7. Hierarchies & Trees

Visualizing topological relations

Thorsten Büring, 06. Dezember 2007, Vorlesung Wintersemester 2007/08
Outline

- Hierarchical data and tree representations
  - 2D Node-link diagrams
    - Hyperbolic Tree Browser
    - SpaceTree
    - Cheops
    - Degree of interest tree
    - 3D Node-link diagrams
  - Enclosure
    - Treemap
    - Ordered Treemaps
    - Various examples
    - Voronoi treemap
    - 3D Treemaps
  - Circular visualizations
    - Space-filling node-link diagram
Hierarchical Data

- Card et al. 1999: data repository in which data cases are related to subcases
- Many data collections have an inherent hierarchical organization
  - Organizational Charts
  - Websites (approximately hierarchical)
  - File system
  - Family tree
  - OO programming
- Hierarchies are usually represented as tree visual structures
- Trees tend to be easier to lay out and interpret than networks (e.g. no cycles)
- But: as shown in the example, networks may in some cases be visualized as a tree

Figure 1: Visualization of the Gnutella network.

Yee et al. 2001
Tree Representations

- Two kinds of representations
- Node-link diagram (see previous lecture): represent connections as edges between vertices (data cases)
- Enclosure: space-filling approaches by visually nesting the hierarchy

http://www.icann.org
Outline

- Hierarchical data and tree representations
  - 2D Node-link diagrams
    - Hyperbolic Tree Browser
    - SpaceTree
    - Cheops
    - Degree of interest tree
    - 3D Node-link diagrams
  - Enclosure
    - Treemap
    - Ordererd Treemaps
    - Various examples
    - Voronoi treemap
    - 3D Treemaps
  - Circular visualizations
  - Space-filling node-link diagram
Node-Link Diagram

- Most conventional layout
  - Tree-depth is mapped to an ordinal Y-axis
  - X-axis is nominal – mainly used to separate siblings
- Can also be turned around
- Circular layout – root in the center with levels growing outward
Node-Link Diagram

- Unlike space-filling methods node-link diagrams provide an effective overview of the topology of a tree.

- **Problems**

  - Large trees require an extreme aspect ratio
    - Example: branching factor of 2
    - Trees get wider approximately proportionally $2^n$ ($n=level$)
    - Taller only proportionally to $n$
    - Large trees become to resemble a straight line

  - Trees usually contain considerably empty space (about 50%)

- InfoVis approaches to address these problems
  - Interaction
  - Distortion

Hyperbolic Tree Browser

Lamping et al. 1995

Comparable to fisheye distortion
- Nodes in the center are displayed at higher granularity
- Neighboring nodes are displayed in diminishing size

Maximum number of nodes displayed in a 600 x 600 pixel window
- Standard 2D hierarchy browser: typically 100 nodes with 3 characters text labels
- Hyperbolic browser: can display 1000 nodes with 50 nearest the focus can show from 3 to dozens of characters text labels

Approach exploits hyperbolic geometry
- Lay out hierarchy on hyperbolic plane and map plane onto a circular display region
- Property of hyperbolic plane: circumference of a circle grows exponentially with its radius
- Hierarchies tend to expand exponentially with depth
- Elegant match!

Inspiration:
Circle Limit IV by M.C. Escher
Hyperbolic Tree Browser

- **Navigation**: users select nodes to become the new center node (animated transitions)
- **Potential problem with orientation**:
  - nodes rotate during pure translation, e.g. node “Lowe” moves from top right to bottom right
  - Not suitable to present data such as organizational charts
- **Small-scale user test** (4 subjects, within-subjects design, IV: type of browser, DV: number of actions, time, preference)
  - No significant performance advantage over a 2D hierarchy browser with horizontal tree layout
  - Participants preferred the hyperbolic tree browser - provided “weaker sense of directionality of links”, but helped to “get(ting) a sense of the overall tree structure”

Lamping et al. 1995
SpaceTree

- Plaisant et al. 2002
- Mechanisms to facilitate large tree exploration / navigation
  - Dynamic rescaling of branches to fit the screen
  - De-composed animated transitions
  - Optimized camera movement
  - Preview icons summarizing branches collapsed (see top-down order)
    - Shading of triangle is proportional to the total number of nodes in the subtree
    - Height of triangle represents depth of subtree
    - Base of triangle proportional to average width (number of items / depth)
  - Search and filter functionality

- Demo
SpaceTree

- Experiment comparing 3 tree-browsing interfaces
  - Microsoft Explorer
  - Hyperbolic tree browser
  - SpaceTree
- Counterbalanced repeated-measures within-subject design
- 18 participants
- Tree with 7000 nodes
- Three task types
  - Node searches
  - Search of previously visited nodes
  - Answering topology questions
- Results
  - Hardly significant performance differences between the interfaces
  - Users found MS Explorer significantly less attractive then the other two interfaces
Cheops

Beaudoin et al. 1996
Exploring and navigating large graphs
- Maintain context
- Provide easy access to details

Cheops provides effective compression by reusing visual components based on interaction

Compress the hierarchy by tessellation of triangles
- In the example triangle 5 could represent either node E or node F
- If triangle 2 is selected, triangle 5 will become node E ...
- Overlapping triangles to indicate larger hierarchy
- The example shows an expansion by adding 5 children per parent

But: users cannot compare topologically remote parts of a structure
Cheops

- Another example
- Three triangles in the last level represent more than one logical node
- If a parent node (e.g. B) is selected the visual components become unambiguous
- Selection of a node implies previous selection of all its parent nodes
- Nodes are represented as paths of visual objects going down from the root – not isolated triangles
Cheops

- Visual cues and terminology to aid interpretation of the compressed visualization
- Selection: deployment of branches
- Pre-selection: direct access to any node in the tree
Degree of Interest Tree

- Nation et al. 2002
- For interactive display of hierarchies within a web browser
- Based on Furnas Degree-of-interest function
  - Each node is assigned a value
  - Degree-of-interest value is determined by a function of a node’s distance from the root of the tree and its distance from the focus of interest
  - Topic of later lecture on focus+context presentation techniques!
- DOITree
  - Upon selection: focused node is allocated most space
  - Remaining space is allocated to nodes based on their DOI values
  - Nodes with more space present more details
Degree of Interest Tree

- Animated transitions
- Reset the tree layout by clicking on the root node
- Tree does not fit the screen in the Y-dimension
  - Prune parts of the tree below a given DOI threshold
  - Pruned branch is represented by a symbol
- Tree does not fit the screen in the X-dimension – visually compress peripheral nodes

3D Approaches

- Why not use an additional dimension to visualize nodes that would otherwise be pruned / collapsed?
  - Cone Tree
  - H3Viewer

- HCI research produced mixed results about the usability of 3D interfaces

- Ongoing research question: do 2D interfaces better exploit the abilities of the human perceptual system?
  - Utilize spatial memory?
  - Controlling 3D navigation with 2D input devices?

- 3D node-link approaches have been mainly researched in the 90s
Cone Tree

- Robertson et al. 1991
- Use depth to make more effective use of screen space
- Hierarchies laid out uniformly in three dimensions
- When a node is selected by a user the tree rotates to bring the node to the front
- Animation to make the users comprehend the rotation
- Problem: still clutter and occlusion
- Movie
Cone Tree

- Usability evaluation by Cockburn & McKenzie 2000
- Compare Cone Tree to conventional explorer-like 2D tree browser
- User test with 12 participants
- Independent variables: depth, density of tree, interface type
- Dependent variables: task-completion time, user preference

Results
- Users were slower in locating data using the Cone Tree
- Performance deteriorated rapidly with a growing branching factor
- But: participants clearly preferred the Cone Tree...
H3Viewer

- Munzner 1997
- H3Viewer supports interactive exploration of large graphs (> 100,000 edges)
- Graph is presented in 3D hyperbolic space
- Child nodes are distributed on the surface of a hemisphere
- Users can drag and rotate graph

Demo

Java 3D implementation and gallery:
http://www.caida.org/tools/visualization/walrus/
Outline

- Hierarchical data and tree representations
- 2D Node-link diagrams
  - Hyperbolic Tree Browser
  - SpaceTree
  - Cheops
  - Degree of interest tree
  - 3D Node-link diagrams
- Enclosure
  - Treemap
  - Ordered Treemaps
  - Various examples
  - Voronoi treemap
  - 3D Treemaps
- Circular visualizations
- Space-filling node-link diagram
**Treemap**

- Johnson & Shneiderman (1992)

**Basic idea**
- Map hierarchical data to rectangular 2D display area by recursively partitioning the screen into rectangular boxes representing nodes
- Utilize 100% of the screen

**Less good for analyzing the topology of a tree**

**Advantages**
- Very effective when focusing on leaf nodes and their attributes
- More suitable for additional encoding via color, size, shape
- Present large hierarchies on a single screen

Image taken from Spence 2007
Treemap

Nested versus non-nested Treemaps
Treemap

- Shneiderman 1992
- Slice and dice algorithm
  - Use parallel lines to divide a rectangle representing an item into smaller rectangles representing the item's children
  - Each child is allocated a size proportional to some property (additional encoding by color)
  - At each level of the hierarchy switch the orientation of the lines (vertical vs. horizontal)
- Example application: file browser
  - Size: file size, color: file type
  - Users can easily identify large file
  - Detect duplicate directories
  - ...

Treemap

- Problems with this layout?
- Creates layouts that contain many rectangles with a high aspect ratio
- Thin rectangles are hard to see, select, label and compare in size
- Which of the blue rectangles is bigger?
Treemap

Several algorithms to create more useful tree-maps by reducing the overall aspect ratios of the map rectangles

Cluster algorithm (Wattenberg1999): employ both vertical and horizontal partitions at each level of the hierarchy

Squarification algorithm (Bruls et al. 2000)

Sorts and adds the input rectangles ordered by size

Problem of both algorithms

- Changes in the data set can cause dramatic layout changes (hard to track items given dynamic data)
- Given ordering of items is not preserved (as indicated by shading)

Subdivision algorithm for squarified algorithm (Bruls et al. 2000)
Ordered Treemap

- Seek compromise between smooth updates and low aspect ratios
- Items are given as a list ordered by index and have varied areas
- Items that are next to each other in the given order should be approximately adjacent in the tree-map
- Shneiderman & Wattenberg 2001
- Pivot-by-size & Pivot-by-middle
  - Partition area into 4 regions
  - Pick pivot element Rp
    - Size: largest item
    - Middle: middle item
  - Depending on the aspect ratio of R, place Rp in horizontal oder vertical middle
  - R1: items earlier in the list than pivot (sublist L1)
  - R2: items in list before R3 such that their overall size makes Rp have aspect ratio closest to 1 (sublists L2, and L3)
  - Apply steps recursively for areas R1, R2, and R3
Ordered Treemap

- Strip treemap - Bederson & Shneiderman 2002
- Modification of squarified algorithm
- Produces better readability than basic ordered treemap algorithms and comparable aspect ratios (only slightly worse than unordered squarified algorithm)
- Rectangle is filled stepwise with strips
- Strip is filled stepwise with rectangles as long as the average aspect ratio of the strip decreases or stays the same
- Otherwise a new strip is added
Ordered Treemap

- Test with several generated data sets
- Table shows results for three levels of hierarchy and eight items at each level
- 100 trials of 100 steps each
- Comparing the algorithms by **average aspect ratio** and **average layout distance change** (how much do rectangles move as data is updated) and **readability** (how easy it is to visually scan a layout to find a particular item)
- Tradeoff between low aspect ratios and smooth updates!

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Aspect Ratio</th>
<th>Change</th>
<th>Readability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slice-and-dice</td>
<td>26.10</td>
<td>0.46</td>
<td>1.0</td>
</tr>
<tr>
<td>Pivot-by-middle</td>
<td>3.58</td>
<td>1.21</td>
<td>0.42</td>
</tr>
<tr>
<td>Pivot-by-size</td>
<td>3.31</td>
<td>4.14</td>
<td>0.33</td>
</tr>
<tr>
<td>Pivot-by-split</td>
<td>3.00</td>
<td>2.37</td>
<td>0.35</td>
</tr>
<tr>
<td>Strip</td>
<td>2.83</td>
<td>1.09</td>
<td>0.51</td>
</tr>
<tr>
<td>Cluster</td>
<td>1.79</td>
<td>7.67</td>
<td>0.26</td>
</tr>
<tr>
<td>Squarified</td>
<td>1.74</td>
<td>8.27</td>
<td>0.26</td>
</tr>
</tbody>
</table>
Ordered Treemap

- Applying the algorithms to real-world data - confirmed prior test results
- Set of 535 publicly traded companies, market capitalization as the size attribute
- Gray scale indicates ordering within each industry group that is the last level of hierarchy in this data set
Ordered Treemap

- Compare algorithms

- History of treemaps
  http://www.cs.umd.edu/hcil/treemap-history/

- Java 1.1 library for five Tree-map algorithms:
Ordered Treemap

- Bederson et al. 2002
- User study of layout readability
- Compared the squarified, pivot-based, and strip treemap algorithms
- 20 Participants had to identify a specific rectangle by clicking on the rectangle with the requested numerical ID
- Repeated-measures design
- Independent variable: treemap algorithm
- Dependent variable: time, subjective user rating
- Time: significant difference between squarified algorithm and the other two
- Preference: significant difference between all three algorithms
- Validates readability metric used
Map of the Market

- Wattenberg 1999
- Cluster treemap to reduce overall aspect ratios
- http://www.smartmoney.com/marketmap/
- 500 stocks updated every 15 minutes
- Each rectangle represents a company
  - Size: company’s market capitalization
  - Color: price performance
- Double-ended multiple hue color coding
  - Green: stock price is up
  - Red: stock price is down
  - Black: neutral, no change
- Detailed information on-demand
- Demo

SAP Pays $6.8 Billion for Business Objects
Treemap 4.1

- Human-Computer Interaction Lab – University of Maryland
- Demo
Some Treemaps Online

NewsMap

Peet's Coffee: Coffee Selector

iTunes Top 100
TennisViewer

- Jin & Banks 1997
- Visualize a tennis match using a treemap
- Match tree
  - Root node – the tennis match
  - Match node subdivides horizontally into sets
  - A set subdivides vertically into games
  - A game subdivides horizontally into points
- Color mapping of rectangles show node ownership (who won what?)
- Translucent child rectangles are layered over parent rectangles
  - Point of a winning effort
  - Point of a losing effort
TennisViewer

Magic Lens to explore ball traces
Example: the return of a service goes out of bounds
FundExplorer

- Csallner et al. 2003
- To support the diversification of mutual fund portfolios, i.e. how to find funds with little overlap in their investments
- Also show stocks with zero investment
- Movie
Voronoi Treemap

- Balzer et al. 2005
- Treemap consisting of arbitrary polygons instead of rectangles
  - Aspect ratio of polygons converges to 1
  - Polygons are distinguishable due to the irregular shapes
  - Avoid that edges of different objects run into each other
Cushion Treemap

- Wijk & van de Wetering 1999
- Treemaps usually fall short to visualize the structure of the tree
- Worst case: a balanced tree, where each parent has the same number of children and each leaf has the same size
- Outcome: regular grid
- Nested treemap may reduce this problem, but:
  - Margins require screen space
  - Deeply nested trees are difficult to read
- Idea: add shading and texture to help convey the structure of the tree
Cushion Treemap
Cushion Treemap

- SequoiaView


- Visualizes the contents of your hard drive
StepTree

- Bladh et al. 2004
- Convey tree structure via third dimension
Outline

- Hierarchical data and tree representations
  - 2D Node-link diagrams
    - Hyperbolic Tree Browser
    - SpaceTree
    - Cheops
    - Degree of interest tree
    - 3D Node-link diagrams
  - Enclosure
    - Treemap
    - Ordered Treemaps
    - Various examples
    - Voronoi treemap
    - 3D Treemaps
- Circular visualizations
- Space-filling node-link diagram
Information Slices

- Andrews & Heidegger 1998
- Visualization is based on one or more semi-circular discs
- Each disc represents multiple levels (5 to 10, configurable) of a hierarchy
- Files and directories deeper in the hierarchy are drawn further from the center
- Child nodes are drawn within the arc subtended by their parents
- For deeper hierarchies multiple discs are cascaded
- Example shows Solaris JDK, 6158 files in 502 directories, maximum depth of 9 levels
- Blue: directories, other colors: file type
Sunburst

- Stasko & Zhang 2000
- Full circular visualization to give each element more space
- Navigating the tree should not lead to significant node position changes (e.g. hyperbolic browser)
- Three animated approaches to provide a focus area while maintaining context
  - Angular detail method
  - Detail outside method
  - Detail inside method
- Comparative evaluation of sunburst vs. treemap did not show significant differences in task completion times, but participants strongly preferred sunburst (Stasko et al. 2000)
- Radial visualizations may better depict the structure of the tree, but are not as space-efficient as treemaps
- Movie
Interring

- Yang et al. 2002
- Multiple foci (circular distortion + radial distortion)
Circular Treemaps

Kay Wetzel

Do not fill space completely

But

- Aspect ratio stays the same for all elements
  - easy comparison of sizes
- Good visibility of nesting (though at the cost of unused space)
- Rather beautiful layout!

Visualization of a file system with color mapping for creation data
Circular Treemaps

- Grokker - http://live.grokker.com/
- Clustering search results as nested circles
- Demo
Outline

- Hierarchical data and tree representations

- 2D Node-link diagrams
  - Hyperbolic Tree Browser
  - SpaceTree
  - Cheops
  - Degree of interest tree
  - 3D Node-link diagrams

- Enclosure
  - Treemap
  - Ordered Treemaps
  - Various examples
  - Voronoi treemap
  - 3D Treemaps

- Circular visualizations

- Space-filling node-link diagram
Enclosure + Connection

EncCON: Nguyen & Huang 2005

Connection (node-link)
- Gives immediate perception of data relationships and the tree structure
- Not efficient regarding display space utilization: most pixels are wasted as background

Enclosure (e.g. treemaps)
- Space-filling approach allows the display large trees on a single glance
- Focus on the leaf nodes but hardly conveys the tree structure

Idea: combine enclosure and connection approach

- Child nodes are not embedded but placed around parent nodes using a circular, space-filling division method
- Focus+context navigation

Java SDK visualization – 9500 directories and files
Obligatory Literature


- Benjamin B. Bederson & Ben Shneiderman, "Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies“, 2002.