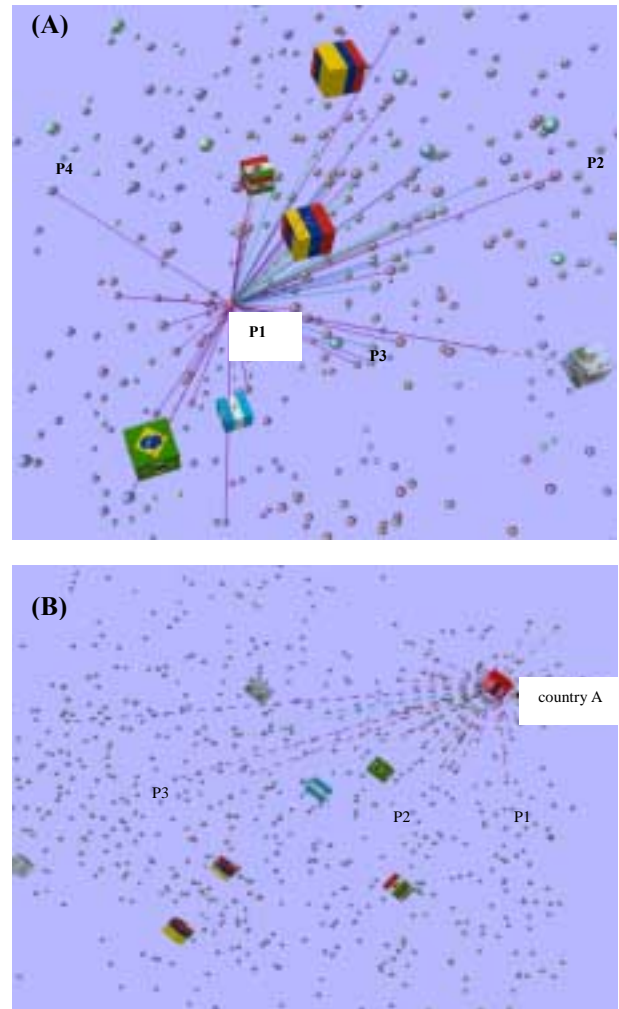


Figure 2: Sales Transaction Analysis.

3.2 Sales Transaction Analysis

In Figure 2, the sales transaction analysis application offers two different relationships:

- (A) Product and Product Relationship
Products are laid out based on how often they are bought together at same time (indicated by solid colored lines).
- (B) Country and Product Relationship
Countries are laid out with respect to products sold. Related products or counties are linked with colored dashed lines (color represents quantities).

4. Conclusion

This poster presents a new iterative freezing technique enabling analysts to quickly visualize large multi-attribute web transactions and to understand service patterns and correlations.

5. References

- [1] M. Kreuzler, N. Lopez, and H. Schumann: "A Scalable Framework for Information Visualization", Proc. Information Visualization 2000, Salt Lake City, UT.
- [2] T.C. Sprenger, R. Brunella, and M.H. Gross, "H-BLOB: A Hierarchical Visual Clustering Method Using Implicit Surfaces", IEEE/VIS2000, Salt Lake City, UT.

Interactive Poster: Visualization of vector Fields using Line Integral Convolution and volume rendering

Anders Helgeland

Oeyvind Andreassen

Norwegian Defence Research Establishment (FFI)

Abstract: Line Integral Convolution (LIC), introduced in 1993 by Cabral and Leedom [1], is a powerful texture-based technique for vector field visualization. Based upon blurring of an input texture locally along the field lines of a given vector field, it generates images that convey the directional information of the field. Due to the computational expense of generating the 3D textures and the difficulties of effectively displaying the resulting output textures, LIC has most commonly been used to depict vector fields in 2D or over a surface in 3D. We propose and study methods and strategies for more efficient visualization of three-dimensional vector fields with LIC.

For large 3D data sets (512^3 or greater), even the Fast LIC algorithm proposed by Stalling and Hege [3] is very computationally intensive. We present a new method for computing 3D LIC textures called *Seed LIC*. This method exploits the sparsity of the input texture by calculating field lines and computing the convolution starting from a set of distributed points (the seed points) only. The seed points can be chosen utilizing certain properties of the field to be visualized, for example the vector magnitude. Sparse input textures have been used together with LIC to reveal the interior structures of the field [2]. The Seed LIC algorithm can be faster than the Fast LIC algorithm by more than an order of magnitude, and can thus be used in an interactive setting. Instead of a computational time of several hours, the CPU time used for generating the LIC textures can be reduced to a few minutes or even seconds. Earlier, 3D LIC have been thought of as a tool only to visualize vector fields in those instances where high image quality is desired and the ability to generate images at an interactive rate is not required.

To reveal the depth relation among the field lines, a shading, or halo technique called limb darkening is used. To emphasize the halo effect and to reduce aliasing effects due to the use of voxels to represent the field lines, the textures are oversampled by a factor of 2 to 4 in each dimension and then convolved with an isotropic $3 \times 3 \times 3$ filter. This approach smears the field lines outwards, making the strokes in the output texture thicker and the 3D shape more clear.

Finally, we show how direct volume rendering applied to 3D LIC can be used to display additional information about related scalar fields when visualizing a vector field. This includes the presentation of a new two-field visualization technique, which we have called *candy cane visualization* (see figure 1). In the example, we use this technique to depict the directional structures of the vorticity field inside vortices obtained from a simulation of stratified shear turbulence by Werne and Fritts [4].

REFERENCES

- [1] B. CABRAL and C. LEEDOM. Imaging Vector Fields Using Line Integral Convolution. In *Computer Graphics Proceedings*, volume 27 of *Annual Conference Series*, pages 263–270, July 1993.
- [2] V. INTERRANTE and C. GROSCH. Recent Advances in Visualizing 3D Flow with LIC. ICASE Report No. 98-26, NASA Langley Research Center, July 1998. <http://citeseer.nj.nec.com/interrante98recent.html>.
- [3] D. STALLING and H.-C. HEGE. Fast and Resolution Independent Line Integral Convolution. In *Computer Graphics Proceedings*, Annual Conference Series, pages 249–256, August 1995.
- [4] J. WERNE and D. C. FRITTS. Stratified shear turbulence: Evolution and statistic. *Geophysical research letters*, 26(4):439–442, February 1999.

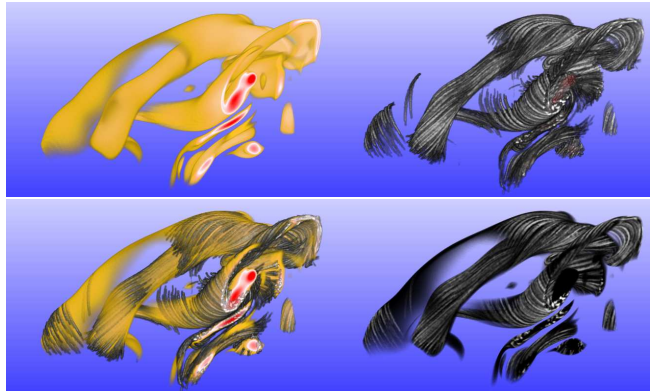


Figure 1: Visualization of two fields, obtained from a simulation of stratified shear turbulence, simultaneously: *vorticity* ($\omega = \nabla \times \mathbf{u}$), represented by the LIC texture and *enstrophy* ($\eta = \|\omega\|^2$). Top left: Visualization of the enstrophy field. The color varies from yellow to red with increasing enstrophy value. Top right: Visualization of the vorticity field using Line Integral Convolution. Bottom left: Two-field visualization, using two independent sets of color and opacity tables. Both fields are displayed simultaneously. Bottom right: Two-field visualization, using enstrophy to define the opacity and the LIC texture displaying the vorticity field to define the color. The field lines of the vorticity field twist around the vortices, resulting in “objects” similar to “candy canes”. The resolution of the textures are $(244 \times 394 \times 224)$.

Interactive Poster: geeViz (Grid-Enabled Environment Visualization) Realtime Visualization of Distributed Systems

Joseph A. Insley Michael E. Papka

*Mathematics and Computer Science Division,
Argonne National Laboratory, Argonne, IL 60439, USA
{insley, papka}@mcs.anl.gov*

1. Introduction

A growing number of applications use collections of geographically distributed resources, with scope and membership changing over time. This increased complexity makes it important to understand interactions among the various components of a distributed system.

One way of gaining insight into the behavior of such systems is to visualize their activities. Seeing when and where jobs are executed and how data is transferred among jobs can help users discover performance bottlenecks and diagnose erroneous behavior. It also enables one to monitor the progress of a computation and to observe the overall state of the system. Visualizing network connectivity among the resources on which distributed applications execute can reap similar benefits. Such visualizations can also serve as a valuable demonstration tool, making it easier to explain complex systems to someone unfamiliar with them.

We present geeViz, a tool that integrates application-specific performance data with network routing and connectivity data to create interactive, quasi-real-time visualizations of Grid applications that communicate the geographical location of application components and the resources that they use, and the interactions among them.

We use three example applications to illustrate the capabilities of our geeViz prototype. Our experience with these applications persuades us that interactive, dynamic display and geographical location information can be a powerful combination for understanding (and communicating) Grid application behavior and performance.

2. Visualization System

The geeViz visualization system consists of a flat map of the world on which are placed 3D primitives representing resources and network connections between those resources. Data transfers are represented by spheres that move along network connections. An interactive interface allows the user to zoom in and out, pan across the map, and select resources to obtain more detailed information. A graphical user interface (GUI) provides additional controls and more textual information.

Display clutter has been identified as a serious problem when displaying large networks [1]. The application runs on either a desktop workstation or a tiled

display. The advantage of using the tiled display over a workstation is that the large format and increased resolution enable a higher level of detail of the map, thus reducing clutter.

To provide a geographically realistic depiction of resource location, we use a variety of sources to obtain latitude/longitude information for placement of resources.

2.1. Traceroute

We address the problems of how to determine and represent properties of the network links that connect resources. A variety of systems exist for determining the physical topology of network connections and both end-to-end and link-by-link characteristics. In our initial work, we use traceroute [2] to obtain both topology and latency information. We run traceroute from one host to another and visualize the route that is returned, by placing a cone on the map where each router in the trace is located and connecting these cones with tubes to represent the links between hops. Links are color-coded based on the latency for their trace segment. Spheres move along the links at varying speeds; the time to traverse a link is proportional to the latency of that link. Other systems, such as GTrace [3] and VisualRoute [4], have similar visualization capabilities but alone lack the range of information that we hope to provide with our tool.

2.2. Grid Monitoring

One challenge to visualizing the progress and performance of actual applications is obtaining appropriate data from these applications. We have developed a custom event infrastructure for use in our work. This infrastructure comprises a logging server to which applications can send events, as well as APIs for sending events to and receiving events from this server. In our initial prototype we defined two types of events: a *transfer type*, which gives performance measurements associated with data transfers, and a *job type*, which indicates the number (and state) of jobs running on a compute resource.

When geeViz receives the first transfer event for a stream, it plots the endpoints of the transfer on the map, adds a link between the two sites, and stores the statistics for the transfer (total number of bytes transferred so far, current bandwidth, and average bandwidth). These values are updated for subsequent events for that transfer.

Spheres are moved along the link in the direction of the data flow. Each sphere represents a particular amount of data, which is controlled from a slider on the GUI, and moves at a constant speed along the link. Additional spheres are placed onto the link based on the current bandwidth. When multiple transfers move in the same direction on a link, each transfer can be displayed as a different set of spheres (each set a different color) or as a single set representing the total bandwidth of all transfers.

For job events, geeViz plots the site where the jobs are running and keeps track of the number submitted, completed successfully, and failed. The site is highlighted while jobs are actively running there.

Selecting a site on the map brings up a text region with information about the site, hosts who have jobs that are being monitored, and transfers into and out of this site.

3. Application Examples

We use three examples to illustrate the capabilities of geeViz: a data transfer system, a remote computing application, and a collaborative environment.

3.1. GridFTP

We exploited logging capabilities incorporated in the GridFTP data transfer tools [5] to enable visualization of wide area data transfers. In a demonstration at the SC2001 conference we visualized the activities of a prototype GridFTP proxy server. With a GridFTP server running on each node of two eight-node clusters, the front-end proxy server receives all initial connections and forwards transfer requests to the backend server with the lightest load. We used our visualization tool to monitor 58 hosts across 21 sites as they transferred data to the proxy server.

3.2. Remote Computing Application

We have used geeViz to monitor a prototype remote computing application developed within the National Fusion Collaboratory project [6]. In this application, data is transferred from a storage location to several compute resources that perform some computation. The results are then transferred back to the storage resource. This new data is then transferred to another resource where a visualization client displays it.

3.3. Access Grid Beacon

The NLANR Multicast Beacon [7] is a suite of tools for monitoring the connectivity between hosts in a multicast session. We have extended geeViz to support the visualization of this data. Cones are again placed on the map to represent sites where Beacon Clients are being run and information about a site is displayed when it is selected on the map.

Links are drawn from each site to all other sites and are color-coded based on the value of the parameter being displayed: loss, delay, jitter, order, or duplicate. The user can select which parameter to display on the links. A

legend shows the values represented by each color for the current parameter.

The map can quickly become cluttered and unreadable if all links are shown at once. Therefore, there are controls for selecting which links to display. One can easily show only those links that have valid data or, conversely, only those with stale data. This can be useful when trying to identify sites in the multicast session that are having connectivity problems. The user can also toggle on/off all links into/out of individual sites.

The Access Grid (AG) is a system for distributed group-to-group collaboration [8], including communication through audio and video streams broadcast via multicast. The AG community uses the multicast beacon to monitor connectivity between AG nodes. In addition to network measurements, the AG beacon contains information about the hosts running the beacon, such as the architecture of the machine and versions of the software. We have used geeViz to successfully monitor all of this AG beacon data. The additional information is made available on the GUI of our tool. Using our tool to visualize this data not only gives insight into the connectivity issues of the AG but also helps illustrate the extent to which the AG has grown by showing the distribution of sites around the world.

Acknowledgments

Thanks to all of the members of the Distributed Systems and Futures Laboratories at Argonne National Laboratory.

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38 and included partial support under the auspices of the Scientific Discovery through Advanced Computing Initiative.

References

- [1] S. G. Eick, Aspects of Network Visualization. *IEEE Computer Graphics and Applications*. March 1996, pp. 69 -72.
- [2] V. Jacobson, Traceroute source code and documentation, 1988. Available from ftp.ee.lbl.gov/traceroute.tar.gz.
- [3] R. Periakaruppan and E. Nemeth, GTrace – A Graphical Traceroute Tool. *Proceedings of 13th Systems Administration Conference - LISA 1999*.
- [4] VisualRoute. www.visualroute.com/visualroute/index.html
- [5] W. Allcock, A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke, The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Journal of Network and Computer Applications*, 23:187-200, 2001.
- [6] K. Keahey, T. Fredian, Q. Peng, D. P. Schissel, M. Thompson, I. Foster, M. Greenwald, and D. McCune, Computational Grids in Action: The National Fusion Collaboratory, accepted for publication in the *Journal of Future Generation Computer Systems*.
- [7] NLANR Multicast Beacon. dast.nlanr.net/Projects/Beacon/
- [8] L. Childers, T. Disz, R. Olson, M. E. Papka, R. Stevens, and T. Udeshi, Access Grid: Immersive Group-to-Group Collaborative Visualization, *Proceedings of the Fourth International Immersive Projection Technology Workshop*, June 19-20, 2000.

Interactive Poster: The Great Potato Search: The Effects of Visual Context on Users’ Feature Search and Recognition Abilities in an IVR Scene

Cullen D. Jackson David B. Karelitz Sean A. Cannella David H. Laidlaw
Department of Computer Science, Brown University, Providence, RI 02912
{cj, dbk, scannell, dhl}@cs.brown.edu

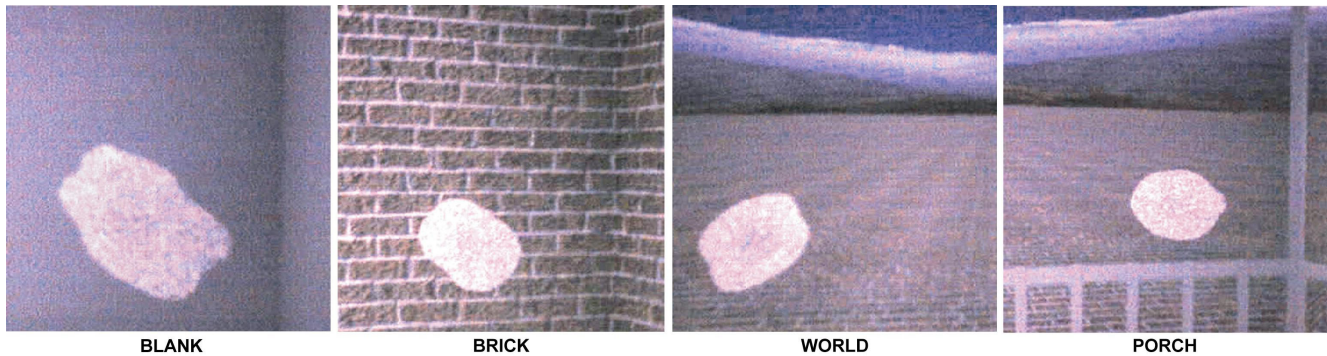


Figure 1. The four visual contexts, BLANK, BRICK, WORLD, and PORCH, displayed in our study of user search performance. The white objects in each context are examples of those displayed during the search task.

Abstract

We present a user study of the effects of visual context and interaction methods on visual search performance in an immersive virtual reality (IVR) system. Contrary to previous research on 2D visual search, our results suggest that visual context does not significantly contribute to a user’s ability to find and identify features on an object in a scene. However, we did find that using a hand-held device to manipulate the object during search facilitated user performance more than simply walking around a static object.

1. Problem

Previous research suggests that visual context facilitates participants’ ability to find and recognize objects in real-world scenes [1], [2]. These “real-world” studies suggest that humans use visual context to assess object congruence with the background, at least when the scene is briefly shown.

The current study aims to examine the role of visual context on a visual search task in an immersive virtual reality (IVR) environment. IVR allows users to fully interact in an actual 3D scene, replete with color, texture, and stereoscopic depth information. We can carefully control the parameters that describe each context, with the aim of understanding those variables that best optimize user performance. A secondary goal is to determine if there is a relationship between the visual context and the method of user interaction (walking around or using a hand-held device).

2. Hypotheses

We hypothesize that the previous results from the 2D visual search studies will generalize to a 3D setting, and that user performance will vary across visual context. Specifically, contexts which provide good motion parallax information and that are projected coincident to the physical CAVE walls will facilitate user performance. Furthermore, we hypothesize that users will be more accurate in finding and identifying features while using a hand-held device to manipulate the object (no locomotion) than while walking around the object; users are able to examine the objects at more orientations and viewpoints with the hand-held device than when the objects are in a fixed position in the environment, and do so more thoroughly.

3. Methods

Fourteen people ($mean_{age} = 24.14 \pm 3.37$ yrs.) participated in the study. The study took place in a CAVE [3] environment with three walls and a floor. The study was split into two 60-trial sets, one for each of the interaction types. Each set contained 12 practice and 48 test trials varied according to the context and the feature presented. In the BLANK context, all of the walls were the same shade of gray. A brick texture was drawn on the walls in the BRICK context. The WORLD context was composed of a brick floor (same as used in the BRICK context) extending to a horizon with mountains and trees and a blue sky above. The PORCH context was identical to the WORLD context with a white porch in the foreground, coincident with the physical walls of the CAVE (see Figure 1).

Users were instructed to find and identify a randomly placed feature (square or triangle) on the object. Each object had a noisy surface and floated in the middle of the CAVE in each trial. The features extended off the surface of each object at a fixed height while the heights of the noise components were random with a fixed maximum height. The shape of each object was determined by a series of spherical harmonics such that the objects were of similar ‘lumpiness,’ but not identical, between trials. Users received feedback during practice trials only. The two interaction methods used for the search task were walking around the object and using a hand-held device to manipulate the object while standing still.

4. Results and Discussion

Our hypotheses suggest that the BRICK context should facilitate user search performance more than the other contexts, followed by the PORCH and WORLD contexts respectively. The BLANK context, with a lack of any texture or depth information (sans the physical CAVE walls) should not facilitate user performance at all.

Results are reported for 9 of the 14 participants. The data of five participants were excluded because they timed out of more than eight of the 48 total test trials on at least one of the test sets (average number of timed-out trials was 12.5 ± 3.6).

No relationship was found between the main factors of the experiment (context, interaction method, and feature type) and there was no effect of context on user performance (shown by a four-way $4 \times 2 \times 2$ ANOVA). However, there was a significant increase in recognition performance when using the wand to manipulate the object instead of walking around the object [$F(1, 7) = 6.37, MS_e = 250.39, p < .05$]. The relationship between the three within-subject factors is shown in Figure 2. In this figure, note the difference in performance across the two plots, each of which represents an interaction type. The mean accuracy for the two interaction types are shown as dashed lines in Figure 2 ($mean_{walk} = 79.1\%, mean_{wand} = 86.4\%$).

These results are contrary to our initial hypotheses concerning the visual contexts; no differences were found between the contexts. One reason may be that the dependent measure was not sensitive enough to capture subtleties in user performance. Another reason may be that users were focused only on the object. We might find that context influences performance in a more exocentric task that requires users to investigate more than one area of the virtual environment.

Assuming a sensitive enough measure, context *did not matter* in terms of this visual search task. This result is contrary to 2D studies showing that context aids object search and recognition. However, those tasks were performed with 2D images and over brief presentation times. The current study utilized 3D scenes, over much longer presentation times, with motion.

The results of this study suggest several things. First, motion cues provide robust configural information about the scene, as well as providing strong depth cues. Stereo also helps disambiguate information in the scene, particularly the structure of the feature with respect to the object. While we did not explicitly test the effects of motion cues or stereo viewing on the task (artifacts of head-tracked stereo IVR), previous research suggests that users make fewer errors comprehending visual data while using hand-coupled or head-coupled motion with stereo viewing [4]. Also, long presentation times permit the user to

integrate a large amount of the available visual information in the scene. We believe that these strong cues, lacking in previous 2D visual search studies, likely contributed to the absence of differences between the four visual contexts in this study.

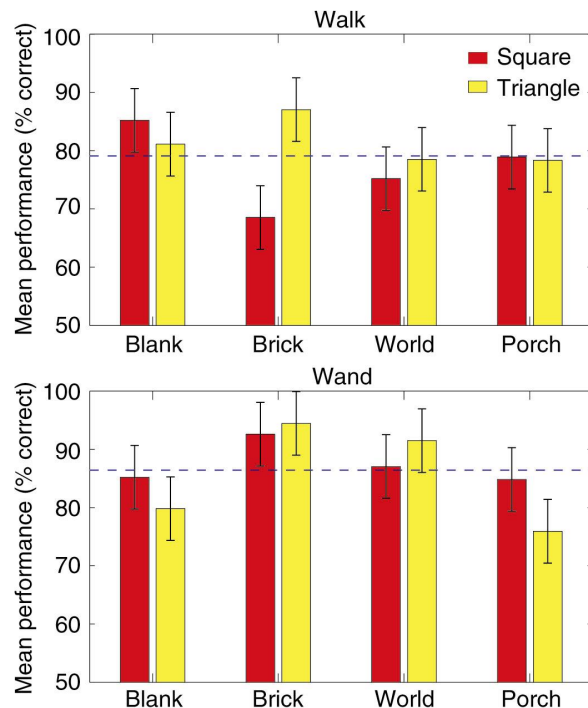


Figure 2. Influence of context, feature type, and interaction method on task performance. Context did not affect performance, while interaction type did. Dashed lines show mean performance over interaction type ($mean_{walk} = 79.1\%, mean_{wand} = 86.4\%$). The error bars are ± 1 SEM.

5. Acknowledgements

We wish to thank Morriah Horani for her helpful comments. This work was partially supported by a NSF ITR grant (CCR-0086065) and the Human Brain Project (NIDA and NIMH).

References

- [1] Biederman, I., Glass, A. L., and Stacy, E. W. Searching for objects in real-world scenes. *Journal of Experimental Psychology*, 97(1), 1973, pp. 22-27.
- [2] Boyce, S. J., Pollatsek, A., and Rayner, K. Effect of background information on object identification. *Journal of Experimental Psychology: Human Perception and Performance*, 15(3), 1989, pp. 556-566.
- [3] Cruz-Neira, C., Sandin, D. J., and DeFanti, T. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In *The Proceedings of ACM SIGGRAPH '93*, 1993, pp. 135-142.
- [4] Ware, C. and Franck, G. Evaluating stereo and motion cues for visualizing information nets in three dimensions. *ACM Transactions on Graphics*, 15(2), 1996, pp. 121-139.

Interactive Poster: VITA Visual Interface for Text Analysis

Zachary Jacobson,
Health Canada, Ottawa Canada
With
Ben Houston & Olivier Dagenais
Carleton University, Ottawa, Canada

Keywords: Text visualization; massive text corpora; semantic search engines; interactive demonstration

VITA, a "Visual Interface for Text Analysis", is a tool used to direct computer-based document searches. It allows a user, via mouse and keyboard action, to interact with search mechanisms - e.g. search engines on the Internet, such as Google and AltaVista - to present documents of potential interest and to arrange them visually. VITA has control features that allow visual clustering of like documents, thus enabling quick refinement of the search process. The visual features of VITA also support the observation and investigation of the [sometimes unexpected] relationships among documents.

VITA has been developed as a research testbed for identifying better methods of visualising relevant document clusters and identifying their relationships. As such, there have been various prototypes created as different ideas emerge. VITA is intended to be search mechanism independent. To bring in a new search engine, minimal programming is needed to designate the relevant HTML or XML fields in the code returned from the search engine for input to the interface.

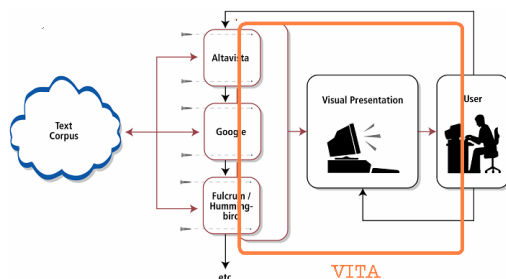


Figure 1

Figure 1 gives a conceptual view of where VITA fits in the scheme of things. The user operates

VITA as a 3D, VR, visually-oriented interface, to control the search parameters (e.g. keywords, concepts), search mechanisms and results presentation. If the task is to find specific documents, then VITA can be used recursively to refine the search.

A query can be broken down into a) the question, b) what the question means in terms of the search engine [i.e. the concepts involved], and c) the returned items [search results]. In VITA, these are represented on planes in the display. In this early VITA example, the question is not actually displayed but the concepts that are implied by the question are shown as objects on the top plane, the extracts from the resultant documents that are relevant to the concepts are on a second plane, and the documents themselves are on a third plane. The lines joining the objects indicate what concepts are involved with which extracts and which extracts are taken from what documents. The objects are constrained to move within their planes, but within their planes they move under the control of an "artificial gravity" clustering algorithm; thus an object moves under the influence of associated objects in the adjoining planes. Brushing an object, using a mouse, reveals the contents of the object.

In more recent versions of VITA, three planes are used to indicate the question, the concepts involved and the identified documents.

There have been a number of VITA prototypes created during the life of the project. The latest version is VITA- ϵ [epsilon]. Each version has been based on the experience gained with the preceding versions. Examples of the β [beta] and γ [gamma] versions are shown in the poster. The β version introduced the "Crown of Thorns"

display and the idea of the separate planes for the concepts and the search results. The γ version used the same principles but was based on a different search engine.

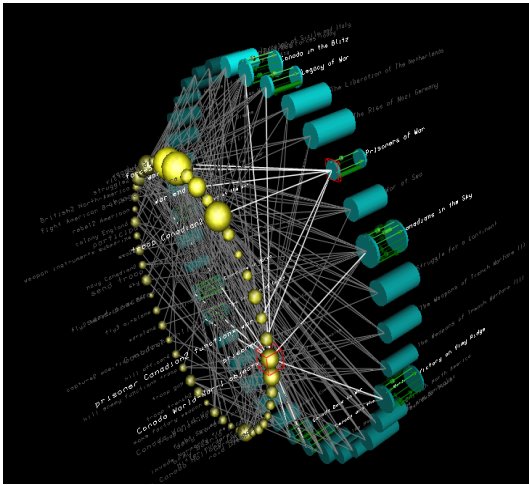


Figure 2

Figure 2 is a typical "Crown of Thorns" display produced by VITA- β . In this display, two planes are used, one for the concepts (gold balls) associated with the query, and one for the documents (blue cylinders) resulting from the search. The size of a gold ball relates to the number of documents that were found that included the particular concept. The red cubic frame surrounding a concept indicates a mouse click highlighting that concept. Information relating to that concept - what it is, what documents it pertains to and the actual references within the documents - are noted by highlighted text, lines and exploded cylinders

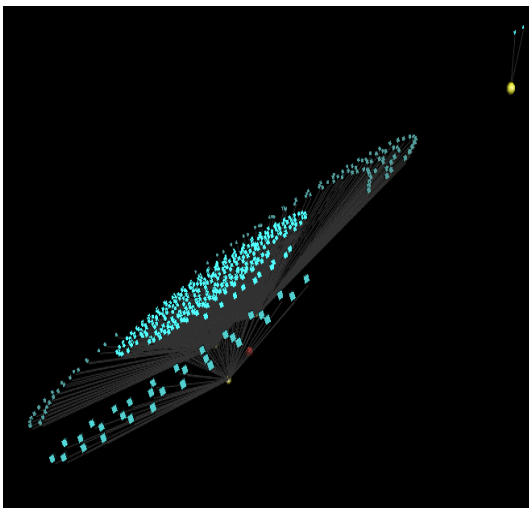


Figure 3

Figure 3 shows an example using VITA- γ , showing the results of a query regarding network intrusions. The query, shown as the small red cube, is linked to the related concepts [gold balls] which in turn are linked to the "hits" [blue cubes]. The artificial gravity clustering algorithm has been switched on by the user to try to identify related documents. The central hits are discursive network intrusion records; the other clumps are less related sources. The totally unconnected "solar system", moving out of the galaxy, is a relatively unrelated concepts and related hits. In this way the tool serves to make visible the structures inherent in large datasets.

Figure 4 shows a "God's-eye" view down through the presentation planes showing the instantaneous "artificial gravity" vectors associated with element clustering. Red shows instantaneous repulsion vectors while green shows instantaneous attraction. Note that the red and green vectors are about equal and opposite, as the presentation has stabilized.

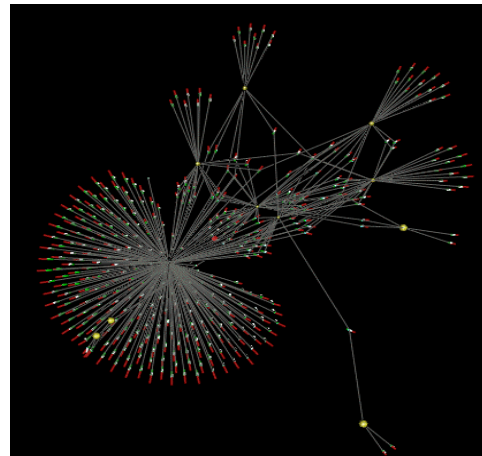


Figure 4

We call the two current versions, to be shown interactively, Δ and ϵ . The former is written in Visual Basic for "on the fly" modification in order to be able to try out new ideas quickly. Operationally it would be a bit slow when large result data sets are expected and it is not particularly robust. VITA- Δ has been interfaced to multiple search engines, such as Google and Hummingbird. In general, the level of effort required to write the "glue code" to interface other search engines to this version of VITA is considered to be minimal. VITA- ϵ 's functionalities are based on that of VITA- β & Δ . Written in C++, it is designed to be faster operationally and to be more robust than earlier versions.

Interactive Poster: Remote Exploration of Isosurfaces with Point-Based Non-PhotoRealistic Rendering

Guangfeng Ji, Han-Wei Shen and Jinzhu gao
Department of Computer and Information Science
The Ohio State University
Columbus, Ohio 43210
E-mail: {jig/hwshen/gao}@cis.ohio-state.edu

Abstract

We present a non-photorealistic rendering technique for visualizing remote isosurfaces. Instead of using conventional smooth shading techniques to render the surfaces, we propose a point-based system that can effectively illustrate the important features of the surface such as silhouettes and illuminations. With our technique, the communication cost between the server and the client is significantly reduced because only a small portion of the visible isosurface that contains the features is transmitted. In addition, we propose to use the discretized marching cubes algorithm to approximate the isosurface triangles so no actual triangle vertex or connectivity information needs to be transmitted. Our algorithm can adjust the size of network messages by changing the illumination threshold, so it is highly adaptable to the underlying network speed.

1 Introduction

Data sets from typical scientific applications are growing rapidly. Data volumes from 3D medical imaging like CT scans are approaching 512^3 or even 1024^3 . As data sets increase in size, it becomes less feasible to move the data to a local workstation for display. The low cost, high speed graphics hardware, although becoming very popular, are still incapable of processing data sets of very large size. As a result, high-end graphics workstations with direct access to the data storage are still heavily used for large scale data analysis.

To facilitate remote exploration of large scale data sets, numerous visualization algorithms have been proposed. Remote visualization typically involves visualization servers which are responsible for retrieving data and computing visualization results in the form of geometry or images. The results are transmitted to one or multiple clients for display.

In recent years, non-photorealistic rendering has drawn great attention from computer graphics researchers. Non-photorealistic rendering has its advantage over photorealistic rendering on that it facilitates the communication of complex shape information by abstracting visual features in a comprehensible and effective manner.

2 Remote Non-Photorealistic Rendering of Isosurfaces

Our work is motivated by the observation that in pen-and-ink or pencil drawing artwork, the artists rarely need to shade the entire object to convey its shape. Frequently, white space is used to depict regions that are highly illuminated, while darker tones are used to represent regions that receive less light. For instance, even though most of the space in the drawings in Figure 1 is left as white, we can still comprehend the shapes of the objects without any difficulties.

Based on the above observation, we devise a point-based non-photorealistic technique for remote visualization of isosurfaces. We assume the server has a direct access to the data set, and can compute isosurfaces at an interactive rate. Instead of sending the actual triangles that comprise the entire isosurfaces, the server computes and sends just enough information for the client to quickly produce images to convey the shape of the isosurfaces.

Our point-based non-photorealistic rendering conveys the shape of a surface using two important visual cues, silhouettes and shading. Silhouettes are to depict the outlines of the surface, while shading provides cues of surface orientations. Since edge-face adjacency information is not readily available for isosurfaces generated by the Marching Cubes algorithm [1], silhouette finding algorithms based on the idea of identifying edges shared by the front and back triangles are thus not suitable for interactive isosurface exploration. To solve the problem, we detect silhouettes by first computing the dot product of the vertex normal and the view vector at each of the isosurface triangle vertices. If a triangle contains vertices that have different signs in the dot product results, the triangle contains a silhouette. We then use linear interpolation of the dot product results to find the exact silhouette locations.

In our point based shading algorithm, points with different densities are used to convey different tones over a surface, where tone refers to the amount of visible light reflected toward the observer from a given area on the surface. We compute the tone by evaluating the Phong illumination model. To ensure a smooth transition of the tone over the surface, the following formulas are used:

$$E = ((T - Tone) \div T)^{Power} \quad (1)$$

$$NP = A \times E \quad (2)$$

Where T is the shading threshold. If the tone of a given area is greater than T , the area is considered as highly illuminated, therefore no points will be assigned and that area will appear as highlight in the final image. Tone, NP and A are the tone, number of points and the screen projection size of a given area, respectively. Power controls the variation of point density among areas with different tones. A typical image generated by our algorithm is shown in Figure 2.

The remote visualization system works as follows. The server is assumed to have a direct access to the data set, and is responsible for extracting all the visible isosurfaces [3]. Silhouettes and point shading information are generated from the visible isosurfaces based on the algorithm described above. The shading threshold T is determined based on the network bandwidth and the required frame rate. Note that if a triangle is highly illuminated and thus does not receive any points, the server can completely discard the triangle and not send any information about the triangle to the clients. To send the silhouette and shading information to the clients, instead of naively transmitting the precise shading point positions over the network,

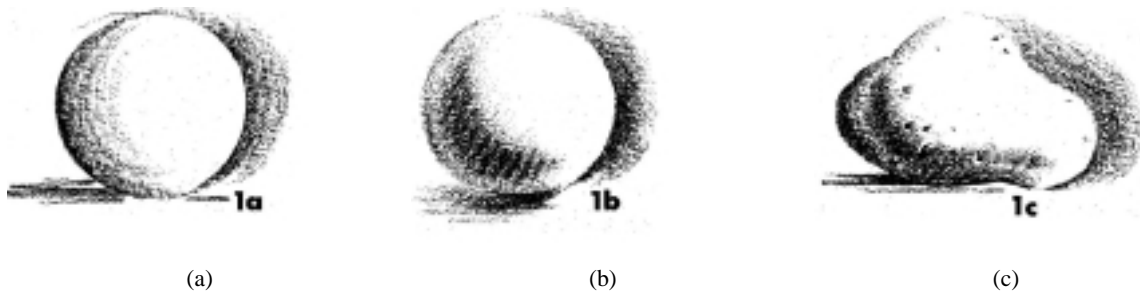


Figure 1: The examples show that even though most of the space in the drawings is left as white, we can still comprehend the shapes of the objects without any difficulty[4].

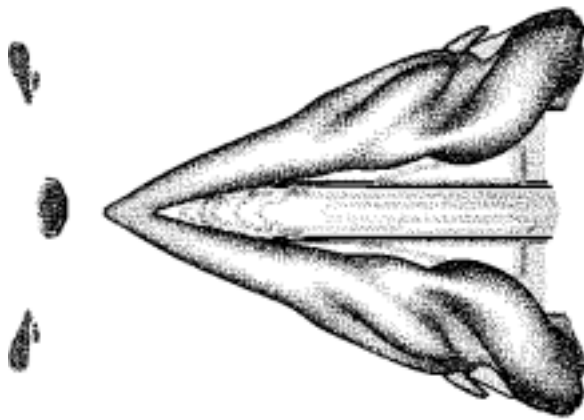


Figure 2: A delta wing image generated by our point-based NPR algorithm

which will incur very high communication overhead, we use the discretized marching cubes algorithm [2]. In discretized marching cubes algorithm, isosurface triangle vertices are always at the middle of two adjacent voxel corners. The position of the isosurface triangles inside a cell can be easily predicted if the marching cubes case number is known. Therefore, the shading information of a triangle can be encoded by the isosurface cell id, the marching cubes case number, and the total number of points that will be rendered within the cell. To send the silhouette positions, rather than representing a silhouette line as two end points which require 6 floating point numbers, we transmit the id of the cell that contains the silhouette, the marching cubes case number, and the silhouette triangle index. The clients can approximate the silhouette lines by placing a large number of points in the silhouette triangle. This way, the communication cost is further reduced.

3 Results and Discussion

Our remote visualization algorithm can reduce the communication cost by not sending any information for isosurfaces in the high-light area and utilizing discretized marching cubes algorithm to fur-

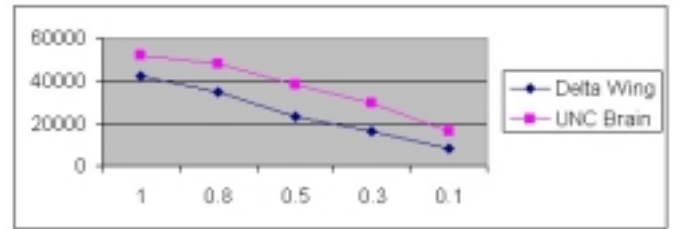


Figure 3: The effect of shading threshold on the message size (in bytes) using two datasets.

ther reduce the message size. The message reduction rate using our algorithm is approximately 40 times compared with the system sending the whole isosurface and approximately 10 times compared with the system sending only the visible portion of the isosurface.

Figure 3 shows the effect of different shading thresholds on the message size. As discussed previously, we use the shading threshold T to control the point distribution. When the shading threshold decreases, the area occupied by highly illuminated regions (white space) increases, and thus the message size decreases. The figure shows this trend when the shading threshold changes from 1.0 to 0.1. Our remote algorithm can adjust the size of the network messages by changing the illumination threshold at run time, so it is highly adaptable to the underlying network speed.

References

- [1] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, July 1987.
- [2] C. Montani, R. Scateni, and R. Scopigno. Discretized marching cubes. In *Proceedings of Visualization '94*, pages 281–287. IEEE Computer Society Press, Los Alamitos, CA, 1994.
- [3] Jinzhu Gao and Han-Wei Shen. Parallel view dependent isosurface extraction using multi-pass occlusion culling. In *ACM/IEEE Symposium on Parallel and Large Data Visualization and Graphics*, Oct 2001.
- [4] J. Hamm. Drawing scenery: Landscapes and seascapes. 1982.
- [5] D. Ebert and P. Rheingans. Volume illustration: Non-photorealistic rendering of volume models. In *Proceedings of Visualization '2000*, pages 1951–202. IEEE Computer Society Press, Los Alamitos, CA, 2000.

Interactive Poster: Showing Shape with Texture

Sunghye Kim, Haleh Hagh-Shenas and Victoria Interrante
University of Minnesota

Abstract

Studies have shown that observers' judgments of surface orientation and curvature are affected by the presence of surface texture pattern. However, the question of designing a texture pattern that does not hide the surface information nor conveys a misrepresentation of the surface remains unsolved. This poster describes experiments done in our lab to investigate the role that texture might play in helping or hindering users' abilities in perceiving and understanding the underlying shape of the objects being represented.

CR Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism — Texture.

Additional Keywords: Visual Perception, Visual Representation, Shape-from-Texture.

1. INTRODUCTION

If we could design the perfect texture pattern to apply to any smooth surface in order to enable observers to more accurately perceive the surface's shape, what would the characteristics of that texture pattern be? The answers to this question have important potential impact across a wide range of visualization applications, from molecular modeling to radiation therapy treatment planning, in which scientists need to attain an accurate, intuitive understanding of the shapes of complicated, smoothly curving surfaces in their data. Over the past several years, we have carried out a series of experiments intended to investigate the impact on shape perception of various characteristics of surface texture patterns. In this poster we report the results of our most recent study, in which we compare performance on a surface attitude probe adjustment task under three principal direction pattern conditions and a control condition in which no texture was present. The goal of this study was to investigate the potential benefits of using a texture pattern that follows both rather than just one of the principal directions across a surface.

2. PREVIOUS WORK

Prior to the work described in this poster we conducted two previous experiments, which provided the foundation for the current work. Our first experiment [2] was designed to examine the effect of the direction of luminance texture pattern anisotropy on the shape judgments of the observers.

Stimuli and Task: Complicated, smoothly curving level surfaces extracted from a volumetric dataset were used. Over this data we generated four different texture patterns via 3D line integral convolution. One of these patterns was isotropic and the other three were anisotropic. In one the texture flow followed a single uniform direction in the tangent plane to the surface to exhibit zero geodesic curvature. Under another condition, the flow followed a vector field that swirled over the surface, exhibiting non-zero geodesic curvature. In the third pattern the flow followed the direction of greatest normal curvature (first principal direction) at every point on the surface.

Observers were asked to manipulate an array of surface attitude probes, which was super imposed on top of the surface. The manipulation was done by the users until they were satisfied that the circular base of each probe appeared to lie in the tangent plane at the probe center and the perpendicular extension pointed in the direction of the normal to the surface at that point. The process was repeated with the same set of surfaces displayed in stereo.

Results: An analysis of the 3D angle errors of the probe settings confirmed the hypothesis that texture pattern anisotropy impedes surface shape perception when the elongated markings are oriented in a way that is different from the principal direction.

In a followup experiment [3] we again investigated the effects of texture orientation on surface shape judgements, but this time we used a more general texture synthesis method [1] and a four alternative forced choice task, in which we asked participants to identify the quadrant in which two simultaneously displayed B-Spline surfaces, illuminated from different random directions, differed in shape.

Results: We found that the accuracy of quadrant identification increased fairly consistently with increasing magnitude of the shape disparity but that the characteristics of this increase differed under the different texture orientation conditions. Participants were able to more reliably perceive smaller shape differences when the surfaces were textured with a pattern whose orientation followed one of the principal directions rather than when the surfaces were textured with a pattern that either gradually swirled in the surface or followed a constant uniform direction in the tangent plane.

3. CURRENT EXPERIMENT

From our previous experiments we concluded that the orientation characteristics of an anisotropic texture pattern do matter. In our latest experiment we tried to address other questions such as: what kind of principal direction texture is best for showing shape and can we characterize what helps in making people's judgments more accurate? In this experiment we used a modified surface attitude probe adjustment task similar to that used in our first experiment to evaluate the relative effectiveness of three different principal direction oriented texture patterns for facilitating accurate shape perception. We also added a control condition of no texture to make sure the texture is not getting in the way of perception. Specifically we were interested to find out whether shape perception is facilitated better using a texture pattern that follows both principal directions than with a texture that follows only one of the above directions. Additionally we were interested to examine the potential effects of other texture characteristics beside orientation. Thus we decided to run our experiment with two one-directional patterns: a set of continuous lines, and a singly oriented line integral convolution like pattern. All the patterns spanned an equivalent range of spatial frequencies and were nearly equal in mean luminance. We made height field surfaces using a base B-Spline surface whose control points we defined randomly and textured these surfaces using the new texture synthesis method [1] developed in our lab.

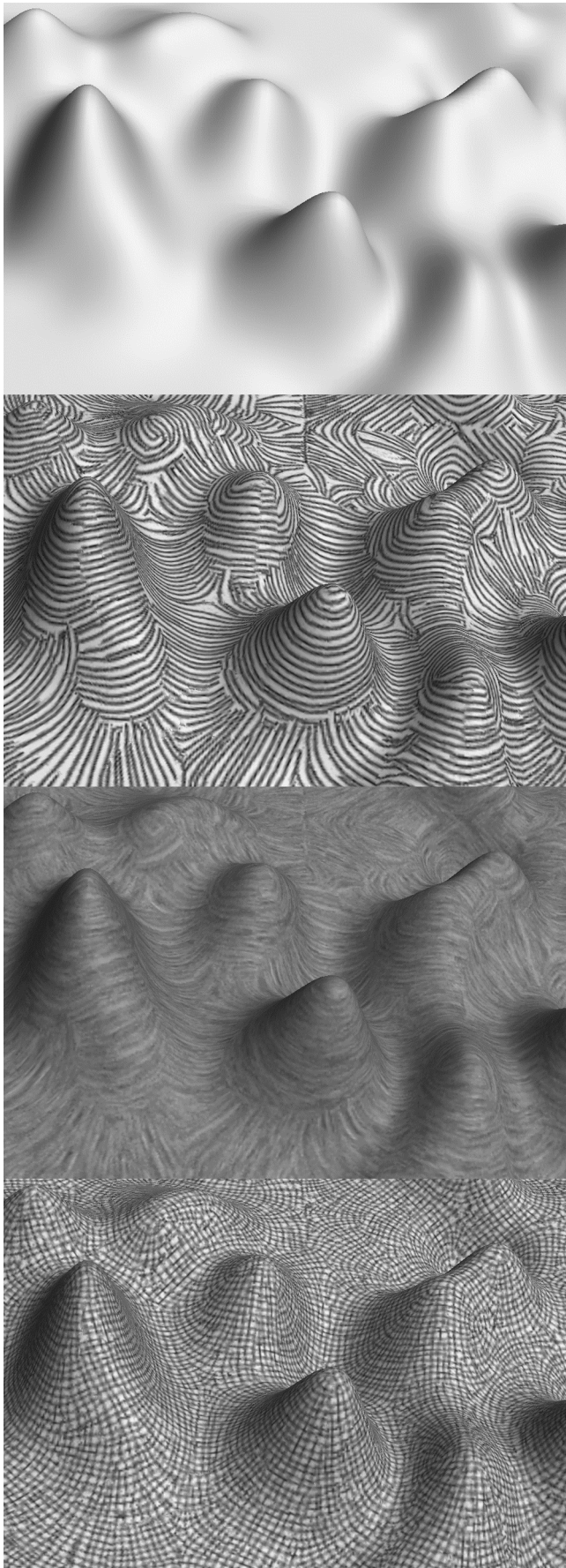
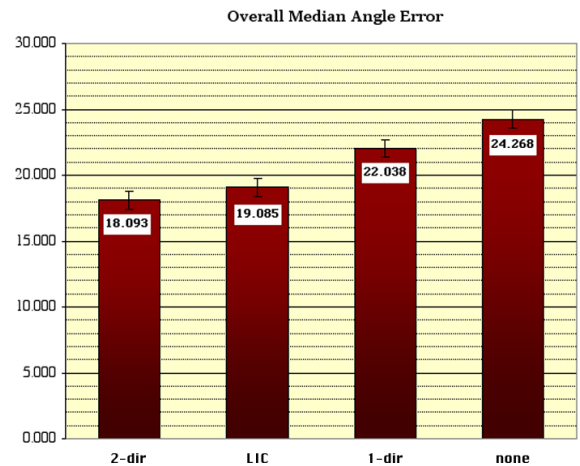


Figure 1: A test surface under each of the four texture conditions.

We also used a random process to predefine the set of fixed probe locations at which shape judgments were made. Both the surface and the probe were modeled in 3D and displayed in perspective projection, with hidden surface removal turned off before the probe was rendered. The experiment consisted of 200 trials per subject: 5 surfaces, 2 probe positions per surface, 3 texture types plus 1 no-texture condition and 5 repeated measures.

Task: The observers were asked to manipulate each probe normal until they were satisfied that it was aligned with the surface normal at that surface point. The viewing angle and lighting conditions were kept constant throughout all trials. We also designed a training procedure that each subject was required to pass before s/he could proceed to the actual experiment.

Results: We used the statistical software package “MacAnova” developed in the Department of Statistics at the University of Minnesota to perform a three-way, within subjects mixed analysis of variance of the results. We found significant main effects of probe location and texture type and a significant two-way interaction between texture types and probe locations. We used Tukey’s HSD method to perform post-hoc pairwise comparisons of the means of the angle errors under the different texture conditions. We found that the following differences were statistically significant at the 0.01 level: 2-dir < 1-dir, 2-dir < none, 1-dir < none, and lic < 1-dir and lic < none.



4. FUTURE WORK

In future experiments we plan to further investigate the interaction between the probe location and the magnitude of the error each observer makes. In addition we hope to apply our texturing method to broader categories of real datasets such as medical and molecular data and run experiments involving application specific tasks and employing domain experts as users.

References

- [1] Gabriele Gorla, Victoria Interrante and Guillermo Sapiro (2002). Texture Synthesis for 3D Shape Representation, IEEE Transactions on Visualization and Computer Graphics, to appear.
- [2] Victoria Interrante and Sunghee Kim (2001). Investigating the Effect of Texture Orientation on Shape Perception, Human Vision and Electronic Imaging VI, SPIE 4299, pp. 330-339.
- [3] Victoria Interrante, Sunghee Kim and Haleh-Hagh-Shenas (2002). Conveying 3D Shape with Texture: Recent Advances and Experimental Findings, Human Vision and Electronic Imaging VII, SPIE 4662, pp. 197-206.

Interactive Poster: 3D Visualization of MIDI Dataset

Reiko Miyazaki Issei Fujishiro

Graduate School of Humanities and Sciences, Ochanomizu University, Japan

reiko@imv.is.ocha.ac.jp; fuji@is.ocha.ac.jp

1. Background and Purpose

When editing and playing a MIDI dataset, users usually rely on so-called sequence software systems, which offer a variety of editwindows, such as *staff* and *piano-roll*, for different tasks. Since a limited number of related parameters appear in each of the editwindows, the users are forced to open multiple windows of small sizes side by side, and thus making it difficult for them to grasp the global musical structures embedded in a multi-channel sequence of note events. In this poster, we present a system, called comp-i (Comprehensible MIDI Player – Interactive), which provides a designated 3D virtual space, where the users are allowed to perform visual exploration of a given MIDI dataset in an immersive and intuitive manner. Although a few studies on 3D music visualization can be found in the literature (e.g., [1]; [2]), the existing systems offer restricted functionalities of overview and detailed analysis. We have carefully designed a spatial substrate and retinal properties for the comp-i virtual space, along with a rich set of operations according to Shneiderman's *Visual Information Seeking Mantra* [3].

2. System Overview

The comp-i system accepts a standard MIDI file (SMF), and selects from the file, three primary MIDI elements – namely, note-on, note-off and set-tempo, to construct a corresponding 3D virtual space. Note-on and note-off are fundamental MIDI messages to start and stop a single note sound, respectively, and include channel, pitch, and sound volume parameters. Set-tempo is a MIDI event that sets the tempo parameter.

Fig.1 shows a complete view of the comp-i virtual space, where multiple channel layers are stacked along the depth axis. Each layer contains point nodes for bar lines, and a series of cylinders each of which corresponds to a single note sound. For visibility and distinction, cylinders on different layers are assigned different semi-transparent colors. The three parameters pitch, volume, and tempo of a note sound are encoded as the height, diameter, and color saturation of the corresponding cylinder, respectively. As a given MIDI dataset is played through a 3D virtual sound device, a scan-plane orthogonal to the layers is moving from left to right to indicate the current position of the note sounds.

The comp-i system offers one or more operations for each of Shneiderman's four major information seeking task categories [3]. Indeed, the users can take advantage of perspective view to grasp the entire dataset within the virtual space. 3D illumination gives object shadows on the floor, and conveys the right information of MIDI object

geometry (*overview*), whereas the users can make their viewpoints as close to selected objects as they like (*zoom*). The users are also allowed to permute the channel layers of interest, control the visibility of retinal/auditory properties of the objects, and choose arbitrary positions to start/stop playing (*filter*). Furthermore, the scales and quantitative properties are possible to be displayed along with the channel layers and the scan-plane (*Details-on-Demand*) (Fig.2).

After the users comprehend the outline of a given MIDI dataset, they are allowed to alter the projection to orthographical for accurate editing work. The system provides two ways to edit the MIDI dataset: direct manipulation of objects for novice users and textbox-based specification for experts. Choice of particular projections of the virtual space supersedes typical GUIs of sequence software systems. For example, the top orthographical view can give a clear view of multi-channel information involving both volume and tempo (Fig.3 (a)-(a')). The side view of the virtual space makes it possible for the users to look over both pitch and volume of multiple channels simultaneously (Fig.3 (b)-(b')). These visualizations cannot be obtained easily through the traditional sequence software systems.

3. Towards Visualizing Musical Structure

We are still extending the comp-i system so that it can provide the users with a focus+context display mechanism to perform visual MIDI data mining. A circular form is a good candidate for a new spatial substrate to visualize the global structure of a long MIDI sequence effectively. In addition, a ConeTrees[4]-based multi-resolution technique is adopted to support interactive similarity search reflecting the hierarchical structure of music pieces (e.g., sentence, phrase, and motif) (Fig.4).

References

- [1] S.M. Smith, G.N. Williams, "A Visualization of Music", In *Proc. IEEE Visualization '97*, October 1997, pp.499-503.
- [2] H.G. Kaper, S. Tipei, "Manifold Compositions, Music Visualization, and Scientific Sonification in an Immersive Virtual-Reality Environment", In *Proc. International Computer Music Conference*, October 1998, pp.339-405.
- [3] B. Shneiderman, *Designing the User Interface Strategies for Effective Human-Computer Interaction*, Addison-Wesley, 3rd edition, 1998, Chapter15.
- [4] G.G. Robertson, et al., "ConeTrees: Animated 3D Visualizations Hierarchical Information", In *Proc. CHI'91, ACM Conference on Human Factors in Computing System*, April 1991, pp.189-194.

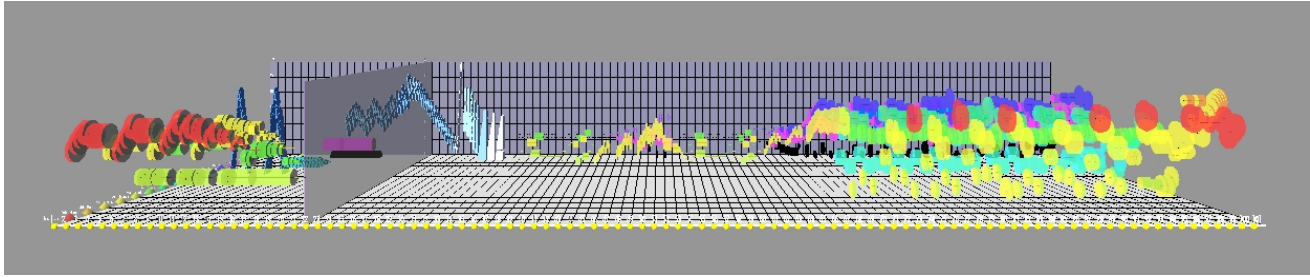


Figure 1: Overview of comp-i virtual space.

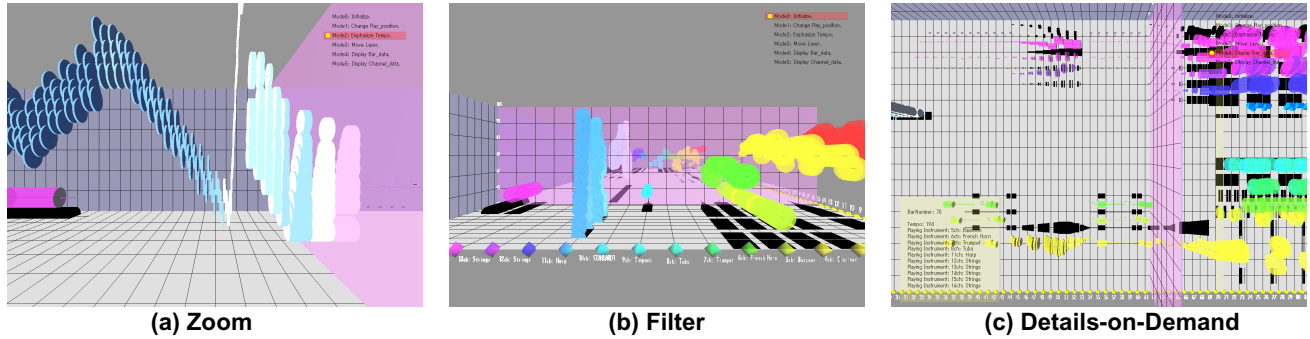


Figure 2: Information seeking task categories supported by comp-i.

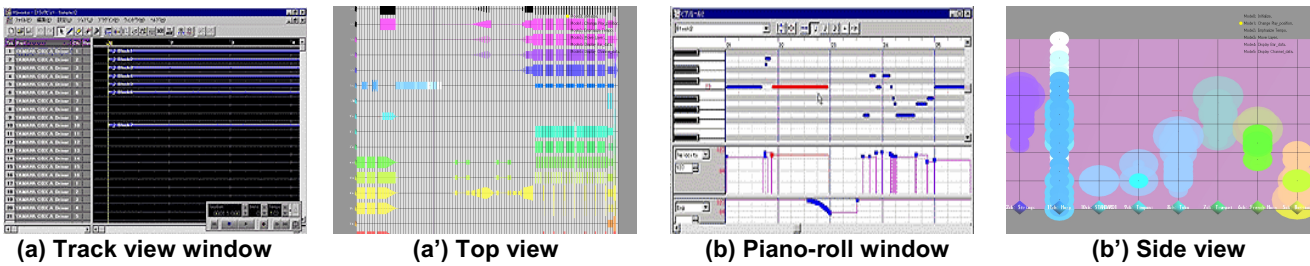


Figure 3: Choice of particular projection of the comp-i virtual space supersedes typical GUIs of a sequence software ((a)(b): sequence software, (a')(b)': comp-i).

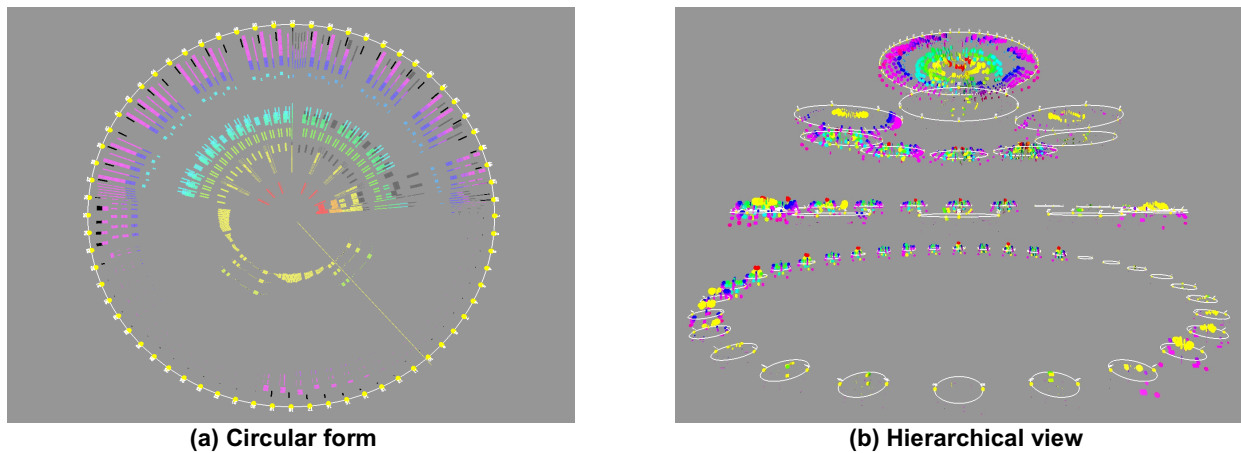


Figure 4: Exploiting musical structure within ConeTrees-based spatial substrate.

Interactive Poster: A Powerful Communication Framework For Distributed Virtual Reality

Bernd Reuther*, Achim Ebert*, Paul Müller*, Hans Hagen*

* Department of Computer Sciences, University of Kaiserslautern, Germany, {reuther|mueller}@informatik.uni-kl.de

* Intelligent Visualization and Simulation Systems, DFKI GmbH Kaiserslautern, Germany, {ebert|hagen}@dfki.de

KEY WORDS: Communication, distributed VR, virtual early prototyping.

1. INTRODUCTION

The objective of our project AIT-Vepop (Advanced Information Technology - Virtual Early Prototyping) is the demonstration of the impact of virtual reality technology on distributed design situations and work practices using standard hardware and software components combined with low cost solutions. Using distributed virtual reality provides new means for directive feedback and interactive evaluation of product data extracted from existing CAD/PDM company infrastructures. Additionally this facilitates the involvement of suppliers and customers ranging from the beginning of the product life cycle to modification studies and marketing. The underlying communication system has to fulfill two different tasks: session-on-line distribution of events (for distributed virtual design sessions) and session-off-line distribution of documents (to prepare design sessions and update model and document databases).

The communication framework for distributed VR presented in this paper provides a powerful but simple to use communication service for interactive data exchange within groups. It addresses communication connections in LAN as well as connecting different companies via WAN and firewall tunneling. Reliable group communication, distribution of interactive voice and video data, tunneling of IP multicasts, optional data encryption and user authentication complete the services of the communication framework, which have been implemented in pure Java.

This research is funded by the European Community under the "Information Society Technologies" program (IST-1999-13346) and is done in close collaboration with the involved project partners.

2. RELATED WORK

Several other developments, which can be distinguished by their service level, aim to support group communication. Some projects are focusing on development of protocols for group communication. These projects consider network problems like low bandwidth, high delay, temporary loss of connections and the implementation of reliable multicasts [1,2]. Furthermore, a few software tools were developed to support the implementation of the communication and management tasks for group communication (a well known example is JSDT [3]). Other developments are defining frameworks for the implementation of group communication applications. These

projects focus on how to develop or implement a distributed application rather than how to implement the communication itself [4].

3. SERVICES

The Virtual Reality Communication System (VRCS) presented in this paper provides services to manage groups and to perform communication tasks. The aim is to provide these services in an easy to use way and to consider limitations of WAN. The VRCS user does not have to deal with network connections or any protocol details. Nevertheless VRCS provides services for performant firewall tunneling and secure data transfer. Therefore VRCS is much easier to use than for example JSDT, but it is still more suitable for WAN communication.

3.1 Management

Each VRCS domain has a VRCS – Management Service (VRCS-MS), which is responsible for managing groups and authorization of users. Further, there are Group Servers which are responsible for managing the members of a specific group and the data distribution within that group. The Group Servers are launched and terminated by the VRCS-MS transparently when a group is created or destroyed. At any time a user may retrieve information about existing groups and their group members.

3.2 Data Transport

After a VRCS user has become a member of a group he is able to communicate with other members of this group. A channel concept is provided in order to distinguish different traffic flows within a group. A group member might send unicast messages or multicast messages (including or excluding himself). VRCS guarantees that multicasts are delivered in the same order to all group members. For each channel a user may specify to use secure/insecure and reliable/unreliable data transfer or to lock/unlock the channel¹. Furthermore, a request-respond mechanism enables synchronous communication between two VRCS users. Information about the current network performance enables applications to adapt to networks with low bandwidth.

3.3 IP Multicast Tunneling

The IP protocol already provides an efficient multicast service, which is used by some applications. For example the Mbone tools VIC and RAT [5] which can be used for videoconferencing. Unfortunately IP multicasts are usually limited to a LAN or even parts of a LAN. Therefore VRCS provides a mechanism to tunnel IP multicast traffic to other group members, whereby loops are avoided automatically.

¹ to reduce the incoming or outgoing amount of data

4. IMPLEMENTATION

4.1 Architecture

The VRCS is realized in three layers (see Figure 1), whereby the VRCS Services layer implements the services for users and provides a configuration interface. There is one component for each service: VRCS-MS, Group Server and Group Member. The Communication Manager realizes the distribution of messages based on identifiers and hides nearly all networking details to the service implementation. The Connection Manager is responsible for transporting data between processes, establishing and terminating logical connections. There will always be only one logical connection between two VRCS processes.

Each service component in the VRCS is uniquely identified by an identifier pair: a group identifier and a group member identifier. Such an identifier pair is used as an address for each component. The Communication Manager has knowledge of all identifiers of all local and remote components.



Figure 1: VRCS Architecture

The Connection Management distinguishes between logical connections which keep the communication status between two processes and network connections for the access of a certain protocol. Currently VRCS supports the protocols: TCP, UDP, SSL and HTTP.

4.2 Firewall Tunneling

Today most LANs are protected by firewalls to avoid that undesired traffic from the internet disturbs the LAN or the hosts in the LAN. This means that only well-known traffic from the Internet is allowed to enter a LAN. Since the proprietary protocol that is used within VRCS is not well-known, VRCS has to use firewall tunneling techniques whenever a firewall separates two VRCS components. This is typically done by using HTTP.

Unfortunately, HTTP does not achieve the same performance over long distances as TCP. Therefore VRCS uses several parallel HTTP connections to improve the performance. The number of parallel HTTP connections might be configured. A separate protocol on top of HTTP performs reordering of data and flow control so that several parallel connections behave the same way than one connection.

5. USING VRCS

VRCS provides powerful but easy to use services for group communication. A few lines of code are sufficient to utilize the services and techniques described in the previous chapters.

5.1 API

VRCS offers an object oriented API, including objects representing a User, the VRCS system, a Group, a GroupMember, Channels and component descriptions (see Figure 2). First, the user must successfully authenticate with the VRCS system in order to retrieve information about the system. An application may create a new group or get access to an existing group. Usually one will join a group by creating a new GroupMember. After that an application is able to communicate

with other group members. Further Channel objects may be created to simplify the communication tasks.

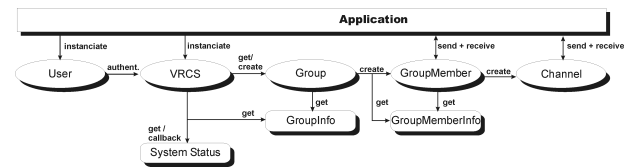


Figure 2: Object oriented API of VRCS

5.2 Configuration

Before VRCS can be used some configuration data must be provided, for example the address of the VRCS-MS and some information about the firewall. These and other parameters must be configured only once. This will usually be done by an administrator. A configuration wizard is available which simplifies and automates the configuration tasks.

5.3 Application Example

An application for virtual early prototyping using VRCS has been developed within the AIT-VEPOP project. This application enables a distributed group of persons to interactively modify the visualized geometry data, exchange documents, see each other and talk to each other.

6. CONCLUSION

In this paper we have presented a communication framework for distributed VR that provides a powerful but simple to use communication service for data exchange within groups for interactive applications in LAN and WAN (including firewall tunneling). Remarkable features are reliable group communication, distribution of interactive voice and video data, tunneling of IP multicasts, optional data encryption and user authentication, as well as pure Java implementation.

REFERENCES

- [1] L.E. Moser, P.M. Melliar-Smith, D.A. Agarwal, R.K. Budhia, C.A. Lingley-Papadopoulos. Totem: A Fault-Tolerant Multicast Group Communication System. Communications of the ACM, vol.39, no.4, 1996.
- [2] Y. Amir; J. Stanton. The Spread Wide Area Group Communication System. Technical Report CNDS-98-4 Johns Hopkins University.
- [3] <http://java.sun.com/products/java-media/jstd>
- [4] I. Marsic. DISCIPLINE: A Framework for Multimodal Collaboration in Heterogeneous Environments. ACM Computing Surveys, Vol. 31, No. 2es, Article No. 4, June 1999.
- [5] <http://www-itg.lbl.gov/mbone>

Interactive Poster: A Software Framework for Medical Visualization

Bartosz von Rymon-Lipinski, Thomas Jansen, Nils Hanssen, Marc Lievin, Erwin Keeve
Surgical Systems Laboratory, research center caesar
{lipinski, jansen, hanssen, lievin, keeve}@caesar.de

Introduction

Image-guided surgery becomes more and more indispensable these days. The vision of reality is enhanced, using information from CT, MR and other medical scans to help the surgeon in his every-day work. Several software products are available, such as specialized non-profit systems (e.g. [1]) or commercial applications (e.g. [2]). Our goal is to develop a general software framework for a complete medical processing-pipeline, including data-processing, visualization and navigation in pre-operative planning and intra-operative guidance. It should be possible to use this framework as a test-platform for research, providing a higher-level visualization infra-structure with extendible interfaces for easy implementation of new algorithms and rendering techniques. Our system has to support also easy integration of third-party software libraries for medical processing and rendering (e.g. ITK and VTK) [3, 4]. To reach these goals, we have developed the cross-platform software framework Julius, which is described in this abstract.

1. Software architecture

The general framework architecture of *Julius*, including the graphical user interface (GUI), was presented in [5]. In this abstract we will focus on the improved support for medical visualization (see figure 1).

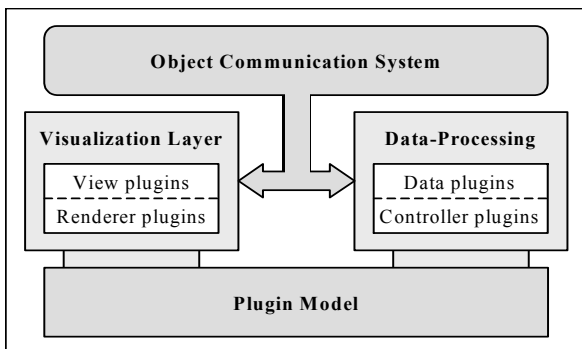


Figure 1. Overview of the visualization infra-structure of Julius. The plugin model is the basis for component implementation, in particular for data-processing and the visualization layer; the communication system manages the co-operation between both.

The main part of Julius is the object-oriented *JCore* library, based on the multi-platform application toolkit Qt [6]. The *JCore* C++ classes implement the following features:

The *plugin model* makes it possible to connect new software components to the framework at run-time, without recompilation of the source code (see [7]). Available plugin versions can be easily activated or deactivated, using the plugin configurator tool of the GUI, which facilitates the testing of new implementations.

Our *Julius object communication system* (JOCS) features message-based information exchange between distributed objects in the framework. The sender can attach data to each message and select between different addressing modes to minimize the number of potential listeners. Therewith, developers have the possibility to implement efficient update schemes for rendering of dynamic scenes. The JOCS features also posting of messages for safe communication in a multi-threaded environment. Extending the communication system is simple, because messages are represented by objects, derived from standard Julius classes.

The *data-processing* is a fundamental part of the *JCore* library. Our concept is based on a separation between data representation and operation. Information for the medical processing pipeline is stored in plugins, derived from the *data plugin* base class. Examples are volumes, polygonal meshes and abstract objects like anatomical properties, finite element nets etc. Different data plugins can be linked together to build a complex and modular hierarchy, like a 3D scene graph. Operations are implemented by the so-called *controller plugins*. Examples are segmentation, registration and mesh generation. Each time a controller creates or modifies data it broadcasts automatically a data modification message via the JOCS. The attachment of this messages contains bit-flags, identifying the portion of the modified data plugin. Other objects that receive this message can update efficiently their internal state, like a hardware-accelerated volume renderer (e.g. on SGI graphics hardware), which does not need to upload the whole voxel data to texture memory, if just the associated color mapping has been modified.

Another important part of the core library is the *Julius visualization layer (JVL)*. The JVL is responsible for the visualization of results from the data-processing, introducing two plugin types: The *view plugin* is linked to a Qt widget in the GUI and encapsulates the context for a rendering library (like OpenGL, VTK). It manages a list of *renderer plugins*, whereas each one is responsible for the visualization of a data plugin in the current rendering context. Different but compatible renderers can be combined dynamically in the same view, resulting in a maximum of configurability and flexibility, e.g. for rendering of a volume together with an overlaid label volume and 3D cutting planes. A view plugin can be set to the so-called active mode. On each incoming data modification message, it updates automatically the rendering or adds new compatible renderers for the modified data. Such a mechanism is useful especially for displaying updates during interaction and interim results from a long data-processing pipeline.

2. Software components

In the following part of the abstract we will present some examples of software components for medical visualization that we have integrated into the Julius framework.

One important component is the surface view plugin. It supports OpenGL and VTK-based renderer plugins for polygonal data. This includes a standard renderer of VTK poly data objects, a renderer for 3D cutting planes and a volume renderer, based on hardware accelerated 3D texture mapping. Primarily, we use this view plugin (with various constellations of renderer plugins) for execution of long controller sequences, as required during complex medical procedures. The co-operation between the Julius data-processing and the visualization layer provides visual feedback and interfaces for data manipulation in each step of the sequence.

Another plugin is the volume view. This component works with a VTK-based volume ray casting renderer plugin. It produces high-quality images, without focusing on real-time feedback and interactivity. Both, the surface and the volume view, can be linked together by setting the same input data for their renderer plugins: transformations and deformations in one view will cause automatic update in the other view.

Finally, the slice view plugin is fundamental for medical applications. It supports OpenGL and VTK-based renderer plugins showing a 2D cut through a volume or polygonal mesh. The slice is represented by a data plugin that can be shared among different views. So,

moving the cut, e.g. via a slider in the GUI, results in a synchronized 2D and 3D display. Additionally, the slice view offers interfaces to connect blending functions between multiple slice images. These feature is useful particularly for semi-automatic registration. It is implemented, using our plugin model, which allows the extension with user-defined functions.

3. Conclusions and future work

With Julius we have developed a general and powerful software framework for medical visualization. The crucial elements of this framework are the cross-platform portability, modularity and application driven configurability. The plugin model, including the data-processing concept and the visualization layer, make Julius easily extendible with new algorithms and rendering techniques. Our communication system is used to coordinate efficiently medical processing and the rendering components, especially during interaction and navigation. Therefore, using Julius as a platform for rapid application development benefits to the end-user and the scientific developer. Our framework is in daily-use and acts as a base for other research.

We plan to integrate additional rendering libraries, including scene-graph support (e.g. via OpenSG) [8]. The JCore library will provide interfaces for distributed communication and visualization. Julius becomes public available in September this year.

4. References

- [1] D.T. Gering, *A System for Surgical Planning and Guidance using Image Fusion and Interventional MR*, Master's Thesis, MIT, 1999
- [2] Analyze Software System from Mayo Clinic, <http://www.mayo.edu/bir/Software/Software.html>, 2002
- [3] Insight Segmentation and Registration Toolkit, <http://www.itk.org>, 2002
- [4] W. Schroeder, K. Martin and B. Lorensen, *The Visualization Toolkit – 2nd Edition*, Prentice Hall PTR, New Jersey, 1998
- [5] E. Keeve et al., "Julius – An Extendable Software Framework for Surgical Planning and Image-Guided Navigation", MICCAI, Utrecht, October 14-17, 2001
- [6] K. Dalheimer, *Programming with Qt (2nd Edition)*, O'Reilly, 2002
- [7] C. Szyperski, "Component Software: Beyond Object-Oriented Programming", Addison-Wesley, 1998
- [8] D. Reiners, G. Voss and J. Behr, "OpenSG – Basic Concepts", OpenSG Symposium, Darmstadt, January 29, 2002

Interactive Poster: Effects of Relative Particle Sizing and Bin Consolidation Enhancements to a 5D Volume Visualization Algorithm

Rhonda J. Vickery[†]
Mississippi State University
rvickery@erc.msstate.edu

Joerg Meyer[§]
University of California, Irvine
jmeyer@uci.edu

Joel P. Martin[†]
Mississippi State University
jmartin@erc.msstate.edu

Timothy R. Keen[‡]
Naval Research Laboratory
keen@hurricane.nrlssc.navy.mil

Randy J. Brou[†]
Mississippi State University
rjb2@erc.msstate.edu

Stephanie M. Doane[†]
Mississippi State University
sdoane@erc.msstate.edu

Robert J. Moorhead[†]
Mississippi State University
rjm@erc.msstate.edu

Ashley M. Noble[†]
Mississippi State University
amr@erc.msstate.edu

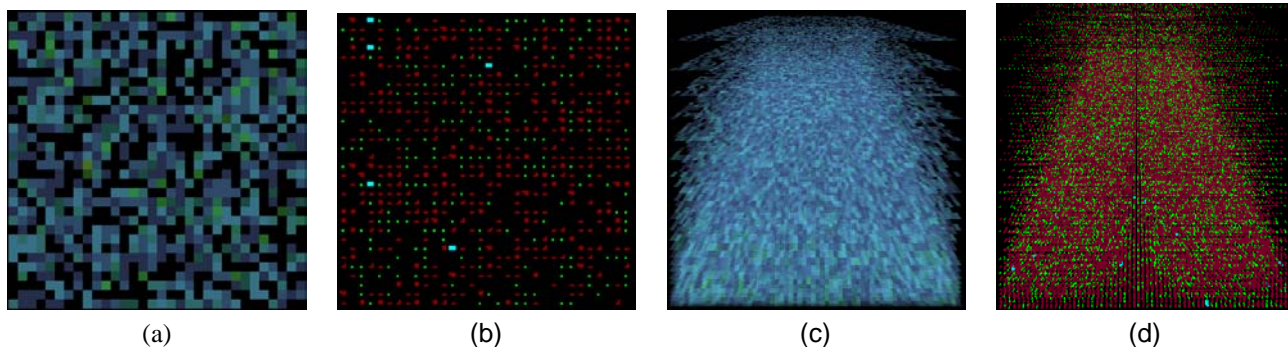


Figure 1: Difference in color separation and particle size is shown in a flat single bottom layer texture synthesized by the original (a) and enhanced (b) algorithms for the hill location. (c) and (d) show this information as stacked volumetric columns.

ABSTRACT

This work presents the results of relative particle sizing and bin consolidation enhancements to a 5D volume visualization algorithm for a coupled bottom boundary layer-sedimentation model.

1 BACKGROUND

We enhanced our existing hybrid immersive visualization system for a five dimensional (5D) coupled bottom boundary layer-sedimentation model [1, 4]. This model predicts sediment resuspension, transport, and resulting distributions for shallow water regions on continental shelves. The system is optimized for deployment in a CAVE-like environment. One variable of interest, suspended sediment concentration (SSC), is 5D and varies by longitude, latitude, depth, time, and grain size. At each grid point there are twenty values for SSC in units of Kg/m^3 , representing grain size categories (or bins) ranging from 2.36 to 3306 μm .

*<http://www.erc.msstate.edu/~rvickery/research/pubs/>

[†]Engineering Research Center, PO Box 9627, Mississippi State University, MS 39762

[‡]Oceanography Division, Stennis Space Center, MS 39529

[§]Department of Electrical and Computer Engineering, 644 Engineering Tower, Irvine, CA 92697

Previously we had implemented a volume visualization algorithm which converted the twenty SSC scalar values at each grid point from Kg/m^3 to numbers of particles of each grain size [4]. A 2D texture mapping scheme was employed to display the particles where a texture for each cell layer was synthesized to show the relative numbers of particles as same size pixels using a different color for each bin (twenty colors in all). This method suffered from two major shortcomings: particles of vastly different sizes were displayed as the same size, and there were too many colors to distinctly distinguish between particle bins.

We incorporated a commonly used sedimentation classification guide [2] that allowed us to consolidate the bins from twenty to fourteen, and further classify the bins into general sedimentation types: clay, silt, sand, and gravel. Colors were assigned based on these types, requiring only four colors. A log_2 particle sizing scheme was also implemented such that particles from the different grain size bins are represented as varying size particles. This was accomplished by using a 4×4 megapixel as the basic particle footprint and coloring from one to fourteen pixels to show the relative log_2 sizing of particles from finest, clay, to coarsest, gravel. For our 200 m horizontal resolution model data depicting normal weather conditions for Oceanside, California, the larger gravel sedimentation types were not present, which further reduced the colors to three. A color scheme (including opacity) was chosen such that the numerous smaller clay particles, as well as the fewer larger sand particles, could be seen amongst the most numerous mid-range

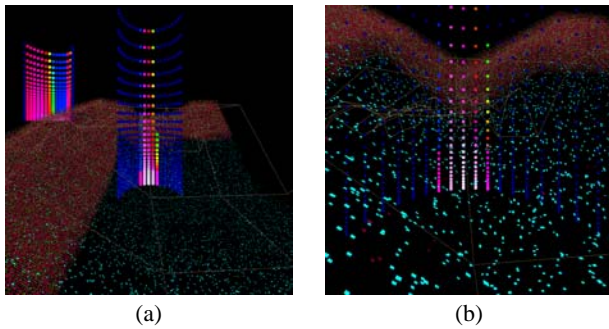


Figure 2: Twelve layer volume visualization using the enhanced algorithm with point detail profiles for both locations (a), and closeup of hole location (b).

silt particles. The resulting breakdown of bins included: clay as 1 green pixel, silt as 2-5 brick red pixels, sand as 6-10 cyan pixels, and gravel as 11-14 dark blue pixels.

2 RESULTS

Snapshot images were taken from the CAVE simulator execution of the enhanced algorithm comparing shallow water columns at two grid locations: one on a hill with a maximum number of smaller particles, and the other in a hole with fewer, larger particles (see Figure 1). These were compared with similar ones taken during execution of the original algorithm.

Differences in color separation and particle size can be seen in the flat single layer texture synthesized by the original (Figure 1a) and the enhanced (Figure 1b) algorithms for the hill location. The relative number of particles in each of the three categories (clay, silt, and sand) is more clearly indicated in the texture for the enhanced algorithm.

A flattened representation of 31 layers of particles for the hill grid location (see Figure 1) shows the volumetric capabilities of the algorithm. Since the model incorporates a sigma coordinate grid [4], layers are densely packed close to the ocean bottom, where sedimentation erosion and deposition are of particular interest. Comparison between the original (Figure 1c) and enhanced (Figure 1d) algorithms indicates more of a distinction between sedimentation types and relative particle sizes.

Figure 2 depicts twelve layers of a volumetric area that includes both the hill and hole locations and shows the overall effect of the enhanced algorithm. This can be compared with the alternate SSC point detail profiles illustrated in Figure 3. The volume visualization clearly shows the difference between a grid location where many smaller particles result in very low sediment concentration (hill), and one where fewer larger particles make up a much larger sediment concentration (hole). This is contrary to what one might think from just looking at the detail profiles. This is due to the differences in the grain sizes (diameters of an assumed spherical particle) between bins, and the part it plays in the mass per volume computation for sediment concentration (Kg/m^3) [4]. Often the smaller particles of suspended sediment are stripped away when a sudden deepening of the water occurs. This is a common effect known as armoring [1].

3 CONCLUSIONS AND FUTURE WORK

The enhancements added to the basic volume visualization algorithm for multiple scalar values of the 5D sediment concentration

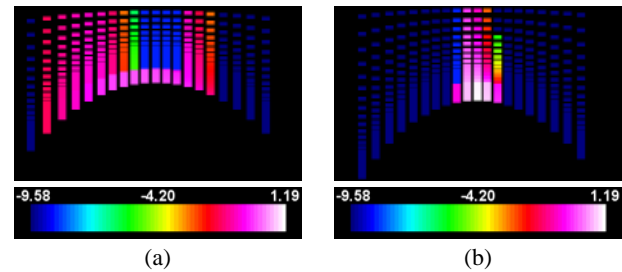


Figure 3: SSC Point detail view with a \log_{10} colormap legend for hill location (a), and hole location (b).

variable provide a clear indication of the relative numbers of particles present at different grid locations from four categories of sedimentation. Current methods for visualizing sedimentation processes cannot provide this same level of information in an integrated display [1]. This method incorporates bin consolidation, relative grain sizing, and distinct color selection to provide an alternate means of visualizing suspended sediment. Combined with a point sediment concentration profile at selected locations, additional information is available that allows the novice user to understand the physical processes involved.

Future work includes a comparison of the volume visualization algorithm with other conventional point-based techniques to see if this method can display more information, especially in the densely packed layers close to the ocean bottom, where dynamic boundary layer processes are of particular interest. The volume-based method will also be combined with boundary layer surface information depicting erosion and deposition processes over time. These combined visualizations will be evaluated to determine their effectiveness in providing more information to the novice user.

This volume visualization method can be applied to other problem domains. Future publications will describe its use in the visualization of dust aerosol transport in global circulation models [3].

4 ACKNOWLEDGMENTS

Special thanks to Derek Irby for his assistance in color selection.

REFERENCES

- [1] Timothy R. Keen, Rhonda J. Vickery, Peter Flynn, Robert H. Stavn, and Walton McBride. Scientific visualization of sediment dynamics in the bottom boundary layer. In *7th Int. Conf. Estuarine and Coastal Model., Proc.*, St. Pete Beach, Florida, November 2001.
- [2] H. O. Pfannkuch and Rick Paulson. *Grain Size Distribution and Hydraulic Properties*. <http://www.cs.pdx.edu/ian/geology2.5.html>.
- [3] Ina Tegen and Andrew A. Lacis. Modeling of particle size distribution and its influence on the radiative properties of mineral dust aerosol. *Journal of Geophysical Research*, 101(D14):19237–19244, August 27 1996.
- [4] Rhonda J. Vickery, Timothy R. Keen, Robert J. Moorhead, Randy J. Brou, Daniel W. Carruth, and Stephanie M. Doane. Volume visualization of 5D sedimentation models. In R. F. Erbacher, P. C. Chen, M. Groehn, J. C. Roberts, and C. M. Wittenbrink, editors, *Visualization and Data Analysis 2002*, volume 4665 of *Proceedings of SPIE*, pages 165–176, San Jose, CA, January 2002.

Interactive Poster: Application of DTI Visualization and Analysis on a Data Set from a Brain Tumor Patient

Song Zhang

David H. Laidlaw*

Mark E. Bastin

Saurabh Sinha†

Thomas S. Deisboeck‡

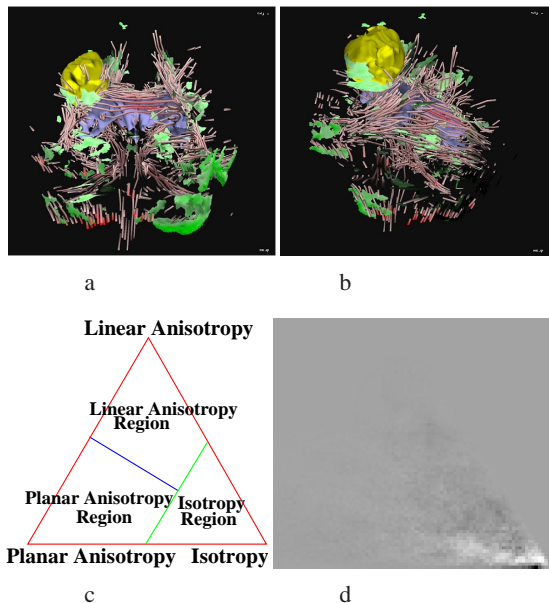


Figure 1: (a,b) Geometric models for visualizing the diffusion anisotropy of water in the brain. Streamtubes and streamsurfaces visualize the diffusion information in linear and planar anisotropic regions respectively. The blue and yellow shapes are isosurfaces of the ventricles and the peritumoral edema. (c) A division of barycentric space into three kinds of anisotropy regions. We generate anisotropy histograms over the barycentric space, and use the division for comparison. (d) The difference of anisotropy histograms over tumor and non-tumor sides of the brain. The brighter area reflects more tumor bearing side data points. The dark region on the top right part of the triangle implies a decrease of linear diffusion anisotropy on the tumor side of the brain. The bright region on the bottom right part implies an increase of planar diffusion anisotropy on the same side. Most of the data points are around the isotropy corner, so we zoom in the picture of the histogram toward that corner.

We demonstrate that 3D reconstruction and visualization can help uncover intriguing features and patterns in a diffusion tensor MR imaging (DTI) data set of a patient suffering from a metastatic brain tumor. We found that the area surrounding both the tumor and its edema is characterized by marked heterogeneity in both linear and planar diffusion anisotropy of water. In fact, similar heterogeneity can be detected throughout the entire brain. Displaying this qualitative diffusion anisotropy information using our 3D immersive virtual reality platform both allow the observer a flexible spatial orientation and enable useful interactive combinations with conventional 2D MRI-sections [1]. Furthermore, by defining the

linear and planar diffusion anisotropy regions on the anisotropy domain we were able to show quantitatively that there is an increase in planar anisotropy as well as a decrease in linear anisotropy on the tumor bearing side of the brain.

1 Methods

In December 2001, the 72-year old male patient presented with left facial seizures, left facial weakness, and slurred speech. CT as well as structural and diffusion tensor MR imaging was performed, and the patient was started on corticosteroid therapy (dexamethasone, 16 mg/day). A craniotomy with excision of the brain tumor followed. Histological evaluation confirmed a metastasis from an adenocarcinoma of the lung.

We visualized the DTI data set by generating geometric models that correlate with the underlying structures of the biological tissues. First, we distinguished between regions of linear and planar diffusion anisotropy of water in the DTI data set, and then we employed streamtubes (red) and streamsurfaces (green), respectively, to visualize these two types of anisotropic diffusion (see Figure 1 (a,b)).

The Cave is an $8 \times 8 \times 8$ foot cube with rear-projected front and side walls and a front projected floor. The graphics system can display the virtual environment so that it appears stationary to the user moving through it. We display geometric models of streamtubes, streamsurfaces, ventricles, and the peritumoral edema in the Cave. Two-dimensional T2-weighted MR sections are also displayed to facilitate the anatomical orientation for the user and their current section and axis can be chosen interactively using a viewing wand.

Streamtubes and streamsurfaces visualize diffusion anisotropy qualitatively. However, we also wanted to measure the diffusion anisotropy quantitatively. In Westin's metric [2], diffusion anisotropy is classified as linear anisotropy (cl), planar anisotropy (cp), or isotropy (cs). The three diffusion anisotropy metrics are normalized so that $cl + cp + cs = 1$. Thus, a two-dimensional barycentric space can be used for the domain of all the diffusion anisotropies, where the three anisotropy metrics are used for the barycentric coordinates [3]. We calculated the histogram of a DTI (or a region of a DTI) over this barycentric space while using the T2-weighted image to filter out the data points that were located outside the brain. Figure 1 (c,d) shows the definition of a barycentric space and the difference between the histograms of tumor side and non-tumor side of the brain.

2 Results

Likely having originated at the gray-white matter junction of the brain, the growing metastasis inevitably impacts the structures in the white matter of the brain, thus affecting the anisotropy of water measured by DTI. Comparing both hemispheres, the following regions of linear and planar diffusion anisotropy can be qualitatively distinguished:

1. The T2 peritumoral edema volume is to a certain extent surrounded by planar diffusion anisotropy, which is not present at the

*Brown University; {sz,dhl}@cs.brown.edu

†University of Edinburgh; {meh,SS}@skull.dcn.ed.ac.uk

‡Massachusetts Institute of Technology and Massachusetts General Hospital, Harvard Medical School; deisboeck@helix.mgh.harvard.edu

same site on the contralateral, left side. The lack of planar diffusion anisotropy towards the skull corresponds with the proximity of the tumor to the surface of the brain. The medial planar diffusion anisotropy along the longitudinal tracts has a different angle as compared to the contralateral, healthy hemisphere. The planar diffusion anisotropy at the level of the posterior horn of the ventricle, ipsilateral to the tumor, is markedly reduced.

2. Indicated by an increase in red coloring, it appears that there is an increase in linear diffusion anisotropy in some distinct paths, e.g. across the corpus callosum towards the contralateral hemisphere (Figure 1 (a)). A similar phenomenon can be found ipsilateral to the tumor, along the longitudinal tracts posteriorly (Figure 1 (b)).

3. There is a noticeable loss of planar diffusion anisotropy around the cerebellum, ipsilateral to the tumor. Conversely, it appears that there is an increase in linear diffusion anisotropy within the area of the cerebellum, primarily ipsilateral to the tumor, towards caudal (Figure 1 (a)).

The quantitative results of the diffusion anisotropy histogram are listed in Table 1. The expansive tumor and its increasing edema volume, respectively, reduce the total linear diffusion anisotropy ipsilateral to the tumor, and increase the total planar diffusion anisotropy on the tumor side of the brain.

	Left	Right(w/metastasis)
<i>linear</i> anisotropy voxels	36,141	26,667
<i>planar</i> anisotropy voxels	63,317	68,677
% of <i>linear</i> anisotropy voxels	17.1%	12.2%
% of <i>planar</i> anisotropy voxels	30.0%	31.5%

Table 1: The results of the diffusion anisotropy histogram are listed for the left as well as for the tumor bearing right side of the brain. We place the green line in Figure 1 (a) at the position that has isotropy value 0.9.

3 Acknowledgments

This work was supported in part by grant CA69246 from the National Institutes of Health, by the Human Brain Project (NIDA and NIMH), by grants from the National Science Foundation (CCR-0086065, CCR-0093238, and EIA-9724347) and by funds from the Cunningham Trust & William Ramsay Henderson Trust Scholarship (S.S). Imaging was carried out at the SHEFC Brain Imaging Centre, University of Edinburgh, UK. The authors would like to thank especially Dr. E. Antonio Chiocca (MGH-Neurosurgical Service, Harvard Medical School) and Professor Ian Whittle (Department of Clinical Neurosciences, Western General Hospital, University of Edinburgh, UK) for inspiring discussions. Thanks also go to Çağatay Demiralp for help develop the CAVE application.

References

- [1] Zhang S., Demiralp C., Keefe D.F., DaSilva M.J., Laidlaw D. H., Greenberg B.D., Basser P.J., Pierpaoli C., Chiocca E.A., and Deisboeck T.S. "An immersive virtual environment for DT-MRI volume visualization applications: a case study." In *IEEE Visualization*, 2001.
- [2] Westin C.F., Peled S., Gubjartsson H., Kikinis R., and Jolesz F.A. "Geometrical diffusion measures for MRI from tensor basis analysis." In *Proceedings of ISMRM*, 1997.
- [3] Kindlmann G.L., and Weinstein D.M. "Hue-balls and lit-tensors for direct volume rendering of diffusion tensor fields." In *IEEE Visualization 99*, 183-190, 1999.

Interactive Poster: Virtual Preoperative Laparoscopic Camera - Visualization and Modeling System

Leonid Zhukov, Igor Guskov[†], Sergey Klibanov[†], Dmitry Oleynikov[‡]

Department of Computer Science, California Institute of Technology, Pasadena, CA

[†]Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI

[‡]Univeristy of Nebraska Medical Center, Omaha, NE

CR Categories: I.3.8 [Computing Methodologies]: Computer Graphics—Applications, I.6 [Computing Methodologies]: Simulation and Modeling, J.3 [Computer Application]: Life and Medical Sciences;

Keywords: Surgical simulations, virtual camera, deformable models

ABSTRACT

Laparoscopy is a minimally invasive procedure which has been an invaluable tool for diagnosing abdominal pathology in patients [1]. However, it still requires the insertion of surgical instruments in to the abdominal cavity and subsequently carries the morbidity of pain and risk of bleeding and infection. Recent advances in CT and MRI technology have allowed virtual examination of the human body without the need for invasive procedures. Virtual Laparoscopy would allow similar diagnostic capability as conventional laparoscopy but without the morbidity associated with surgical procedures [2]. In this paper we developed a virtual laparoscopic camera, based on a *patient specific* CT data, which can be used in the preoperative planning as well as an advanced surgical teaching tool.

We started from the typical clinical abdominal CT data consisted of 56 slices with resolution 512x512 each. Internal organs were then segmented based on tissue density through the deformable model approach [3]. We used atlas based human organs as initial coarse surface models. The organ models (triangular meshes) were then subdivided using interpolating modified Butterfly subdivision [4]. Regular Laplacian smoothing was applied to subdivided meshes to improve triangle aspect ratios.

The models were manually scaled and approximately positioned and oriented within the scan. Final fit was accomplished by iterative automatic procedure, deforming every model according to CT data image forces and model tension elastic forces [5]. Image forces attracted the model to the image features and elastic forces, due to deformation energy of the model, ensured its smoothness. Additionally, we provided several fixed control points on the scan to ensure quality of segmentation.

To simulate the deformation of abdominal cavity insufflation, we considered the abdominal wall as an elastic membrane and computed its displacement under pressure. We described it by the Laplace equation with fixed boundary conditions and constant external forces due to insufflation. Numerical computations were then performed using 2D linear Finite Element Method [6].

In order to achieve visual realism we textured the models with characteristic patterns coming from images of real human organs.

The texturing was performed using the methodology of lapped textures [7]. The mesh was covered with a number of overlapping patches textured independently; when rendering the patches were drawn successively on top of each other. Since each single patch does not have much curvature the textures appear without distortion.

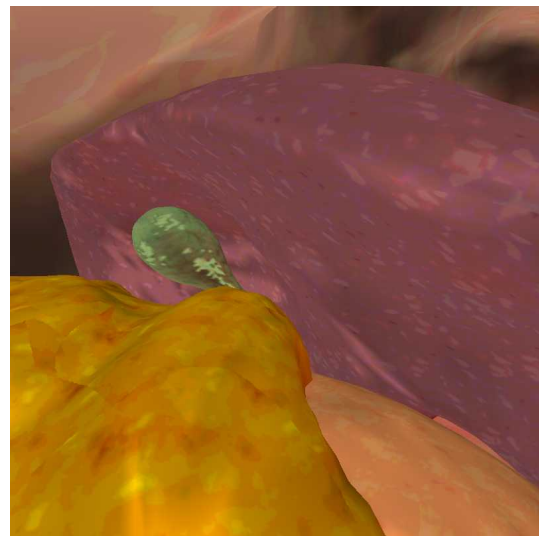


Figure 1: Snapshot from a virtual laparoscopic fly-through. The camera is positioned in abdominal cavity above the upper part of intestine, facing gallbladder and right lower part of kidney.

The final model of the abdominal cavity and selected organs can be then viewed through a virtual camera controlled by a user as a fly-through simulator. Snapshot from a virtual laparoscopic fly-through is presented in Figure 1.

In conclusion, virtual reality preoperative planning camera simulator uses *patient specific* data for the model in order to allow the surgeon to see anatomic relationships as they appear during the actual procedure on the patient and practice a complex surgical intervention before the surgery.

REFERENCES

- [1] D.B. Jones, N.J Soper, “Laparoscopic general surgery: current status and future potential”, *American Journal of Roentgenology*, 163(6), pp 1295-301, 1994
- [2] J. Torkington, S.G. Smith, B.I. Rees, A. Darzi, “Skill transfer from virtual reality to a real laparoscopic task”, *Surgical Endoscopy*, 15(10), pp 1076-1079, 2001

- [3] T. McInerney, D. Terzopoulos, "Deformable Models in Medical Image Analysis: A Survey," *Medical Image Analysis* 1(2), pp91-108, 1996.
- [4] D. Zorin, P. Schrder, W. Sweldens, "Interpolation Subdivision for Meshes with Arbitrary Topology", *SIGGRAPH 1996*, pp 189-192.
- [5] L. Zhukov, J. Bao, I. Guskov, J. Wood, D. Breen "Dynamic Deformable Models for MRI Heart Segmentation". *SPIE Medical Imaging 2002*, Proceedings of SPIE vol 4684, pp 1398-1405
- [6] O.C. Zienkiewicz, R.L. Taylor, "The finite element method : basic formulation and linear problems", vol. 1, 4-th ed., McGraw-Hill, London, UK, 1989.
- [7] E. Praun, A. Finkelstein, H. Hoppe. "Lapped textures", *SIGGRAPH 2000*, pp 465-470