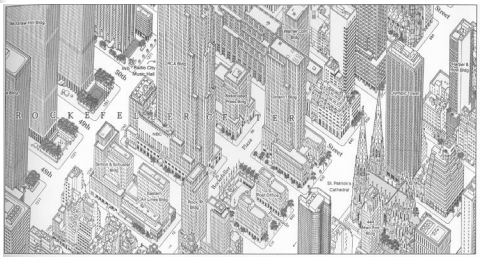
Interacting with Large Data Sets

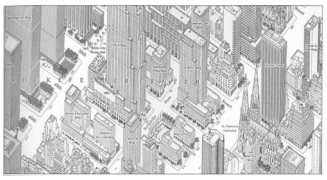
Richard Anderson
Ken Fishkin

Issue is not *amount* of data



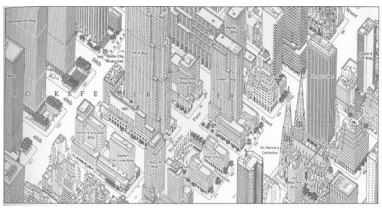
Issue: dimensionality

- date built, property value, architect, owner, wheelchair-accessible, building-code violations, seismic stability, etc., etc.



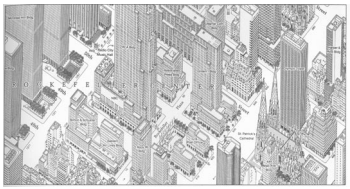
Issue: connections

- buildings with same architect



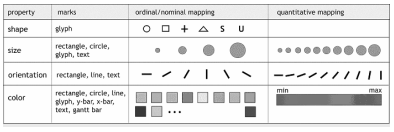
Issue: flexibility

- Everybody cares about different things
- architect vs landlord vs tenant vs ….



Issue: dynamic creation

- Don't have time to carefully and precisely plan each visualization, a la Tufte – need to dynamically generate layout, colors, etc. on the fly.



Topic: How do you

- A) visualize the set
- B) define subset of data to show
  - The 'FROM" and 'WHERE' clauses
- C) visualize an element in the set
  - SQL 'SELECT … FROM' clause
- D) visualize connections between elements
- E) navigate between elements
- F) change query

Two Main Types of Techniques

- Those that focus on *data* (e.g. Stolte)
  - Emphasis is on forming queries, showing results
- Those that focus on *connections between data* (e.g. Ping)
  - Emphasis is on interactive nagivation, showing space of elements
- Let's do connections first (easier!)

Showing connections

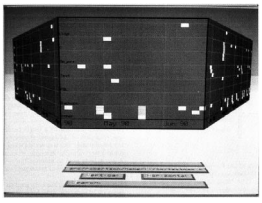
- Many techniques ignore this
- Whole new ball of wax
- Examples of when useful: org. chart, bib. Citations, internet message traffic, gnutella

Connections: (A) showing set

- Need to scale well for large networks
- All employ various "fade-out" techniques
  - Ping: works on arbitrary graphs, so long as 1 conn. Component.

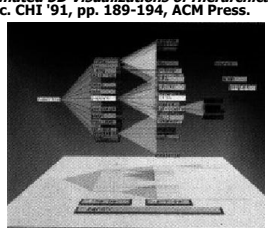
Perspective

- **J.D. Mackinlay, G.G. Robertson and S.K. Card, *The Perspective Wall: Detail and Context Smoothly Integrated*, p. 173-180, Proceedings of CHI'91**



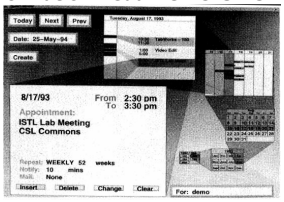
Cone Tree

- **Robertson, G. G., Mackinlay, J. D. and Card, S. K., *Cone Trees: Animated 3D Visualizations of Hierarchical Information*, in Proc. CHI '91, pp. 189-194, ACM Press.**



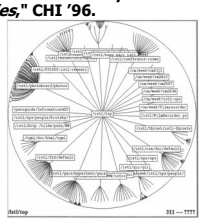
Spiral

- **Mackinlay, J., Robertson, G., and Deline, R. *Developing calendar visualizers for the information visualizer*. UIST '94.**



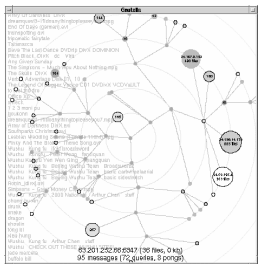
Sphere

- **Lamping, J. and Rao, R., "*The Hyperbolic Browser: A Focus + Context Technique for Visualizing Large Hierarchies*," CHI '96.**



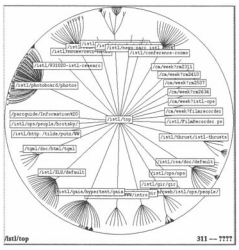
Circle

- (Ping). Works on arbitrary graphs, so long as 1 connected component



Connections: (B) define subset of data to show

- Typically show *entire* set of data (context)
- With a "distinguished element" at the center (focus)
- (not emphasis of Ping paper, this is from Lamping)

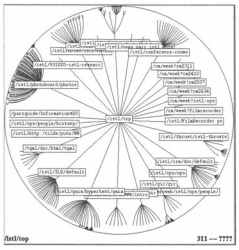


Connections: (C) show an element

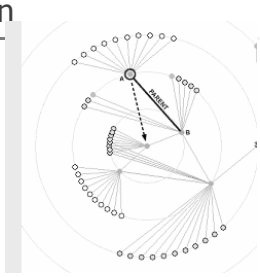
- Not focus of this technique, often minimal:

Connections: (D) show inter-connections

- Many techniques ignore this
- But center of these techniques



Connections: (E) navigate between



Connections: (F) change query

- Navigation *is* implicitly changing the query

Yee Paper: Summary

- How to visualize networks
  - Small amount of data/node
- How to intuitively do transition
- Handles scaling problem by mapping to circle – not infinite, but good
- Allocate children radially, space is F(subtree size).
- Bit of a kludge with non-tree edges (don't "count" in tree size)
- Transition done via polar rotation
- Also vector between old and new root maintained
- Also animation used (slow-in, slow-out)

Digression

- This work started from a class project in a class similar to this

Digression

- "it makes more sense to linearly interpolate the polar coordinates of the nodes, rather than their rectangular coordinates"
  - WHY?
  - Their answers:
    - "Since the nodes are radially positioned".
    - Clustering nodes "reduces effort to understand the animation", since nodes move in chunks
    - Moving in arcs is "an effective technique from traditional animation"
  - My answer: To better match the metaphor
    - This is theme we will expand upon later

Stolte Paper

- Much harder problem. Why?
  - A) handle infinite number of fields
  - B) handle queries and show data of various types, with various visualizations
- They show a "swiss army knife" that can generate many different queries/visualizations.

(But….. It's really even harder)

- In "real life", most tables use numerous levels of 'join', which these techniques largely gloss over.
  - E.g. the query "find customers who ordered a book from a publisher in their same state" will typically require at least: (Customer JOIN order) (JOIN asset) (JOIN book) (JOIN publisher)
- Some queries are virtually impossible to answer any other way than via a complex SQL query, e.g. "find all publishers in New York who have sold > 1000 Bibles in the last year"

Step 1: Pick major axes

- Drag-and-drop 2 fields onto X and Y axes, "shelves"
- Minimal filtering (bounds, undefined subset-choosing).
- All fields are of 2 types:
  - Quantitative (i.e. real #s: price, salary…)
  - Ordinal (i.e. enum's: state, month, …)
- So we get 3 possibilities:

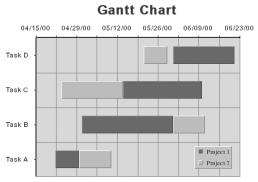
1) Quant vs. Quant

- Classic chart or scatterplot
- Chatterplot: (student height vs. GPA)
- Can turn scatterplot into graph by using SUM, AVG, MIN, etc.: (student height vs AVG(GPA))

2) Ordinal vs. Quant

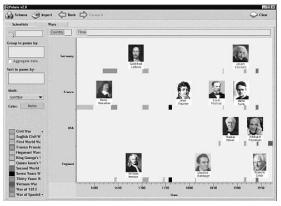
- Bar chart, dot plot, or gantt chart
- Bar chart: (birth state vs. GPA)

Gantt chart: like Microsoft project timelines:



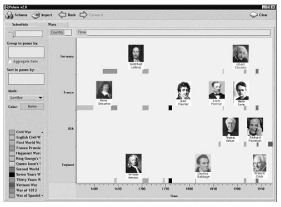
Their example:

- How did they specify the join from country to inventors?
- How did they specify join from inventor to picture?
- How did they get 2 different visualizations on the display?



Their example:

- "Multiple data sources may be combined in a single Polaris visualization: Each data source maps to a separate layer or set of layers"

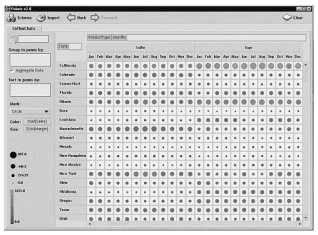


3) Ordinal vs. Ordinal

- Axis variables are typically independent (e.g. birth state vs. birth month)
- You can then introduce additional dimensions

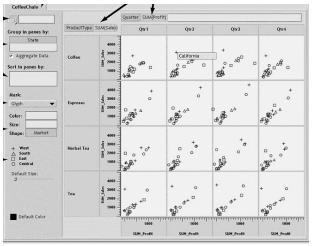
Ordinal vs. Ordinal – table

- Entries in table show some quantity or quantities
- Color:SUM(Sales)
- Size:SUM(Margin)



Going multi-dimensional

- Can add more dimensions by turning 1 or both axes into an "accordion", gives you an array of smaller visualizations.

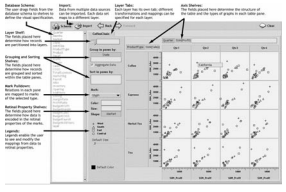


Stolte: (A) showing set

- Not done. You could do a "null query" and get a scatterplot, if you so desired.

Stolte(B) defining subset

- Drag-and-drop fields of interest
- Can use range queries on Quant. Fields



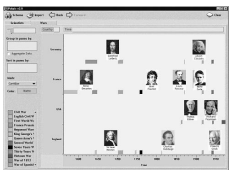
Stolte(C) show an element

- Smorgasbord of graphical techniques, attempt to do it automatically

property	marks	ordinal/nominal mapping	quantitative mapping
shape	glyph	○ □ + △ S U	
size	rectangle, circle, glyph, text		
orientation	rectangle, line, text		
color	rectangle, circle, line, glyph, y-bar, x-bar, text, gantt bar		min max

Stolte(D) show inter-connections

- Can overlay graphs, but not their emphasis.
- Supports "brushing"

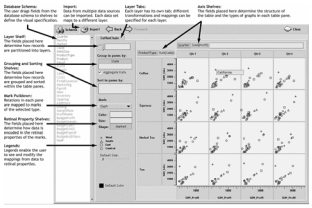


Stolte(E) Navigate between elements

- N.A. – you build up queries and modify them, use GUI.

Stolte(F) Change query

- Use drag-and-drop, pull-downs, etc.



Project

- One reason queries are so complicated is because they work in "SQL space". Could they be more readily described in "tuple space"? What if the "operands" are tuples in the database, and the queries become "find more like these", "compare ones like this to ones like those", etc.

Project

- Amazon.com has made a subset of its database available for Web programming (http://associates.amazon.com/exec/panama/associates/ntg/browse/-/567632) . Here both the elements, and the connections between them, are available and of interest. Explore visualization techniques which show both elements and connections, using the Amazon data.

Project

- All these techniques focus on showing how the data is *now*. Sometimes, what is also (or even mainly!) of interest is showing *trends* in the data and its interconnections over time. Explore visualization techniques that focus on deltas in connections over time (possible IRS project).

Projects from Yee

- Small graphs and large ones had different preferences for transition models. This implies that perhaps a third model would do better. Find it.
- Extend it to work on very large graphs (as they suggest)
- Extend it to show temporal changes – right now, only done if watching animation.

Project

- New "fade-out" technique

Next Time

- No readings!
- Guest Lecturer: Alan Borning, will talk about "Visualization Challenges in Urban Sim"
- This will end the Visualization exploration – next stop, Pen Computing