Scalable Visualization with Accordion Drawing
Tamara Munzner
University of British Columbia
5 February 2005
Vancouver Studies in Cognitive Systems 2005

Accordion Drawing
rubber-sheet navigation
- stretch part of surface
- the rest squishes
- borders nailed down
- helps maintain orientation

guaranteed visibility
- landmarks stay visible
- never offscreen
- visible mark even if very squished
- helps guide navigation choices

Accordion Drawing Framework
infrastructure for motion, marking, rendering

dexample datasets
- trees built-in hierarchical structure
- gene sequences
dense, partially vertically correlated
- transactions in power set space
- very sparse, huge space

Outline

Accordion Drawing

Example Applications
- Trees
- Sequences
- Power Sets

Rubber-Sheet Navigation

Focus+Context technique
- merge overview and details for single combined view
- rectilinear, multiple foci [Sarkar 94, Robertson 91]
Hierarchical Grid

two directions, horizontal and vertical

two valid interpretations for SplitLines
  - linear ordering
  - hierarchical subdivision of space
    child splits parent in two

application maps from 2D layout to grid

Grid Motion

move a set of SplitLines
  - grow several regions simultaneously
  - shrink the rest
    - new $O(k \log n)$ algorithm
      - $k =$ # lines to move
      - $n =$ # lines total
    - robust calculation, move each line only once

Guaranteed Mark Visibility

Marks

regions of interest shown with color highlight
  - differences between datasets
    - search results
    - user selections

guide navigation
  - safe to avoid empty places
    - no false negatives, lack of mark meaningful
  - investigate marked areas
    - squished marks are visible placeholders
    - seeing details still requires navigation

provide landmarks
  - relative positions stay the same
    - “green area I looked at first is underneath blue one”

Guaranteed Visibility

infrastructure needed for efficient computation

relief from exhaustive exploration
  - missed marks lead to false conclusions
  - hard to determine completion
  - tedious, error–prone

compelling reason for Focus+Context
  - controversy: does distortion help or hurt?
  - strong rationale for comparison

constraint to fit everything in viewport
  - instead could show indirectly
    - ideas: Halo [Baudisch 03]

Guaranteed Visibility Previous Work

visibility of abstract information
  - effective view navigation [Furnas 97]
  - critical zones [Jul and Furnas 98]
How Could Marks Disappear?

moving outside viewport
- choose global Focus+Context navigation “tacked-down” borders
  as opposed to free camera motion

occlusion
- choose 2D++ layout
  as opposed to 3D layout

culling at subpixel sizes
- develop efficient check for marks when culling

Focus+Context Previous Work

combine overview and detail into single view

Focus+Context
- large tree browsing
  - Cone Trees [Robertson et al 91]
  - Hyperbolic Trees [Lamping et al 95, Munzner 97]
  - Space Tree [Plaisant et al 03]
  - DOI Tree [Card and Nation 02]
- global
  - Document Lens [Robertson and Mackinlay 93]
  - Rubber Sheets [Sarker et al 93]

our contribution
- scalability, guaranteed visibility

Rendering

rubber sheet navigation challenges
- depth complexity changes quickly
  - can be extremely high, thousands of objects per pixel

guaranteed visibility challenges
- avoid overculling
  - violate guaranteed visibility constraint
- avoid underculling
  - inefficient, overwrite same pixel multiple times

want render time to depend on screen area
- not size of dataset

Near-Constant Rendering Time

Rendering Time: Trees (3.5M)
**Rendering Time: Power Sets (1.5M)**

- Block size: 4 columns: 64
- Block size: 8 columns: 64
- Block size: 16 columns: 64
- Real block size: 4 columns: 64

![Graph showing rendering time vs total transaction size](image)

**Rendering Time: Sequences (40M)**

- Single
- Wide
- Square
- Taza
- Trexila
- Orion
- Murphy

![Graph showing rendering time vs total sequence size](image)

**Preprocessing Time: Sequences (70 sec)**

- Single
- Wide
- Square
- Taza
- Trexila
- Orion
- Murphy

![Graph showing preprocessing time vs total sequence size](image)

**Scalability Limits**

- Memory footprint is limitation
  - Everything must fit into main memory
- Previous systems
  - TJ: 250–500K nodes
  - SJ: 1.7M nodes
- Now
  - TJ: 3.5M nodes
  - SJ: 40M nodes
  - PSV: 1.5M nodes

**Linear Memory Usage**

- Balanced
- Class(1)
- Class(2)
- OpenDirectory(1)
- OpenDirectory(2)

![Graph showing memory usage vs tree size](image)

**Memory Usage: Trees (3.5M)**

- Balanced
- Class(1)
- Class(2)
- OpenDirectory(1)
- OpenDirectory(2)

![Graph showing memory usage vs tree size](image)
Outline

Accordion Drawing

Example Applications
- Trees
- Sequences
- Power Sets

TreeJuxtaposer

active area: hierarchy browsing
- previous work: browsing
- tree comparison was still open problem

bioinformatics application
- phylogenetic trees reconstructed from DNA
- rectilinear layout, following conventions

Phylogenetic/Evolutionary Tree

[Meegaskumbar et al., Science, 298:379 (2002)]

Common Tree Size Now

[Meegaskumbar et al., Science, 298:379 (2002)]
Tree of Life: 10M Species

Comparing Trees
- multiple trees
  - from phylogenetic reconstruction
  - algorithms return many possibilities
- comparing contiguous groups
  - clade: ancestor + all descendants
  - is a clade in one tree also a clade in other?
  - is some group a clade?

Paper Comparison
- focus
- context

Biologists’ Requirements
- reliable detection of structural differences
  - rapid identification of interesting spots
- analyses of differences in context
  - mostly side by side comparison
- manipulation of increasingly larger trees
- support for multiple platforms
  - Java with OpenGL bindings

TreeJuxtaposer Contributions
- first interactive tree comparison system
  - automatic structural difference computation
  - guaranteed visibility of landmark areas
- scalable to large datasets
  - 250,000 to 500,000 total nodes
  - new work: 3.5 million nodes
  - all preprocessing subquadratic
  - all realtime rendering sublinear
- techniques broadly applicable
  - not limited to biological trees
- overall winner: InfoVis Contest 2003

Outline
- Accordion Drawing
- Example Applications
  - Trees
    - Sequences
  - Power Sets
SequenceJuxtaposer
accordion drawing for DNA/RNA

previous work: web–based sequence browsers
- Ensembl, UCSC Genome Browser, NCBI MapViewer
  - heavily used, huge server–side databases
- zoom or pan in jumps
- can’t see context

fluid Focus+Context navigation
guaranteed visibility
- establish when these features useful
- proof of concept prototype, eventually merge

SJ in action
shown on publicly available data
- onion yellows phytoplasma: whole genome
  860 Kbp
- Murphy: 22 genes
  44 mammals x 17000 bp each = 748 Kbp
- Treezilla: single gene
  500 plants x 1428 bp each = 714 Kbp

[videos]

previous paper: 1.7 million nucleotides
currently: 40 million nucleotides

Outline

Accordion Drawing

Example Applications
- Trees
  - Sequences
  - Power Sets

PowerSetViewer: Steerable Data Mining
investigating transaction logs

setting parameters for filtering operations
- classic problem: too much or too little
  - engine allows parameter changes midstream

have a steering wheel: steerable data mining

need a windshield: visualization
- want meaningful spatial layout as parameters change
- scalability issue: what if filter is null?
  - entire log passed through to viz client

Transactions As Sets
market–basket transactions are sets
- A bought (bread, milk, eggs)
  - B bought (bread, chocolate, cat food)

alphabet: universe of possible items to buy
- all items in grocery store

space of all possible transactions
- set of all possible sets: power set
  - huge, but only sparsely populated
  - show distribution of log data within absolute space
    of possibilities
  - accordion drawing preserves relative order
Enumeration of Power Set

order first by cardinality (set size)
within cardinality, order by alphabetical order
[a], [b], [c], [ab], [ac], [bc], [abc]

very long linear list
wrap scanline-style, at a fixed width
- 128 columns, millions of rows

with conventional display, couldn’t see anything
- everything smaller than a pixel

with guaranteed visibility, marks are visible
- construct hierarchical grid on the fly
- add and delete SplitLines as needed
- empty rows collapsed

Distribution of Transactions: 300K Log

- alphabet: items in grocery store
- transactions: items bought at once
- highlighting: sets containing specific item

Distribution of Transactions: 90K Log

- alphabet: available courses
- transactions: courses taken by student in one term
- highlighting: grad CS courses

Future Work

trees with weighted edges

sequence alignment editing

protein sequences

linking tree and sequence navigation

open-source release of power set viewer
- data mining: transaction processing

More information

olduvai.sourceforge.net
- open-source release of Tj, Sj

www.cs.ubc.ca/~tmm/papers.html
www.cs.ubc.ca/~tmm/talks.html
- papers, slides, images, movies

Published Papers

Treejuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility
Tamara Munzner, Francois Guimbretiere, Serdar Tasiran, Li Zhang, and Yunhong Zhou.

Sequencejuxtaposer: Fluid Navigation For Large-Scale Sequence Comparison In Context
James Slack, Kristian Hildebrand, Tamara Munzner, and Katherine St. John.

now PowerSetViewer: joint work with
Qiang Kong, UBC
Raymond Ng, UBC

new TJC, TJC-Q: joint work with
Dale Bremner, Virginia
Greg Humphreys, Virginia