Data Visualization (DSC 530/CIS 602-01)

Data

Dr. David Koop
Focus+Context Overview

**MBED**
- Elide Data
- Superimpose Layer
- Distort Geometry

**Reduce**
- Filter
- Aggregate
- Embed

[Munzner (ill. Maguire), 2014]
Elision: DOI Trees

[Heer and Card, 2004]
Superimposition with Interactive Lenses

(a) Alteration

(b) Suppression

[ChronoLenses and Sampling Lens in Tominski et al., 2014]
Superimposition with Interactive

(c) Enrichment

[Extended Lens in Tominski et al., 2014]
It can be difficult to observe micro and macro features simultaneously with complex graphs. If you zoom in for detail, the graph is too big to view in its entirety. If you zoom out to see the overall structure, small details are lost.

Focus + context techniques allow interactive exploration of an area. Mouseover to distort the nodes.

[M. Bostock, http://bost.ocks.org/mike/fisheye/]
Cartesian Distortion

[M. Bostock, http://bost.ocks.org/mike/fisheye/]
Cartesian Distortion

[M. Bostock, http://bost.ocks.org/mike/fisheye/]
(a) Bring (step 1) – Selecting a node fades out all graph elements but the node neighborhood. (b) Bring (step 2) – Neighbor nodes are pulled close to the selected node. (c) Go – After selecting a neighbor (the green node in Fig. 4(b)), a short animation brings the focus towards a new neighborhood.
Venn Diagram

http://askville.amazon.com/idea-Venn-diagram/AnswerViewer.do?requestId=8420613
Scalability

- How to show the intersection of four sets? 8?
- Euler Diagrams: only show intersections/containments that exist
- Still run into scalability issues
Figure 7: Grouping research articles on a timeline. 

Figure 8: Items can be expanded to reveal a larger image or the article's abstract. The boundary moves to accommodate the larger item and other items move along the y-axis to remain visible and selectable. Sets of hotels, subway entrances, and medical clinics may help them find a hotel that is central to several medical clinics and near a subway entrance.

4.4 Sets over Scatterplots

Scatterplots have clearly defined spatiality due to the numerical positioning of items. We add Bubble Sets to a reimplementation of the well-known GapMinder Trendalyzer. This scatterplot shows fertility rate against life expectancy and is animated over time. Data points represent countries, sized by population. Colour and set membership is defined by the continent. The grouping of the Sub-Saharan Africa countries, highlighted in Figure 6, reveals that while most of the countries in this set had high fertility rates and low life expectancies in 1996–1999, there are two outliers, Mauritius and Reunion, which are islands in the Indian Ocean. As the data set includes data for many years, and since Bubble Sets are calculated at interactive rates, the temporal changes can be convincingly shown through animation.

5. DISCUSSION AND FUTURE WORK

We have presented Bubble Sets, a method for automatically drawing set membership groups over existing visualizations with different degrees of requirements for primary spatial rights. In contrast to other overlaid containment set visualizations, Bubble Sets maximizes set membership inclusion and minimizes inclusion of non-set members. In fact, Bubble Sets can guarantee that all set members will be within one container, as opposed to the more common multiple disjoint containers. While Bubble Sets cannot guarantee non-set member exclusion, the routing algorithm minimizes these occurrences.

Within our isocontour approach we have implemented several heuristics to reduce surface calculation and rendering time, such as grouping pixels for potential calculations and restricting the regions in which items influence the potential field. The current implementation works without noticeable lag; items can be dragged and the surface follows; for our examples, order of 655 nodes, 65–85 sets. For example, it takes on average 65 ms to calculate the virtual edge set, fill the energy field, find the contour, and render the Sub-Saharan Africa set in a window size of 685x685 pixels. That set has 0-64 items and the entire scatter plot has 65–685 points. The majority of this time is spent creating the virtual edge set. An incremental approach, using A* search as in [89], may provide improvements in speed and stability.

As the number of items, the screen resolution, or the number of sets increases, so will the rendering time. Additional techniques, such as grouping close items into larger pseudo-nodes, and caching the energy field values between frames, may increase the capacity of the system.

ACKNOWLEDGMENTS

Thanks to the following organizations for supporting this research (iCORE, CFI, NSERC, and SMART Technologies). [Collins et al., 2009]
To understand the advantages and drawbacks of our technique, we performed a controlled experiment with 13 participants, comparing KelpFusion to Bubble Sets [7] and LineSets [1]. We discovered that KelpFusion improved on Bubble Sets, outperforming the technique in accuracy and completion time. We also found that KelpFusion was on par with LineSets in terms of accuracy but yielded faster completion time.

Figure 1 illustrates three existing methods, Bubble Sets, LineSets, and Kelp Diagrams, in comparison to KelpFusion. Figure 1(a) shows the visualization generated by Bubble Sets, Figure 1(b) shows the visualization generated by LineSets, Figure 1(c) shows the visualization generated by Kelp Diagrams, and Figure 1(d-f) shows the visualizations generated by our KelpFusion implementation.

KelpFusion

(1) Bubble Sets
(2) Kelp Diagrams
(3) LineSets
(4) KelpFusion (dense)
(5) KelpFusion (sparse)

[Meulemans et al., 2013]

To automatically draw Euler diagrams to convey abstract set intersections, several papers have explored the problem of set topology, for example, Simonetto and Auber [14]. Other approaches investigated the use of proximity graphs, for example, Simonetto [16]. Shortest-path graphs have been used to delineate imprecise regions, representing set intersections. In these diagrams, closed curves correspond to sets and overlaps between the curves indicate set intersections. Several papers have explored the problem of constructing a boundary of a region based on points that are likely inside the intended region [2]. Shortest-path graphs allow KelpFusion to fill faces when shortest-path graph and its corresponding boundary can be computed efficiently, enabling interactive manipulation of shape and clusters of a point set. In other words, the use of a spanning graph, KelpFusion introduces the use of a shortest-path graph, for example, Simonetto and Auber [14] and Simonetto [16]. Other approaches investigated the use of proximity graphs, for example, Simonetto [16]. Shortest-path graphs allow KelpFusion to fill faces when shortest-path graph and its corresponding boundary can be computed efficiently, enabling interactive manipulation of shape and clusters of a point set. In other words, the use of a spanning graph, KelpFusion introduces the use of a shortest-path graph.
Projects

- Feedback
- Designs:
  - Show what is done so far and different ideas along the way
  - Sketch/mockup items that haven't yet been integrated
  - Note interactions/types of visualizations
  - Concentrate on the design of the visualization
  - Think about how your design impacts the ability to answer questions
Data Wrangling

- Problem 1: Visualizations need data
  Solution: The Web!
- Problem 2: Data has extra information I don't need
  Solution: Filter it
- Problem 3: Data is dirty
  Solution: Clean it up
- Problem 4: Data isn't in the same place
  Solution: Combine data from different sources
- Problem 5: Data isn't structured correctly
  Solution: Reorder, map, and nest it
Hosting data

- github.com
- gist.github.com
- myjson.com
- Google Drive
- Other services
Why JavaScript?

• Python and R have great support for this sort of processing
• Data comes from the Web, want to put visualizations on the Web
• Sometimes unnecessary to download, process, and upload!
• More tools are helping JavaScript become a better language
Online JavaScript Resources

- [http://learnjsdata.com/](http://learnjsdata.com/)
- Good coverage of data wrangling using JavaScript
LeafletJS Example:
New York City Murder by Precinct