InfoVis Systems & Toolkits

CS 7450 - Information Visualization
September 24, 2012
John Stasko

Background

- In previous classes, we have examined different techniques for presenting multivariate data
  - We’ll continue to show more later too

- Today we look at systems that implement these ideas and provide some of their own new visualization techniques
Agenda

- Systems providing a view or views
  - Improvise, ILOG Discovery, Many Eyes, Polaris
- Toolkits that can be used to build systems
  - prefuse, Piccolo, protovis, D3, ...
- Commercial systems (to come in a future class)
  - Spotfire, InfoZoom, InfoScope, Tableau, ...

Systems/Tools

- Primarily examining academic systems that provide preexisting views
  - Commercial systems next time
**Improvise**

- System used to build visualization tool
- Modular library of visualization components
- Multiple views key
- Coordination model coupled with data processing language at its heart

Weaver
InfoVis '04

---

**Improvise**

- Visualization authoring
  - Put together combinations of well-known views and coordinations
  - Designed for people to build these combinations, but that process is non-trivial (Chris admits)
Key Components

- **Live Properties**
  - For coordinating controls (e.g., views, sliders)
  - Control defines live properties that bind to variables

- **Coordinated Queries**
  - Visual abstraction language
  - Visually encode data records into graphical attributes

http://www.cs.ou.edu/~weaver/improvise/
Application

- Understand spatial and temporal patterns of movements of people and groups of people, understand commerce and travel patterns
- Focus: Social networks of people and historical data
- Alternate: Understand actions and movements of people in terrorist groups

Particulars

- Who – Historical geographers, geneologists
- Problem – Provide platform where these people can do research and more easily understand the movement patterns
- Data – Old hotel records & registers
Data Collection

- Two hotel registers in PA
  - ~10k guest entries
  - Hundreds of place of origin (mostly NE)
- Transcribed into spreadsheets (400 hours)
- Many repeat visitors
- 6% of dataset not legible/usable

Preliminary Data Analysis

- Simple tables generated
- GIS software for mapping geographic locations
Visualization

- How?
  - Consult with historical geographers
  - Iterative design

- Multiple, coordinated views
Looking for patterns of visitation

Showing time of year of visits
Data Analysis

- Through interaction with views, analysts can pose queries and look for answers
- “For these people on this date, did they ever meet prior to that?”
- Can also ask questions for unknown people (match a ?)
Data Attributes Explored

- Weekly circuits
- Weather effects
- Biweekly circuits
- Particular hotel characteristics
- Groups & gatherings
- Non-periodic regular visits

http://www2.ilog.com/preview/Discovery/
Characteristics

- Large collection of views including scatterplots, parallel coordinates, treemaps, heatmaps, bar charts, ...

Polaris

http://www.graphics.stanford.edu/projects/polaris/

http://www.tableausoftware.com

Video

Stolte et al

TVCG ’02
Basis

- Relational databases
- Pivot tables from spreadsheets
- N-dimensional data cubes

- Analytic approach is fundamental
- Provides visual representations of these concepts

Visualization

- Table of data (rows, columns)
- Each axis may have nested dimensions
- Table entry is a pane, and has visual marks to represent data

- Analyst drags and drops fields from database schema onto shelves of display
- Much interaction supported
Graphics

- Formal table algebra provided to describe data
- Visual mappings established from data types to appropriate (good) markings and encodings

Interaction

- Four main types
  - Deriving additional fields
  - Sorting and filtering
  - Brushing and tooltips
  - Undo and redo
Many Eyes

- InfoVis on the web
- Website developed from IBM’s infovis group
- Motivating infovis challenges:
  - Difficulty of creating new visualizations
  - How do you discuss the visualizations?
- Project goals:
  - Enable end-user creation of visualizations
  - Foster large-scale collaborative usage

Features

- Provides data upload capabilities and choice from library of visualizations
- Includes
  - Gallery of recently uploaded visualizations for browsing
  - Chosen highlighted visualizations
  - Attached discussion forums for each vis
Data

- Users upload their own data sets
  - All become public
- Format: table or unstructured text
  - Metadata allowed
- Immutable once uploaded

Visualizations

- Preloaded visualization types
  - Has grown over time
- User chooses one to combine with their data
- Provides named, typed slots that the user maps particular pieces of data to
  - System makes some reasonable guesses too
Social Aspects

- Users identified by login ID
- Can leave comments about different visualizations
- Can take snapshot of visualization state
  - Unique URL
- “Blog this” button

Fig. 3. Three user generated visualizations offering different perspectives on the same dataset on car fuel economy. The grey areas on the top and bottom are automatically generated by the application and allow the user to browse through different dimensions in the data.

Allows the user to control the mapping from data to image
Evaluation

• Quantitative, objective
• 1895 posts of March ‘07
• Wide variety of topics of visualizations and motivations for creating visualizations
• Does seem to be fostering discussion

Viégas et al
HICSS ’08

Evaluation

• Qualitative, subjective
• In-depth interviews with some ME users
• Visualizations used largely for communication and collaboration (not necessarily analysis)
  – Privacy and audience management a concern
• Highlights a number of interesting, non-expected uses of the technology

Danis et al
CHI ’08
Watch More

- Video of talk from Stanford HCI seminar

Thoughts?

- What do you think of the design choices they made?
Sense.us

- Related, follow-up to Many Eyes
- Discussion and visualization of US Census data
- Go beyond Many Eyes in terms of annotation, collaboration, and discussion

Components

- Doubly-linked discussion
  - Can go from visualization to threaded discussion items or vice-versa
- Graphical annotation
  - Simple graphics editor and comments (like transparent layer)
- Bookmark trail
  - Small strip of views
- Comment listings and social navigation
  - Searchable, sortable indices and links
Figure 1. The sense.us collaborative visualization system. (a) An interactive visualization applet, with a graphical annotation for the currently selected comment. The visualization is a stacked time-series visualization of the U.S. labor force, broken down by gender. Here the percentage of the work force in military jobs is shown. (b) A set of graphical annotation tools. (c) A bookmark trail of saved views. (d) Text-entry field for adding comments. Bookmarks can be dragged onto the text field to add a link to that view in the comment. (e) Threaded comments attached to the current view. (f) URL for the current state of the application. The URL is updated automatically as the visualization state changes.

Figure 5. Scatterplot of U.S. states showing median household income (y-axis) vs. retail sales per capita (y-axis). New Hampshire and Delaware have the highest retail sales.

Figure 6. Visualization of the number of teachers. Annotations indicate the start of compulsory education and the rise of teachers in the post World War II era.

Figure 7. Annotated view of stock brokers. The attached comment reads, "Great depression sold a lot of brokers.

Sample annotations
Microsoft Pivot

- System unveiled to great publicity from LiveLabs
  - After shut-down, has mostly disappeared
- Faceted browsing/filtering with image-based UI

Demo

[Image of the interface for Endangered Species]
Toolkits & Infrastructures

- Set of components or capabilities that allow others to put together visualization systems

- Growing trend

InfoVis Toolkit  
http://ivtk.sourceforge.net
Characteristics

• Coherent software architecture and set of Java components
  – Data structure is a table
  – Views: time series, parallel coordinates, scatterplots, node-link diagrams, treemaps
  – Added capabilities in color management, labeling, dynamic queries, ...

Critique

• Primarily limited to views it provides, but good & powerful support for them
• Documentation?
Prefuse

- Toolkit in Java for building information visualizations

Beginning in a series of toolkits from Jeff Heer and his research group

Heer et al
CHI ’05

http://prefuse.org

Demo
Video
Characteristics

- Fine-grained building blocks for constructing visualizations (as opposed to pre-made views)
- Data model is a graph (entities & relations)
- Includes library of layout algorithms, navigation and interaction techniques
- Written in Java2d
- Follows CMS vis process model
Applications

- DOI Trees (AVI ’04)
- Vizster (InfoVis ’05)
- Many more

Flare

- New version of prefuse implemented in ActionScript for running in Flash player

http://flare.prefuse.org

Demos
Protovis

- New toolkit following a declarative model
  - Simple grammar of graphical primitives called marks
  - Marks are associated with data, using mapping to physical properties
  - Properties can be dynamic or static
  - Register event handlers to provide interactivity

Built-in Mark Types

Fig. 5. Examples of built-in mark types. (a-h) Area; Bar; Dot; Image; Line; Label and Bar; Rule and Bar; Wedge.
Examples

```javascript
new pv.Panel().canvas("fig3a")
  .add(pv.Bar)
  .data([1, 1.2, 1.7, 1.5, .7, .2])
  .bottom(0).width(20)
  .height(function(d) d * 80)
  .left(function() this.index * 25)
  .root.render();

new pv.Panel().canvas("fig3b")
  .data(function(d) d)
  .fillStyle(pv.Colors.category19.parent)
  .bottom(function() let { c = this.cousin() }
  .c ? (c.bottom + c.height) : 0)
  .height(function(d) d * 40)
  .left(function() this.index * 35)
  .root.render();
```

Fig. 2. Specifying two simple charts. (a) Bar. (b) Stacked area.

Gallery
**D³: Data-Driven Documents**

- Newest entry in the Heer-Bostock line of toolkits
- “Not just an infovis toolkit”
- Javascript-based
- Very similar to Protovis…
  - Except makes use explicitly of web standards such as Scalable Vector Graphics (SVG) rather than a proprietary “marks” graphics set

**D3 Design Pattern**

- Declarative Syntax like Protovis
- Creating/Modifying selections of the HTML DOM
- “An elegant for-loop with a bunch of useful helper functions”
- Excellent support for changing data
  - Taking advantage of CSS3 Transformations and Transitions
- Integrates seamlessly into any webpage
D3 Website

D3 Examples


- “Show Reel”: http://bl.ocks.org/1256572


- Box Plots: http://mbostock.github.com/d3/ex/box.html
D3 Support

- Active community online
  - Including Mike Bostock often answering questions
- In active development
  - Though the critical features are unlikely to change
- Chad is GT’s “D3-expert-in-residence”

Design Patterns

- Heer used his experience with prefuse etc to describe common design patterns for infovis
  - Reference model
  - Data column
  - Cascaded table
  - Relational graph
  - Proxy tuple
  - Expression
  - Scheduler
  - Operator
  - Renderer
  - Production rule
  - Camera
  - Dynamic query binding

Heer & Agrawala
TVCG (InfoVis) ’06
Piccolo

http://code.google.com/p/piccolo2d

Characteristics

- Graphics toolkit with built-in zooming and panning support
- Very nice for implementing infovis
- Will discuss more later in course...
InfiView

- Environment for developing web-based data visualizations
- Maps, charts, graphs, ...
- Ajax-powered
- Focus on speed and connecting to large databases

Google Chart Tools

http://www.infiview.com

http://code.google.com/apis/visualization/documentation/
Characteristics

- Javascript-based
- Gallery of contributed code segments
- Visualizations are interactive
- Evolving API

Processing

- Java based
- Unlike prefuse & flare, not specifically designed for InfoVis
  - Data Reader? Layout algorithm?
  - But can definitely be used to build visualizations!
- Well documented, lots of tutorials with contributions from many people and even books

Ben Fry
Processing: the idea

- Programming as scripting
  - PDE: processing development environment
  - A program is called a *sketch*
  - written as a list of statements

The Pain of using Java2D

- Graphics repaint
- Animation
- ActionListener, EventListener ...
- Long-winded Java syntax

- Processing makes many of these things much easier
Processing: Useful Functions

void setup() {
    //your own code here
}

void draw() {
    //your own code here
}

• These are built-in functions that are called automatically.
  – The setup() block runs once.
  – The draw() block runs repeatedly: good for animation

Reflection

• What would you seek in a good infovis system or toolkit?
Toolkit Tradeoffs

- **InfoVis-focused**
  - Many fundamental techniques built-in
  - Can be faster to get something going
  - Often more difficult to implement something “different”
  - Doc is often poor

- **Generic graphics**
  - More flexible
  - Can customize better
  - Big learning curve
  - Doc is often better
  - Can take a long time to (re)implement basic techniques

More Next Time...

- Demos of commercial infovis systems
  - Spotfire
  - Table Lens/Eureka
  - SeeIt
  - InfoZoom
  - InfoScope
  - Tableau
Project Topics Feedback

- Returning your proposals
- Get to work – Poster session is in less than a month

Upcoming

- Commercial systems
  - Reading
    - Spenke & Beilken '00

- Interaction
  - Reading
    - Yi et al '07