Hierarchical Data Analysis

Introduction to Trees and Hierarchies

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

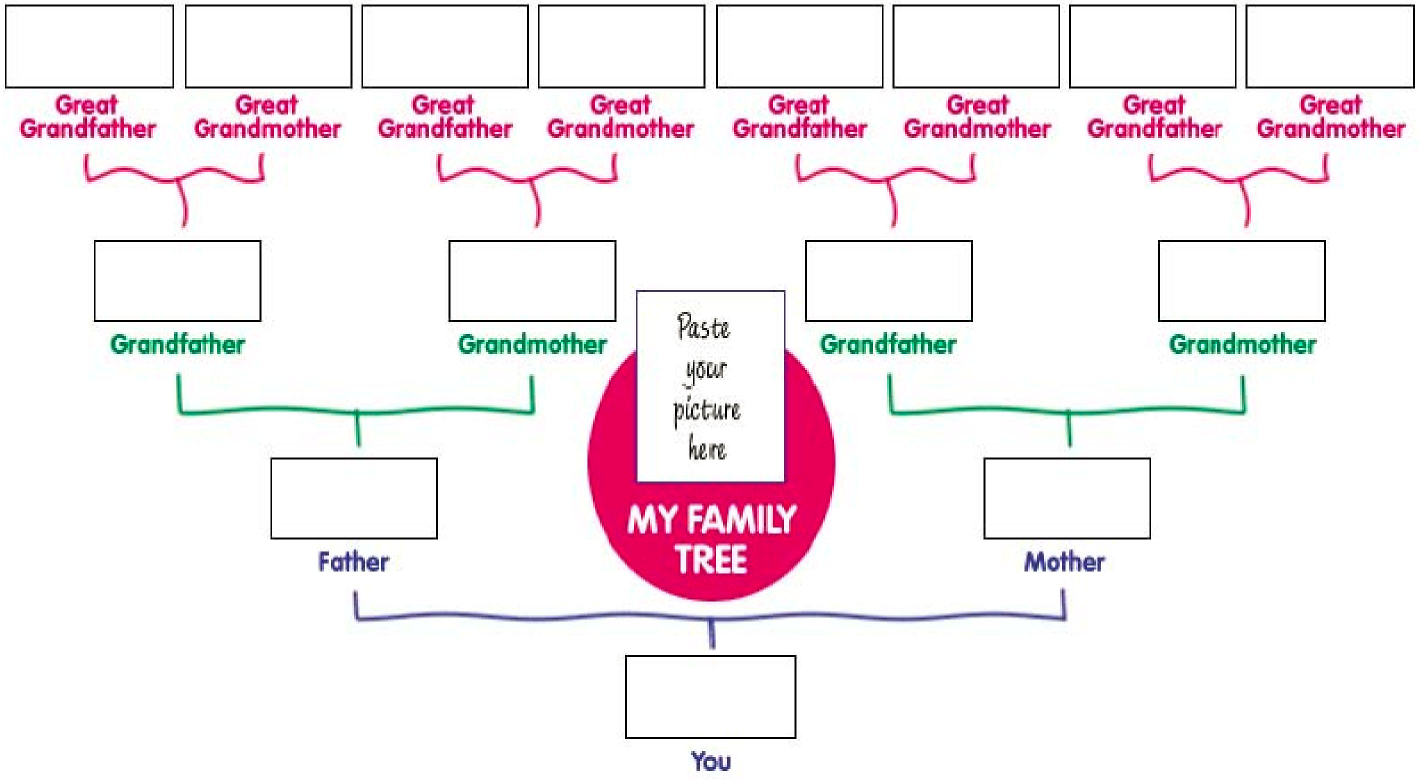
Objective

Apply methods of hierarchical data analysis

Hierarchies

Definition

- Data repository in which cases are related to subcases
- Can be thought of as imposing an ordering in which cases are parents or ancestors of other cases



Parts of this slide borrowed from John Stasko's Information Visualization Course Lecture: Hierarchies and Trees 1(Node-Link): http://www.cc.gatech.edu/~stasko/7450/11s/Talks/tree1.pdf

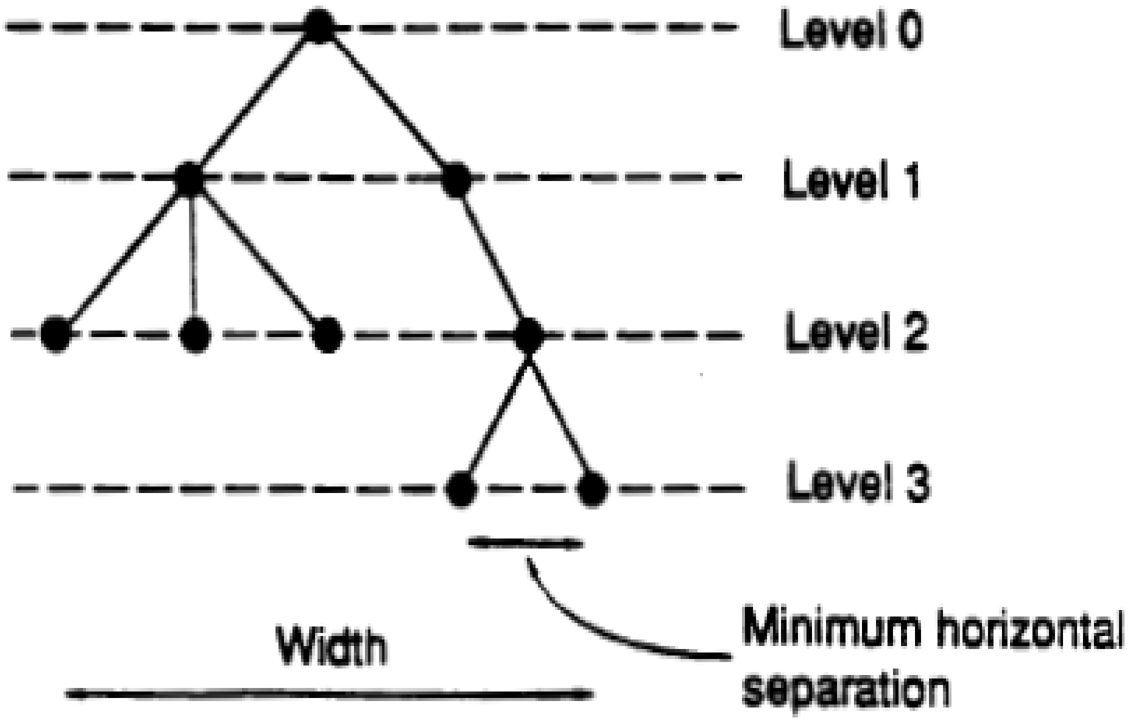
Trees

- Hierarchies are often represented as trees
  - Directed
  - Acyclic

- Two main representation schemes
  - Node-link
  - Space-Filling

Rooted Trees

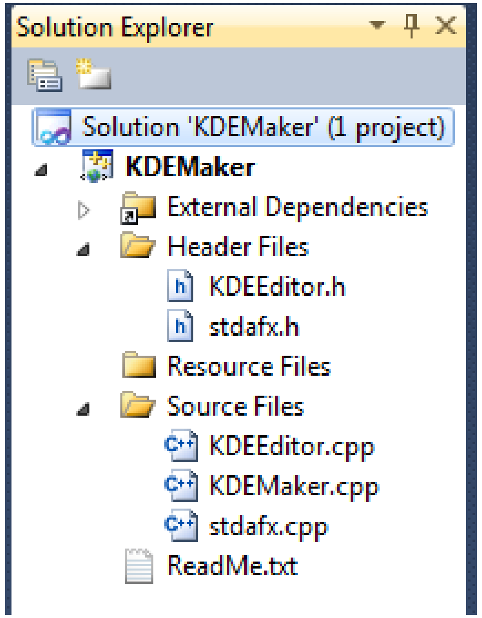
- A graph might be used to represent some hierarchy, so we often utilize a tree metaphor
- Typically these utilize the following aesthetics
  - Vertices are placed along horizontal lines according to their level
  - Minimum separation distance between two consecutive vertices on the same level
  - Width of the drawing is as small as possible

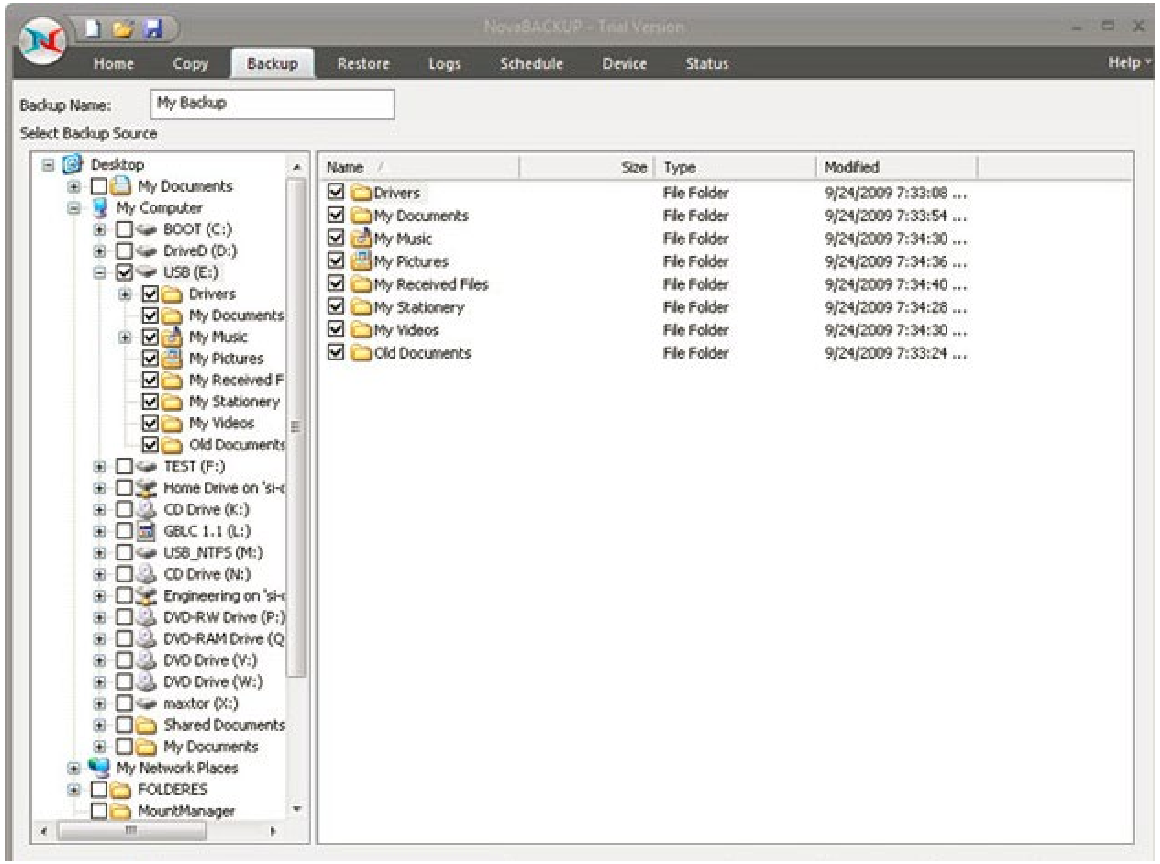


P Eades and R Tamassia, "Algorithms for Drawing Graphs: An Annotated Bibliography," Technical Report No. CS-89-09, Brown University, October 1989.

Using Rooted Trees

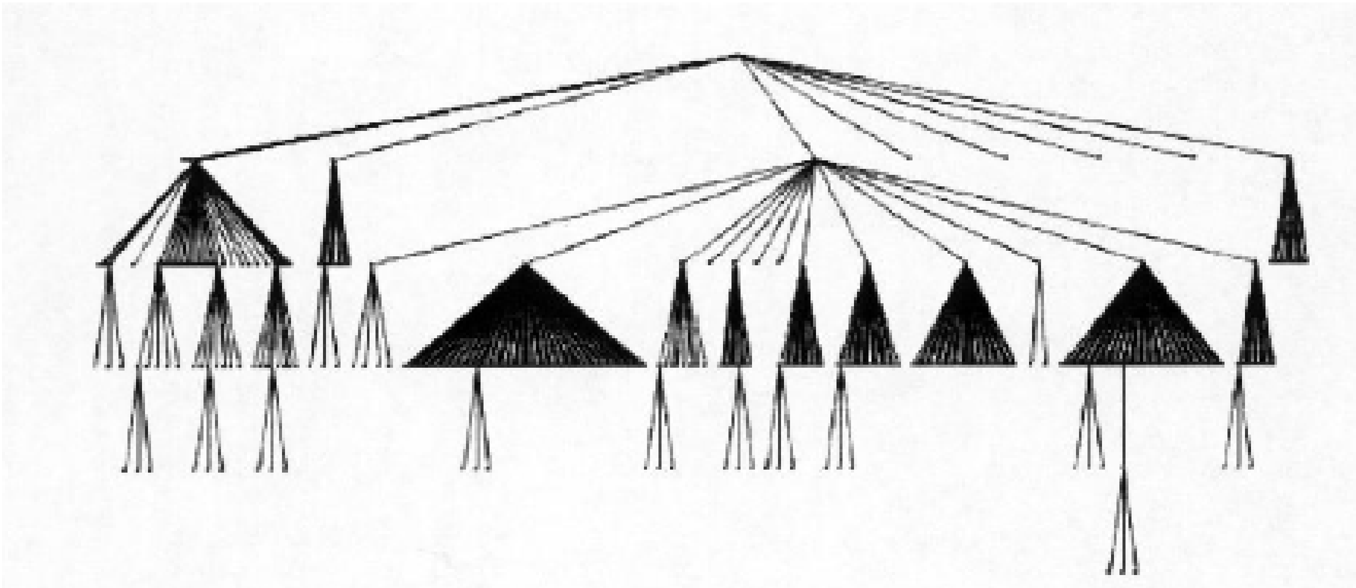
What are such sorts of structures useful for?





Top-Down Approach

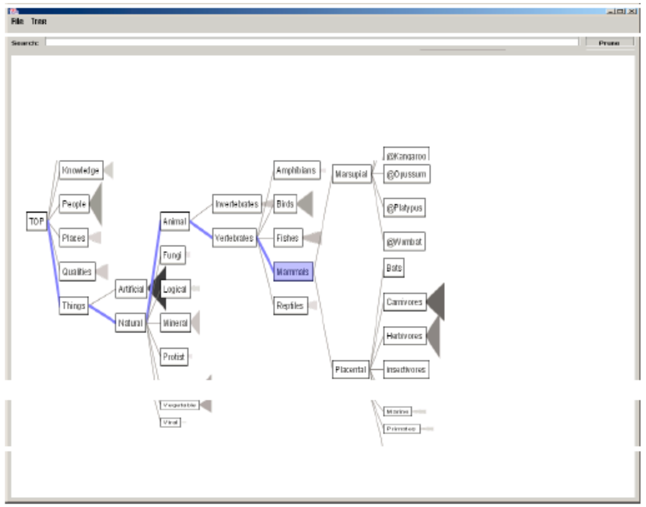
- Width of fan-out uses up horizontal real estate very quickly
  - At level n, there are 2^n nodes
- Tree may grow very long in one branch
- Essentially you can wind up leaving a lot of screen real estate empty



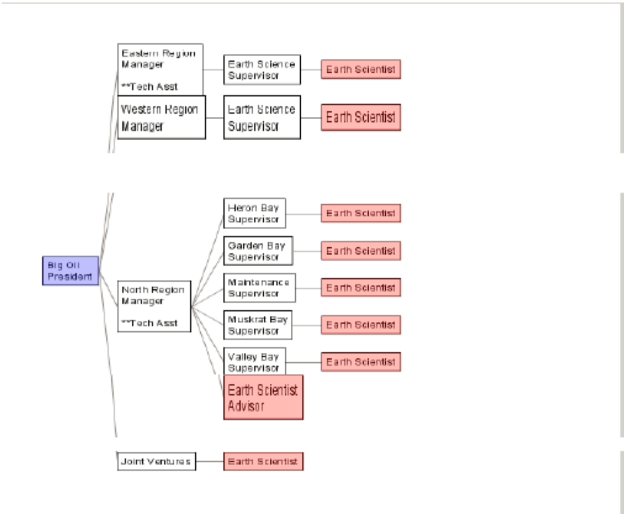
Parts of this slide borrowed from John Stasko's Information Visualization Course Lecture: Hierarchies and Trees 1(Node-Link):
http://www.cc.gatech.edu/~stasko/7450/11s/Talks/tree1.pdf

Space Tree

- Visualization techniques try to overcome some of these issues in node link tree diagrams



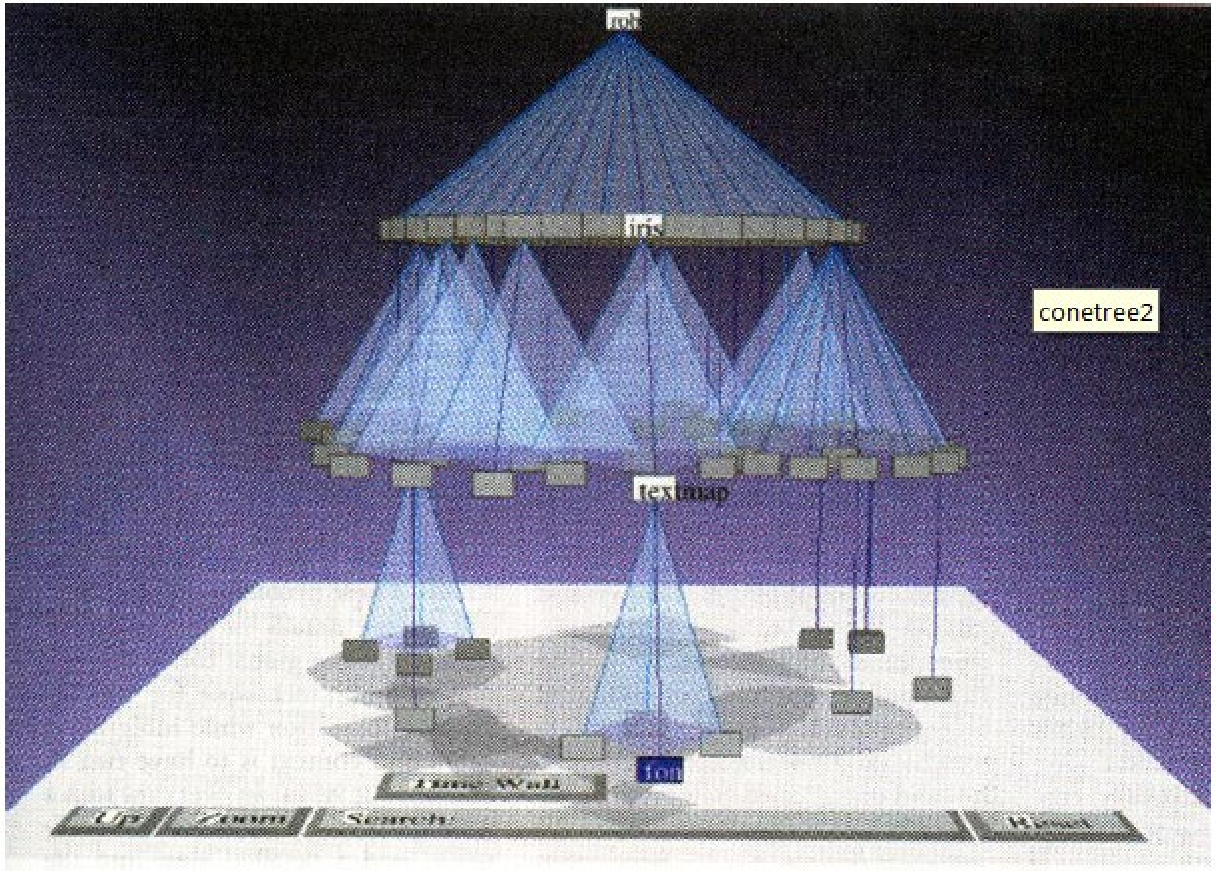
- *Space Tree* by Plaisant et al.
  - Dynamic rescaling of branches to best fit available screen space
  - Utilized preview icons to summarize branch topology



Plaisant, C.; Grosjean, J.; Bederson, B.B.; , "SpaceTree: supporting exploration in large node link tree, design evolution and empirical evaluation," IEEE Symposium on Information Visualization, pp. 57- 64, 2002

Cone Trees

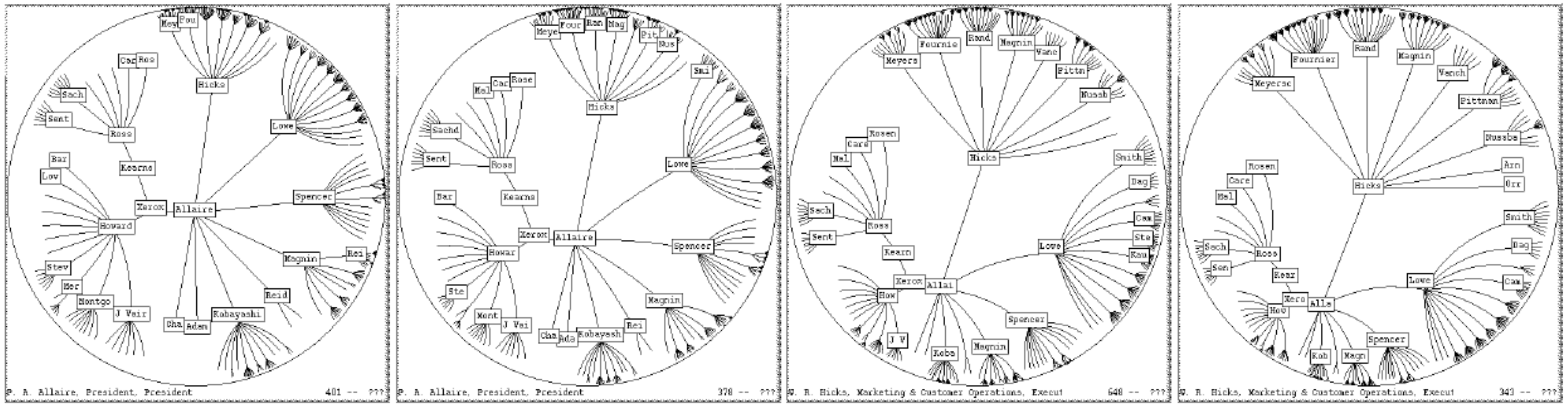
- Add a third dimension for the layout
- Children of a node are laid out in a cylinder below the parent
  - Siblings live in one of the 2D planes



Robertson, G. G. Mackinlay, J. D. Card, S. K. Cone Trees: animated 3D visualizations of hierarchical information, *Proc. Human factors in computing systems conference*, March 1991, 189-194.

Space Tree

- Don’t have to constrain to top-down geometry approach
- Apply a hyperbolic transformation to the space
- Distance between parent and child decreases as you move farther from the center
- Children go in a wedge rather than a circle



Lamping, J., Rao, R., Pirolli; P. (1995) A focus+context technique based on hyperbolic geometry for visualizing large hierarchies *Conference proceedings on Human factors in computing systems,* 1995, 401-408

Node-Link Shortcomings

Difficult to encode more variables of data cases

- Shape
- Size
- Color

- All of these can clash with the basic node-link structure

Parts of this slide borrowed from John Stasko's Information Visualization Course Lecture: Hierarchies and Trees 2(Space Filling): http://www.cc.gatech.edu/~stasko/7450/Notes/tree2.pdf

Hierarchical Data Analysis

Introduction to Tree Maps

Ira A. Fulton Schools of Engineering
Arizona State University

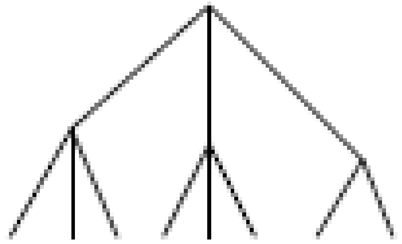
Objective

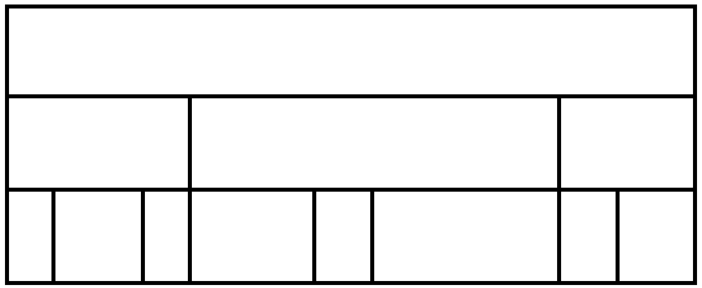
Objective

Explain hierarchical representation schemes

Space-Filling Representation

- Each item now occupies an area
- Children are contained under the parent



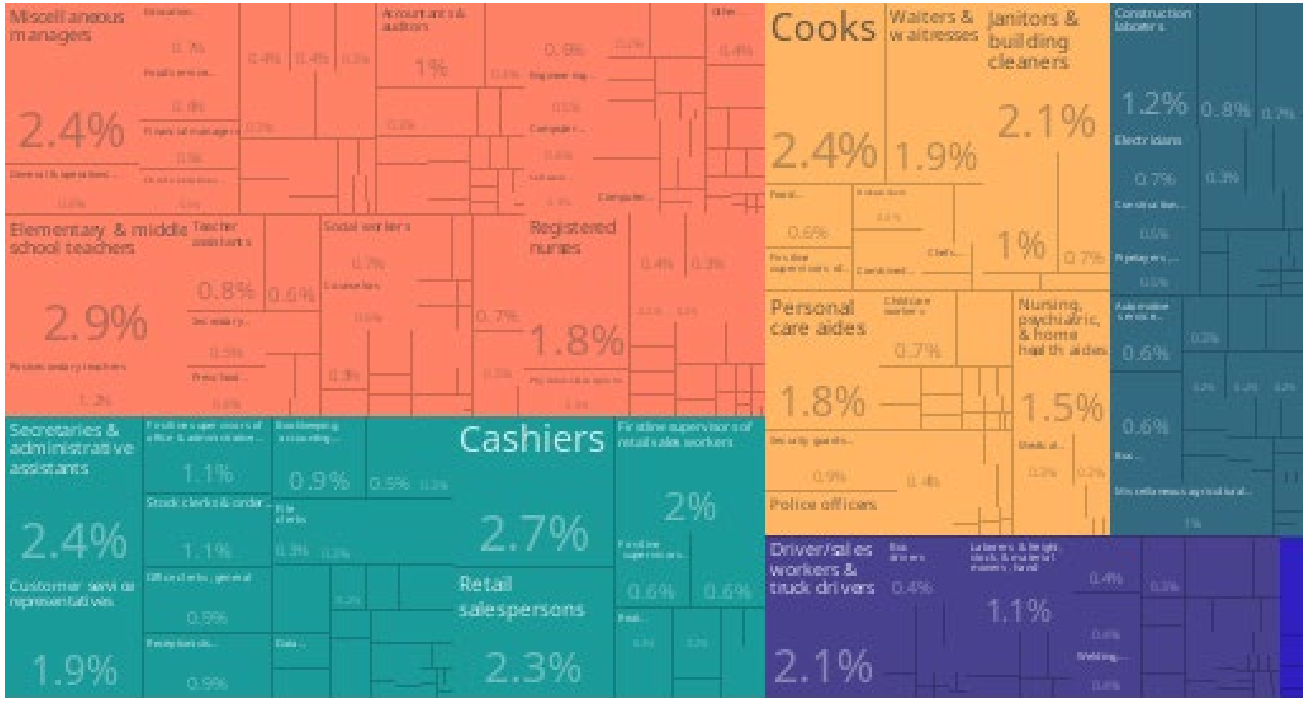


One example: "Icicle plot"

Parts of this slide borrowed from John Stasko's Information Visualization Course Lecture: Hierarchies and Trees 2(Space Filling): http://www.cc.gatech.edu/~stasko/7450/Notes/tree2.pdf

Treemap

- Space filling representation developed by Johnson and Shneiderman
- Children are drawn inside their parent
- Alternate horizontal and vertical slicing at each successive level
- Use area to encode other variable of data items



By Datawheel [CC0], via Wikimedia Commons

Johnson, B. and Shneiderman, B. Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. In Proceedings of the IEEE Information Visualization '91, pages 275–282, IEEE, 1991.

Treemap Algorithm

Draw()

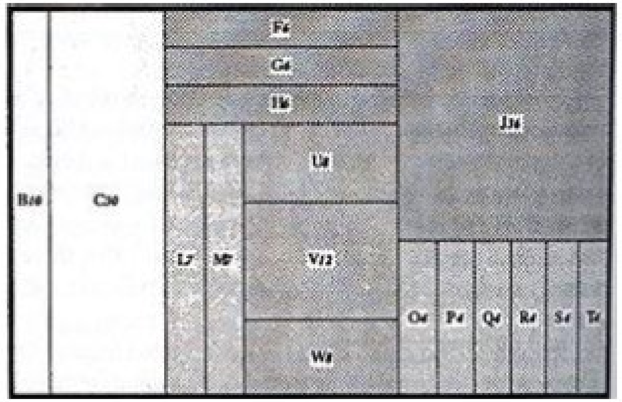
{

1. Change orientation from parent (horiz/vert)
2. Read all files and directories at this leve
3. Make rectangles for each, scaled to size
4. Draw rectangles using appropriate size and color
5. For each directory
   - Make recursive call using its rectangle as focus

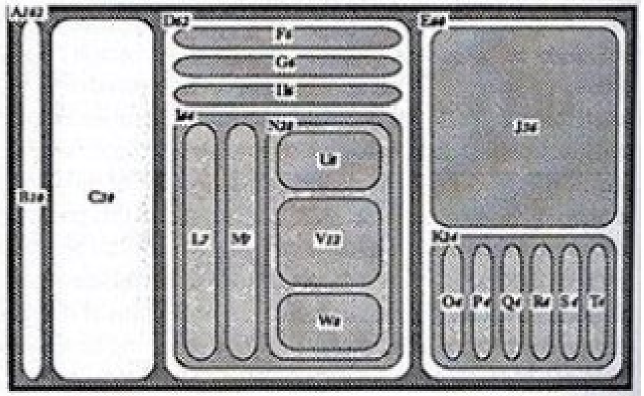
}

Parts of this slide borrowed from John Stasko's Information Visualization Course Lecture: Hierarchies and Trees 2(Space Filling): http://www.cc.gatech.edu/~stasko/7450/Notes/tree2.pdf

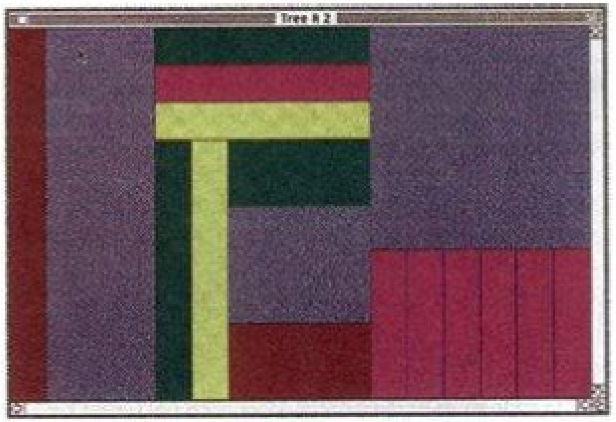
Nested vs. Non-Nested

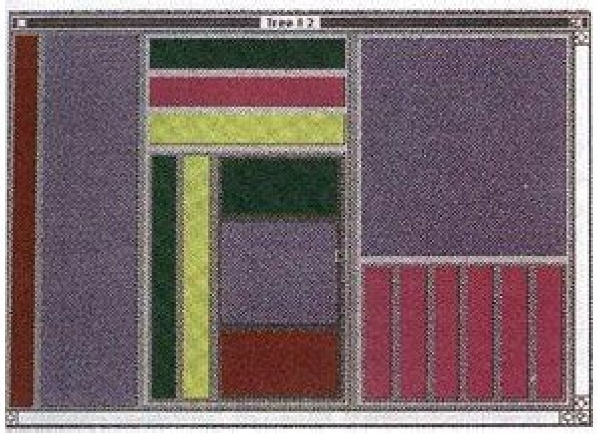


Non-nested Tree-Map



Nested Tree-Map





Parts of this slide borrowed from John Stasko's Information Visualization Course Lecture: Hierarchies and Trees 2(Space Filling): http://www.cc.gatech.edu/~stasko/7450/Notes/tree2.pdf

Treemap Applications

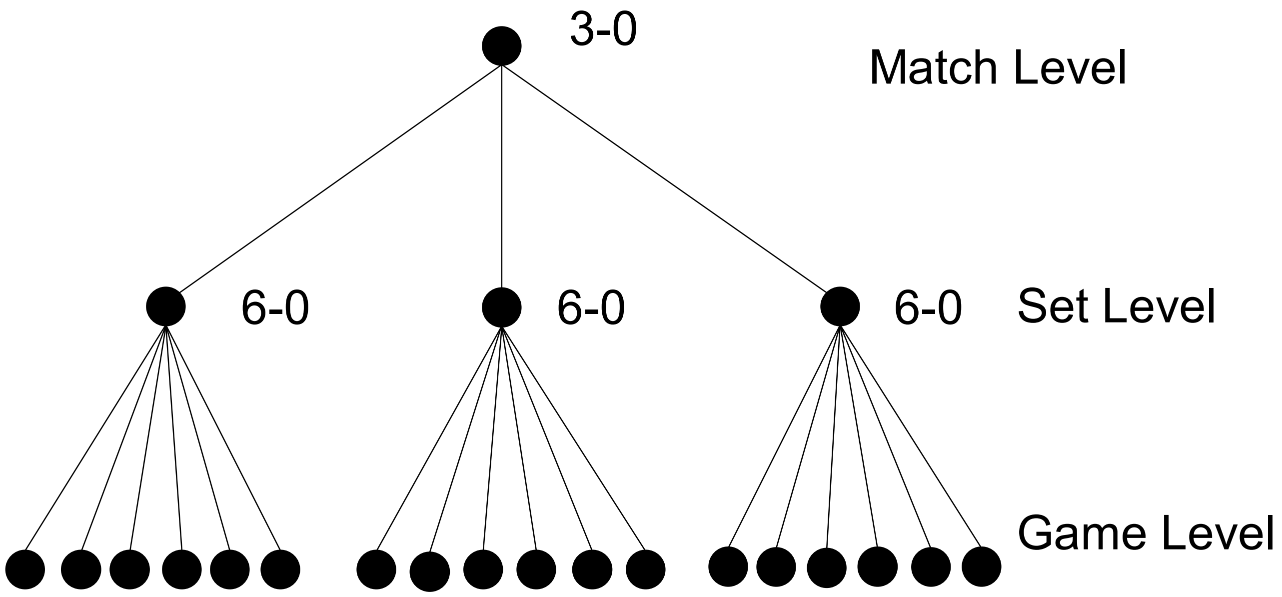
Can use the Treemap idea in a variety of domains

- File/directory structures
- Software diagrams
- Sports analysis
- Stock Market Data

Parts of this slide borrowed from John Stasko's Information Visualization Course Lecture: Hierarchies and Trees 2(Space Filling):
http://www.cc.gatech.edu/~stasko/7450/Notes/tree2.pdf

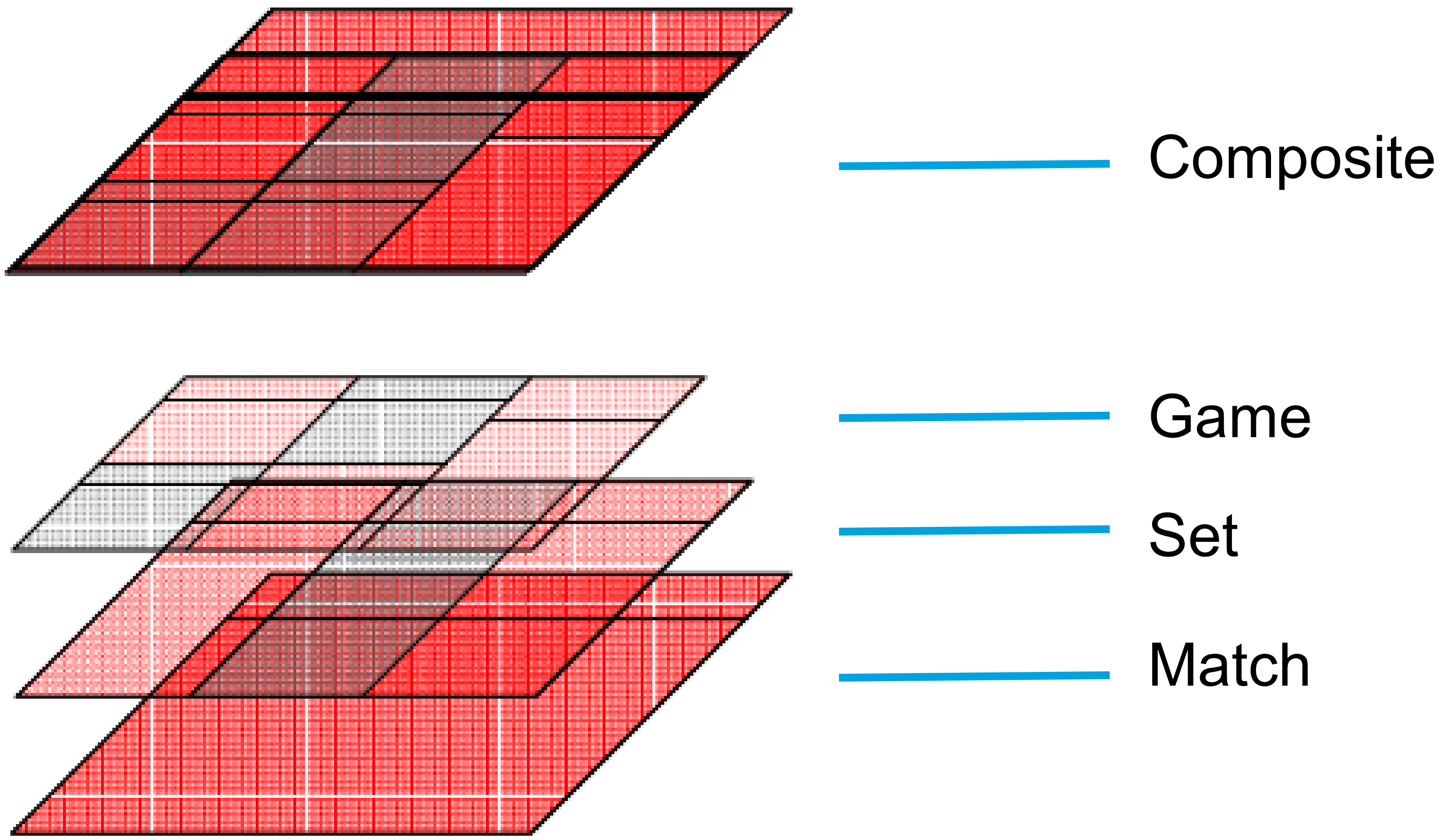
Visualizing a Tennis Match

- Analyze, review and browse a tennis match
- Space-filling/treemap-like hierarchy to show a competition tree
- Show match, sets, game points
- Lens can show shot patterns
- Color encodes player



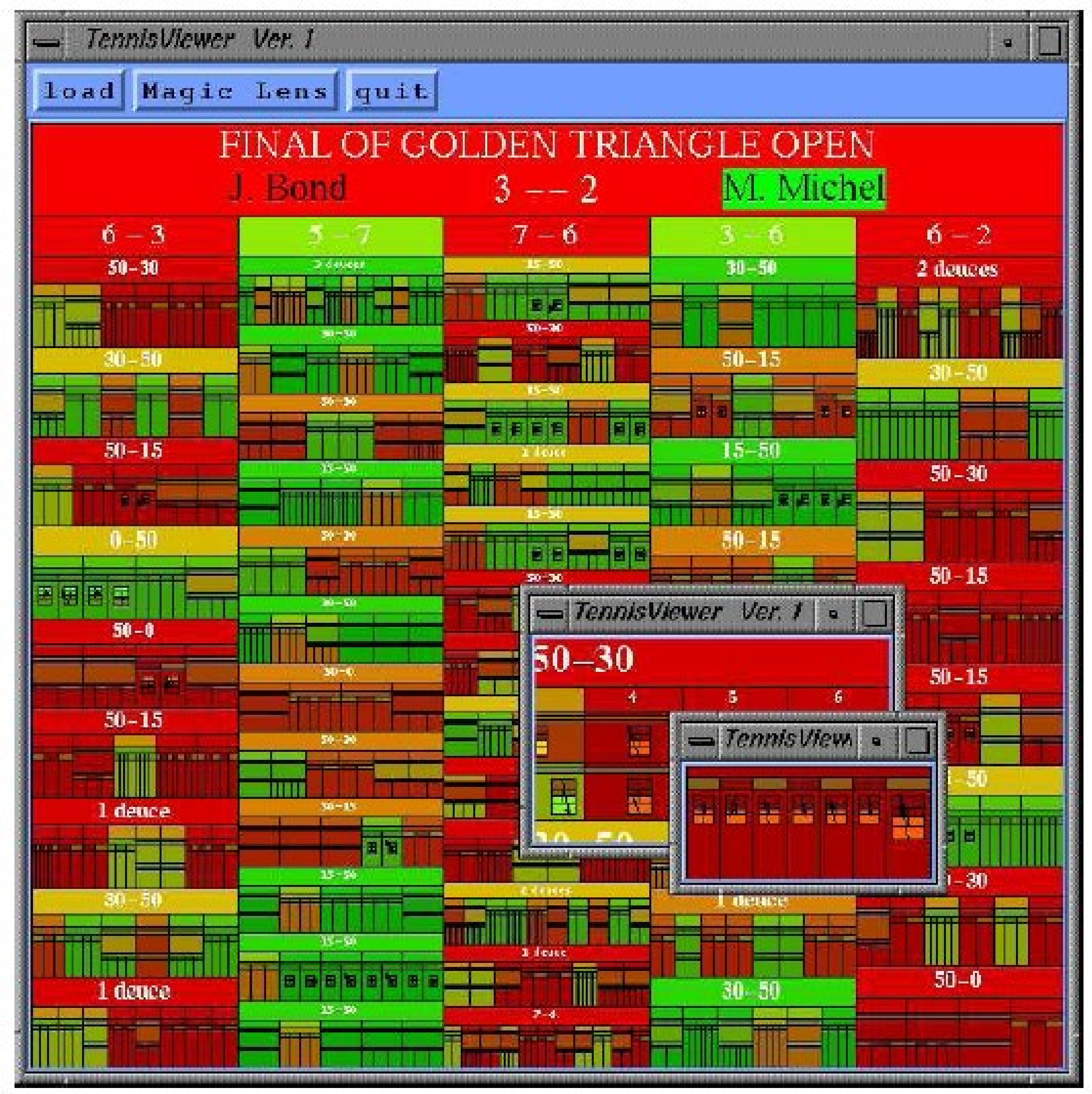
Liqun Jin and David C. Banks, ``TennisViewer: a Browser for Competition Trees,'' IEEE Computer Graphics & Applications, Vol. 21, No. 2 (March/April 1997), pergamon Press, pp. 171-178.

Visualizing a Tennis Match



Liqun Jin and David C. Banks, ``TennisViewer: a Browser for Competition Trees,'' IEEE Computer Graphics & Applications, Vol. 21, No. 2 (March/April 1997), pergamon Press, pp. 171-178.

Visualizing a Tennis Match



Liqun Jin and David C. Banks, ``TennisViewer: a Browser for Competition Trees,'' IEEE Computer Graphics & Applications, Vol. 21, No. 2 (March/April 1997), pergamon Press, pp. 171-178.

Tree Map Benefits

- Good representation of two attributes beyond node-link:
  - Color
  - Area

- Not quite as good at representing structure
  - What happens if the tree is perfectly balanced?
  - Can also get long-thin aspect ratios
  - Borders can help on small trees, but take up too much area on large, deep trees

Parts of this slide borrowed from John Stasko's Information Visualization Course Lecture: Hierarchies and Trees 2(Space Filling): http://www.cc.gatech.edu/~stasko/7450/Notes/tree2.pdf

Hierarchical Data Analysis

Tree Map Algorithms

Ira A. Fulton Schools of Engineering
Arizona State University

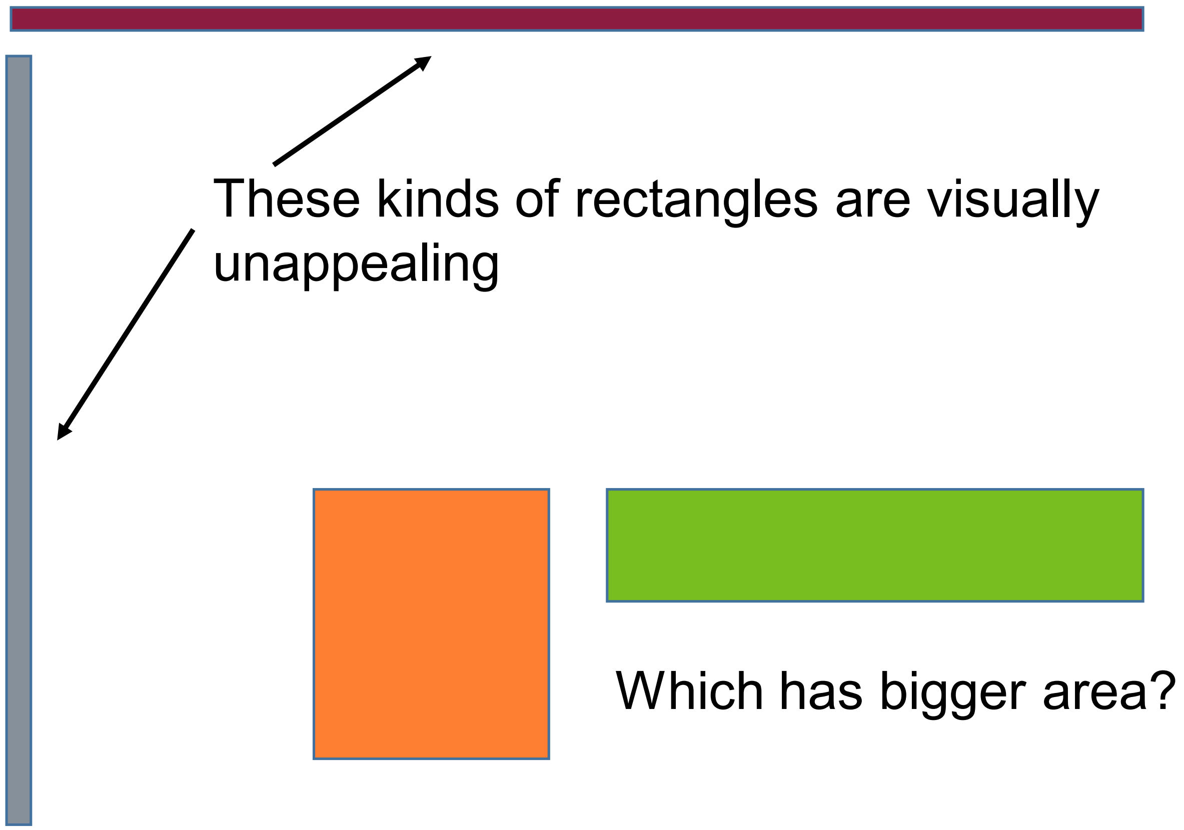
Objective

Objective

Explain hierarchical representation schemes

Aspect Ratios

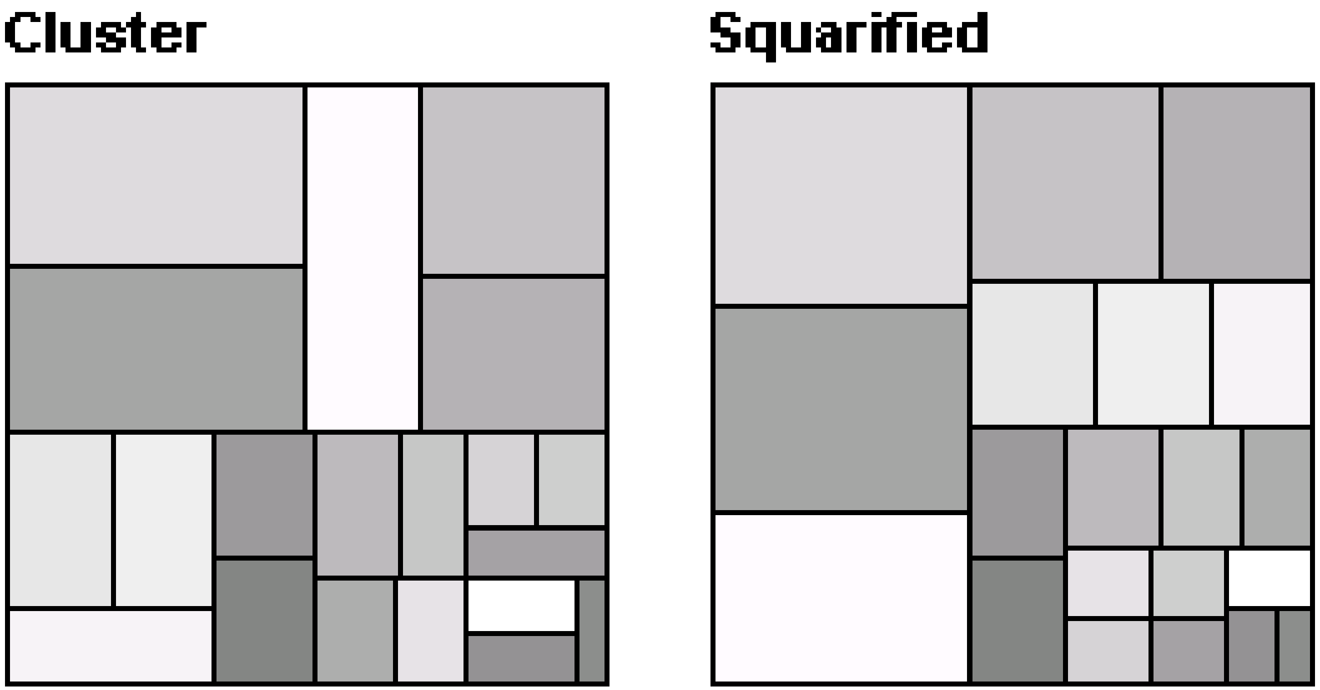
- Cleveland's "Banking to 45"
- Here, the aspect ratio will drastically affect the visualization



Parts of this slide borrowed from John Stasko's Information Visualization Course Lecture: Hierarchies and Trees 2(Space Filling): http://www.cc.gatech.edu/~stasko/7450/Notes/tree2.pdf

Clustered and Squarified Treemaps

- Simple recursive algorithm to reduce overall aspect ratio
- Bruls et al. introduced squarified treemap



1 -Wattenberg, M. “Visualizing the Stock Market,” Proceedings of ACM CHI 99, Extended Abstracts, pp.188-189, 1999.

2 -Bruls, D.M., C. Huizing, J.J. van Wijk. “Squarified Treemaps”. In: W. de Leeuw, R. van Liere (eds.), Data Visualization 2000, Proceedings of the joint Eurographics and IEEE TCVG Symposium on Visualization, 2000, pp. 33-42.

Clustered and Squarified Treemaps

Methods had two major drawbacks

- Changes in the set can cause discontinuities in the layout
  - If treemap data is dynamic large visual changes make data hard to track
- If the data has ordering information this is not preserved

1 -Wattenberg, M. "Visualizing the Stock Market," Proceedings of ACM CHI 99, Extended Abstracts, pp.188-189, 1999. 2 -Bruls, D.M., C. Huizing, J.J. van Wijk. "Squarified Treemaps". In: W. de Leeuw, R. van Liere (eds.), Data Visualization 2000, Proceedings of the joint Eurographics and IEEE TCVG Symposium on Visualization, 2000, pp. 33-42.

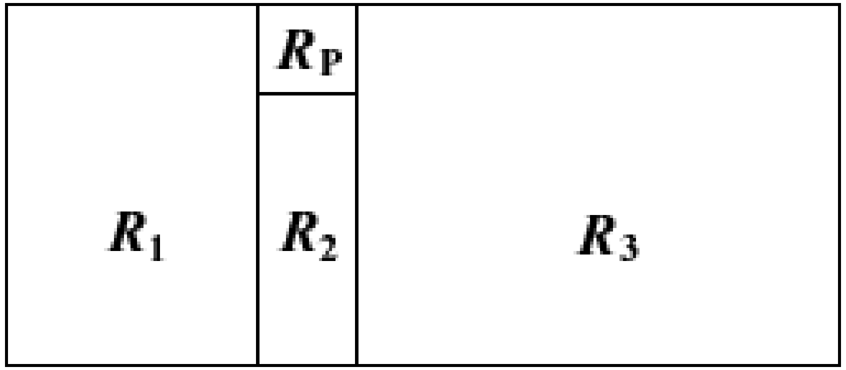
Ordered Treemap

- Shneiderman and Wattenberg introduced the ordered treemap to try and overcome these limitations
- Possible to create a layout in which items that are next to each other in given order are adjacent in the tree map
- Presented two algorithms for ordering a treemap

Ordered Treemap Algorithm

Starting with a rectangle R to be subdivided, first algorithm pivot-by-size, the pivot is item with largest area

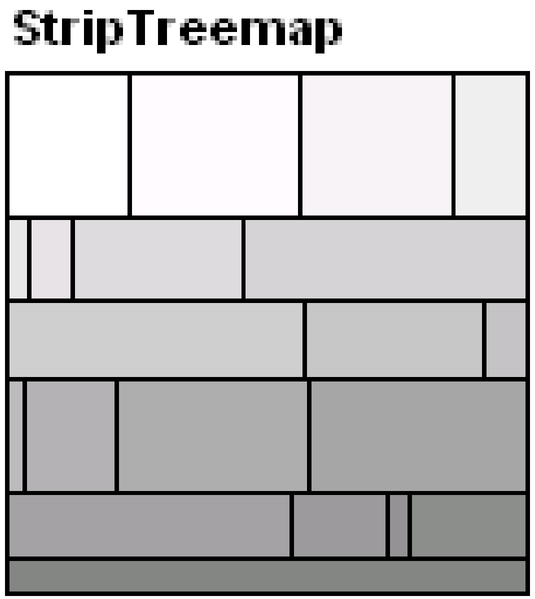
1. Let P, the pivot, be the item with largest area in list of items
2. If width R is greater than or equal to the height, divide R into four rectangle
3. Place P in the rectangle R_p
4. Divide the items in the list, other than P, into three lists, L_1, L_2, L_3 to be laid out in R_1, R_2 and R_3.
5. Recursively lay out L_1, L_2 and L_3 in R_1, R_2 and R_3



Ben Shneiderman, Martin Wattenberg: Ordered Treemap Layouts. INFOVIS 2001: 73-78

Strip Treemaps

1. Scale the area of all rectangles so total area of input rectangles equals that of layout rectangle
2. Create a new empty strip, the current strip
3. Add next rectangle to current strip, recomputing height of strip based on area of all rectangles within the strip and recomputing width of each rectangle
4. If average aspect ratio of the current strip has increased as a result of adding rectangle in step 3, remove rectangle pushing it back onto list of rectangles and go to step 2
5. If all rectangles have been processed stop, else step 3



Benjamin B. Bederson, Ben Shneiderman, Martin Wattenberg: Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. ACM Trans. Graph. 21(4): 833-854 (2002)

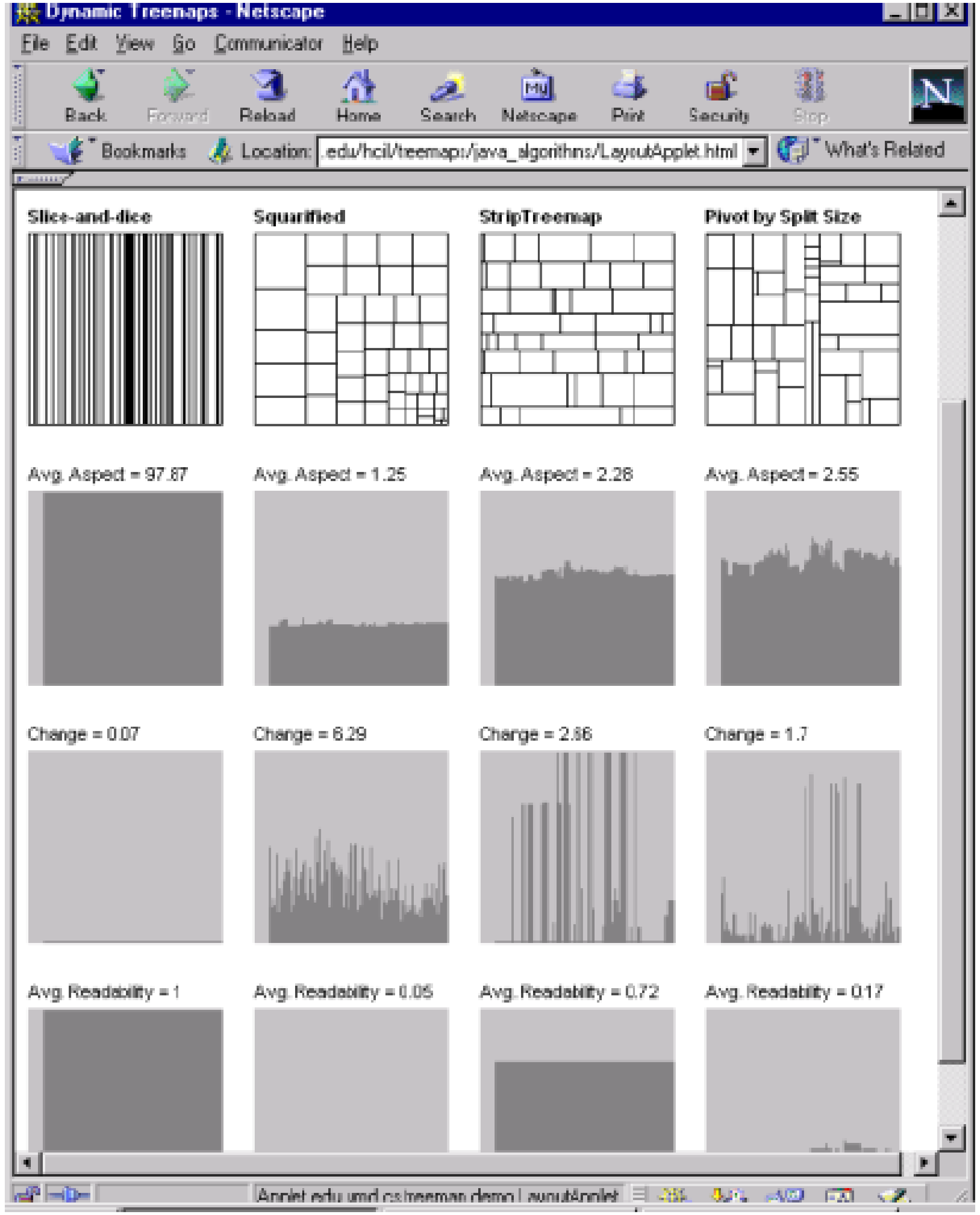
Metrics For Treemaps

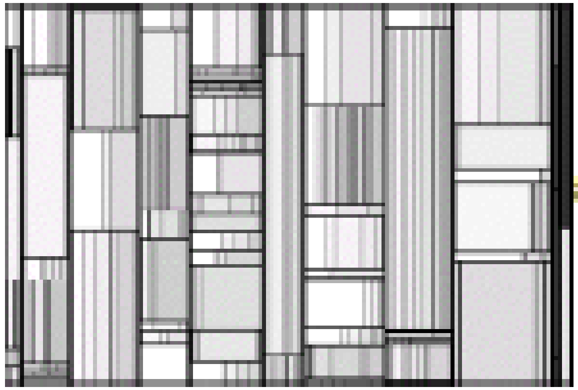
- In order to assess all these different treemap algorithms, we need metrics to define how “good” they are

- Use two metrics:
  - Average aspect ratio of treemap layout
  - Layout distance change function

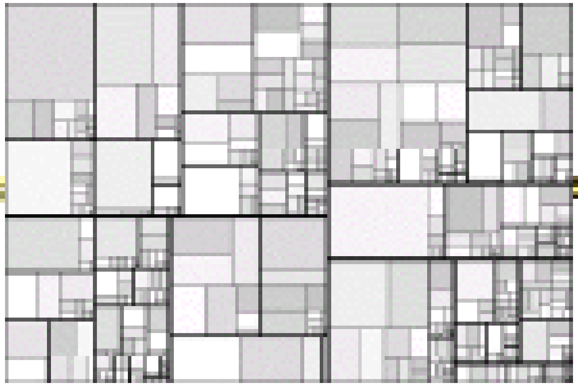
Metrics For Treemaps

- Goal is to have low average aspect ratio and a low distance change as data is updated
- Average aspect ratio is the unweighted average
- Distance change is Euclidian distance of change in width height and corner location of rectangles

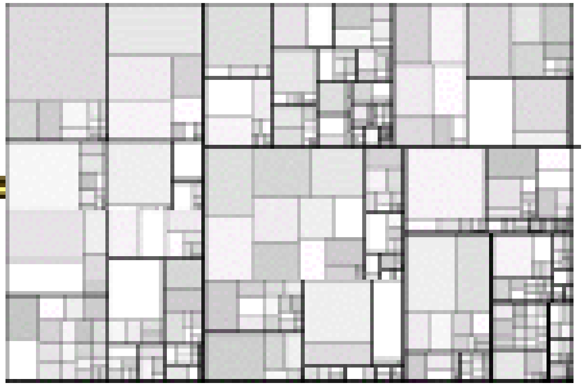

Dynamic Treemaps - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: .edu/hcil/treemaps/java_algorithms/LayoutApplet.html What's Related
Slice-and-dice
Squarified
StripTreemap
Pivot by Split Size
Avg. Aspect = 97.87
Avg. Aspect = 1.25
Avg. Aspect = 2.28
Avg. Aspect = 2.55
Change = 0.07
Change = 6.29
Change = 2.86
Change = 1.7
Avg. Readability = 1
Avg. Readability = 0.05
Avg. Readability = 0.72
Avg. Readability = 0.17



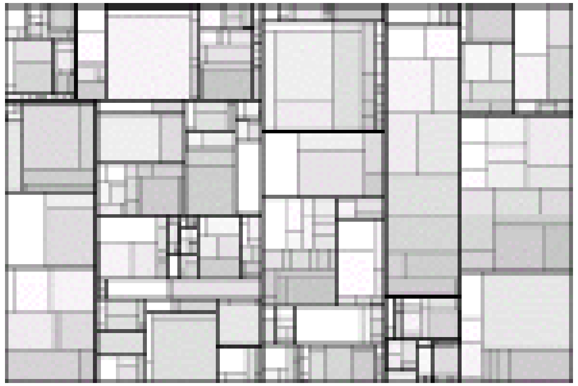
Slice-and-dice



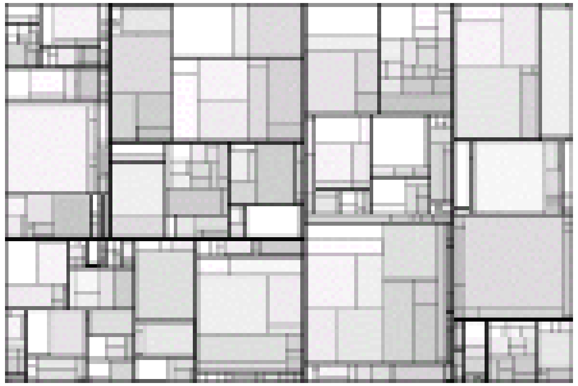
Cluster



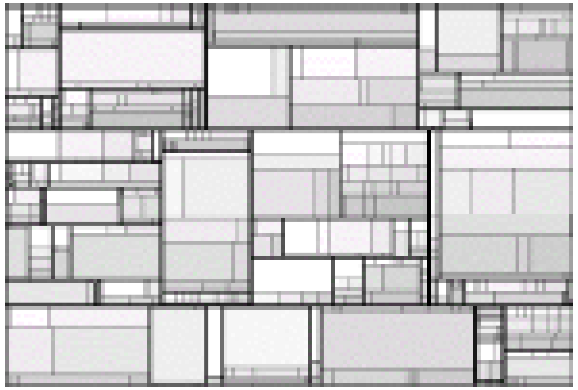
Squarified



Pivot-by-middle



Pivot-by-size



Strip

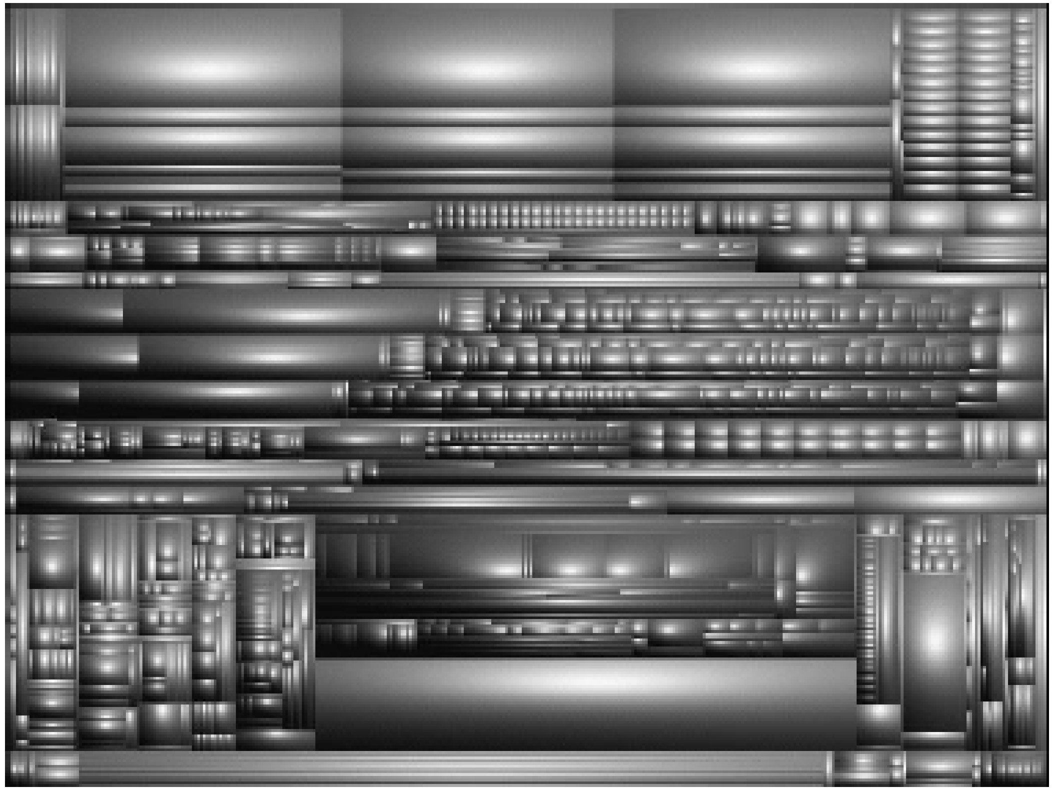
Showing Structure

- Regular borderless treemap makes it challenging to discern structure of hierarchy, particularly large ones
  - Supplement treemap view
  - Change rectangles to other forms

Parts of this slide borrowed from John Stasko's Information Visualization Course Lecture: Hierarchies and Trees 2(Space Filling): http://www.cc.gatech.edu/~stasko/7450/Notes/tree2.pdf

Cushion Treemap

- Use shading and texture to help convey structure of hierarchy



Van Wijk, J.J.; Van de Wetering, H. (1999), "Cushion treemaps: visualization of hierarchical information", Proc. IEEE Symp. Information Visualization (Info Vis '99), pp. 73–78

Another Problem

- What if nodes with zero value are very important?

- If we're mapping areas, how do we map to zero?
  - Example: Stocks portfolios

Parts of this slide borrowed from John Stasko's Information Visualization Course Lecture: Hierarchies and Trees 2(Space Filling):
http://www.cc.gatech.edu/~stasko/7450/Notes/tree2.pdf

Context Treemap

- One way to overcome this is to distort classic treemap visualization

- Distorted treemap can show one more attribute than a classic treemap
  - node area is no longer proportional to attribute being visualized

Christoph Csallner, Marcus Handte, Othmar Lehmann, John T. Stasko: FundExplorer: Supporting the Diversification of Mutual Fund Portfolios Using Context Treemaps. INFOVIS 2003: 203-208

Context Treemap

- Several different implementation strategies for this

  1. Use a regular tree map but add some epsilon to zero value items
  2. Use an exponential mapping area(node)=2^(value(node))
  3. Assign some minimal screen space size to zero nodes

Christoph Csallner, Marcus Handte, Othmar Lehmann, John T. Stasko: FundExplorer: Supporting the Diversification of Mutual Fund Portfolios Using Context Treemaps. INFOVIS 2003: 203-208

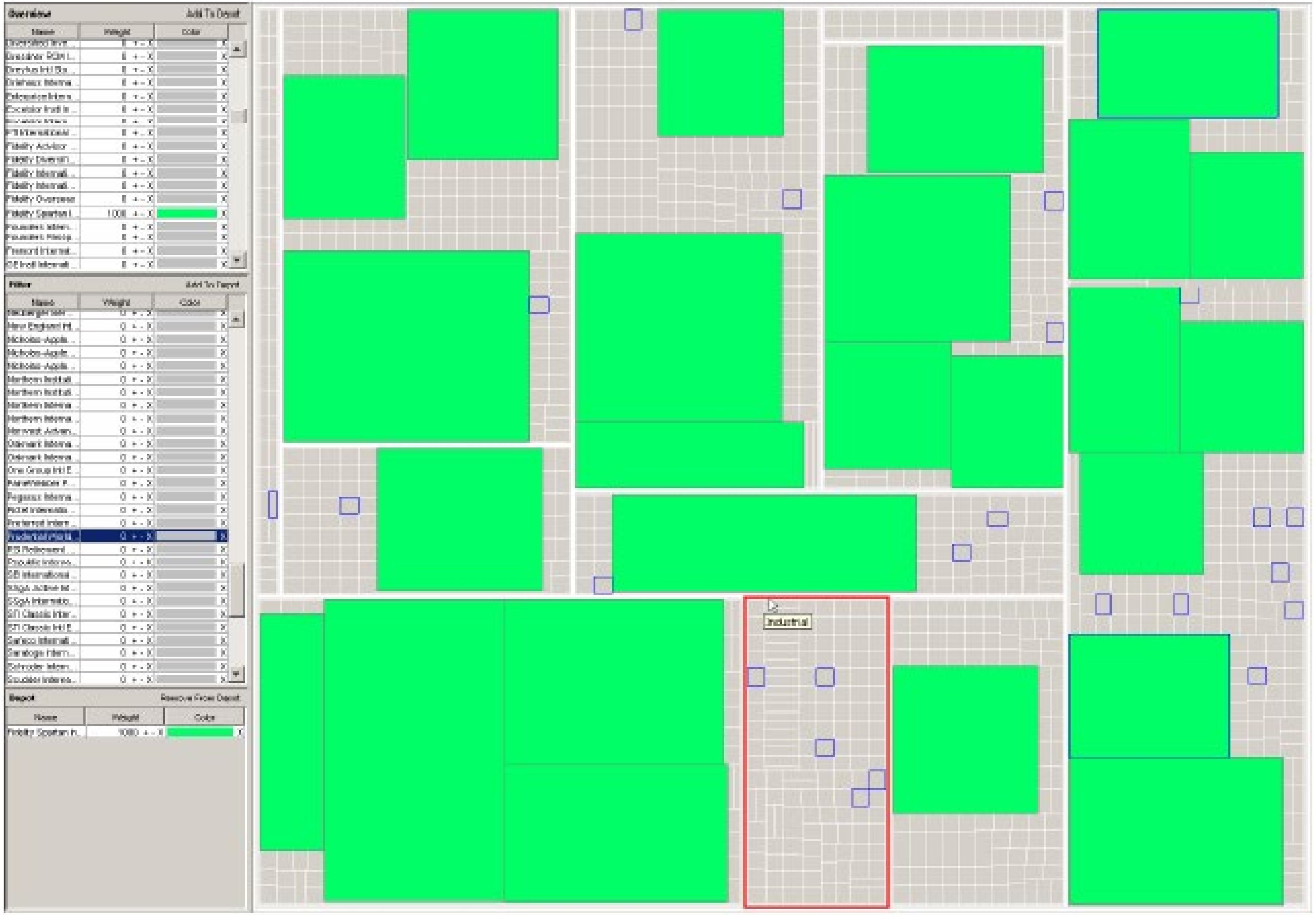
Context Treemap

Final solution was to calculate intermediate values

1. Calculate the total (in this paper it was total invest money)
   - Value(total)
2. Create an additional total with respect to the context
   - Value(total)*v, where v can be modified as a scale factor
3. Split context screen real estate among all empty nodes
   - Value_c = value(total)*v/#empty

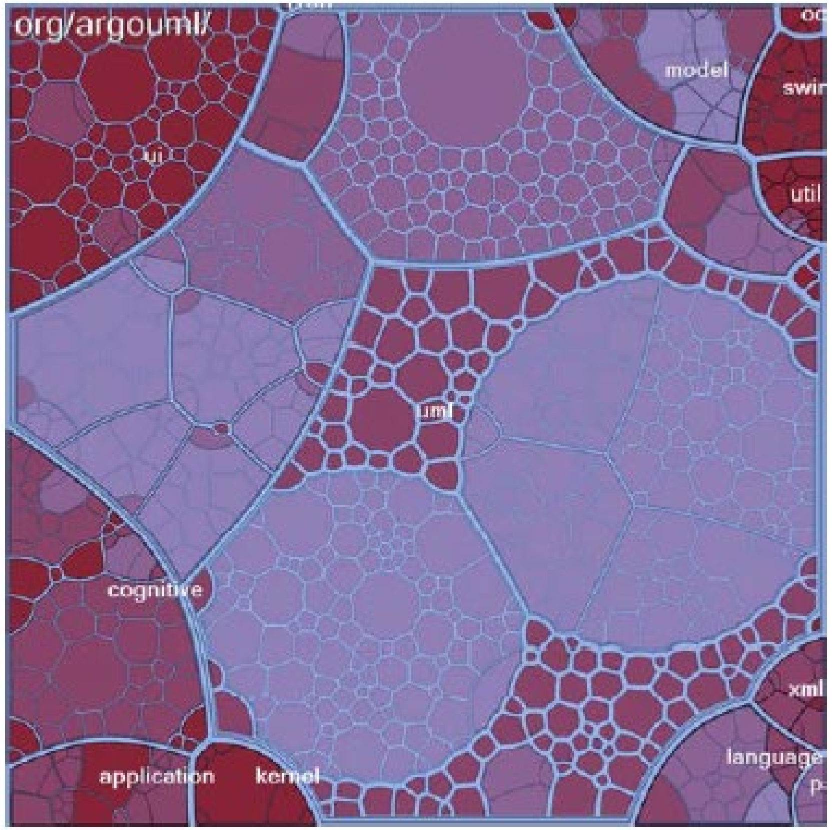
$$value`(node) = \begin{cases} value_c \text{ if } value(node) = 0 \\ value(node) \text{ otherwise} \end{cases}$$

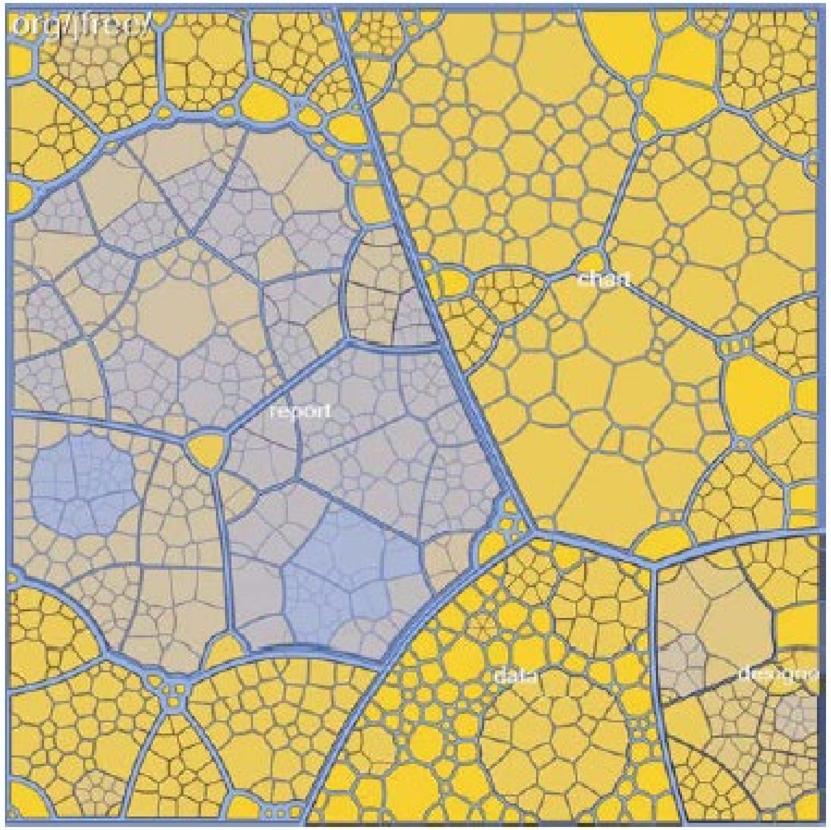
Christoph Csallner, Marcus Handte, Othmar Lehmann, John T. Stasko: FundExplorer: Supporting the Diversification of Mutual Fund Portfolios Using Context Treemaps. INFOVIS 2003: 203-208



- Christoph Csallner, Marcus Handte, Othmar Lehmann, John T. Stasko: FundExplorer: Supporting the Diversification of Mutual Fund Portfolios Using Context Treemaps. INFOVIS 2003: 203-208

Voronoi Treemaps





Michael Balzer, Oliver Deussen, and Claus Lewerentz. 2005. Voronoi treemaps for the visualization of software metrics. In Proceedings of the 2005 ACM symposium on Software visualization (SoftVis '05)

Definition of Voronoi Diagram

Let P be a set of n distinct points (sites) in the plane.

The Voronoi diagram of P is the subdivision of the plane into n cells, one for each site

Steven Fortune. A sweepline algorithm for Voronoi diagrams. Proceedings of the second annual symposium on Computational geometry. Yorktown Heights, New York, United States, pp.313–322. 1986

Definition of Voronoi Diagram

- A point q lies in the cell corresponding to a site $p_i \in P$ *iff*

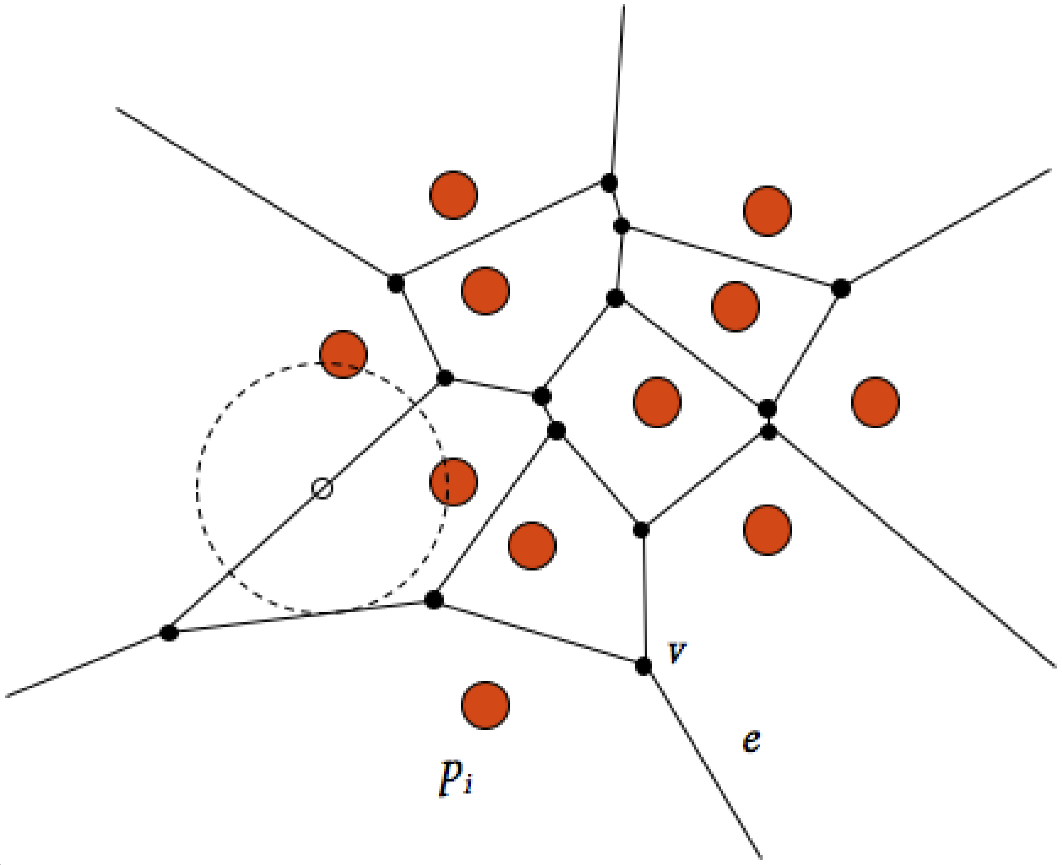
- Main Algorithm is Fortune's Algorithm

Euclidean_Distance(q, p_i) < Euclidean_distance(q, p_j), for each $p_i \in P$, $j \neq i$.

Steven Fortune. A sweepline algorithm for Voronoi diagrams. Proceedings of the second annual symposium on Computational geometry. Yorktown Heights, New York, United States, pp.313–322. 1986

Summary of Voronoi Properties

- A point q lies on a Voronoi edge between sites p_i and p_j *iff* the largest empty circle centered at q touches only p_i and p_j
  - A Voronoi edge is a subset of locus of points equidistant from p_i and p_j



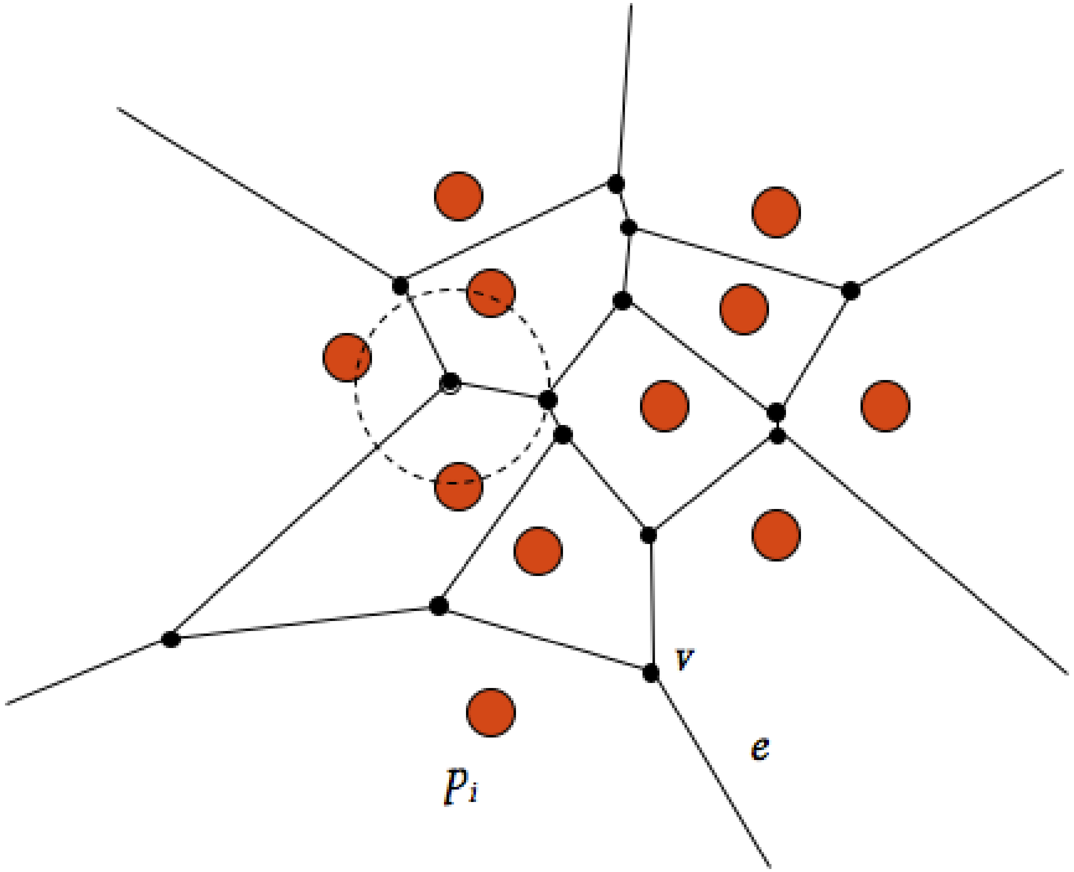
p_i : site points

e : Voronoi edge

v : Voronoi vertex

Summary of Voronoi Properties

- A point *q* is a vertex *iff* the largest empty circle centered at *q* touches at least 3 sites
  - A Voronoi vertex is an intersection of 3 more segments, each equidistant from a pair of sites



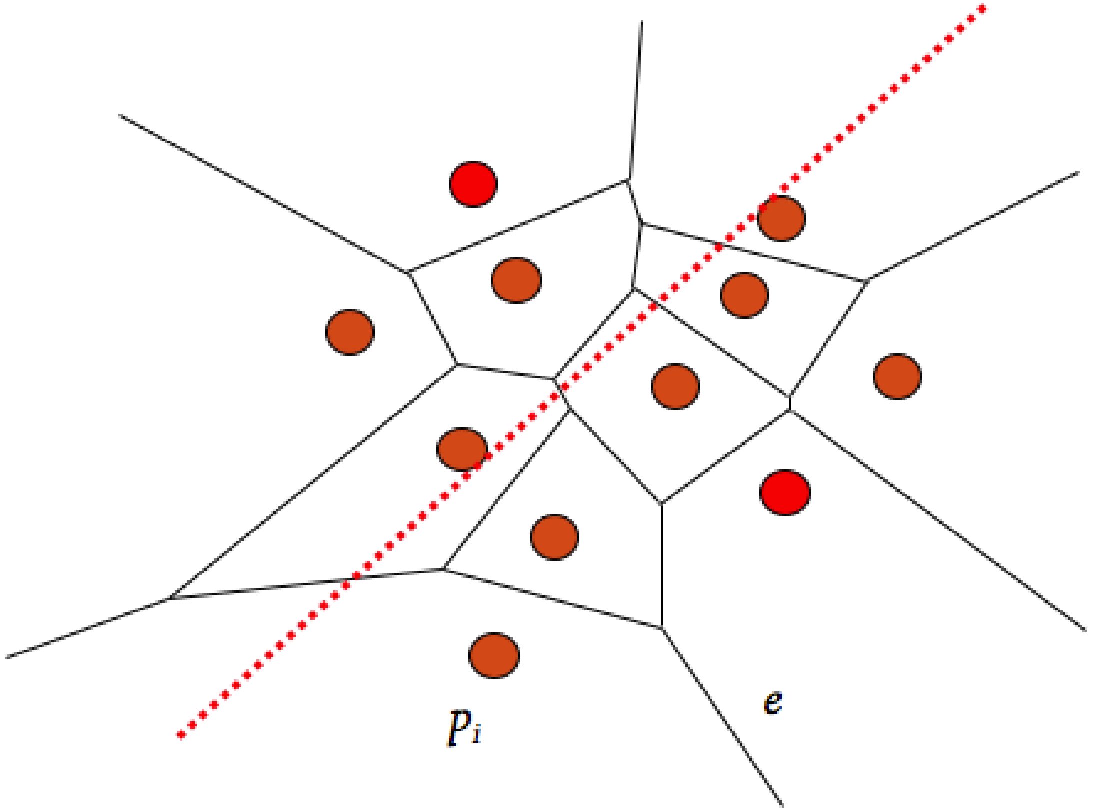
p_i : site points

e : Voronoi edge

v : Voronoi vertex

Summary of Voronoi Properties

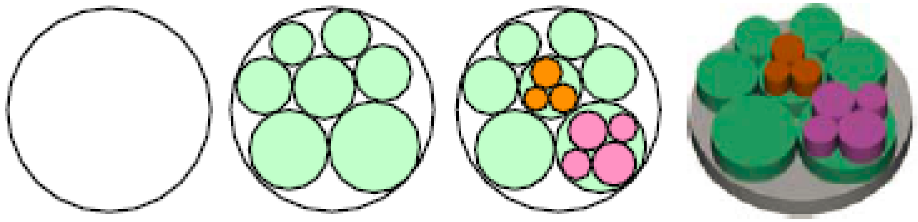
- Voronoi diagrams have linear complexity $\{|v|, |e| = O(n)\}$
  - Intuition: Not all bisectors are Voronoi edges!



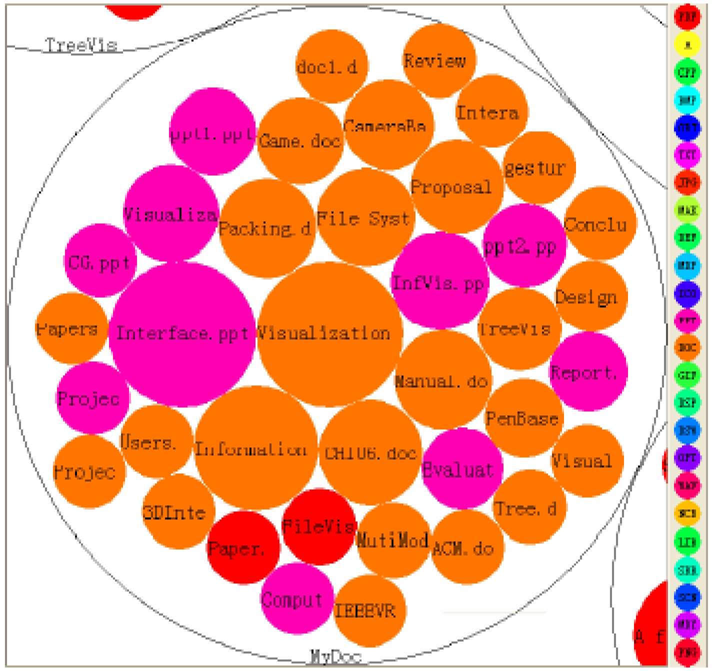
p_i : site points

e : Voronoi edge

Circle Packing



(a) Level 0 (b) Level 1 (c) Level 2 (d) 3D view



Weixin Wang, Hui Wang, Guozhong Dai, and Hongan Wang. 2006. Visualization of large hierarchical data by circle packing. In Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI '06), pg. 517-520.

Hierarchical Data Analysis

What is Hierarchical Clustering?

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

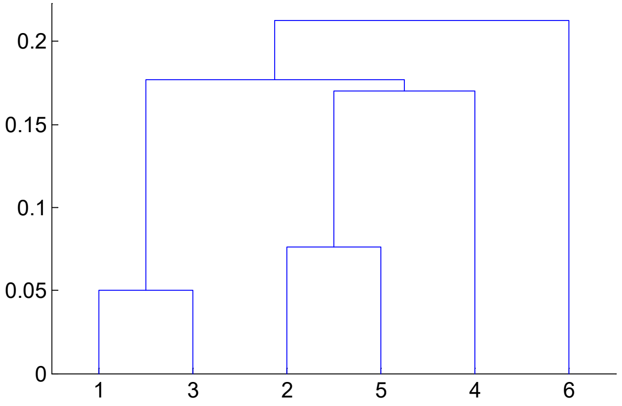
Objective

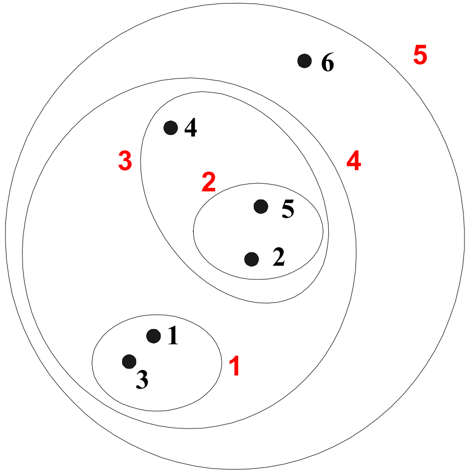
Apply methods of
hierarchical data analysis

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree

- Can be visualized as a dendrogram (along with other options)
  - A tree-like diagram that records the sequences of merges or splits





Strengths of Hierarchical Clustering

- No assumptions on the number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level
- Hierarchical clusterings may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., phylogeny reconstruction, etc), web (e.g., product catalogs) etc

Hierarchical Clustering

Agglomerative:

- Start with the points as individual clusters
- At each step, merge the closest pair of clusters until only one cluster (or **k** clusters) left

Divisive:

- Start with one, all-inclusive cluster
- At each step, split a cluster until each cluster contains a point (or there are **k** clusters)

Traditional hierarchical algorithms use a similarity or distance matrix

- Merge or split one cluster at a time

Complexity of Hierarchical Clustering

- Distance matrix is used for deciding which clusters to merge/split
- Not usable for large datasets
- At least quadratic in the number of data points

Hierarchical Data Analysis

Agglomerative Clustering

Ira A. Fulton Schools of Engineering
Arizona State University

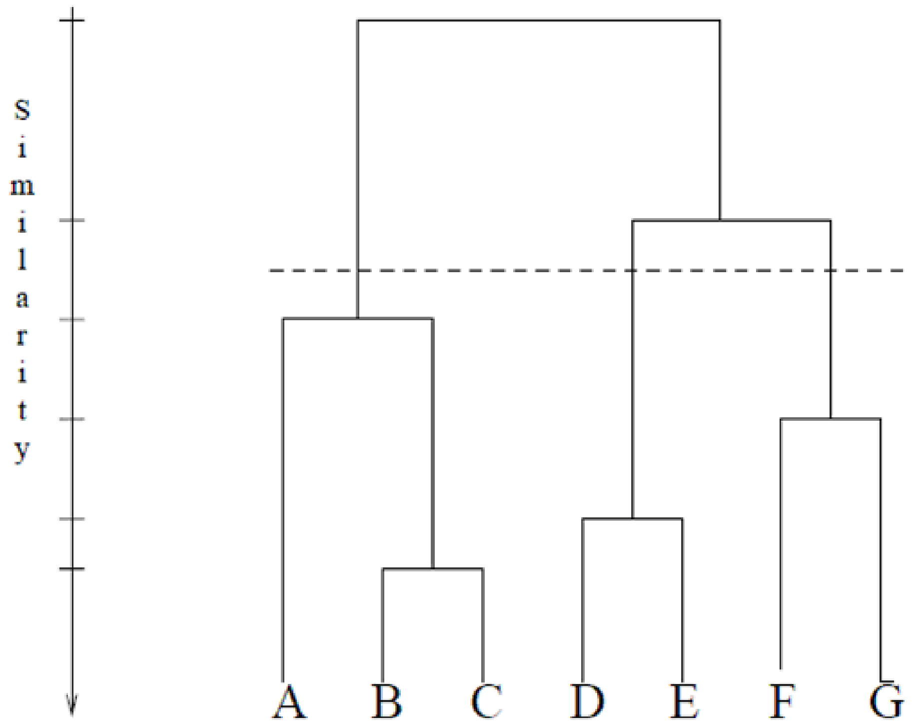
Objective

Objective

Apply methods of hierarchical data analysis

Agglomerative Clustering Algorithm

- Most popular hierarchical clustering technique



Agglomerative Clustering Algorithm

Basic algorithm

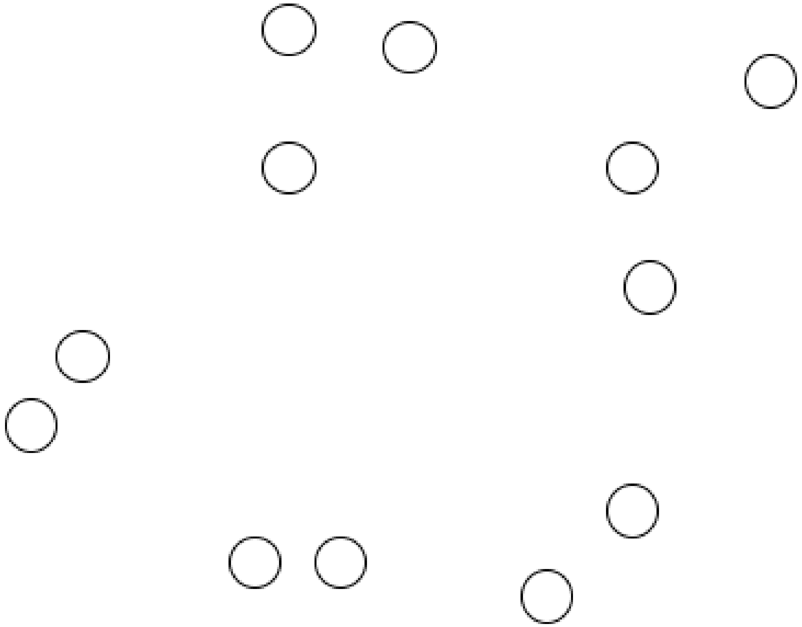
1. Compute the distance matrix between the input data points
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the distance matrix
6. **Until** only a single cluster remains

Agglomerative Clustering Algorithm

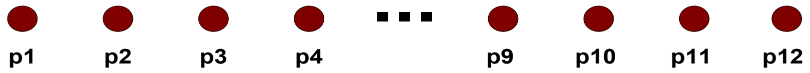
- Key operation is the computation of distance between two clusters
  - Different definitions of the distance between clusters lead to different algorithms

Hierarchical Clustering: Input/Initial Setting

Start with clusters of individual points and a distance/proximity matrix

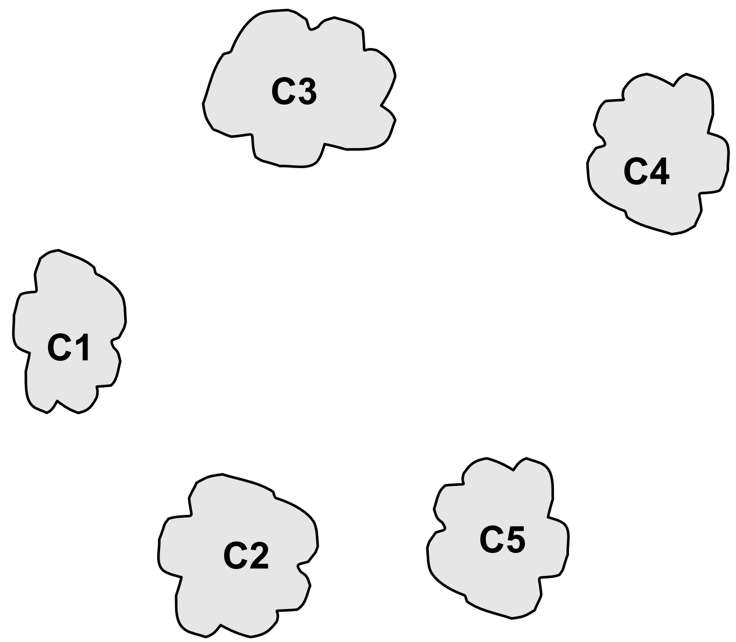


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						



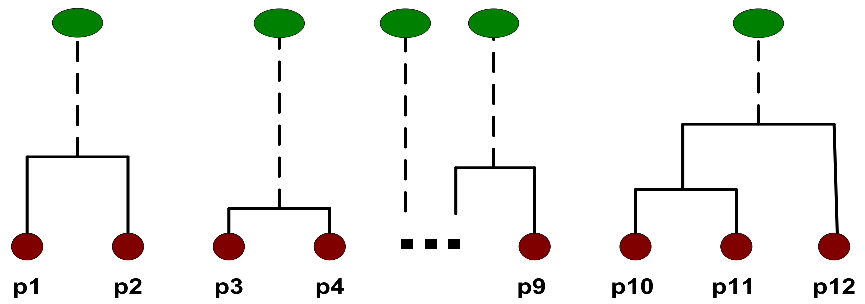
Intermediate State

After some merging steps, we have some clusters



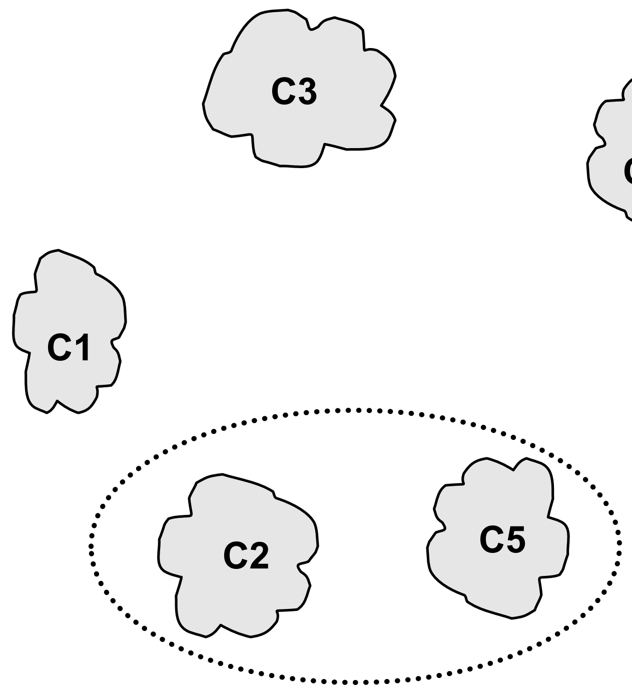
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

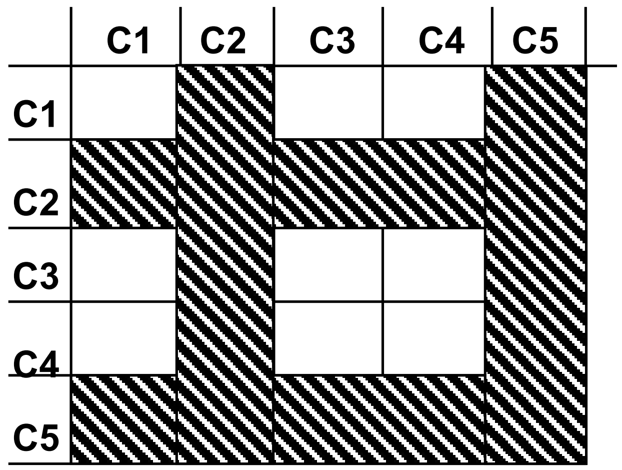
Distance/Proximity Matrix



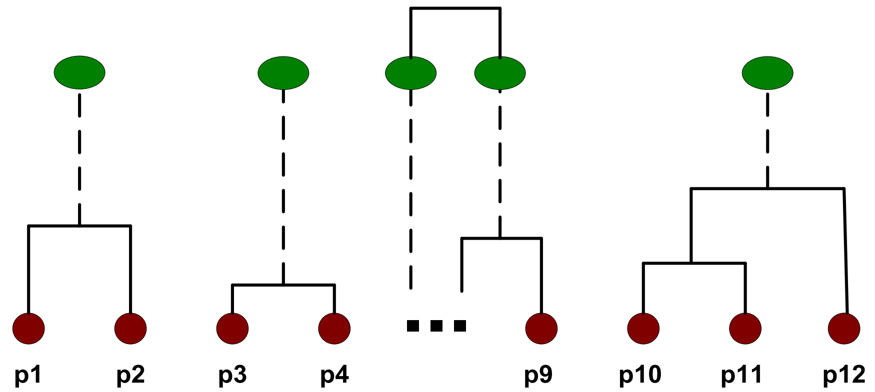
Intermediate State

Merge two closest clusters (C2 and C5) and update distance matrix



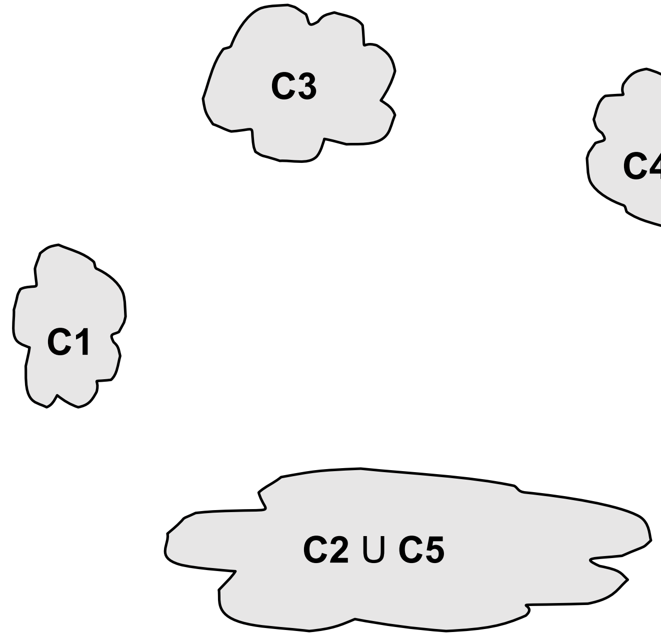


Distance/Proximity Matrix

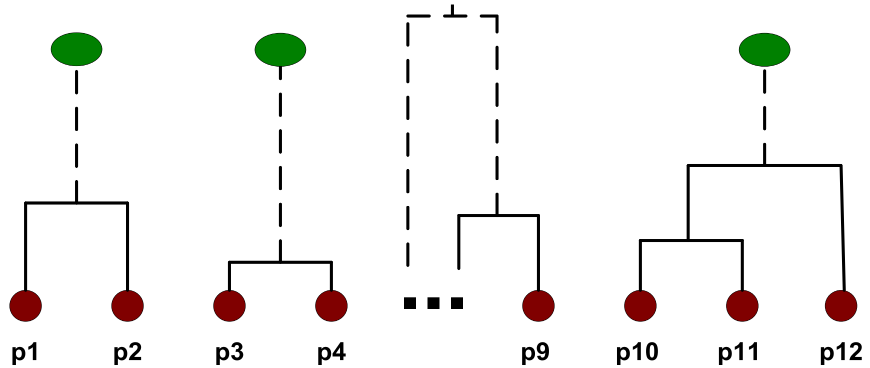


After Merging

"How do we update the distance matrix?"



	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		



Distance between two clusters

- Each cluster is a set of points

- How do we define distance between two sets of points?
  - Lots of alternatives
  - Not an easy task

Distance between two clusters

- Single-link distance between clusters C_i and C_j is the *minimum distance* between any object in C_i and any object in C_j

- The distance is defined by the two most similar objects

$$D_{sl}(C_i, C_j) = \min_{x,y}\{d(x, y) \mid x \in C_i, y \in C_j\}$$

Hierarchical Data Analysis

Distance Metrics in Hierarchical Clustering

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

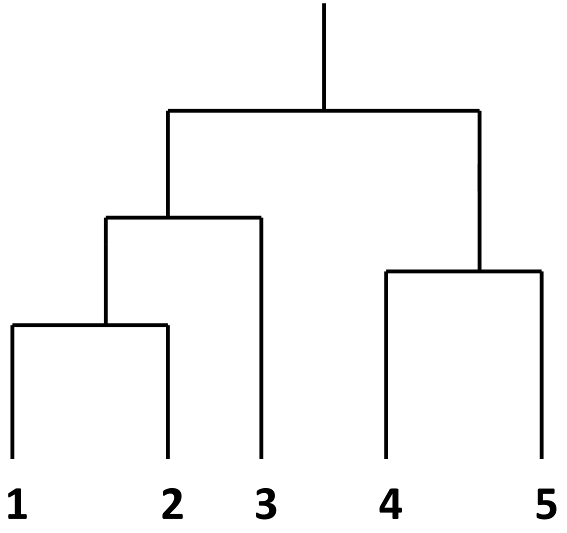
Objective

Apply methods of hierarchical data analysis

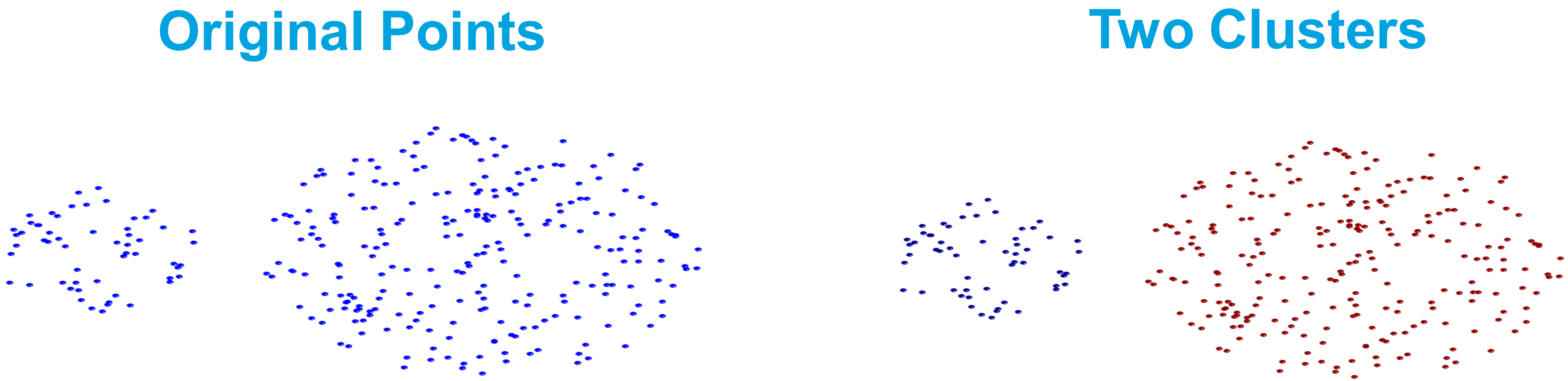
Single-link Clustering: Example

- Determined by one pair of points, i.e., by one link in proximity graph

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



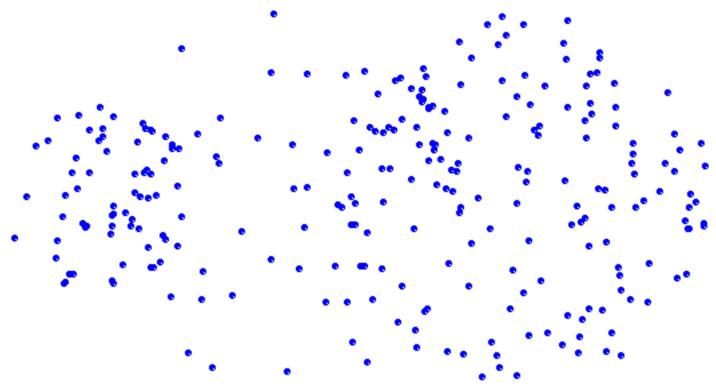
Strengths of Single-Link Clustering



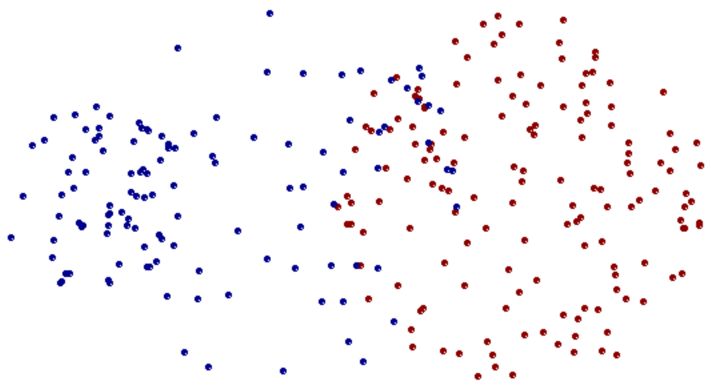
Can handle non-elliptical shapes

Limitations of Single-Link Clustering

Original Points



Two Clusters

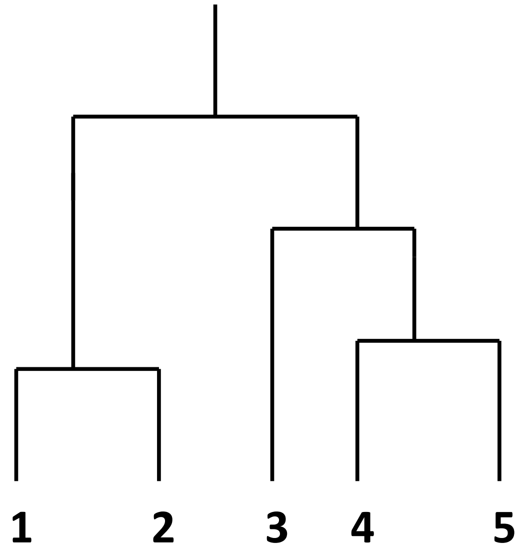


- Sensitive to noise and outliers
- It produces long, elongated clusters

Complete-link Clustering: Example

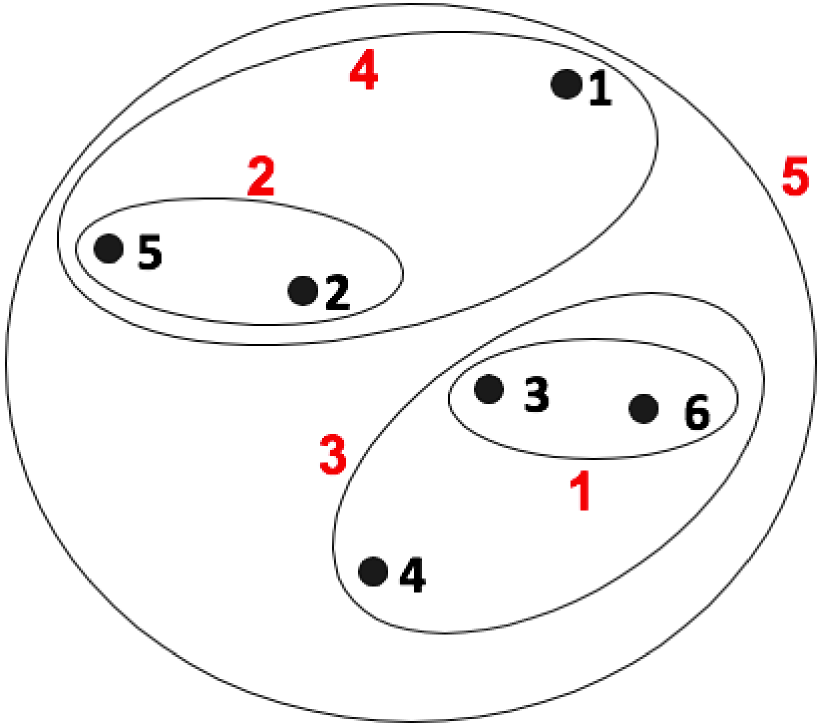
- Distance between clusters is determined by two most distant points in different clusters

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

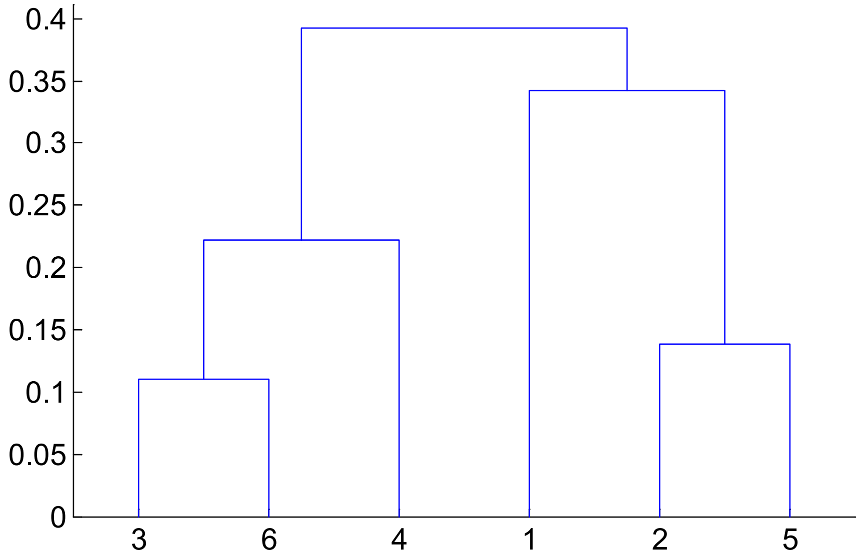


Single-Link Clustering Example

Nested Clusters

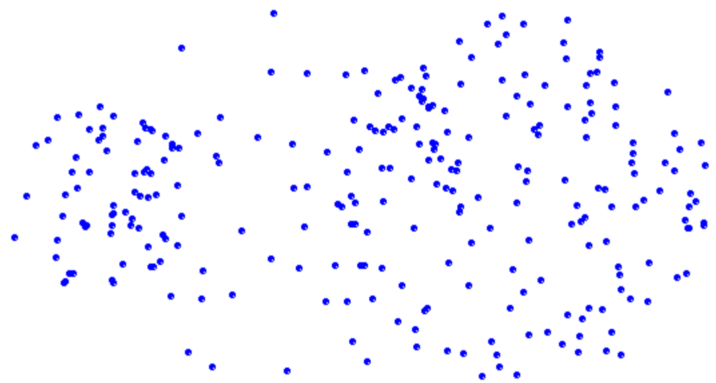


Dendrogram

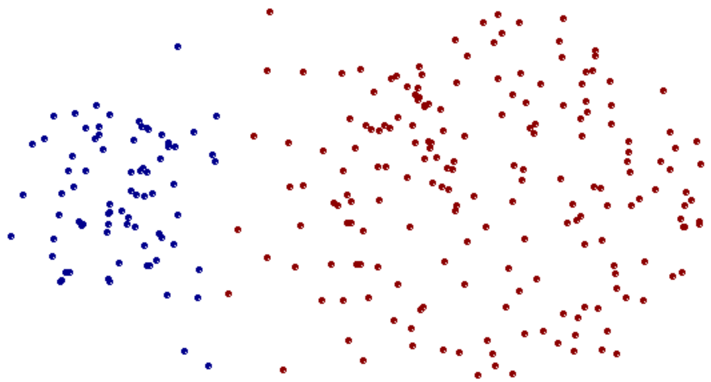


Strengths of Complete-link Clustering

Original Points



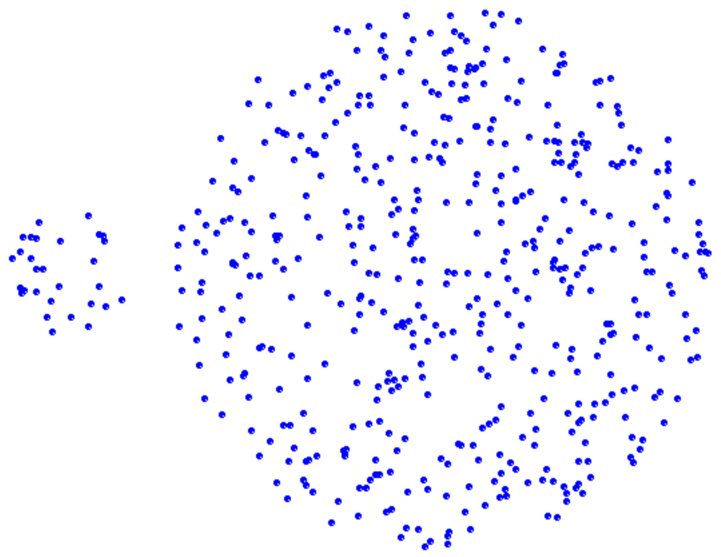
Two Clusters



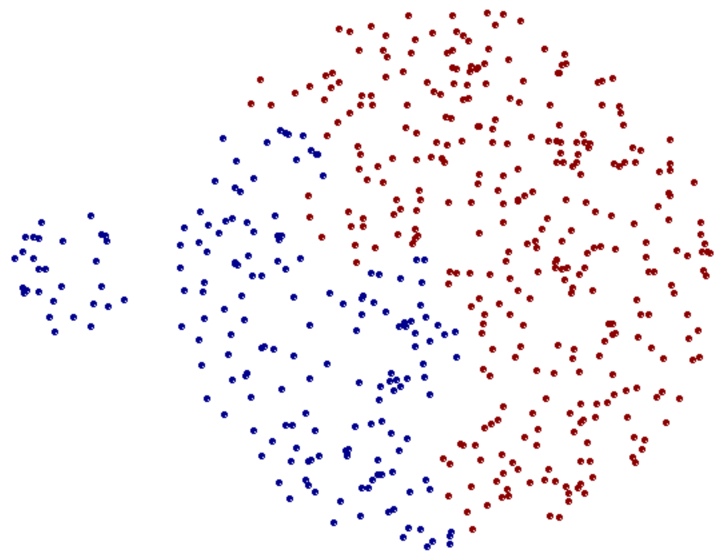
- More balanced clusters (with equal diameter)
- Less susceptible to noise

Limitations of Complete-Link Clustering

Original Points



Two Clusters

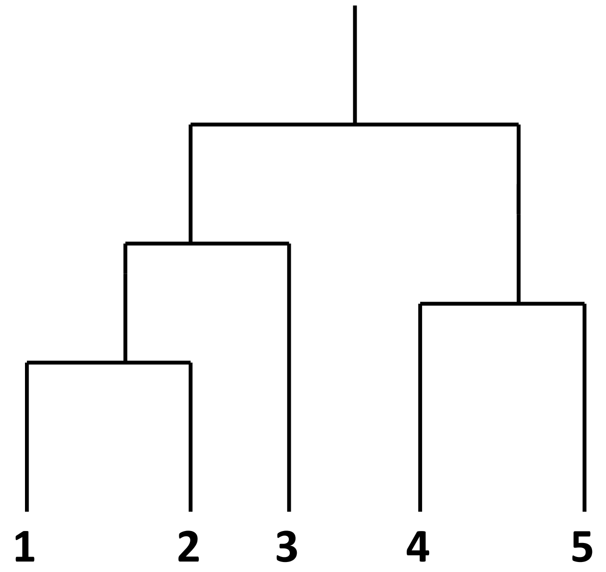


- Tends to break large clusters
- All clusters tend to have same diameter – small clusters are merged with larger ones

Average-link Clustering: Example

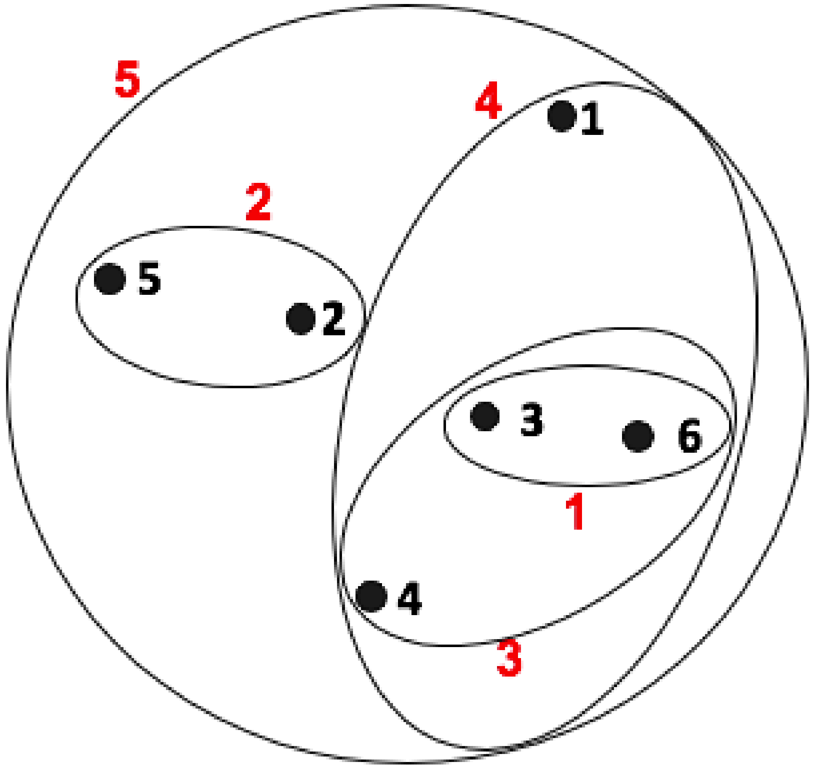
Proximity of two clusters is the average of pairwise proximity between points in the two clusters.

	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

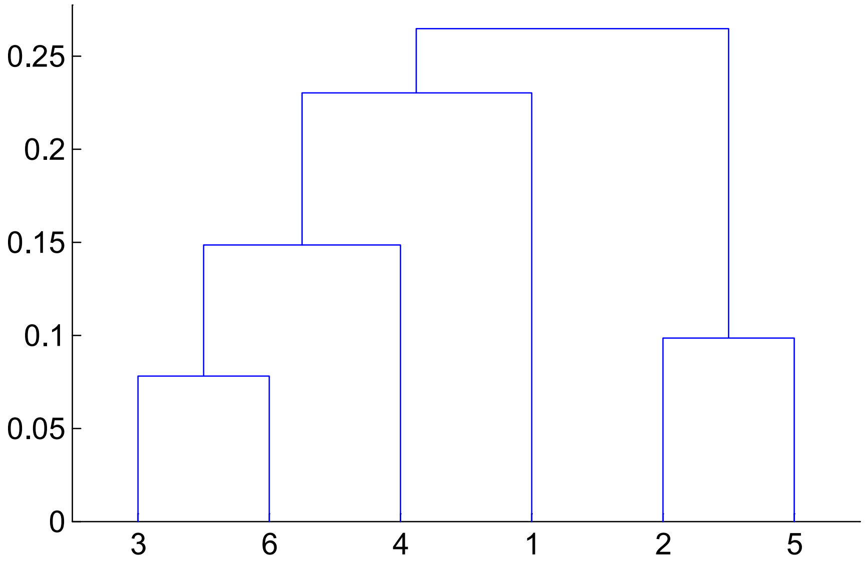


Average-Link Clustering Example

Nested Clusters



Dendrogram



Average-Link Clustering: Discussion

- Compromise between Single and Complete Link

Strengths

- Less susceptible to noise and outliers

Limitations

- Biased towards globular clusters

Distance Between Two Clusters

- Centroid distance between clusters C_i and C_j is the distance between the centroid r_i of C_i and the centroid r_j of C_j

$$D_{centroids}\left(C_i, C_j\right) = d(r_i, r_j)$$

Distance Between Two Clusters

- Ward's distance between clusters C_i and C_j is the *difference* between the *total within cluster sum of squares for the two clusters separately*, and the *within cluster sum of squares resulting from merging the two clusters* in cluster C_{ij}

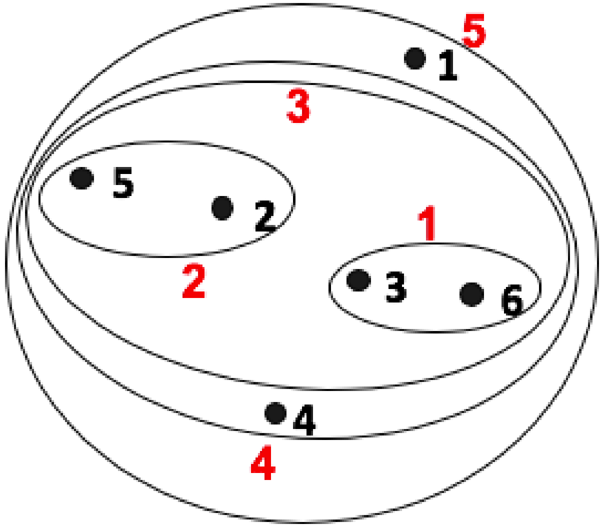
$$D_w(C_i, C_j) = \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 - \sum_{x \in C_{ij}} (x - r_{ij})^2$$

r_i: centroid of C_i
r_j: centroid of C_j
r_{ij}: centroid of C_{ij}

Ward's Distance for Clusters

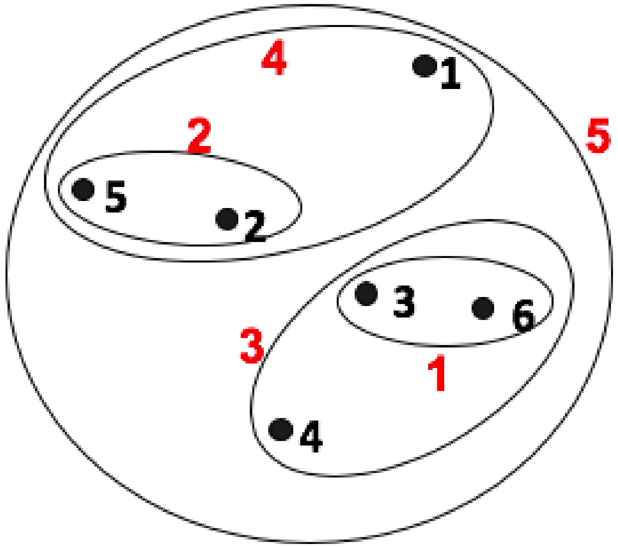
- Similar to group average and centroid distance
- Less susceptible to noise and outliers
- Biased towards globular clusters
- Hierarchical analogue of k-means
  - Can be used to initialize k-means

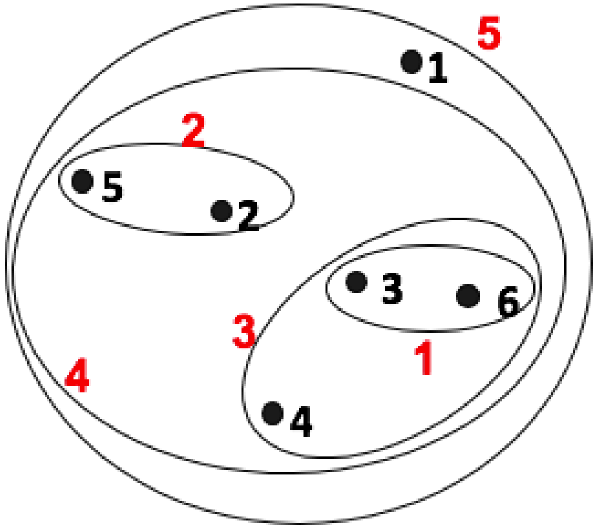
Hierarchical Clustering: Comparison



MIN

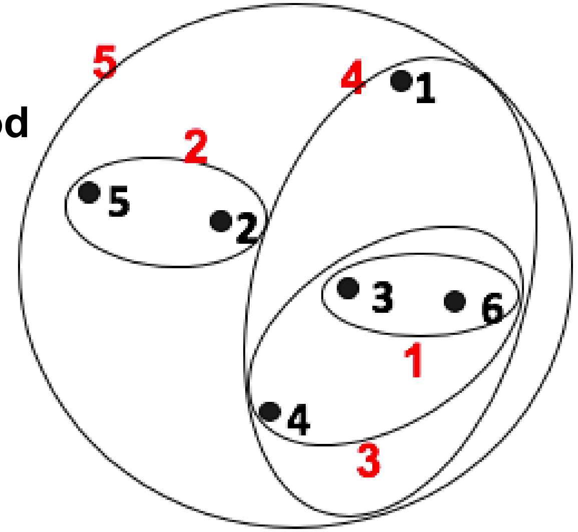
MAX





Group Average

Ward's Method



Hierarchical Clustering: Time and Space Requirements

- For a dataset X consisting of n points
- $O(n^2)$ **space**; it requires storing distance matrix
- $O(n^3)$ **time** in most of the cases
  - There are n steps and at each step the size n^2 distance matrix must be updated and searched
  - Complexity can be reduced to $O(n^2 \log(n))$ time for some approaches by using appropriate data structures

Hierarchical Clustering Issues

- Distinct clusters are not produced
- Methods for producing distinct clusters but involve specifying somewhat arbitrary cutoff values
- What if data doesn't have a hierarchical structure?
- Is HC appropriate?