10

Information Visualization

Overview

Techniques for high-dimensional data

- scatter plots, PCA
- parallel coordinates
- link + brush
- pixel-oriented techniques
- icon-based techniques

Techniques for hierarchical data and networks

- trees: tree maps
- graph clustering
- distortion, focus+context

High-dimensional data

"Dimension" refers often to data channels (attributes), not to true spatial dimension (coordinates).

Roles of data and coordinates can be swapped:

In scatter plots (multi-dimensional histograms) data become coordinates and vice versa

Often no spatial coordinates exist, e.g. in visualization of (relational) data bases.

Scatter plots

(2D) scatter plots are projections to 2D subspaces spanned by pairs of coordinate axes.

n-dimensional data lead to a *n*x*n* matrix of scatter plots.

For small *n*, the matrix can directly serve as a visualization.

Example (*n*=4):

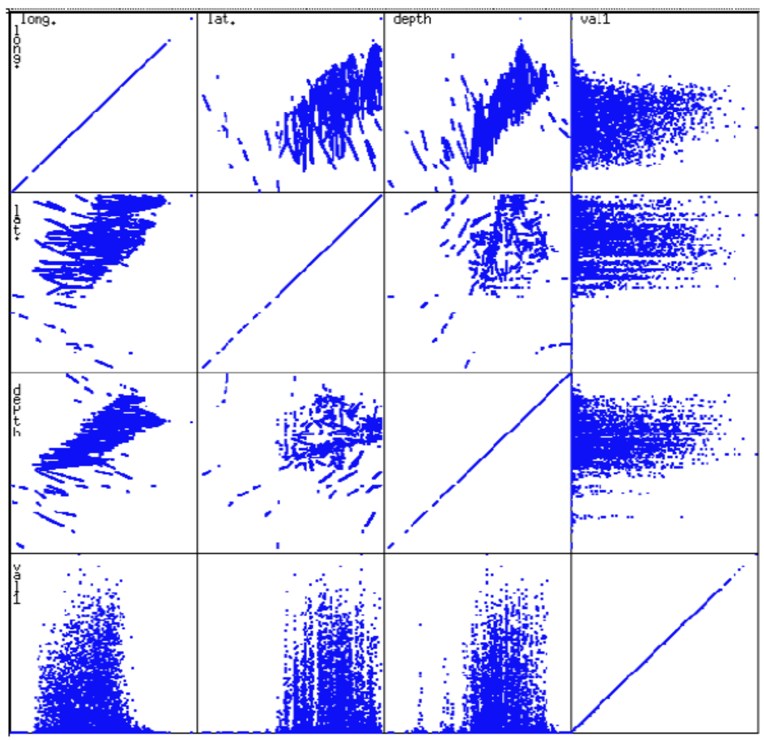


Image credit: M. Ward

Dimension reduction

Often the *n* attributes are not independent and the scatter plot lies almost in a *k*-dimensional linear subspace.

Principal component analysis (PCA):

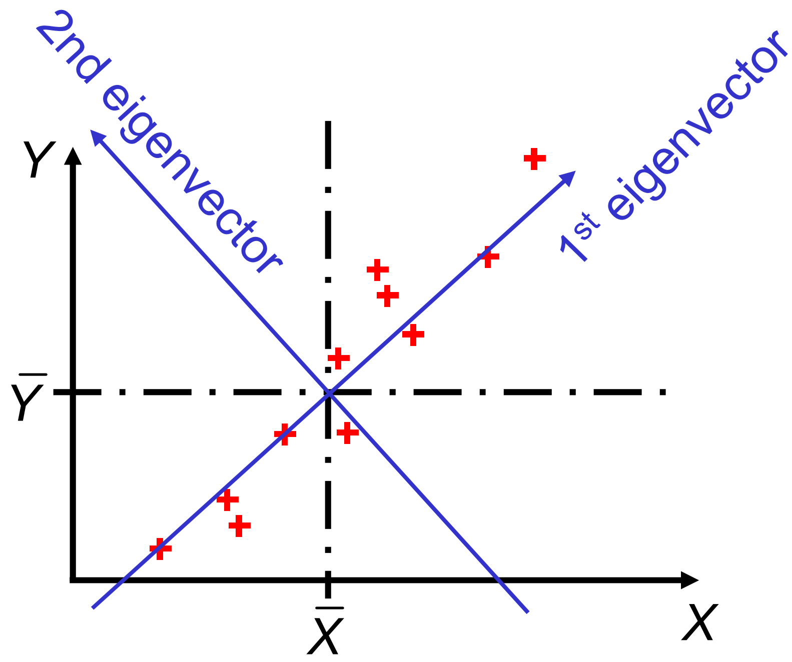
- For each pair (X,Y) of attributes compute the covariance

$$\operatorname{cov}(X,Y) = \frac{\sum_{i=1}^{n} \left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{n-1}$$

resulting in the (symmetric, positive semidefinite) covariance matrix

$$C = \begin{pmatrix} \operatorname{cov}(X,X) & \operatorname{cov}(X,Y) & \operatorname{cov}(X,Z) \\ \operatorname{cov}(Y,X) & \operatorname{cov}(Y,Y) & \operatorname{cov}(Y,Z) \\ \operatorname{cov}(Z,X) & \operatorname{cov}(Z,Y) & \operatorname{cov}(Z,Z) \end{pmatrix}$$

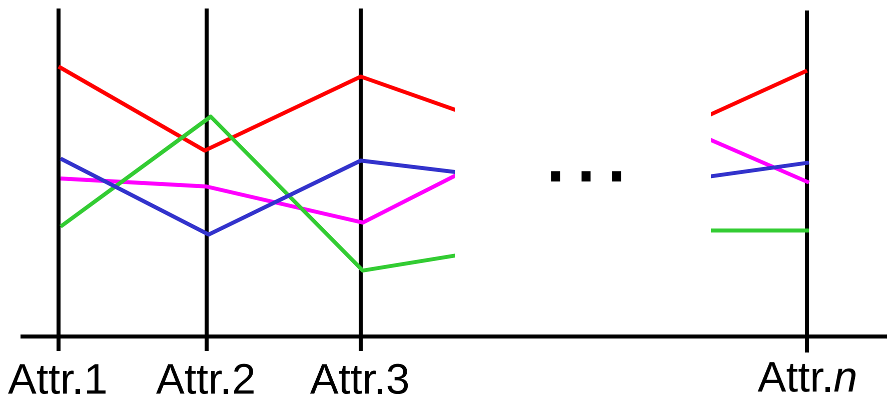
- Compute the eigenvalues (nonnegative) and eigenvectors (orthogonal)
- Sort eigenvectors by descending eigenvalues
- Project data to subspace defined by the mean $(\bar{X}, \bar{Y}, \bar{Z}, \cdots)$ and the first *k* eigenvectors (directions of largest data variation)



Parallel coordinates

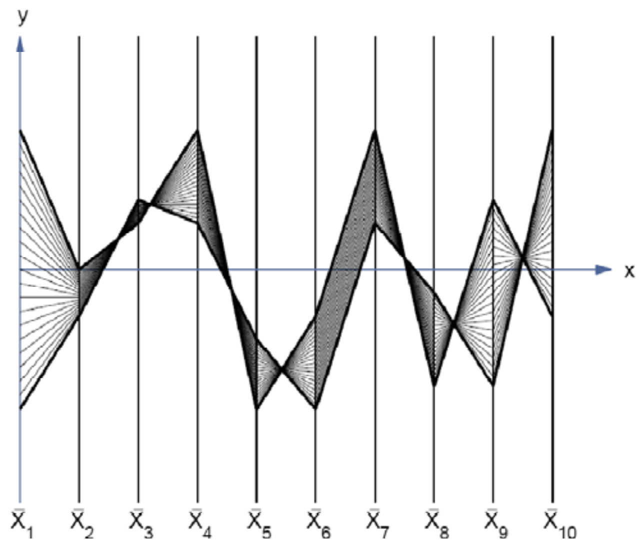
Visualization method of parallel coordinates (Inselberg1985):

- *n* parallel and equidistant axes (one per attribute)
- axes scaled to [min, max] range of corresponding attribute
- every data item is represented by a polyline which intersects each of the axes at the point corresponding to its attribute value

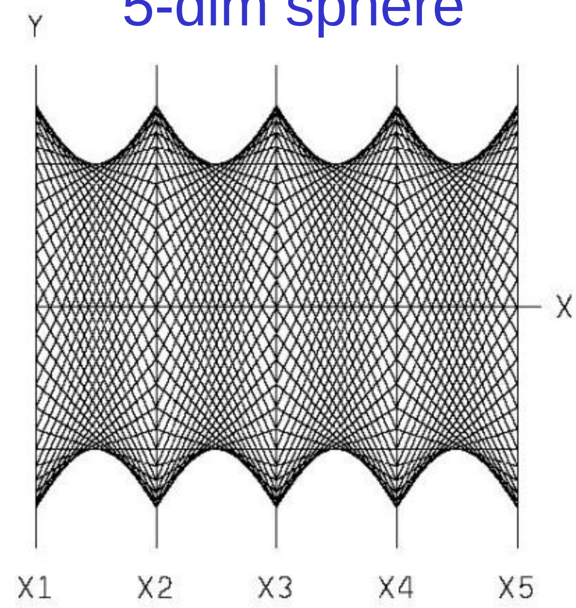


What can be done with parallel coordinates?

line in 10-space



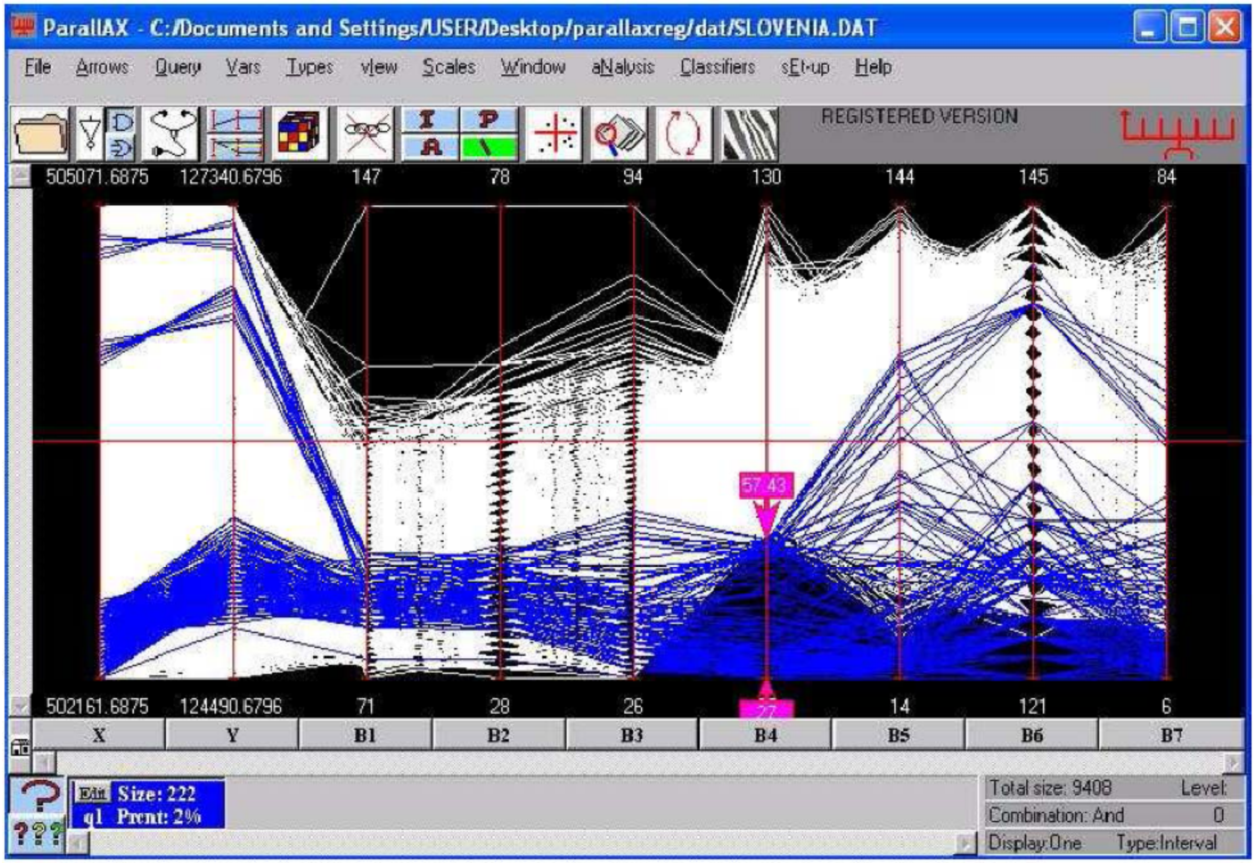
5-dim sphere



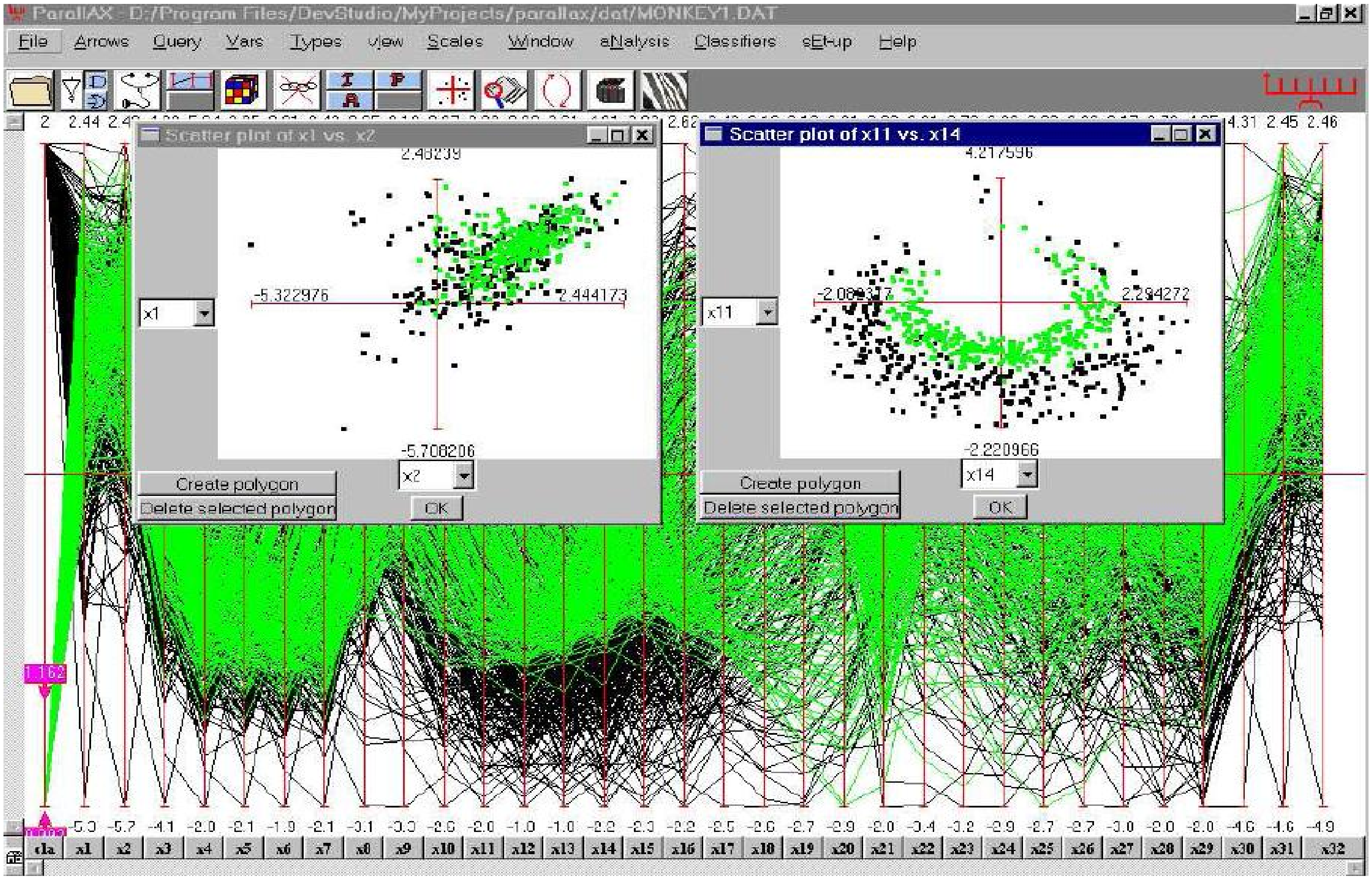
- Linear or spherical arrangement can be "seen" (according to Inselberg)
- Algorithm for testing if a point is in the convex hull of a set of points: check if the polyline is within the two envelopes of the set of polylines

Queries in parallel coordinates:

- Brushing technique (example: "ParallAX" tool by A. Inselberg)



- Linked views (link & brush technique)

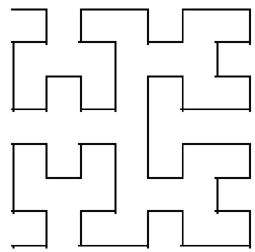


Pixel-oriented techniques

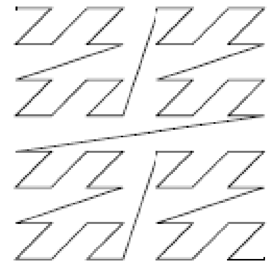
Space-filling curves for query-independent visualization of database

Idea: represent each record by a single pixel

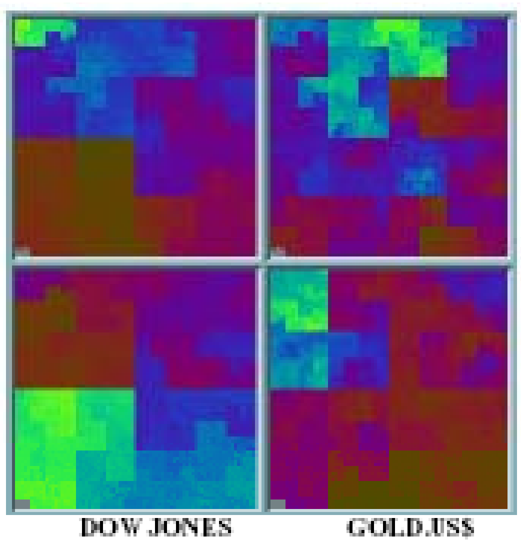
- map one attribute to color, map sorting key to space-filling curve

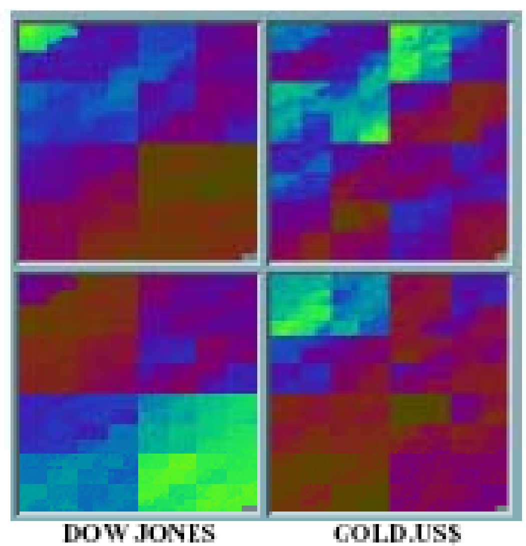


Peano-Hilbert



Morton (Z-Curve)

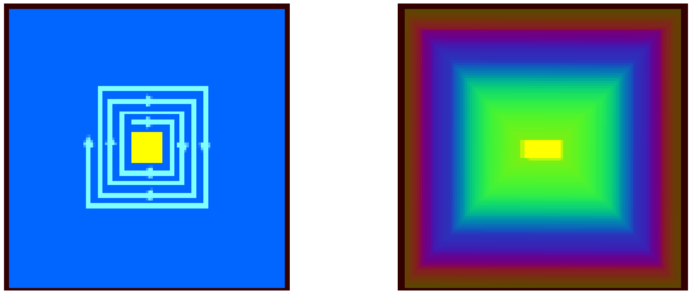




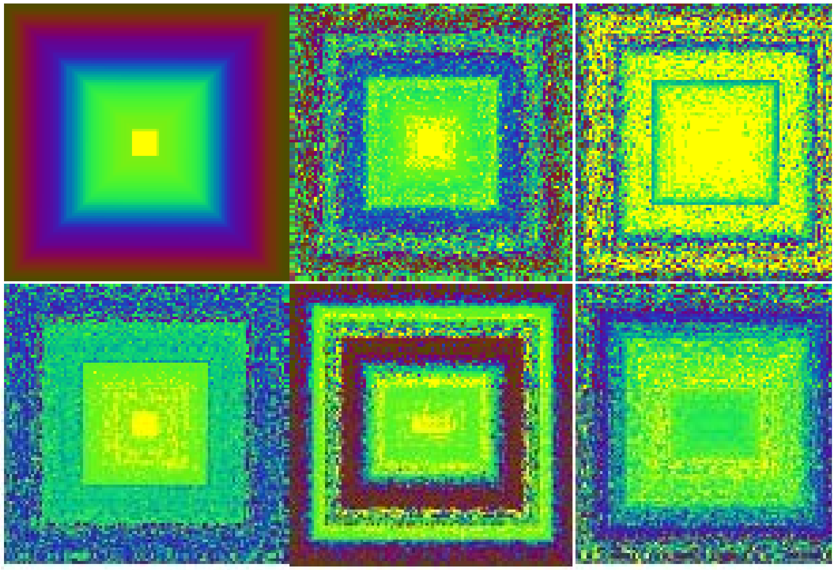
Spiral technique (Keim) for query dependent visualization.

- sort records (near a query point) by distance to query
- map sorted list to spiral

Spiral technique:



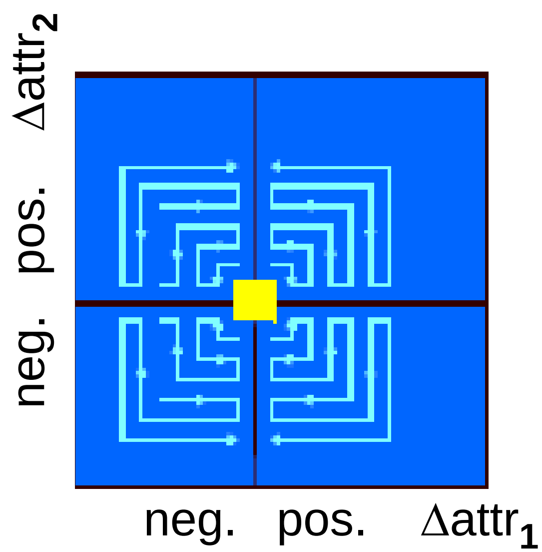
Example: result of a complex query

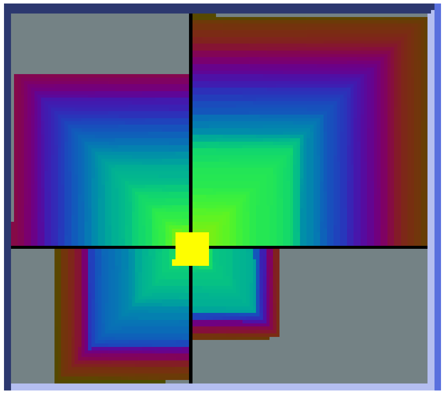


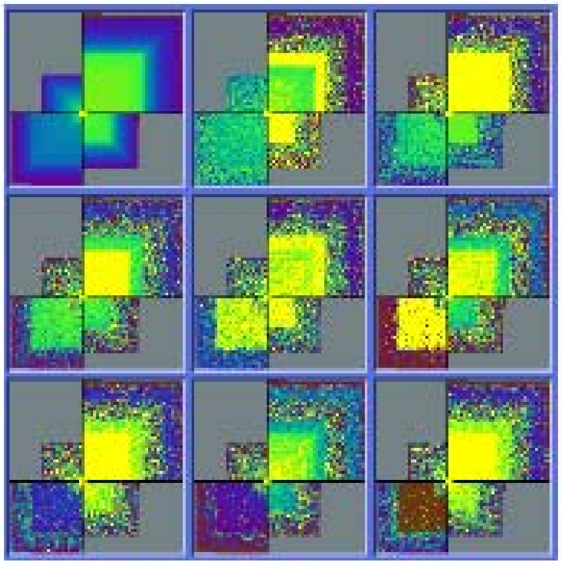
Color coding: Overall distance and distance of attributes 1..5

Axes technique (Keim) for query dependent visualization.

- for two selected attributes, separate space into lower/higher attribute values
- draw spirals per quadrant







Color coding: Overall distance and distance of attributes 1..8

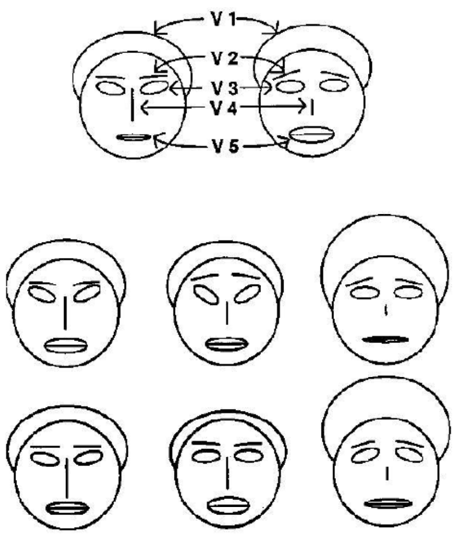
Icon-based techniques

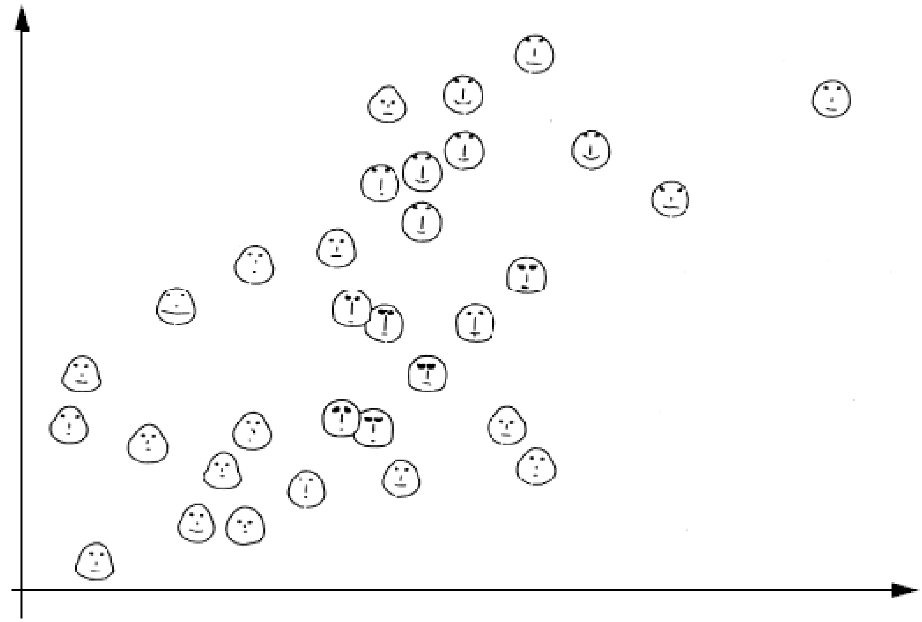
Chernoff Faces:

- two attributes are mapped to the display axes
- remaining attributes are mapped to shape and size of hair, eyebrows, eyes, nose, mouth, etc.

Idea: Use the human ability to recognize and memorize faces.

Example:





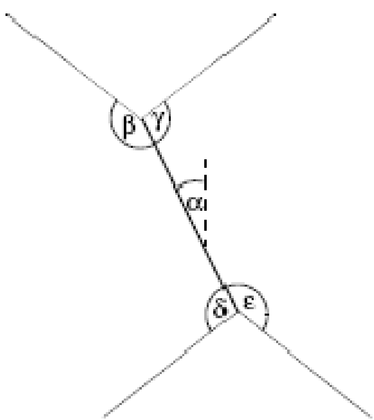
Stick figures (Grinstein)

- two attributes are mapped to the display axes
- remaining attributes are mapped to lengths of limbs or angles between them

Idea: Texture pattern in visualization shows certain characteristics.

Example:

Stick figure icon



a family of stick figures

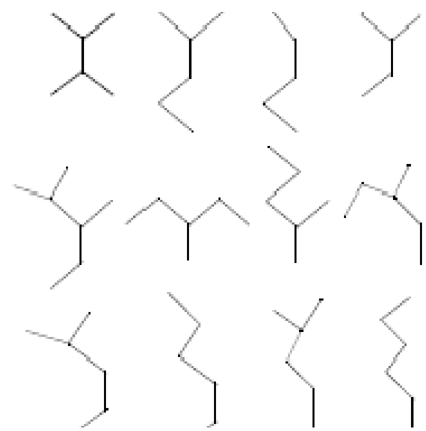
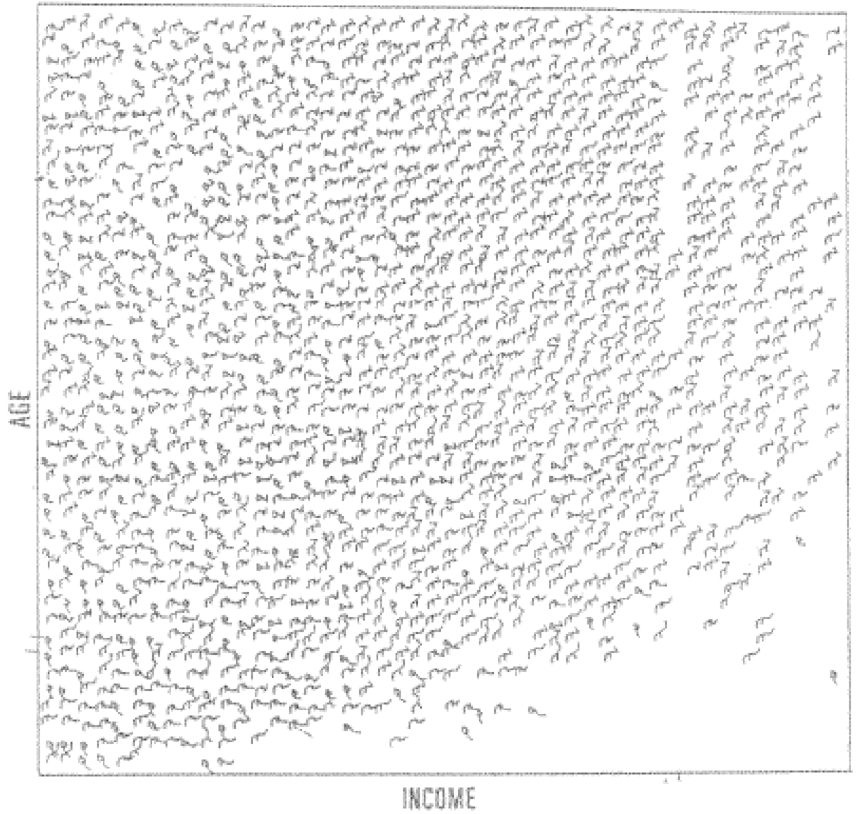


Image credit: G. Grinstein

Example: census data (age, income, sex, education, etc.)



It can be observed that the structure is more homogenous for higher incomes than for lower ones.

Hierarchical and network data

Mathematical description of hierarchies and networks: graphs.

Some important special types of graphs:

- undirected graphs
- directed graphs
- directed acyclic graphs (DAGs)
- rooted trees
- unrooted trees
  (In an unrooted tree, every node can be chosen as the root.)
- forests, etc.

Cone trees

Cone trees (Robertson) are 3D embeddings of trees.

- Children arranged on circular cones
- Navigation by interactive rotation at all hierarchy levels

Useful for trees with high branching (no binary trees!)

Example: file system visualization

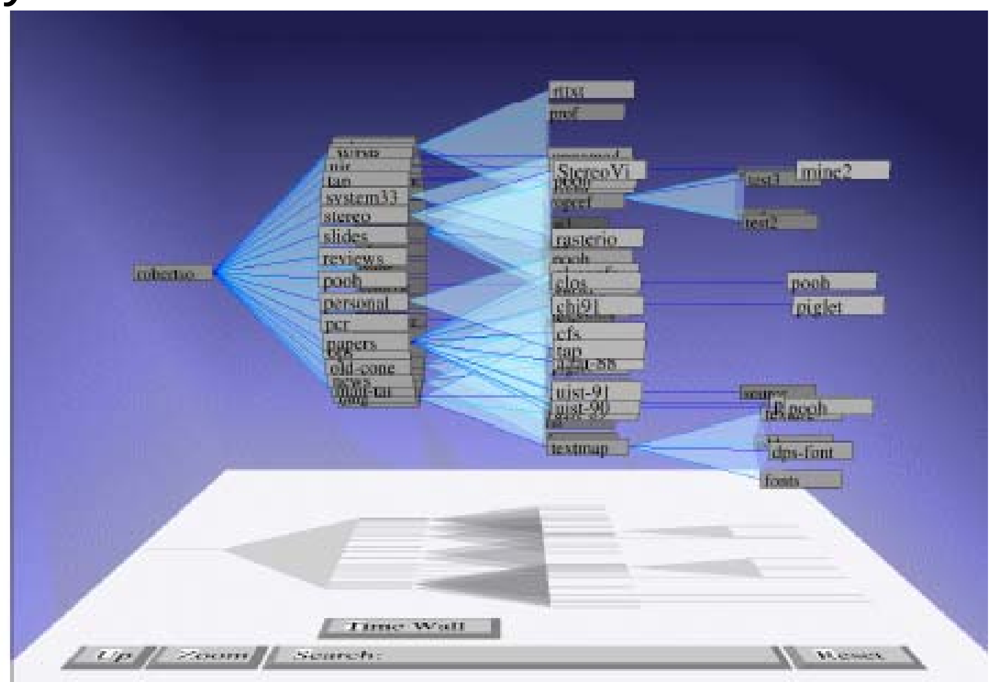


Image credit: S. Card

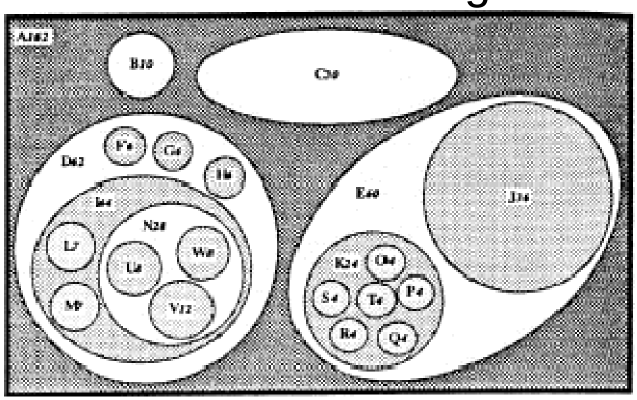
Tree maps

Trees with weight attribute at nodes can be visualized using tree maps (Johnson and Shneiderman).

Tree maps are special Venn diagrams where

- subtrees are represented by rectangles
- rectangle area is proportional to total weight of the subtree
- split direction is vertical/horizontal for odd/even hierarchy level
- nodes can have colors, labels, tool-tip info, etc.

Standard Venn diagram



Tree map

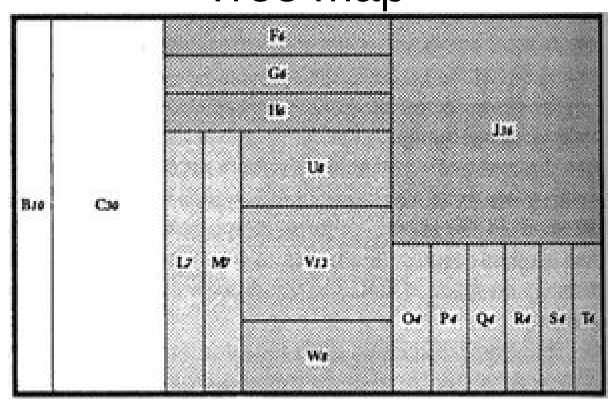


Image credit: B. Shneiderman

Example: file system with 1000 files

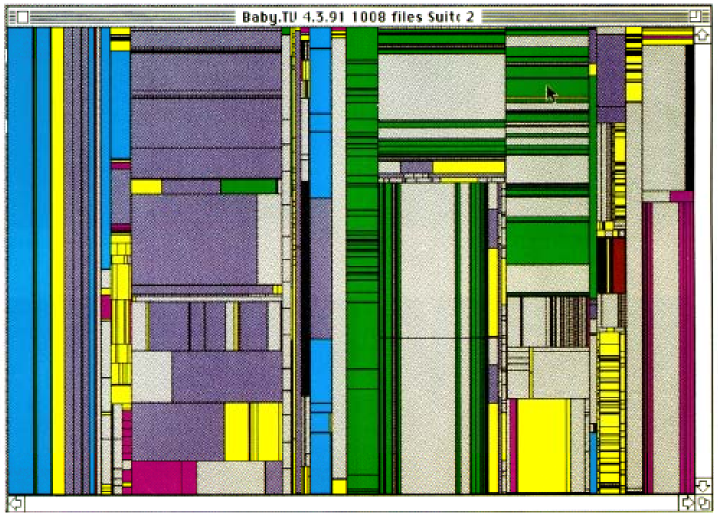
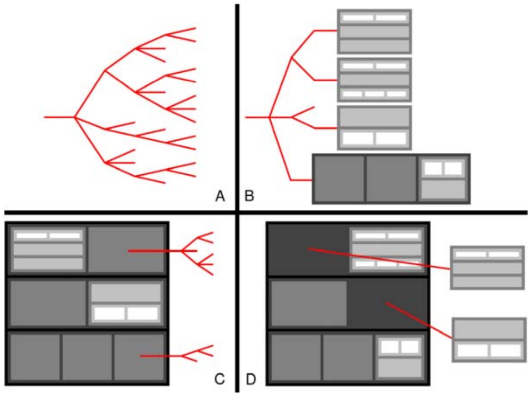


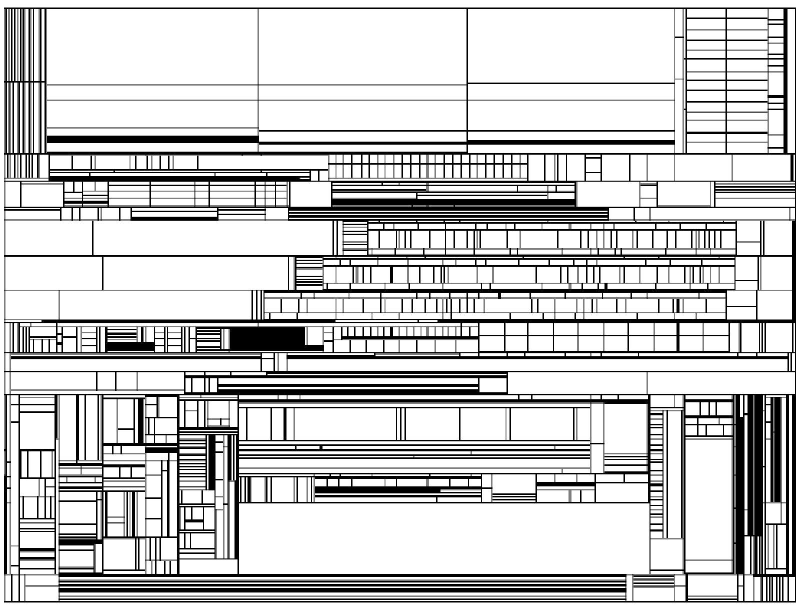
Image credit: B. Shneiderman

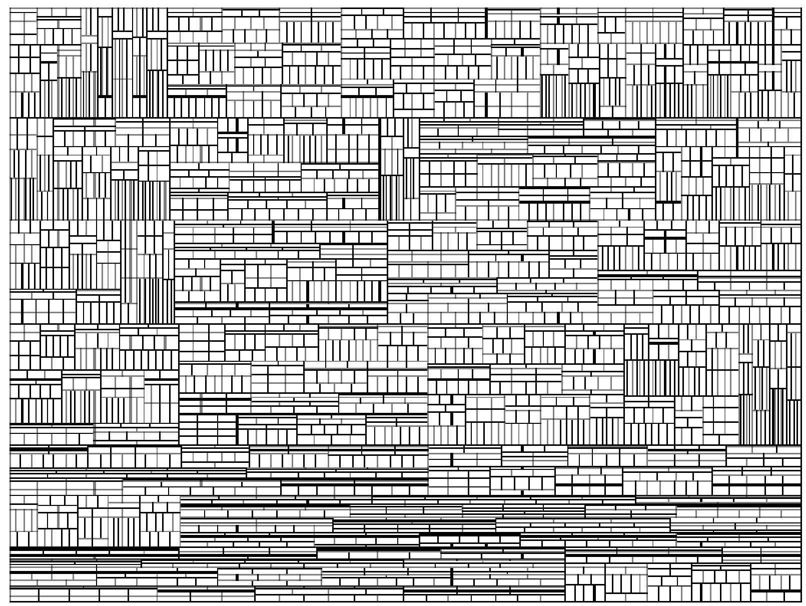
Application: Combining tree maps and node-link diagrams (Zhao)



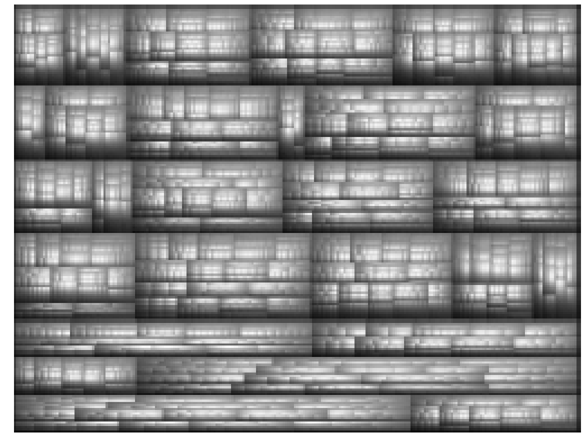
Problem of tree maps:
In large trees, hierarchical levels can be hard to see

Examples "file system" and "organization"



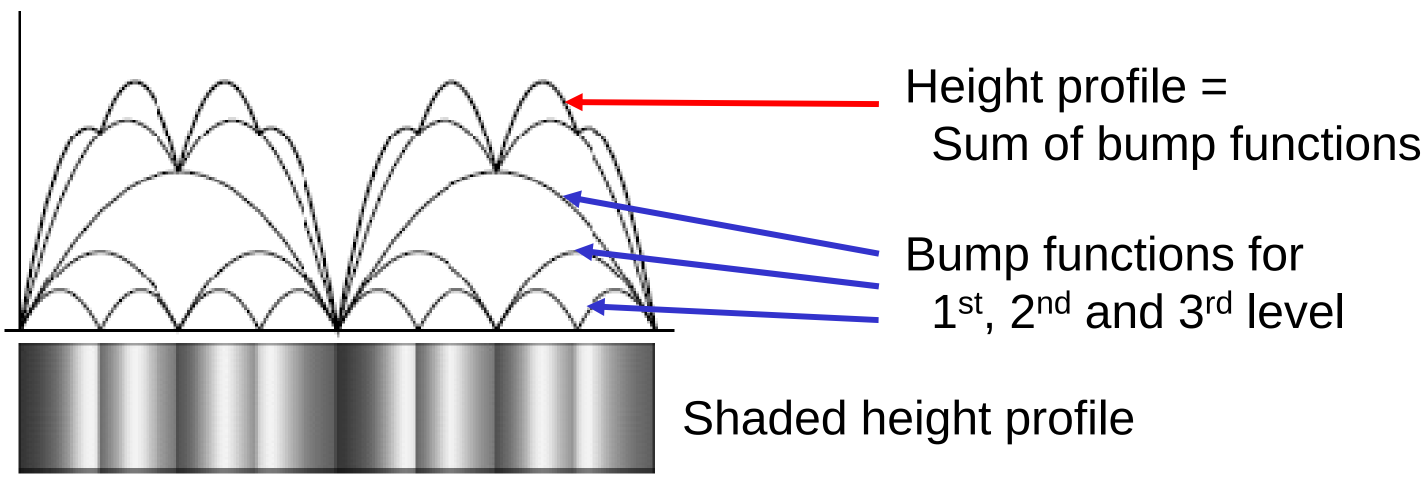


Solution: Cushion tree maps (van Wijk, van de Wetering 99)



Idea: give rectangles a height profile, with height depending on the hierarchy level

Example (1D): height profile for a binary tree.



Height function for an interval $\left[x_1, x_2\right]$ at the ith level

$$\Delta z\left(x\right) = f^{\,i}\,\frac{4h}{x_2 - x_1}\left(x - x_1\right)\left(x_2 - x\right)$$

It defines a bump with a peak height of

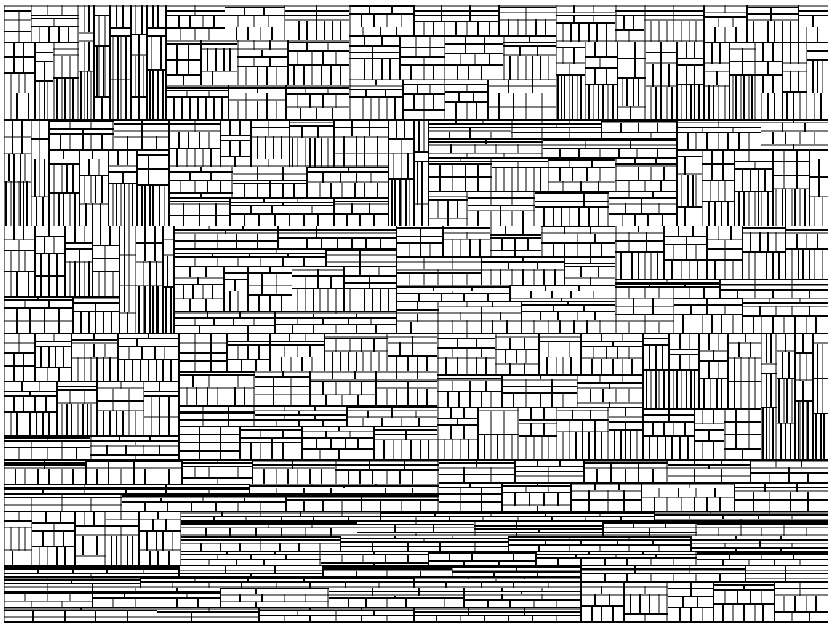
$$f^{\,i} h\left(x_2 - x_1\right)$$

(with user-defined parameters *h* and *f*)

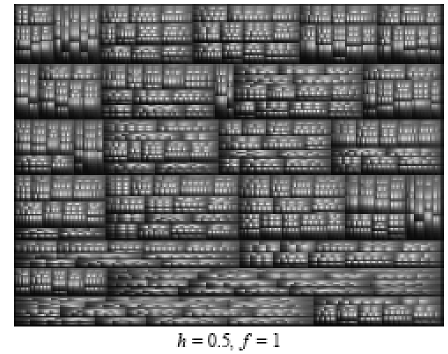
Modification for 2D: Alternate between vertical and horizontal "ridges", i.e. for even numbers *i* use the function:

$$\Delta z\left(x, y\right) = f^{\,i}\,\frac{4h}{y_2 - y_1}\left(y - y_1\right)\left(y_2 - y\right)$$

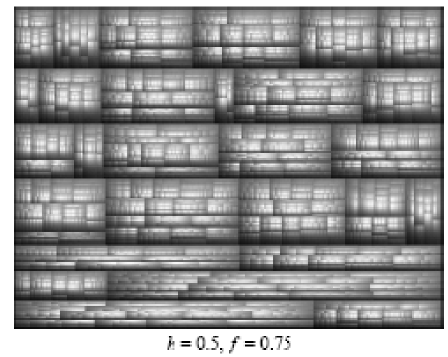
Standard vs. cushion tree maps in "organization" example



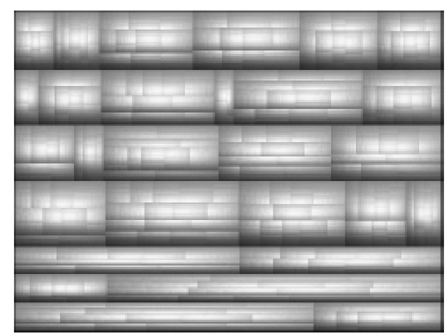
$h = 0.5, \; f = 1$



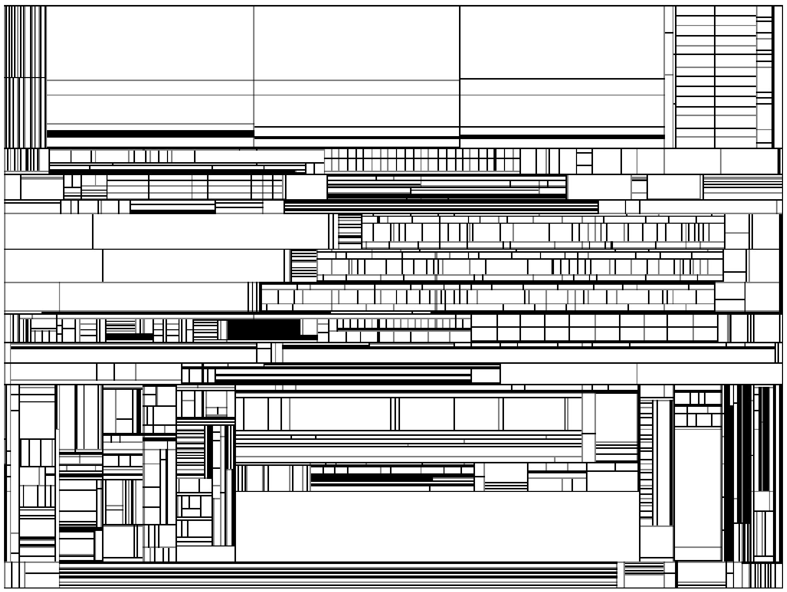
$h = 0.5, \; f = 0.75$

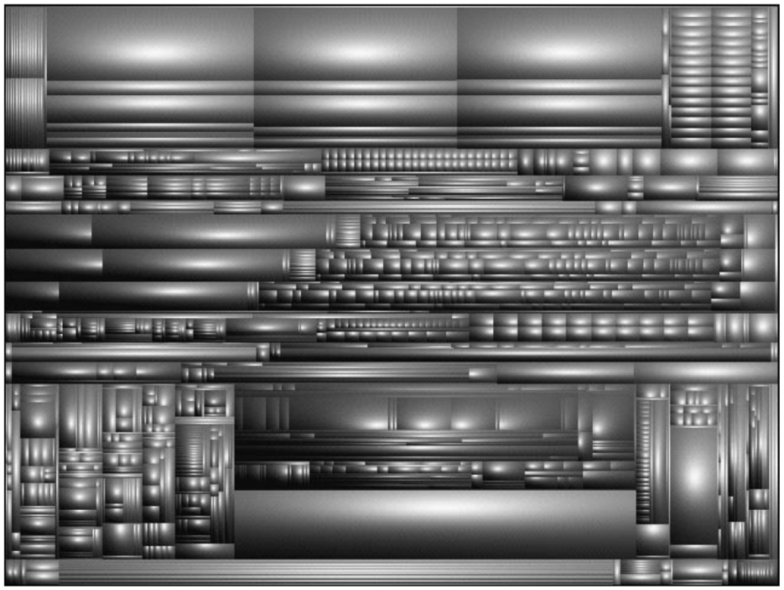


$h = 0.5, \; f = 0.5$



Standard vs. cushion tree maps in "file system" example

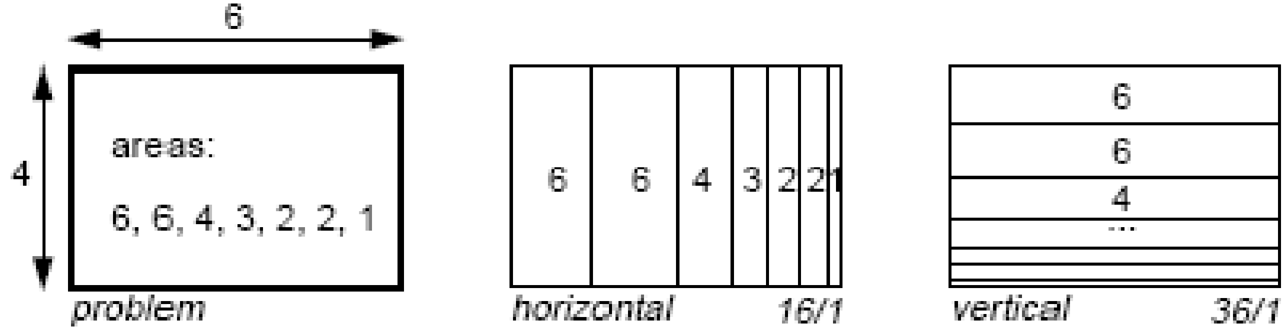




h = 0.5, f = 1

2nd problem of tree maps: bad aspect ratios

Example: 7 children with weights 6,6,4,3,2,2,1:



Worst aspect ratios: 16:1 and 36:1

Solution: Squarified tree maps (Bruls et al.)

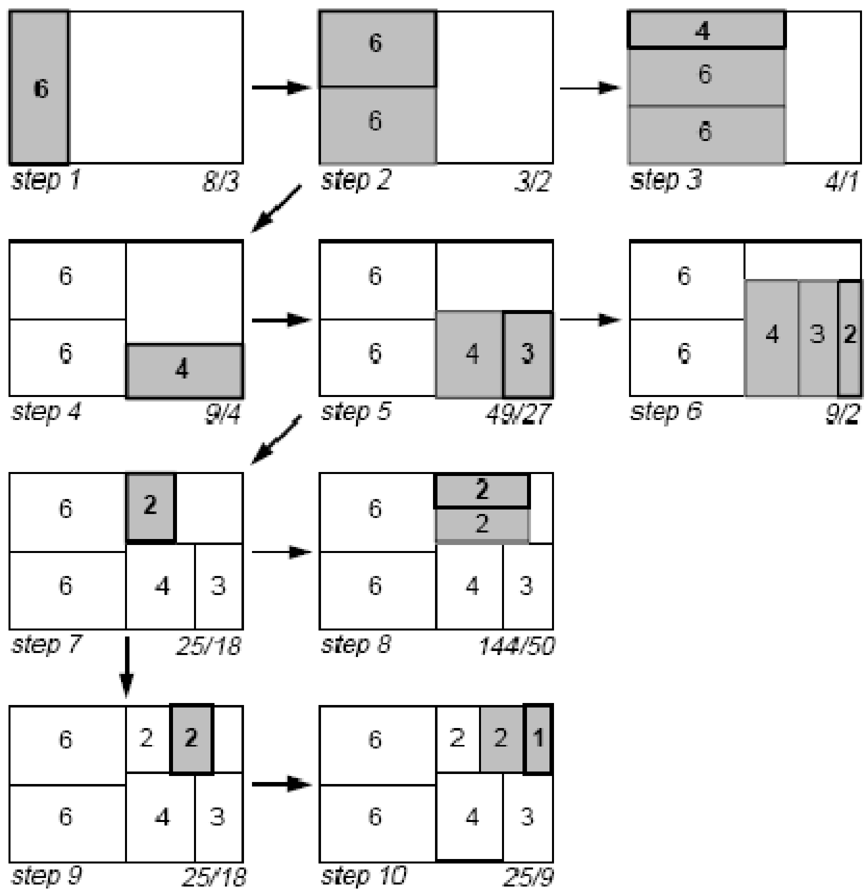
Idea: allow both vertical and horizontal splits within the same level of the tree.

"Squarification" algorithm:

Sort children by descending weight

While list of children non empty

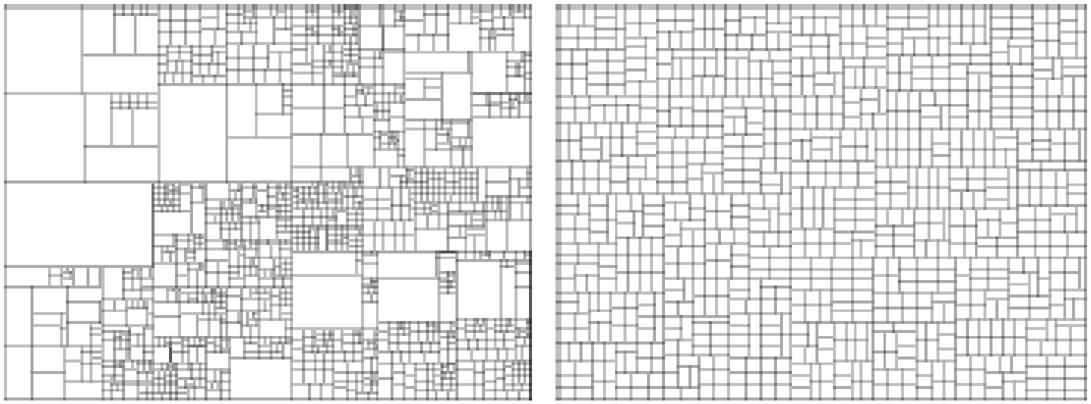
- Insert the first child, splitting the larger edge
- Repeat
  - "Squeeze" the next child into the same "row" (along the shorter edge)
  - If aspect ratio is worse than that of previous step, undo the step (steps 3,6,8 in example) and break out of inner loop



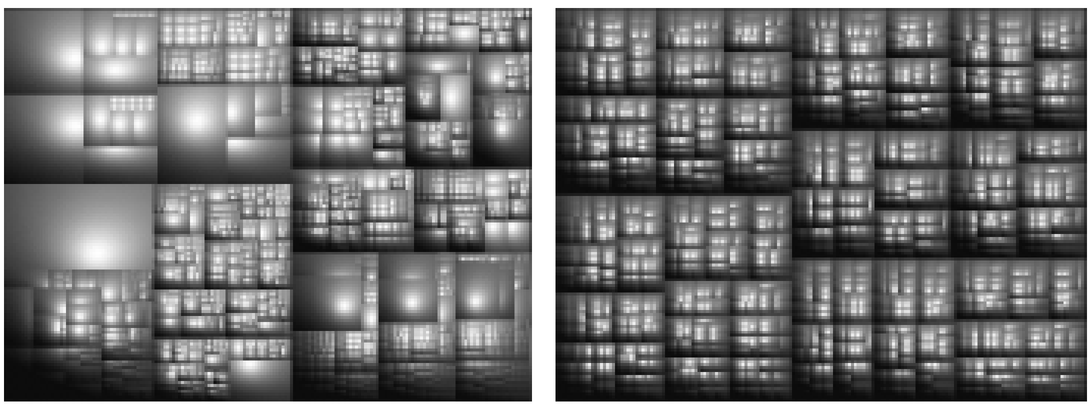
Worst accepted aspect ratio: 25:9

Squarified tree maps need visual cues for hierarchy levels.

Squarified tree maps of "File system" and "organization":

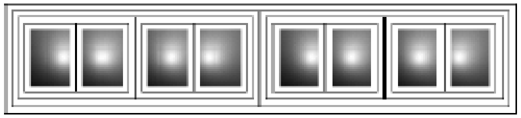


… and with cushions added:

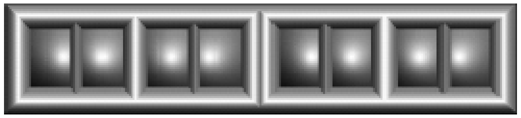


Alternative hierarchy enhancement techniques:

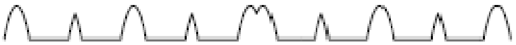
- Nesting



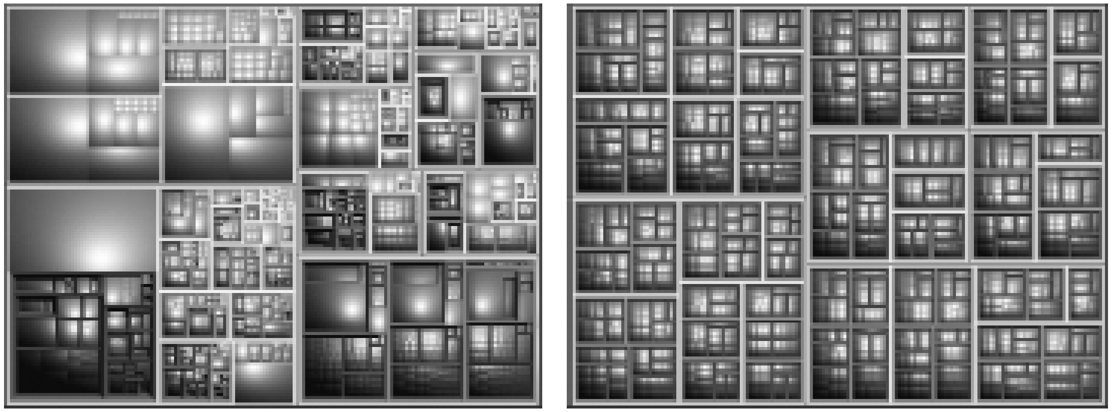
- Frames



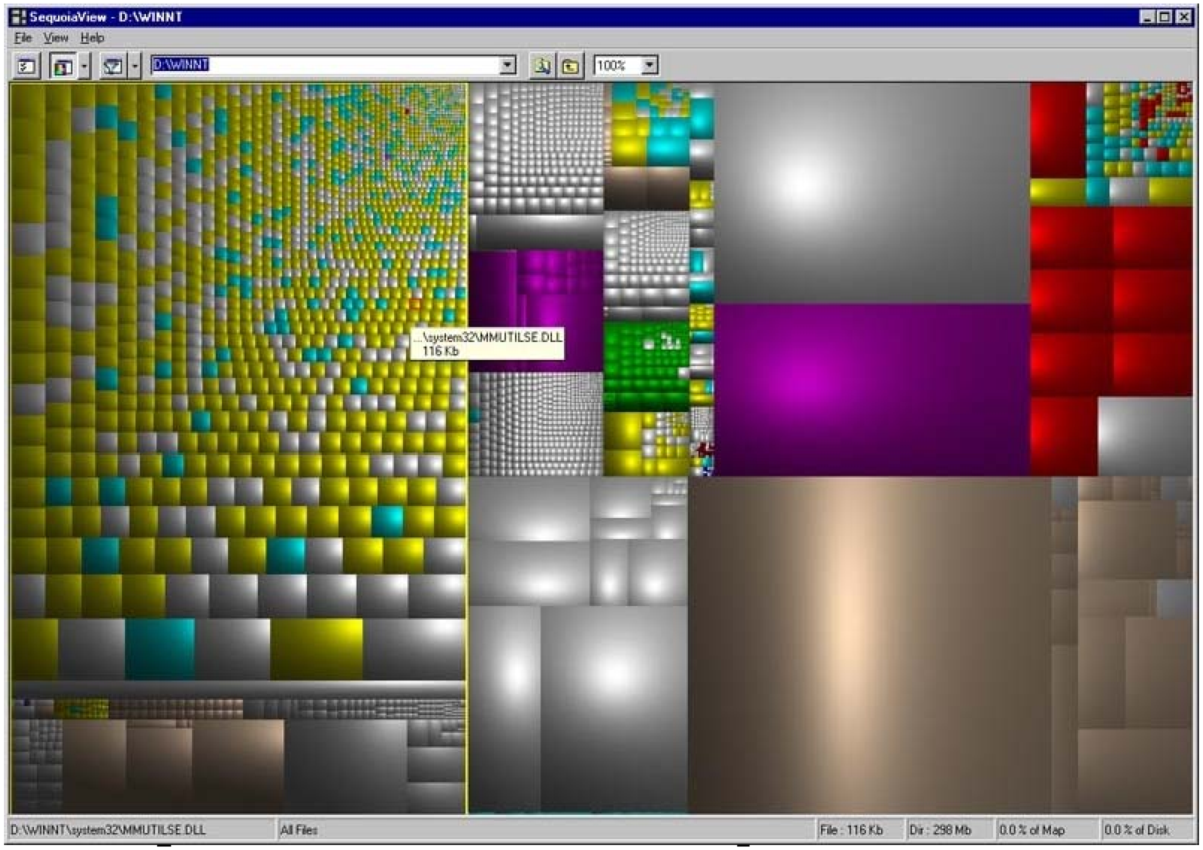
with profiles:



applied to examples:



Application: SequoiaView tool for file system visualization



(http://www.win.tue.nl/sequoiaview)

Clustering techniques

Motivation for clustering in visualization of graphs (networks):

Multiple levels-of-detail are obtained by identifying "highly connected" subsets and representing them by glyphs

Clustering techniques are often based on force models.

Assume an undirected graph $G=(V,E)$ with set of nodes V and set of edges E.

Notation: e_{ij} edge connecting nodes i and j

$\mathbf{p}_i$ position of node i

$$p_{ij} = \left\| \mathbf{p}_i - \mathbf{p}_j \right\|$$

The attractive force is usually Hooke's spring law

$$f(x) = A \cdot (x - x_0)$$

where x_0 is the zero energy length of the spring.

The repulsive force generally follows an inverse square law inspired by electrostatic fields:

$$g(x) = \frac{B}{x^2}$$

The total potential energy is then:

$$P = \frac{A}{2} \sum_{e_{ij} \in E} \left(p_{ij} - x_0\right)^2 - B \sum_{i \neq j} \frac{1}{p_{ij}}$$

Difficult to visualize: small world graphs (Watts and Strogatz).

Small world graphs are connected graphs having

- a small average path length (between pair of nodes), and
- a high clustering index,

both compared to a random graph with the same number of nodes and edges.

The clustering index of a node v is the ratio between

- number of existing edges in the 1-neighborhood N(v) of v
- number of possible edges, which is $k(k\text{-}1)/2$ if $k = |N(v)|$

The clustering index of the graph is the average of the clustering indices of its nodes.

Energy models suited for small-world problems:

- *r*-PolyLog energy models (Noack):

Potential energy:

$$P = \sum_{e_{ij} \in E} \left(p_{ij} - x_0\right)^r - \sum_{i \neq j} \ln\left(p_{ij}\right)$$

Attractive and repulsive forces are obtained by taking derivative.

For 1-PolyLog:

$$f(x) = 1 \qquad g(x) = \frac{1}{x}$$

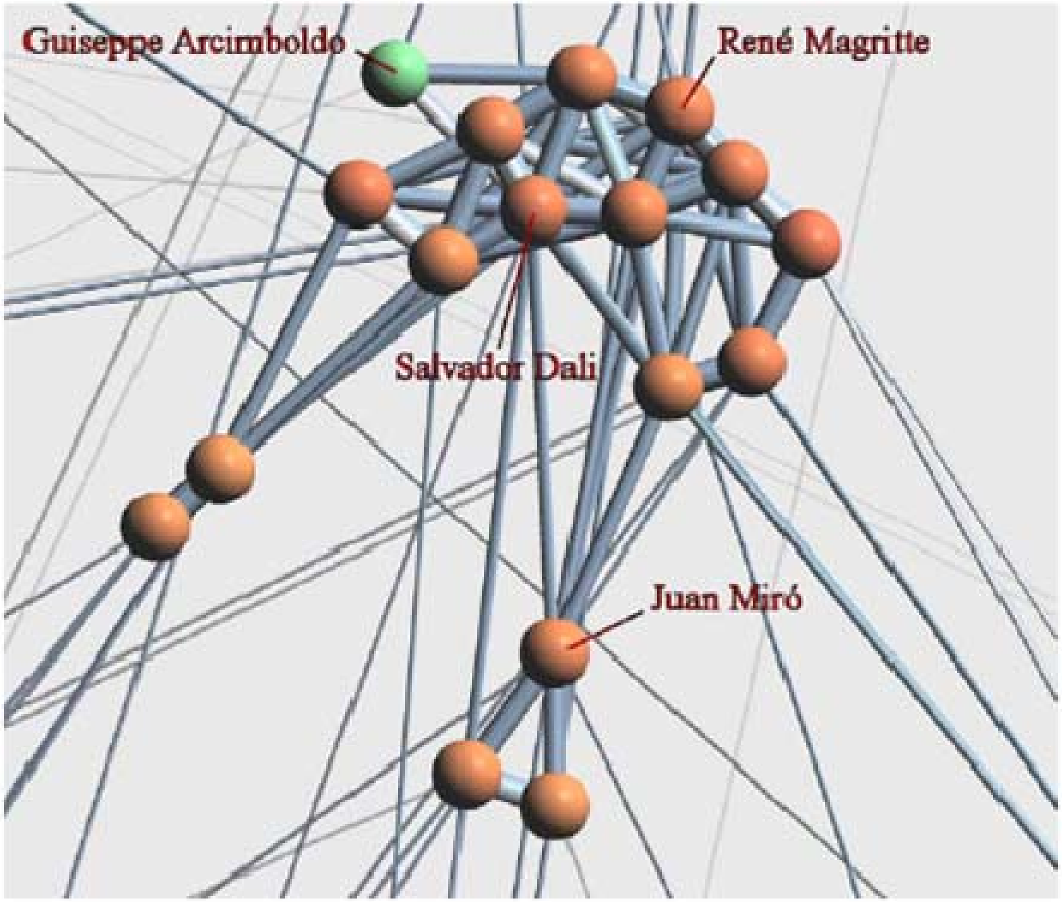
Minimum energy configuration of 1-PolyLog has the property:

- Distance between two clusters C_1 and C_2 is inversely proportional to their coupling:

$$\frac{\left|\left\{e_{ij} : i \in C_1, j \in C_2\right\}\right|}{\left|C_1\right|\left|C_2\right|}$$

Example small world graph:

- 500 painters/sculptors
- 2486 connections
- average path length 4
- clustering index 0.18
  (random graph: 0.0093)

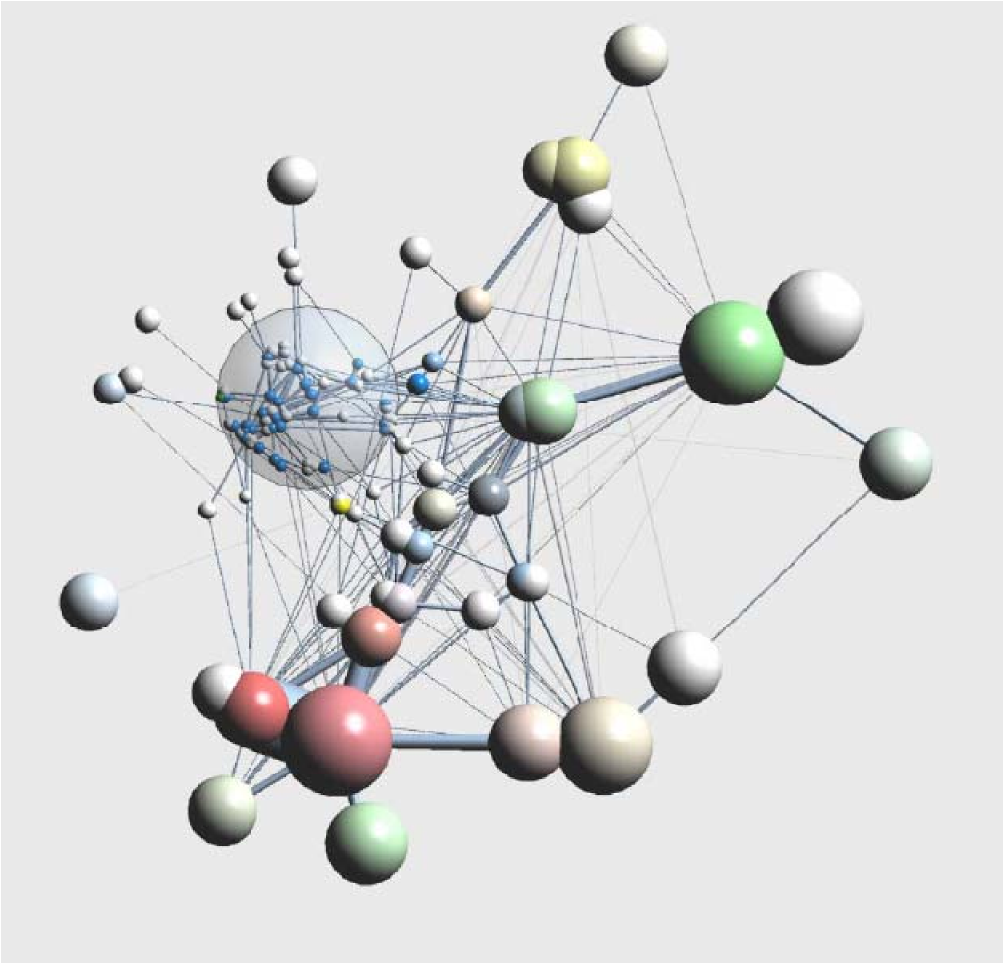


Video credit: F. van Ham, TU Eindhoven

Example: citations between Vis papers

Color coding:

- blue: volume vis
- red: flow vis
- green: terrain, surfaces
- yellow: info vis



Distortion techniques

Perspective wall (Robertson)

Example: documents arranged on a perspective wall

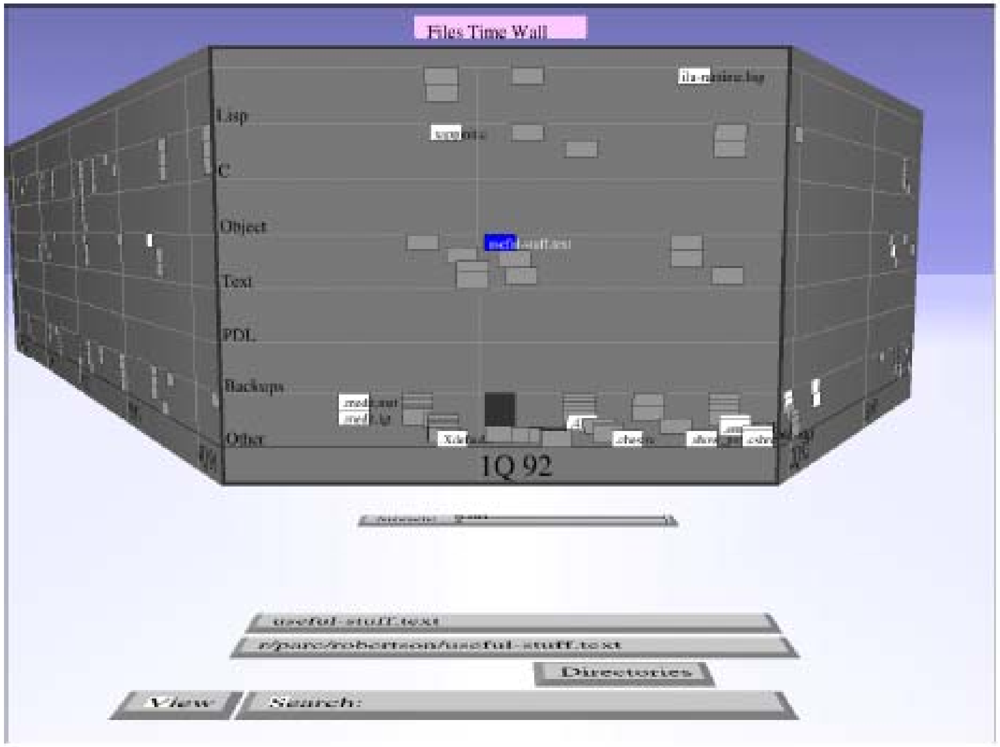


Image credit: S. Card

Table lens (Rao and Card)

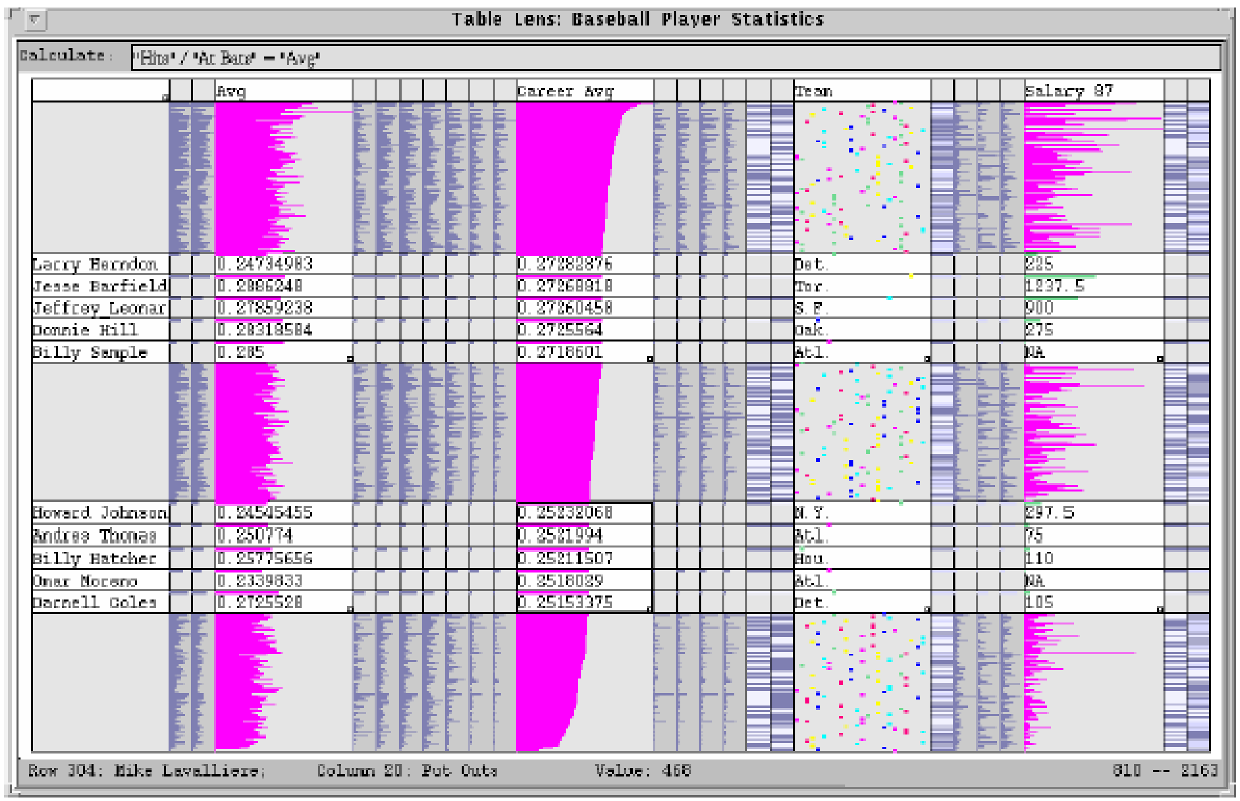
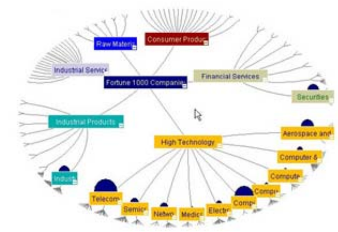
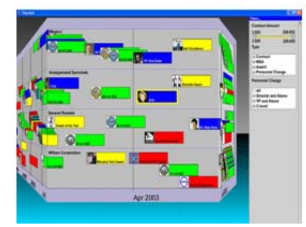


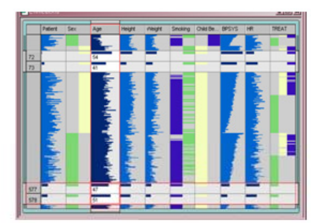
Image credit: R. Rao



Inxight StarTree allows users to navigate and explore hierarchical relationships and drill-down to information of interest.



Inxight TimeWall allows users to analyze trends and relationships over time, discerning patterns to develop predictive models.



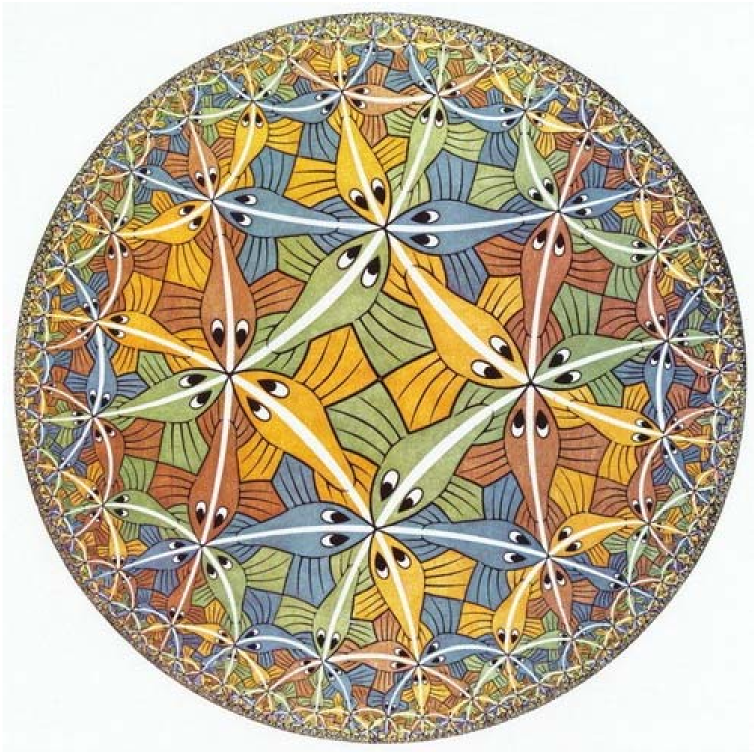
Inxight TableLens allows users to spot relationships and analyze trends in tabular data, easily viewing and manipulating entire data sets.

Inxight software

Hyperbolic trees are based on the Poincaré Disk model (projection) of the hyperbolic space H_2.

In the Poincaré Disk, the role of straight lines is taken by

- circles which intersect the bounding circle $x^2 + y^2 = 1$ orthogonally, and
- diameters of the bounding circle.



M.C. Escher's "Circle Limit III", 1958, illustrates lines (white circles).

Property of Poincaré Disk:

- Triangles have sum of angles < 180°
- It has the metric

$$ds = \frac{\sqrt{dx^2 + dy^2}}{1 - x^2 - y^2}$$

- The bounding circle is at infinity
- Circle perimeter grows exponentially with its radius.

As a consequence, trees can be drawn undistorted in hyperbolic space:

- all edges having about the same length and
- all nodes having the same angle available for their children

Rigid transformations of Poincaré Disk: Möbius transformations of complex numbers:

$$z' = T_{c\theta}(z) = \frac{\theta z + c}{\bar{c}\theta z + 1}, \quad |\theta| = 1, |c| < 1$$

These are

- for c = 0: rotations around 0
- for θ = 1: translations (mapping 0 to c and -c to 0)
- combinations:

$$T_{c_2\theta_2}\left(T_{c_1\theta_1}(z)\right) = T_{c\theta}(z)$$

with

$$c = \frac{\theta_2 c_1 + c_2}{\theta_2 c_1 \bar{c}_2 + 1}, \quad \theta = \frac{\theta_1\theta_2 + \theta_1 \bar{c}_1 c_2}{\theta_2 c_1 \bar{c}_2 + 1}$$

Hyperbolic tree technique (Lamping et al.).

Change of focus, i.e. moving a different node towards the center, is achieved by performing a translation in hyperbolic space.

Example: Visualization of a large organizational hierarchy in hyperbolic space with different foci

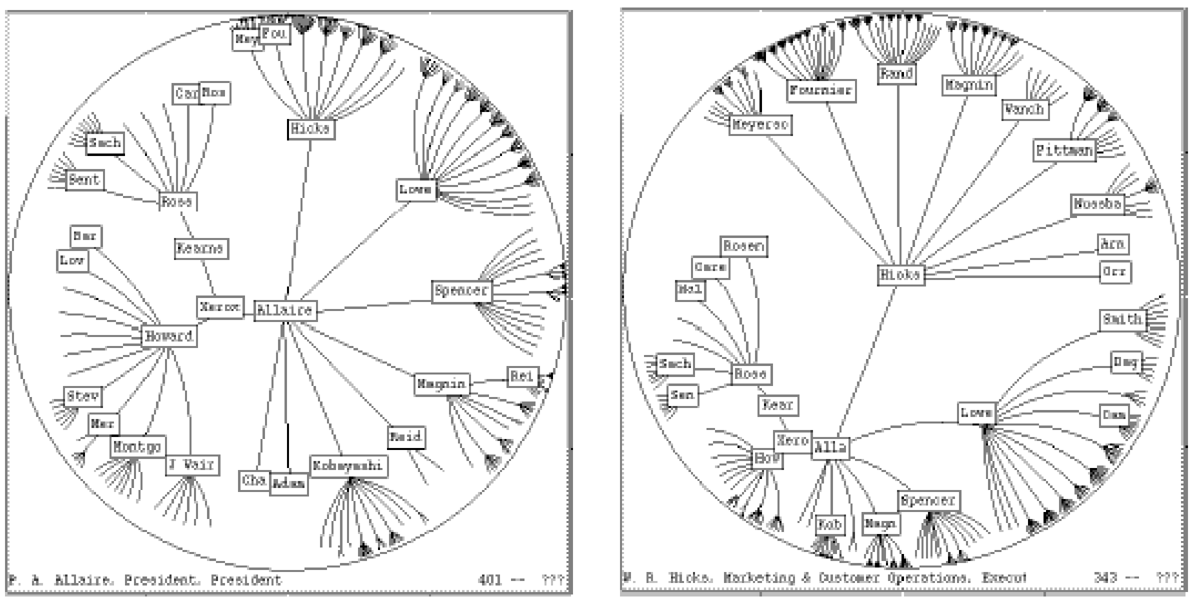


Image credit: R. Rao