Methods for visualizing

~ two- and three-dimensional data sets ~

29.05. 2014

What is visualization?

- Understanding and seeing
- We talk about trying to make our thoughts clear, to bring them into focus
- "To visualize" – really means "to construct a visual image in the mind" – but now also refers to "a graphical representation of data or concepts"
- Defn of Visualization: The use of computer-supported, visual representation of data to amplify cognition

- The purpose of visualization is insight, not pictures
- Visualizations can be said to be natural or visually efficient

External Cognitive Aids

- External cognition
  - Internal and external representation and processing weave together in thought
- External cognitive aids can enhance cognition
- Examples:
  - Slide rule
  - For multiplication, use of paper reduces the time required by a factor of 5 (for most people) – Why?
- An important class of external cognitive aids that make us smart are graphical inventions
  - Charts for navigation
  - Diagrams

Using Vision to Think

- A graphic picture may be used to communicate a preexisting idea or thought
- But graphical aids can also be used in formulating ideas and thoughts (“Using Vision to Think”)

Reading a diagram

- 3 levels of reading:
  - Read Fact
  - Read Comparison
  - Read Pattern

How Visualization Amplifies Cognition

Different ways that visualizations could help amplify cognition:

1. By increasing memory and processing resources available
2. By reducing the amount of time to search
3. Enhancing the detections of patterns and enabling perceptual inference operations
4. Aid perceptual monitoring
5. By encoding information in a manipulable medium

Mapping Data to Visual Form

Different forms of data:

- ❑ Data tables
- ❑ Meta data (descriptive data about data)
- ❑ Hierarchies, heterarchies

Types of Data

- ❖ *Entities* – objects of interest
- ❖ *Relationships*
  - ❑ form structures that relate entities
  - ❑ many kinds of relationships
- ❖ *Attributes* of entities or relationships
  - ❑ if they cannot be thought of independently
- ❖ *Operations* (*Colin Ware*)
  - ❑ Operations (actions) can also be considered as data (the merging of two data objects is an action we may wish to visualize, and operations are actions that operate on the data – distinct from the data, relationships and attributes.)

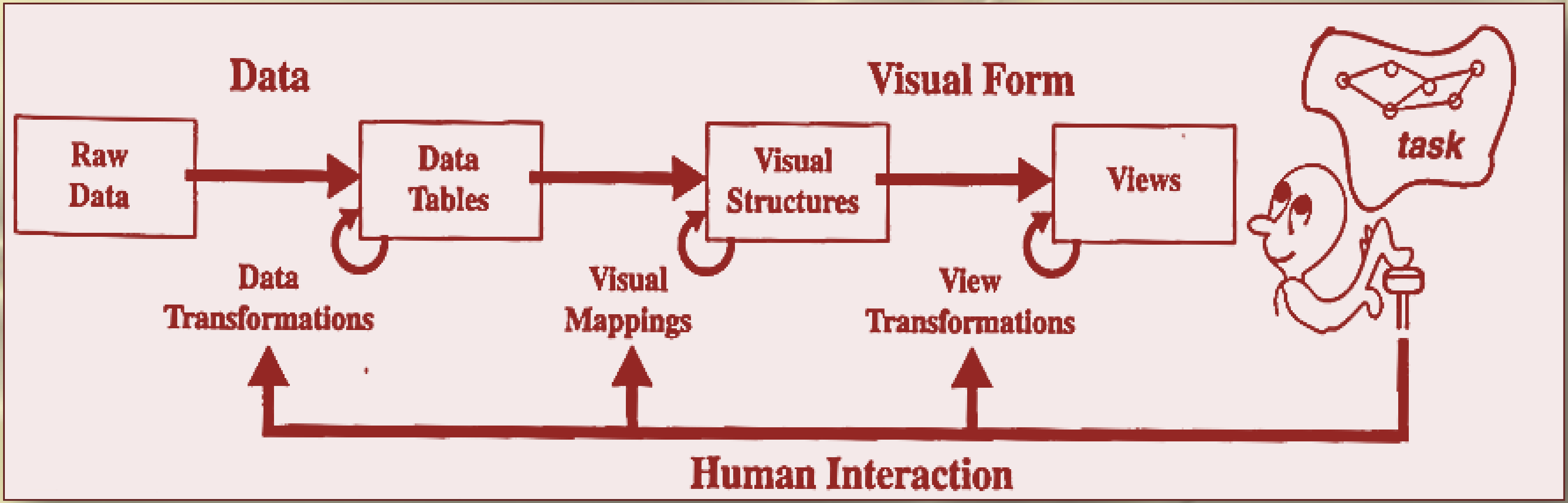
Attribute quality (variable types)

- *Nominal* data
  - Labeling function (e.g. apple, orange), category data that can be directed compared (=)
- *Ordinal* data
  - Sequencing things, ranking (<,>)
- *Quantitative* data
  - Real numbers (e.g. object A is twice as big as B, can do arithmetic on them, ratios)
  - And Interval data
    - Able to derive gap between data values (e.g. time of departure and arrival of an aircraft)

We can sometimes transform one type of data into another

Interaction and Transformation Controls

- Raw Data → Data Table (*filtering*)
- Data Table → Visual Structure (*pick mappings)*
- Visual Structure → Views (*probes, viewpoints, distortions*)



A User Study to Compare Four Uncertainty Visualization Methods for 1D and 2D Datasets

Many techniques have been proposed to show uncertainty in data visualizations. However, very little is known about their effectiveness in conveying meaningful information. We present a user study that evaluates the perception of uncertainty amongst four of the most commonly used techniques for visualizing uncertainty in one-dimensional and two-dimensional data. The techniques evaluated are traditional error-bars, scaled size of glyphs, color-mapping on glyphs, and color-mapping of uncertainty on the data surface. The study uses generated data that was designed to represent the systematic and random uncertainty components. Twenty-seven users performed two types of search tasks and two types of counting tasks on 1D and 2D datasets. The search tasks involved finding data points that were least or most uncertain. The counting tasks involved counting data features or uncertainty features.

Insight and quality assurance can be improved by recording uncertainty along with data. The practical benefits of understanding uncertainty in a scientific process can be manifold. Some uncertainty visualization techniques seem to appear more effective than others, however, little comparison has been done to evaluate the effectiveness of most of these techniques. Keeping this in mind, we categorize uncertainty visualization techniques for one-dimensional, two-dimensional, three-dimensional and temporal data. Based on this categorization, we constructed a user study to evaluate the effectiveness of four commonly used uncertainty visualization techniques:

- *Size of the uncertainty glyphs*
- *Color of the uncertainty glyphs*
- *Color of the data surface*
- *Error-bars*

Uncertainty Visualization Framework:

Data Dimension	*Technique*			
	Glyphs-size	***Blurring***	***Transparency***	***Other Methods***
0D	Point size	Point fadeout	Point visibility	...
1D	Glyphs on the line	Line fadeout	Line visibility	...
2D	Glyphs on surface	Surface fadeout	Patch visibility	...
3D	Volumetric glyphs	Volumetric fadeout	Regional visibility	...

Inter-class

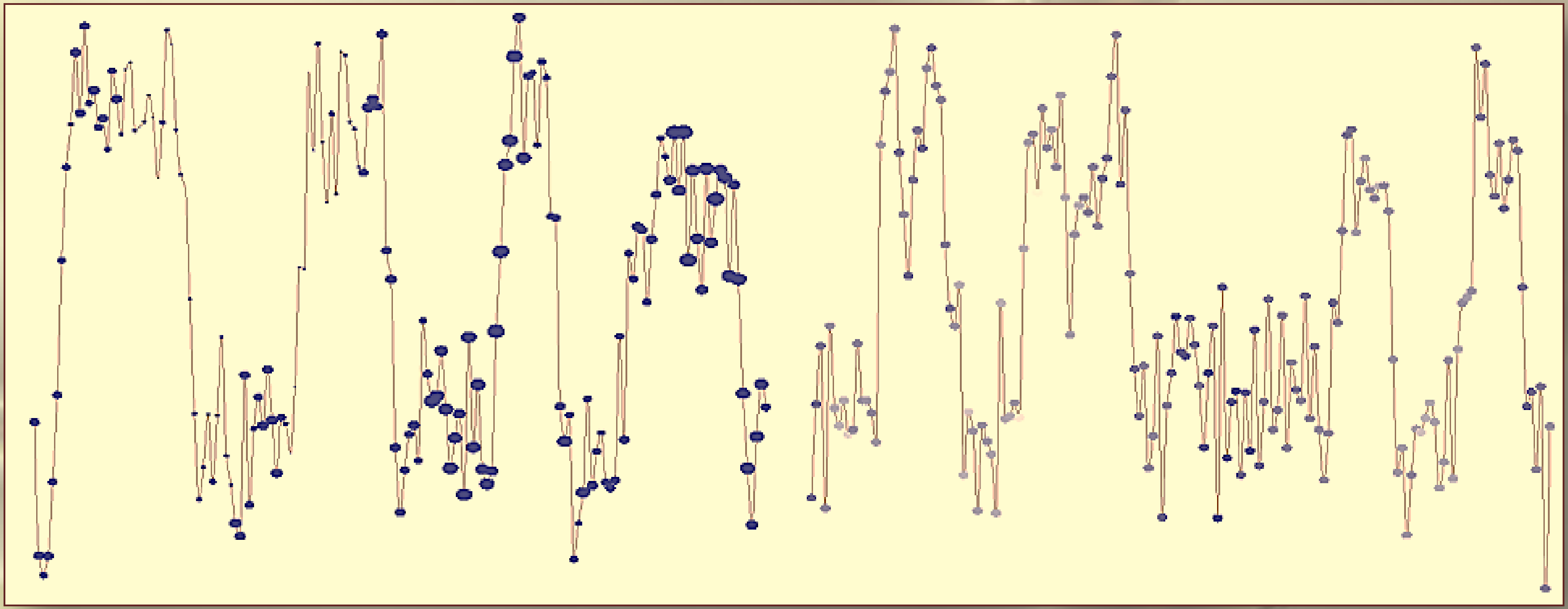
Time

Scalars, vectors and tensors

Intra-class

Uncertainty visualization techniques explored for 1D datasets

a) Scaling the size of glyphs: b) Altering the color attribute of glyphs:



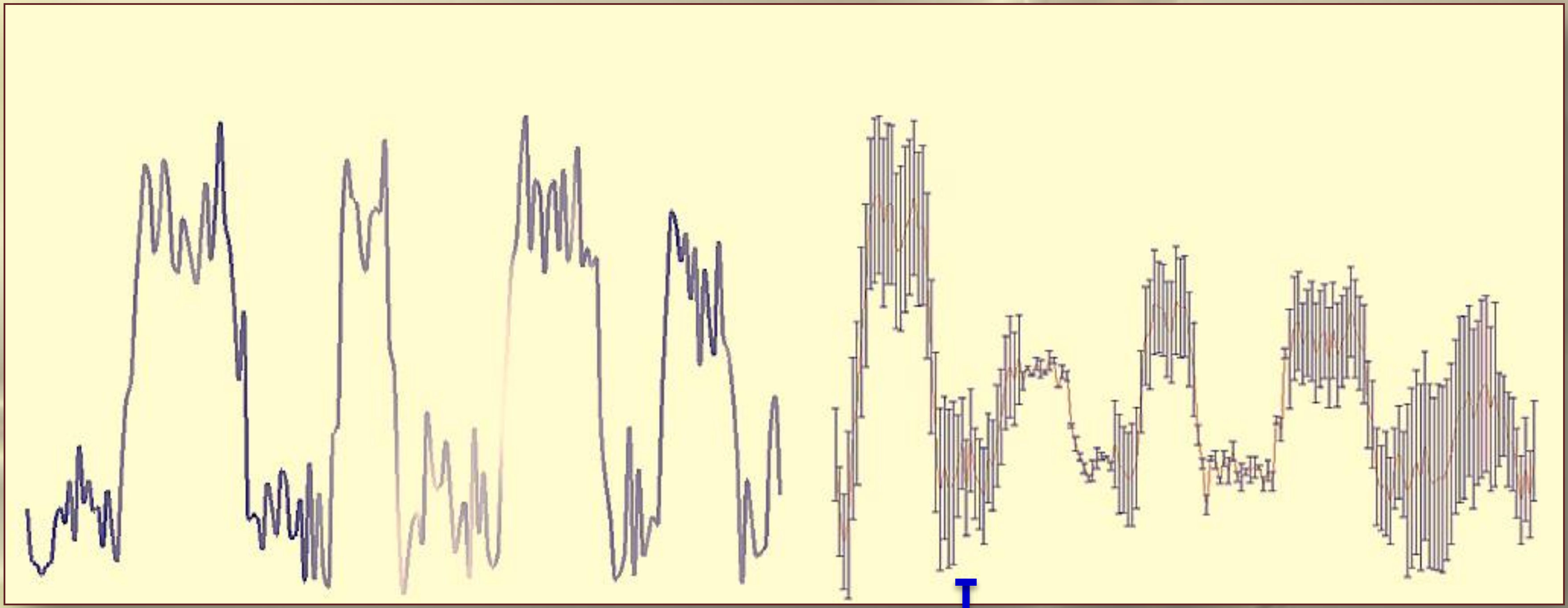
High uncertainty
Low uncertainty

High uncertainty
Low uncertainty

… Uncertainty visualization techniques explored for 1D datasets

c) Color-mapping the surface of data with uncertainty

d) Using the traditional error bars:

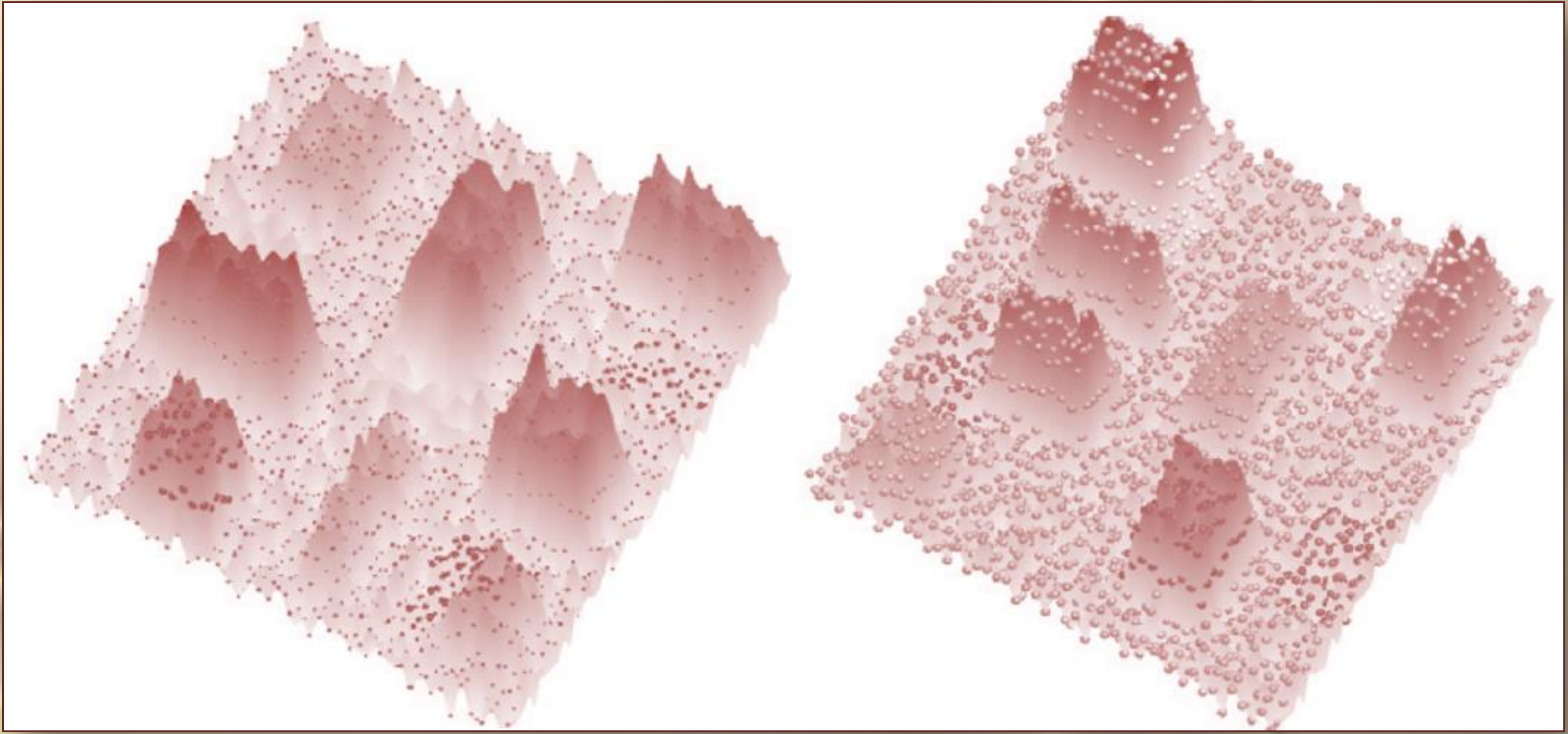


High uncertainty
Low uncertainty

High uncertainty
Low uncertainty

Uncertainty visualization techniques explored for 2D datasets

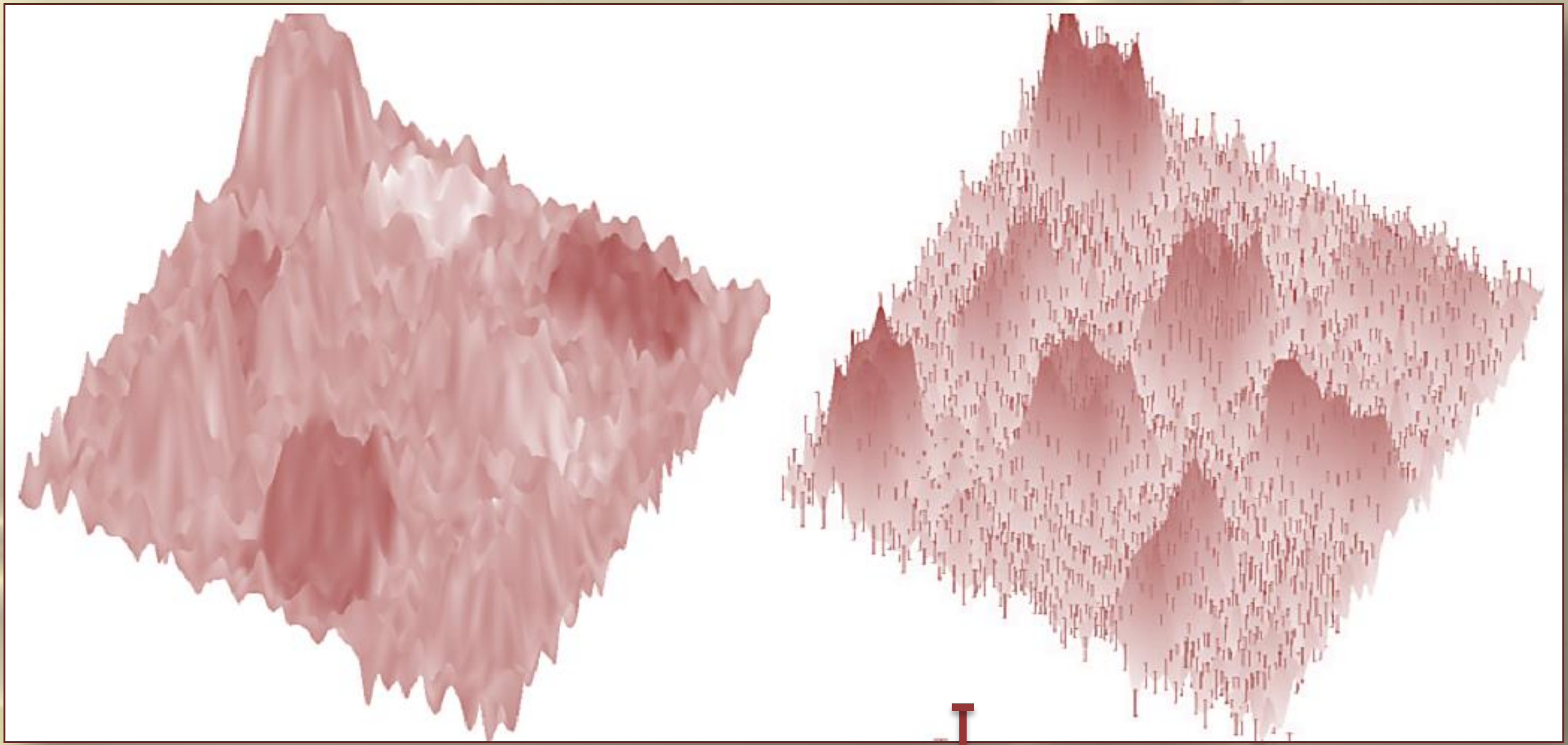
a) Scaling the size of glyphs: b) Altering the color attribute of glyphs:



… Uncertainty visualization techniques explored for 2D datasets

c) Color-mapping the surface of data with uncertainty:

d) Using the traditional error bars.:



High uncertainty
Low uncertainty

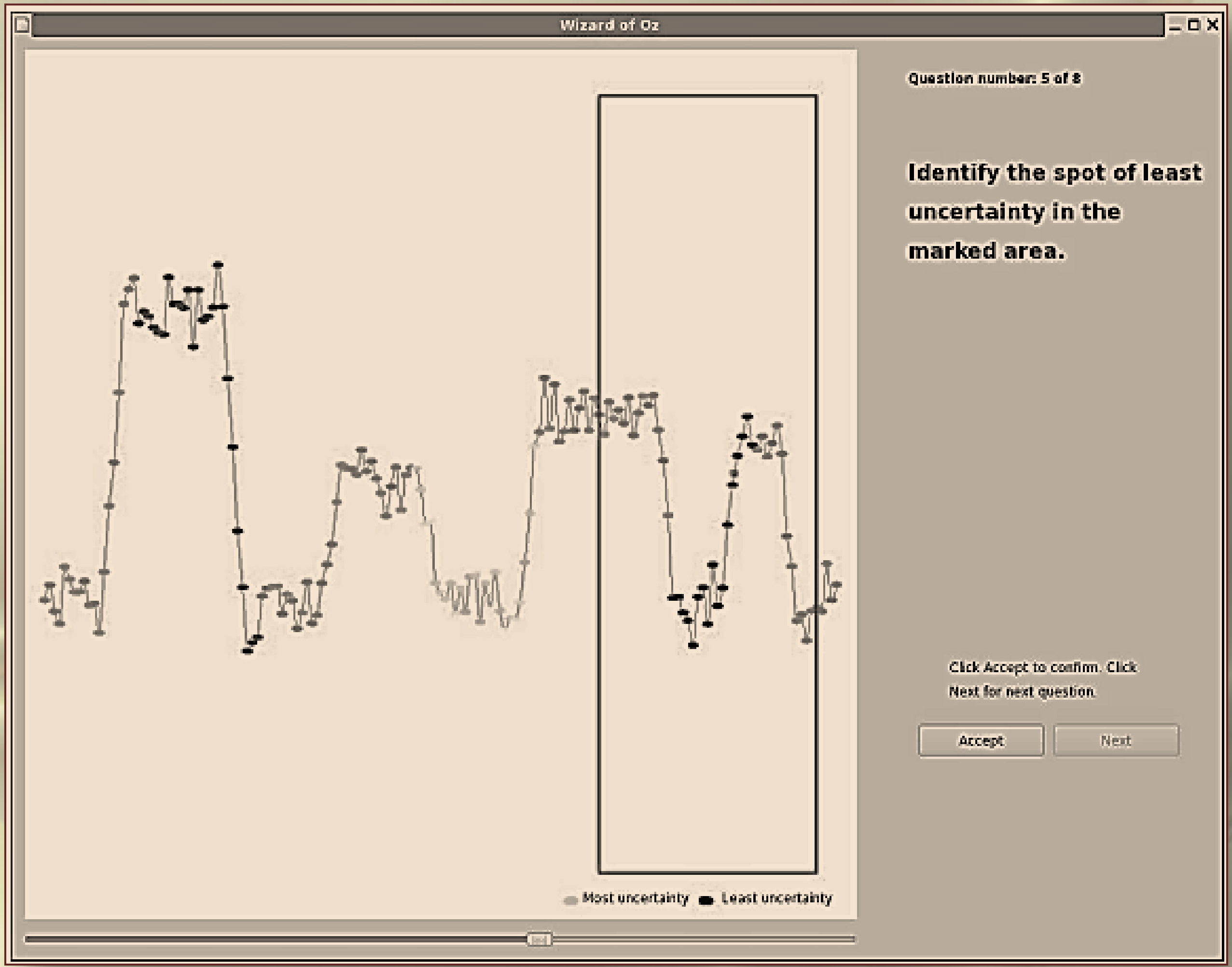
High uncertainty
Low uncertainty

The Main Study:

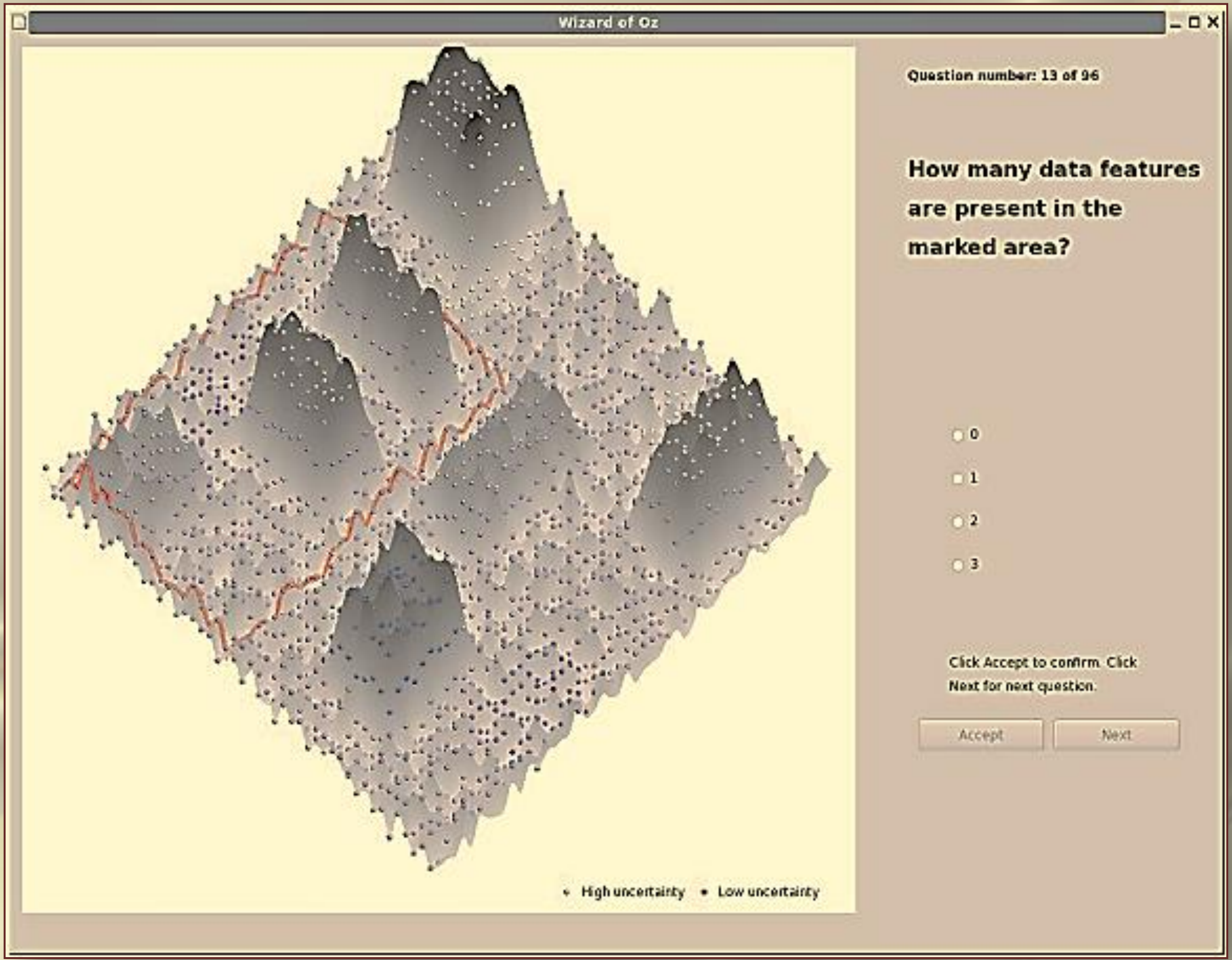
The four questions asked were:

- How many data features are present in the marked area?
- How many uncertainty features are present in the marked area?
- Identify the spot of least uncertainty in the marked area.
- Identify the spot of most uncertainty in the marked area.

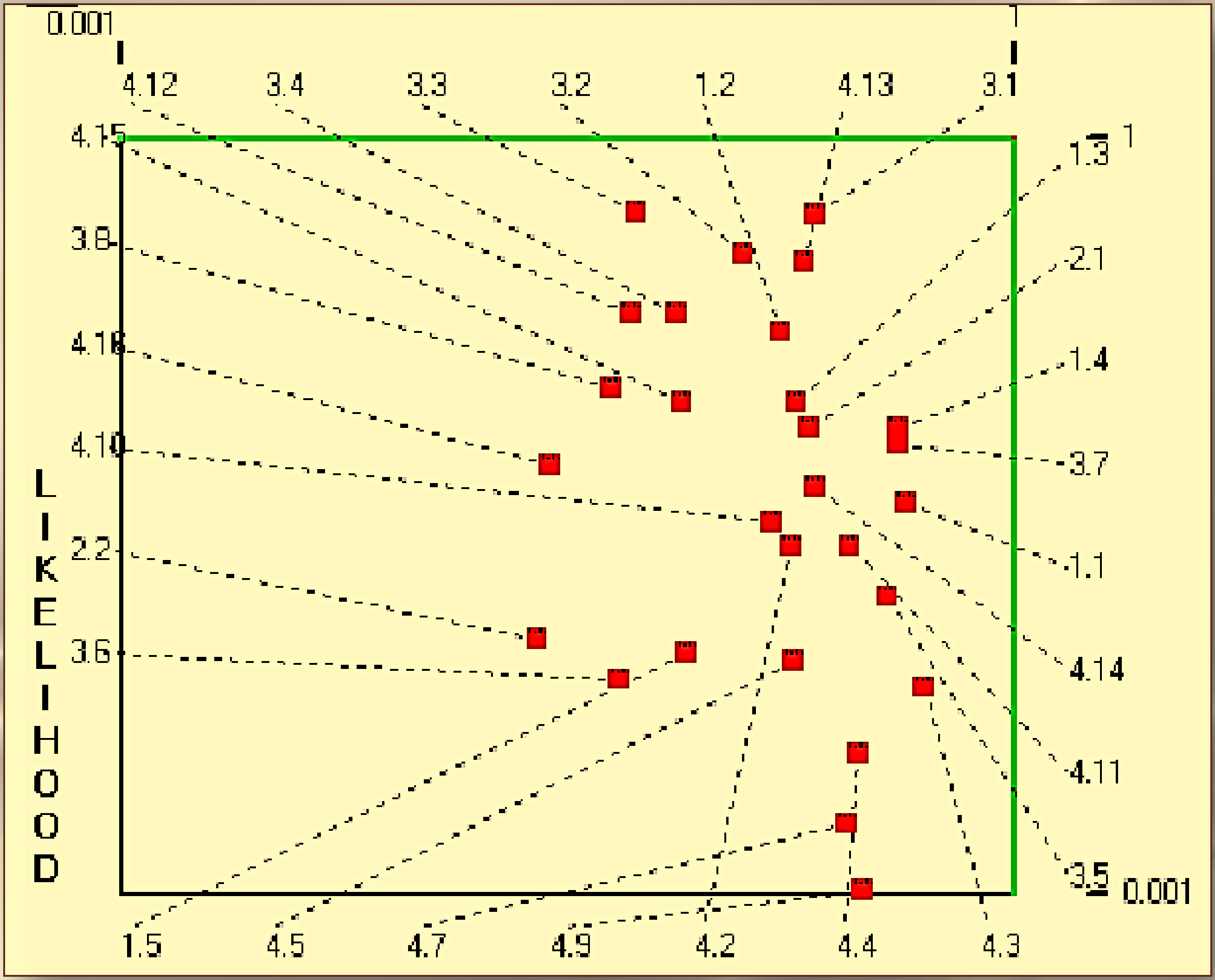
User interface for a search task on a 1D dataset:



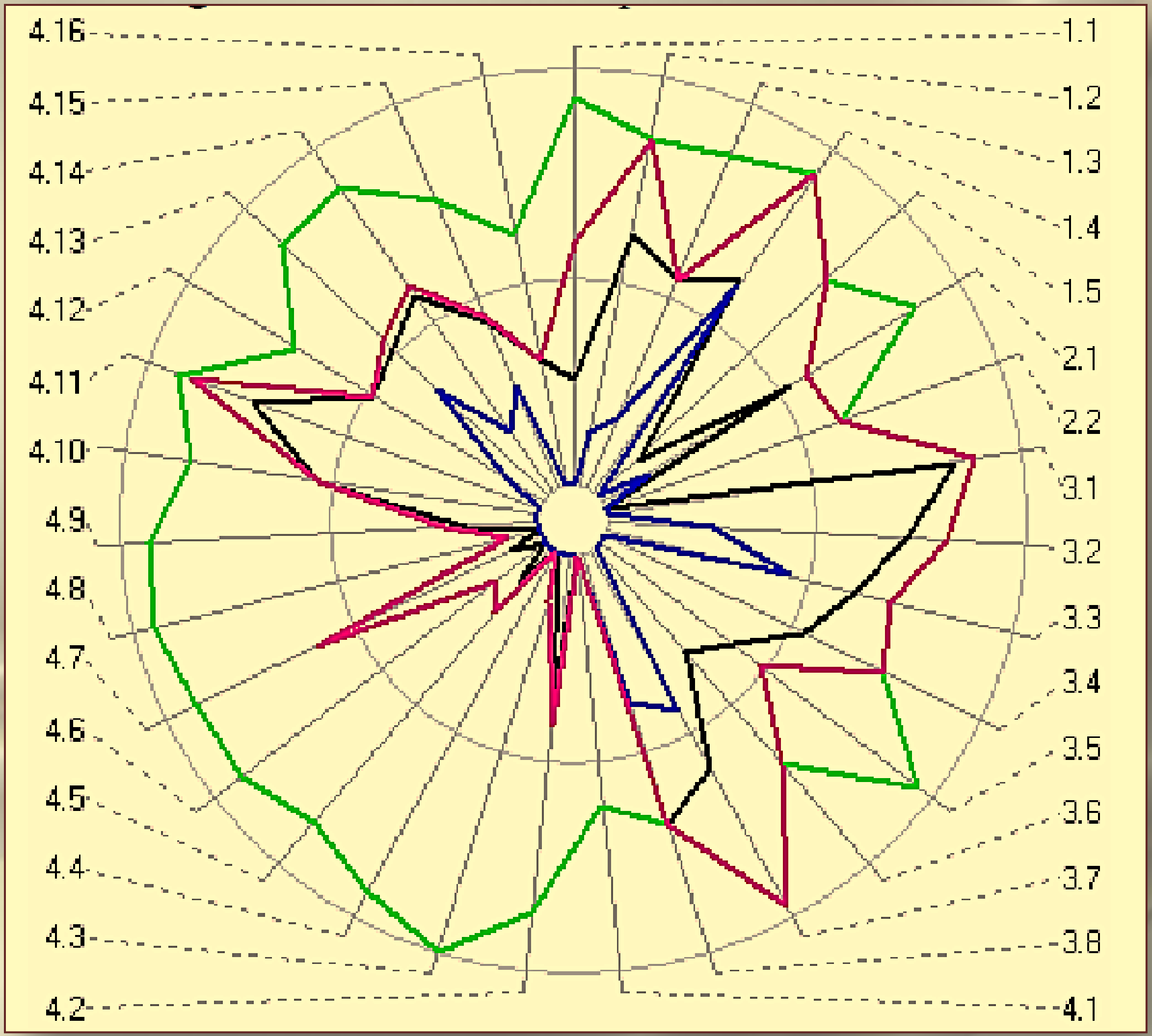
User interface for a counting task on a 2D dataset:



2-D chart of risks:



Kiviat chart of several designs' risks:



A Taxonomy of Visualization Techniques using the Data State Reference Model

The taxonomic analysis will show, many of the techniques share similar operating steps that can easily be reused (shows that the Data State Model not only helps researchers understand the space of design, but also helps implementers understand how information visualization techniques can be applied more broadly.

We will present a detailed analysis of a large number of visualization techniques using the Data State Model.

The analysis of the information visualization design space is the most detailed and thorough to date. It is more detailed in the sense that we have broken each technique down by not only its data type, but also by its processing operating steps. It is thorough in that it categorizes the well-known techniques.

SUMMARY OF DATA STATE MODEL

Previously, we also proposed a taxonomy of information visualization techniques based not only on data types, but also on the processing operators that are inherent in each visualization technique. We showed that information visualization techniques could be described using the *Information Visualization Data State Reference Model* (*Data State Model*).

Figure 1 shows an overview of the *Data State Model,* which breaks down each technique into four Data Stages, three types of Data Transformation and four types of Within Stage operators. The visualization data pipeline is broken into four distinct *Data Stages: Value, Analytical Abstraction, Visualization Abstraction,* and *View* (Table 1). Transforming data from one stage to another requires one of the three types of *Data Transformation operators: Data Transformation, Visualization Transformation,* and *Visual Mapping Transformation* (Table 2).

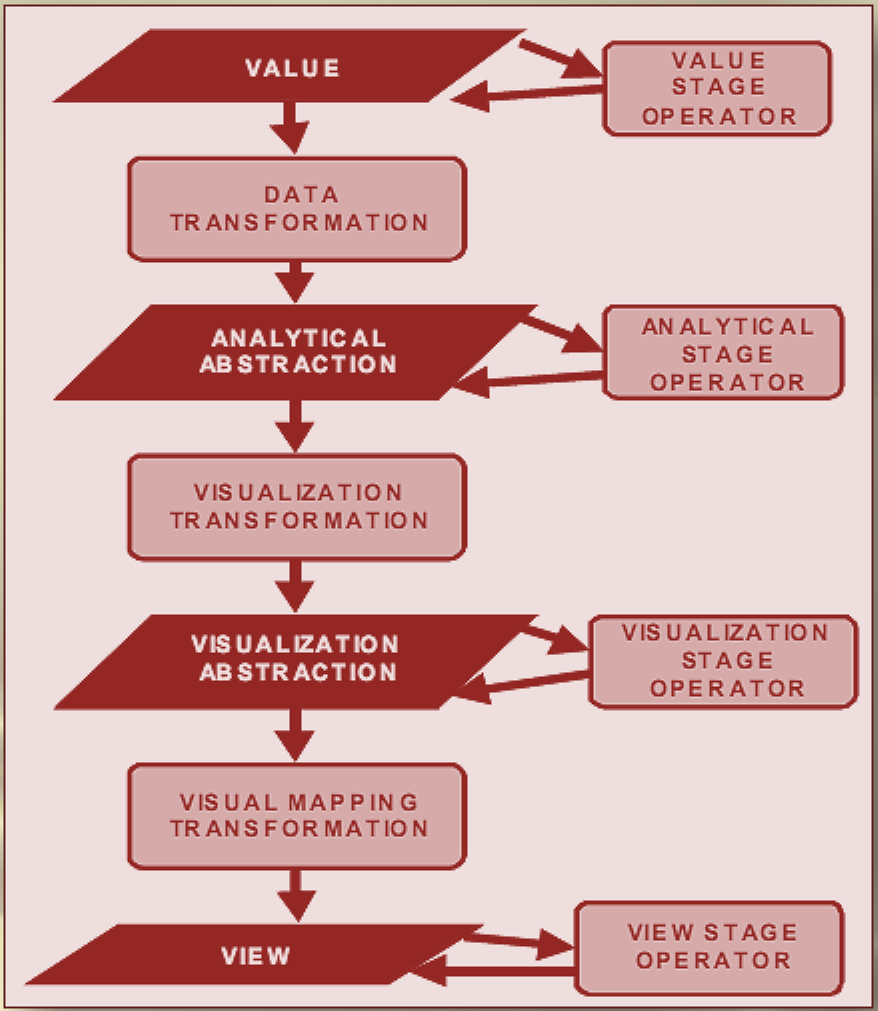


Figure 1: *Information Visualization Data State Reference Mode*

Table 1: *Data Stages in the Data State Model*

Stage	Description
Value	The raw data
Analytical Abstraction	Data about data, or information, a.k.a. meta-data
Visualization Abstraction	Information that is visualizable on the screen using a visualization technique
View	The end-product of the visualization mapping, where the user sees and interprets the picture presented to her

Table 2: *Transformation Operators*

Processing Step	Description
Data Transformation	Generates some form of analytical abstraction from the value (usually by extraction).
Visualization Transformation	Takes an analytical abstraction and further reduces it into some form of visualization abstraction, which is visualizable content.
Visual Mapping Transformation	Takes information that is in a visualizable format and presents a graphical view.

Within each Data Stage, there are also operators that do not change the underlying data structures.

These are the Within Stage Operators, of which there are four types, corresponding to the four Data Stages:

1. *Within Value,*
2. *Within Analytical Abstraction,*
3. *Within Visualization Abstraction, and*
4. *Within View.*

Figure 2 shows an example of the *Data State Model* applied to the problem of visualizing the connections between a set of *Web pages*. This example shows that:

(1) some operators create new kinds of data sets, whereas some operators create filtered subsets, which is the difference between Transformation and Within Stage operators, and

(2) that the same Visualization Abstractions can be mapped using a variety of Visual Mapping Transformation operators. For example, Disk Trees or Cone Trees can both be applied to a hierarchy of interconnected nodes.

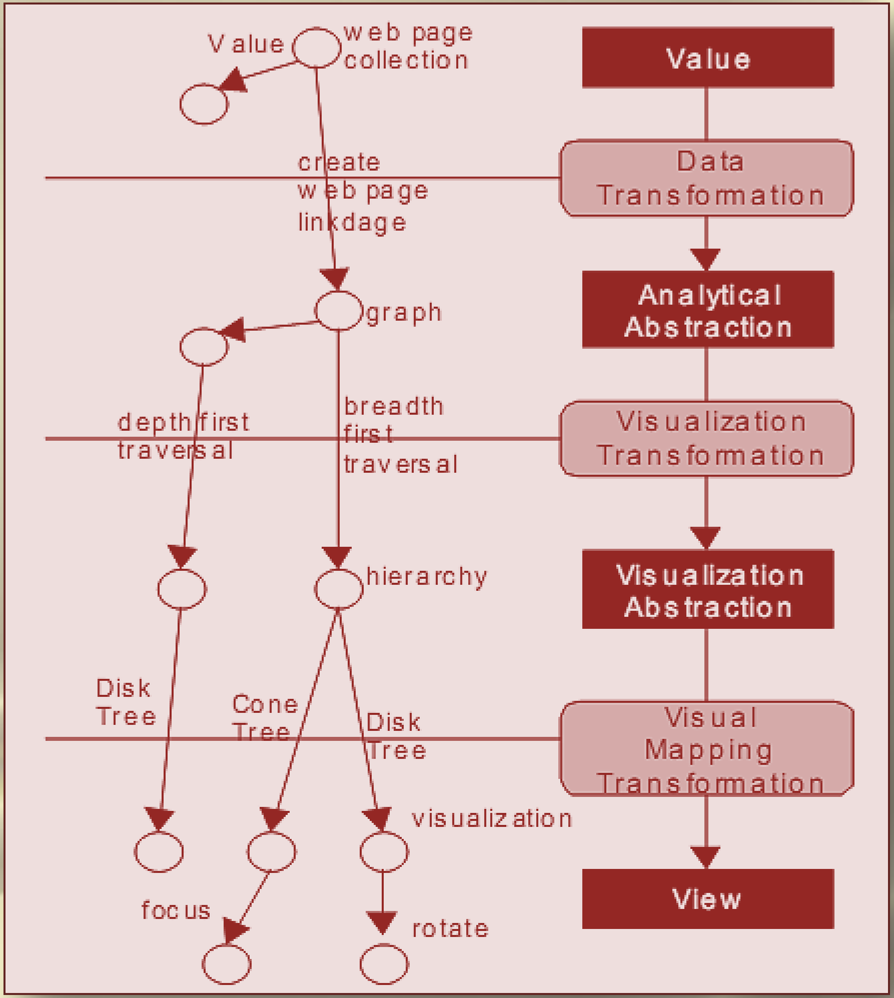


Figure 2: *Data State Model applied to Web sites*

TAXONOMY

By isolating dependencies, we can more easily reuse different parts of the pipeline to construct new information visualizations. Therefore, we have taken this model and used it to taxonomize various visualization techniques.
The idea is to analyze the various techniques, thus increasing our knowledge of how each technique can be built using various operators. In the following, we used this model to analyze some 36 visualization techniques.
With a clearer understanding of the interactions between the data and the operators, implementers will be more equipped to construct new interactions or new visualizations. In practice, these analysis techniques have been applied in a system called the *Visualization Spreadsheet* and have enabled reuse and rapid development.

The following **table** presents the ***taxonomy*** using the ***Data State Model***.

A row represents a single visualization technique or system. ***The cells*** in that row describe the operators that comprises that technique. *Non-italic* items refer to the operators, while *Italic* items refer to example data sets within that ***Data Stage***.

The columns are the seven types of operators described in **Figure 1**: within and non-within stage operators.

In certain cases below, cells marked with → have no corresponding ***Abstractions*** or ***Operators*** at that stage, because the data is already in a visualizable format.

Visualization Technique	Within Value	Data Transformation	Within Analytical Abstraction	Visualization Transformation	Within Visualization Abstraction	Visual Mapping Transformation	Within View
Some example Scientific Visualizations							
Visible Human Project [NLM]	*Data:Image Scans of slices of human*	Marching Cubes: create voxels	*Abstraction: Voxels*	→	→	Create slices of Volumes	Specifying slices using sliders
MapQuest [MapQuest.com]	*Data: Geographical Road Maps*	Parse information into records	*Abstraction: Parsed record set* Dynamic value-filtering of records	Create linear list of records	*Visualization Abstraction: Linear list with item features*	Icons depicting different locations and their types: restaurants, etc.	Scroll, Zoom, View-filtering of interested locations.
Ozone Layer Visualization [Treinish94]	*Data: Ozone layer geographical information over time*	Extract geographical samples into quantitative variables	Normalize samples and quantitative values	Direct spatial mapping of quantitative values to longitude, latitude, height	*Abstraction: Earth with overlaid info.*	Map quantitative variables to longitude, latitude, and height; Map ozone level to color	Rotate, Scale, Animate; Change colormap
Geographical-based Info Visualization							
Profit Landscape [Visible Decisions]	*Data: Profit statistics linked to geographical regions*	Extract into quantitative variables	Normalize sample	Direct spatial mapping of geo-coordinate variables	→	Map geo-coordinate variables onto a geographical map; Map profit variable to glyph (size of lines)	Rotate, Scale, Animate; Change colormap
2D							
TileBars [Hearst95]	*Example data: text documents*	Parse into feature vectors.	Search through vector, compute intersection of vectors.	Each rectangle corresponds to a document.	→	Squares represent text segments; darkness indicates frequency of terms	Browse
ValueBars [Chimera92]	*Example data: text documents, file system records*	Parse into feature vectors, then choose one attribute.	Allow multiple attributes to be chosen for several ValueBars.	→	→	Lines represent the value of the attribute of an item in the text document.	Scroll
Information Mural [Jerding96]	*Example data: software code, documents, stock prices, sun spot data*	Parse into feature vectors	Dynamic Value-Filtering	→	→	Lines represent the value of the attribute of an item in the document; color maps another value or type.	Scroll; zoom
LifeLines [Plaisant96]	*Example data: medical and court records*	Parse into feature records	Dynamic Value-Filtering	Create lines on 2D plot	Dynamic value-filtering; Apply unmapped variable filtering	Icons indicate discrete events; Line colors and thickness indicate relation or significance	Dynamic view-filtering
Multi-dimensional Plots							
Dynamic Querying [Ahlberg94]	*Example data: Home, Movies sales data*	Parse into feature records	→	Create multi-dimensional point sets	Dynamic value-filtering; Apply unmapped variable filtering	Map into scatter plot; Choosing variables-to-axes mappings	Dynamic view-filtering
Parallel Coordinates [Inselberg97]	*Example data sets: production run of VLSI chip yield and its defect parameters*	Extract corresponding yield and parameter feature set	Choosing a subset of records using dynamic value-filtering	Create point set from records	*Visualization Abstraction: Point set*	Plot point set using parallel coordinates	Dynamic view-filtering; Sorting of axis; Interactive permutation of axis

Visualization Technique	Within Value	Data Transformation	Within Analytical Abstraction	Visualization Transformation	Within Visualization Abstraction	Visual Mapping Transformation	Within View
World-Within-World [Feiner 93]	*Data: High-dimensional point set or surfaces* Dynamic value-filtering	→	*Abstraction: Point set or surfaces.* Normalize samples	→	→	Map high dimensional surface to local area	Dynamic view-filtering; Rotate, Scale, Focus
Information Landscapes and Spaces							
Perspective Wall [Mackinlay91]	*Example Data: Schedule, File system*	Parse information into records	*Abstraction: Parsed record set* Dynamic value-filtering of records	Create linear list of records	*Visualization Abstraction: Linear list with item features*	Create wall panels in 3D with glyphs, with focus+context distortion-based display	Focus on a particular wall; Focus an item; Dynamic view-filter; Choose different levels of detail
Pad++ [Bederson94]	*Example data: Many types including text documents, file system, drawings.*	→	*Abstraction: windows, lines, icons, points, polygons.*	→	*Many Abstractions are compatible.*	Many representations.	Zoom: some objects will fade in, some will fade out Scroll
Elastic Windows [Kandogan96]	*Example Data: Mail reader, WWW Browser, Window based interfaces*	→	*Abstraction: windows*	→	*Many kinds of windows are compatible.*	Create Elastic Window mapping to screen (space-tiling)	Change focus, enlarge, zoom-out
WebBook and WebForager [Card96]	*Data: URLs for web pages*	Retrieve web pages; Generate images of each Web page	*Abstraction: Images of HTML pages generated by getting the Web pages*	Create linear list of pages; Aggregate into a book or a pile; Place pile on book shelf (creating list of lists); Crawl from a URL and create a book from the collection	*Abstraction: Linear page lists, Collection of page lists.* Merge page lists; Merge sets of page lists	Create books with multiple pages; View using Document Lens; Create bookshelf, table, piles	Focus on a book; Focus on a page; Flip through pages in a book; View book using Document Lens; Put onto history pile
Trees							
Cone tree [Robertson91], Hyperbolic Browser Lamping95], TreeMap [Johnson91], DiskTree [Chi98], Cheops [Beaudion96], WebTOC [Nation97], Information Cube [Rekimoto93]	*Data: File system; Organization charts; Hypertext or Web linkage structure*	Extract into graph	*Abstraction: Graph* Apply dynamic value-filtering of nodes or edges	Do breadth first traversal	*Visualization Abstraction: Tree hierarchy*	Layout using 3D cones; hyperbolic tree; Disk Tree; space filling TreeMap; Cheops approach; Expanding trees; Using Information Cube technique with semi-transparent cubes.	Focus node; Hide subtree; Change orientation and position of tree; Apply Dynamic level-filtering
Network							

Visualization Technique	Within Value	Data Transformation	Within Analytical Abstraction	Visualization Transformation	Within Visualiza-tion Abstraction	Visual Mapping Transformation	Within View
GraphViz [AT&T]	*Example data: Software code modules, File system, all kinds of graphs*	Extract edges and nodes into a graph.	*Abstraction: graph*	→	→	Sophisticated graph layout algorithm that places nodes on 2D plane intelligently	View: graph with minimized edge crossings.
GV3D or lpNV3D [Ware93]	*Example data: Software code modules*	Extract connections between code	filter nodes	Form nested graphs from earlier extracted graphs.	Dynamic value-filtering; Apply unmapped variable filtering	Map code modules into cubes in 3D, with linkages between cubes specifying relationships	Dynamic view-filtering
SeeNet [Becker95] Comment: aggregation is mentioned as implemented using data management software, several different views of the data sets.	*Example data sets: phone calls made; Internet packet flows; Email communication patterns*	Parse into source and destination links.	*Analytical abstraction: parsed records of source and destination and associated feature sets* Unmapped variable value-filtering; Choose variables of displayed statistics; Aggregate records	Transform into graphs and networks.	*Visualization Abstraction: Graphs, and Networks*	Display graph as matrix, geographical linkmaps, or nodemaps	For all three views: Sound feedback; Unmapped variable view-filtering (they called it 'conditioning') For matrix display: Threshold time view-slider; Permute rows and columns For nodemaps and linkmaps: Change Size, Color, Zoom; Parameter focusing; Identification by brushing; Change animation speed; Change line thickness, or line length; Dynamic query threshold view-slider For nodemaps: Change symbol size; Use color sensitivity view-slider
Text							
AlignmentViewer [Chi96]	*Data: Similarity reports from comparing a single sequence against a database of many other sequences*	Parsing textual reports; Addition, Subtraction between different reports; Unmapped variable value-filtering	*Abstraction: Alignment records (data structure representing parsed information)*	Extracting information from records	*Visualization Abstraction: Feature point set with vector*	Map into comb-glyphs	Rotation, Translate Zoom; Focus on a single alignment; Detail-on-demand; Animation (by using an iterator over the view-filtering)
ThemeScape and Galaxies [Wise95]	*Data: CNN news stories*	Create textual word frequency vector; Choose an item and then perform weighted query	*Analytical Abstraction: Text vectors*	Multi-dimensional scaling (MDS); Principal component analysis	*Visualization Abstraction: 2D positions from MDS*	Map into surfaces of hills and valleys	Zoom, Rotate; Focus on detail spot For ThemeScape: Create slices For Galaxies: Animate scatter plot
Web Visualization							
WebSpace [Chi94]	*Data: web site*	walk web site and create web linkage graph	value-filtering	Create breadth first traversal tree	*Visualization Abstraction: Tree*	Layout using cone tree	Dynamic view-filtering
3D Hyperbolic [Munzner95]	*Data: web site*	walk web site and create web linkage graph	value-filtering	Create breadth first traversal tree	*Visualization Abstraction: Tree*	Layout using 3D Hyperbolic Tree	Dynamic view-filtering
WebMap [Domel94]	*Data: web site traversal history*	Extract user path from traversal history graph	*Abstraction: traversal history graph*	Form navigation spanning trees.	*Visualization Abstraction: Tree*	Map to Tree layout, circle layout, rectangle layout, Horizon tree layout	Dynamic view-filtering

Visualization Technique	Within Value	Data Transformation	Within Analytical Abstraction	Visualization Transformation	Within Visualization Abstraction	Visual Mapping Transformation	Within View
Time Tube [Chi98]	*Data: web structure evolving over time and its associated usage statistics (Content, Usage, and Topology of the web site)*	Create graph from web structure by crawling the web site	*Analytical Abstraction: Evolving graph represented as ordered collection of graph*	Do breadth first traversal with global node position over time	*Visualization Abstraction: Evolving tree as ordered list of trees*	Create Time Tube, which is represented using an aggregation of Disk Trees (invisible tube--like shelf)	Recognize gestures for: Focus on a slice; Bring slices back into the Time Tube; Zooming focus on the connectivity of a node by right-clicking on it; Rotate slices; Brushing on pages by highlight URL on all slices; Animate through the slices
Visualization Spreadsheets							
Table Lens [Rao94a,Rao95]	*Data: baseball player statistics*	Parse statistics into numeric records	*Analytical Abstraction: Numeric records* Sort records	Construct numeric table from records	*Visualization Abstraction: Constructed numeric table*	Represent number using bars, with focus+context distortion-based table	Change distortion focus
Spreadsheet for Images [Levoy94] (direct mapping from data to view)	*Data, Analytical and Visualization Abstraction: pixels, voxels* Rotate Image; Filter; Change color scale; and other image processing mechanisms	→	→	→	→	→	*View: images from pixels, volumes from voxels* Rotate image; Filter; Change color scale; and other image processing operations; Rocking the volume visualization
FINESSE [Varshney96]	*Data: Financial data*	Compute call and put option prices	*Abstraction: Matrix records, Mathematical functions* Change parameter of functions; Change arithmetic relationships	Compute curves from math function models	*Visualization Abstraction: Matrix, Computed curves*	Create heat map; Create surfaces in 3D; Plot using 3D bar charts; 2D line plots; Create text for filenames; Represent variables using value sliders	Change orientation of geometric objects; Change to common colormap or font; View using same geometric orientation; Show cell dependency relationships; Picking a data item, Input math function
Spreadsheet for Information Visualization [Chi97infovis] Comment: allows value and view Dependencies between cells	*Example data sets: Point sets; Matrix; Sequence similarity reports; Web structure, Web usage pattern; etc.*	Normalize matrices; Parse textual reports; Create random point sets; Create graph from web structure by crawling the web site	*Abstraction: Normalized matrix and point sets; Value tuples; Evolving graph represented as ordered collection of graph* Dynamic value-filter; Algebraic data set operators	Perform Delaunay Triangulation; Extract data features from records; Do breadth first traversal with global node position over time	*Visualization Abstraction: Point set; Matrix; Triangulated surface; Point set with feature vector; hierarchy, list of trees; etc.*	Create heat map, matrix cube visualization, matrix bar visualization, Cone Tree, Disk Tree, glyphs, scatter plot; Choosing variable-to-axes mapping; Change cells to share same visual mapping Transformation	Dynamic view-filter; Change object position and orientation; Pixel image addition between cells; Geometric object addition between cells; Animation; Coordinated direct manipulation
Web Analysis Visualization Spreadsheet [Chi99]	*Date: Web site usage analysis* Filter-Value	Extract linkage information; Extract usage information	Cluster nodes	Breadth First Traversal	Perform usage frequency pattern algebra; Apply Spreading Activa-tion pattern algebra	Display Disk Tree; Display Cone Tree; Apply Coloring Pattern; Display Pattern Glyph	Apply geometric operators; Detail-on-demand Zoom; Animation

XmdvTool: integrating multiple methods for visualizing multivariate data

Much of the attention in visualization research has focussed on data rooted in physical phenomena, which is generally limited to three or four dimensions. However, many sources of data do not share this dimensional restriction. A critical problem in the analysis of such data is providing researchers with tools to gain insights into characteristics of the data, such as anomalies and patterns. Several visualization methods have been developed to address this problem, and each has its strengths and weaknesses. This paper describes a system named XmdvTool which integrates several of the most common methods for projecting multivariate data onto a two-dimensional screen. This integration allows users to explore their data in a variety of formats with ease. A view enhancement mechanism called an N-dimensional brush is also described. The brush allows users to gain insights into spatial relationships over N dimensions by highlighting data which falls within a user-specified subspace.

Multiple methods of visualizing the yeast vacuole permit evaluation of its morphology and inheritance during the cell cycle

The vacuole of the yeast Saccharomyces cerevisiae was visualized with three unrelated fluorescent dyes: FITC-dextran, quinacrine, and an endogenous fluorophore produced in ade2 yeast. FITC-dextran, which enters cells by endocytosis, had been previously developed as a vital stain for yeast vacuoles. Quinacrine, which diffuses across membranes and accumulates in acidic compartments in mammalian cells, can also be used as a marker for yeast vacuoles. ade2 yeast accumulate an endogenous fluorophore in their vacuoles.

Using these stains, yeast were examined for vacuole morphology throughout the cell division cycle. In both the parent cell and the bud, a single vacuole was the most common morphology at every stage.

… Multiple methods of visualizing the yeast vacuole permit evaluation of its morphology and inheritance during the cell cycle

Two or more vacuoles could also be found in the mother cell or in the bud; however, this morphology was not correlated with any stage of the cell division cycle. Even small buds (in early S phase) often contained a small vacuole. By the time the bud was half the diameter of the mother cell, it almost always bore a vacuole. This picture of vacuole division and segregation differs from what is seen with synchronized cultures. In ade2 yeast, the bud usually inherits a substantial portion of its vacuole contents from the mother cell. We propose that vacuolar segregation is accomplished by vesicular traffic between the parent cell and the bud.

Systems and methods for visualizing interior tissue regions using expandable imaging structures

~ Patent 05848969 ~

A system for imaging an interior body region comprising a probe including an axis and an imaging structure having a periphery adapted to selectively assume an expanded geometry and a collapsed geometry, and an array of spaced apart ultrasound transducers attached to the periphery of the imaging structure, wherein members of the array move radially outward and inward relative to the axis as the periphery assumes the expanded geometry and the collapsed geometry respectively.

… Systems and methods for visualizing interior tissue regions using expandable imaging structures

A method for visualizing body tissue comprising the steps of providing a probe with an imaging structure having a periphery adapted to selectively assume an expanded geometry and a collapsed geometry, and an array of spaced apart ultrasound transducers attached to the periphery of the imaging structure, guiding the probe into an interior body region while causing the imaging structure to assume the collapsed geometry, causing the imaging structure to assume the expanded geometry upon arrival in the interior body region, and operating the imaging structure, while in the expanded geometry, to visualize tissue in the interior body region.

… Systems and methods for visualizing interior tissue regions using expandable imaging structures

A system for imaging an interior body region, comprising a probe having an imaging structure adapted to selectively assume an expanded geometry and a collapsed geometry, and an array of ultrasound transducers on the imaging structure, wherein members of the array move away from one another as the imaging structure assumes the expanded geometry.

A system for imaging an interior body region, comprising a probe including a distal portion, an imaging structure having a predetermined geometry on the distal portion, a plurality of arrays of ultrasound transducers attached to the imaging structure in a predetermined arrangement for acquiring images along a plurality of sectors, and a support structure on the probe and substantially surrounding the imaging structure, the support structure being adapted to assume an expanded geometry and a collapsed geometry.

References

1. *XmdvTool: integrating multiple methods for visualizing multivariate data,* Matthew O. Ward, Worcester Polytechnic Institute, Worcester, MA, Proceeding VIS '94 Proceedings of the conference on Visualization '94, Pages 326-333, IEEE Computer Society Press Los Alamitos, CA, USA ©1994;
2. *Multiple methods of visualizing the yeast vacuole permit evaluation of its morphology and inheritance during the cell cycle,* L S Weisman, R Bacallao, and W Wickner, October 1, 1987 // JCB vol. 105 no. 4 1539-1547, The Rockefeller University Press, doi: 10.1083/jcb.105.4.1539, ©1987 Rockefeller University Press.
3. *Systems and methods for visualizing interior tissue regions using expandable imaging structures,* D McGee, D Panescu, DK Swanson, H Tenhoff, US5848969 A, Grant, US 08/736,827, Dec 15, 1998.
4. *A Taxonomy of Visualization Techniques using the Data State Reference Model,* Ed H. Chi, Xerox Palo Alto Research Center 3333 Coyote Hill Road, Palo Alto, CA 94301, echi@parc.x
5. *A User Study to Compare Four Uncertainty Visualization Methods for 1D and 2D Datasetserox.com* , Jibonananda Sanyal, Student Member, IEEE, Song Zhang, Member, IEEE, Gargi Bhattacharya, Phil Amburn, Member, IEEE, and Robert J. Moorhead, Senior Member, IEEE, IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL. 15, NO. 6, NOVEMBER/DECEMBER 2009.

Demos and various technologies

a) Visual Attribute Explorer – Java program: http://www.alphaworks.ibm.com/tech/visualexplorer
b) TableLens – Java applet: http://www.inxight.com/products/sdks/tl/ (see demos on right)
c) Spacetree: http://www.cs.umd.edu/hcil/spacetree/ (Available as a downloadable app, Java Webstart, Java Applet)
d) See also Tamara Munzner's work on the SequenceJuxtaposer http://olduvai.sourceforge.net/sj/, and the TreeJuxtaposer work at http://www.cs.ubc.ca/~tmm/papers/tj/
e) Jambalaya available as a Java applet and Java client side (http://www.thechiselgroup.org/jambalaya)
f) http://www.cs.kent.edu/~zhao/vis13/lectures/flowvis.pdf
g) http://infovis.cs.vt.edu/oldsite/papers/Papers.html
h) http://graphics.ethz.ch/teaching/former/scivis_07/Notes/Slides/01-intro.pdf