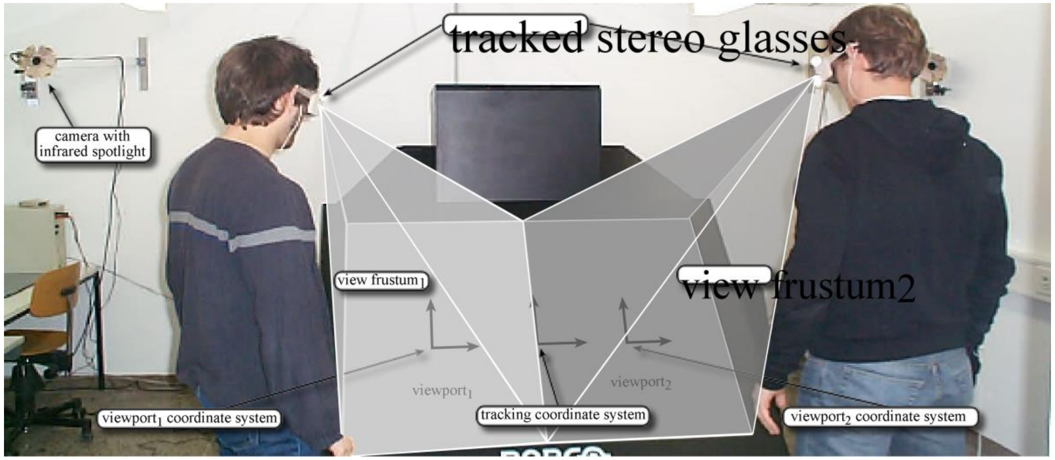
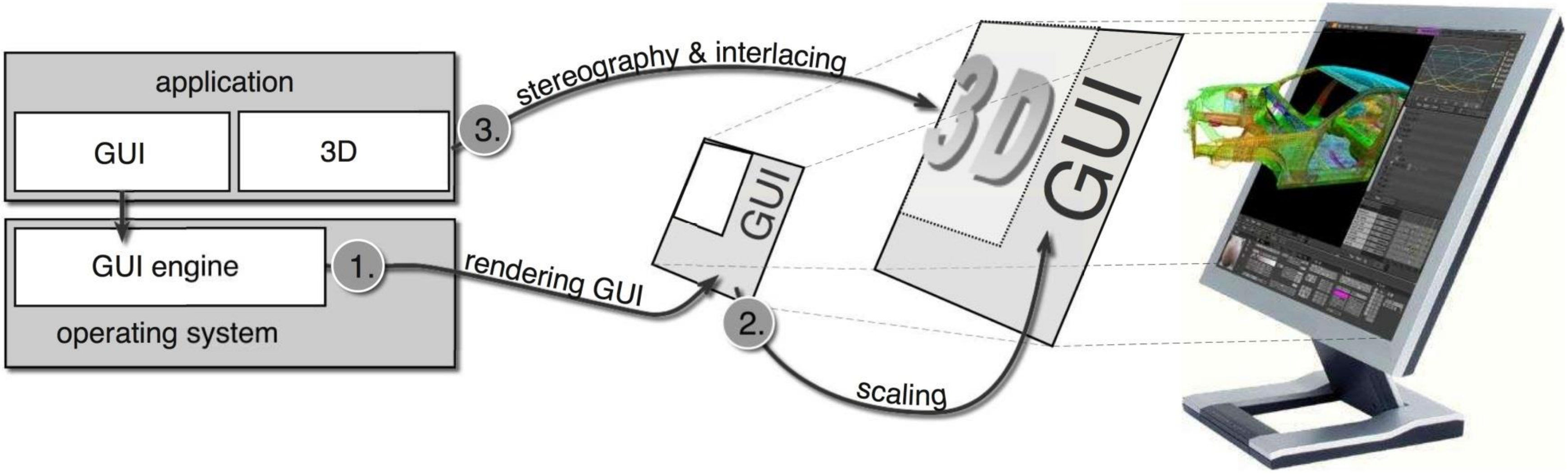

tracked stereo glasses
camera with infrared spotlight
view frustum$_1$
view frustum2
viewport$_1$
viewport$_2$
viewport$_1$ coordinate system
tracking coordinate system
viewport$_2$ coordinate system

viscg.uni-muenster.de

www.cs.uni-paderborn.de

9. Interaction techniques – basic concepts



viscg.uni-muenster.de

Interaction techniques

- **Navigation** – changing camera position, scaling of view
- **Selection** – identification of object, a set of objects, region of interest; applying further operations on them
- **Filtering** – reduction of data size mapped onto screen
- **Reconfiguration** – changing the mapping of data to graphic entities or attributes
- **Change of encoding** – changing the graphics attributes (point size, …)
- **Aggregation** – merging different views, objects
- **Abstracting/specifying** – change of LOD
- **Hybrid techniques** – combinations of above

Navigation operators

- Navigation is used for searching a subset of the input data which the user wants to explore; searching for appropriate view orientation and LOD
- In 3D the navigation is determined by the camera position, view direction, size, and shape of view frustum and level of LOD
- Navigation can be automatic or user-driven

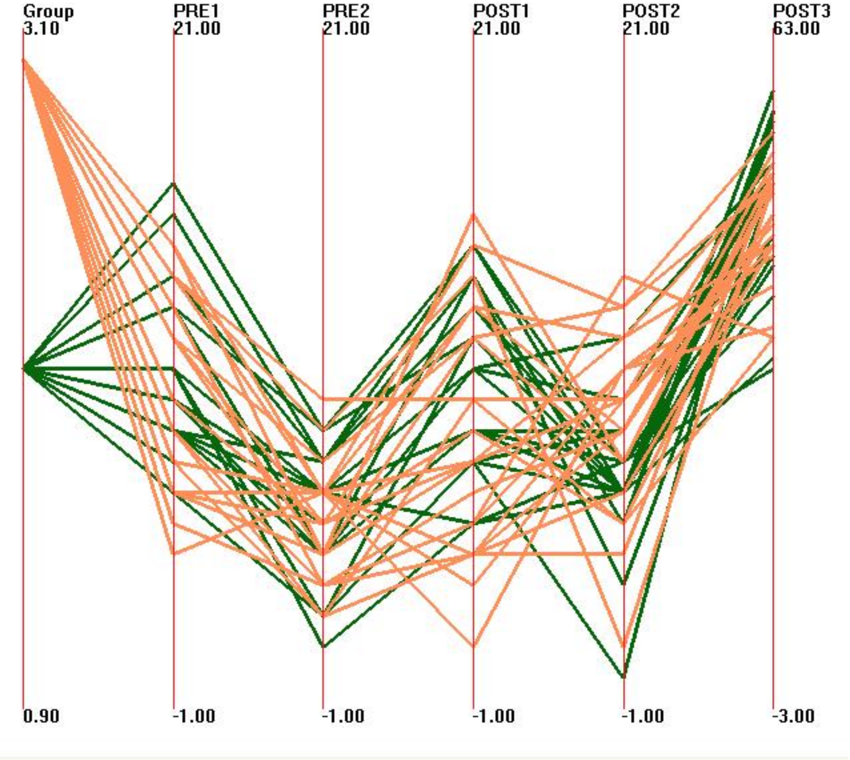
Selection operators

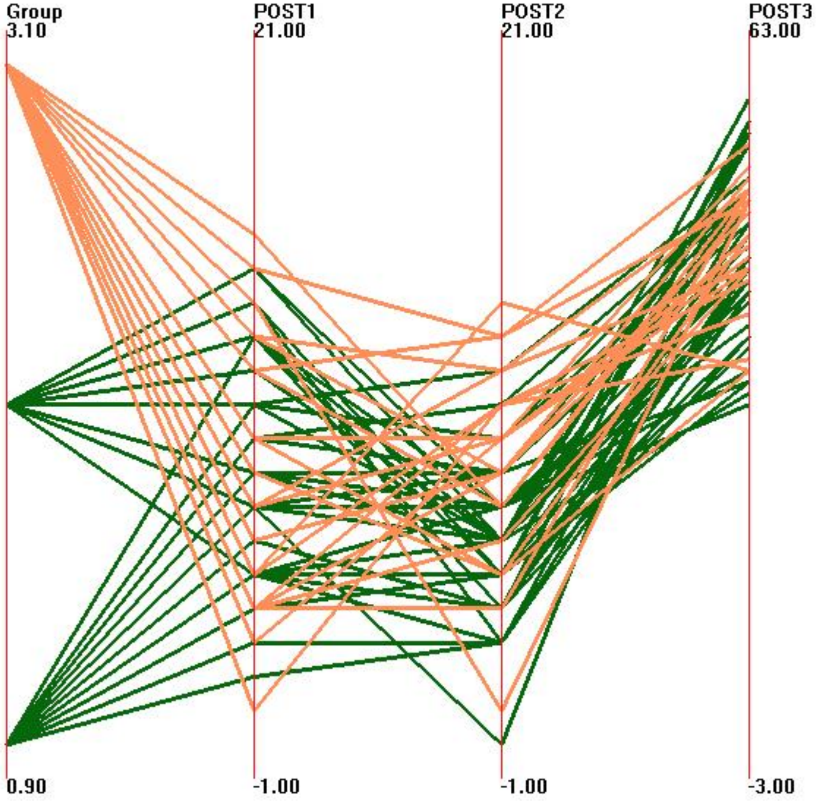
- Isolating a subset of components to be visualized, these are further processed using other operations – highlighting, deleting, masking, …
- We need to know the expected result (e.g., new selection should replace the old one or should add items to the old one?)
- Granularity of selection – size of area influenced by the selection
- Direct selection (by the user) or indirect (fulfilling a set of criteria)

Filtration operators

- Reducing data size to be visualized – by setting limitations
- Determining the region of interest – several methods:
  - Manipulation using sliders, immediate update of visualization
  - Selection of items to be kept/hidden – e.g., hiding columns in MS Excel

Filtration operators





Filtration operators

- Difference between filtration and selection followed by deleting or masking:
  - Filtration is **indirect** – often before data visualization, in separate dialog window (not in the visualization itself)
  - Selection is **direct** – objects are marked directly in the visualization window (e.g., by mouse clicking)

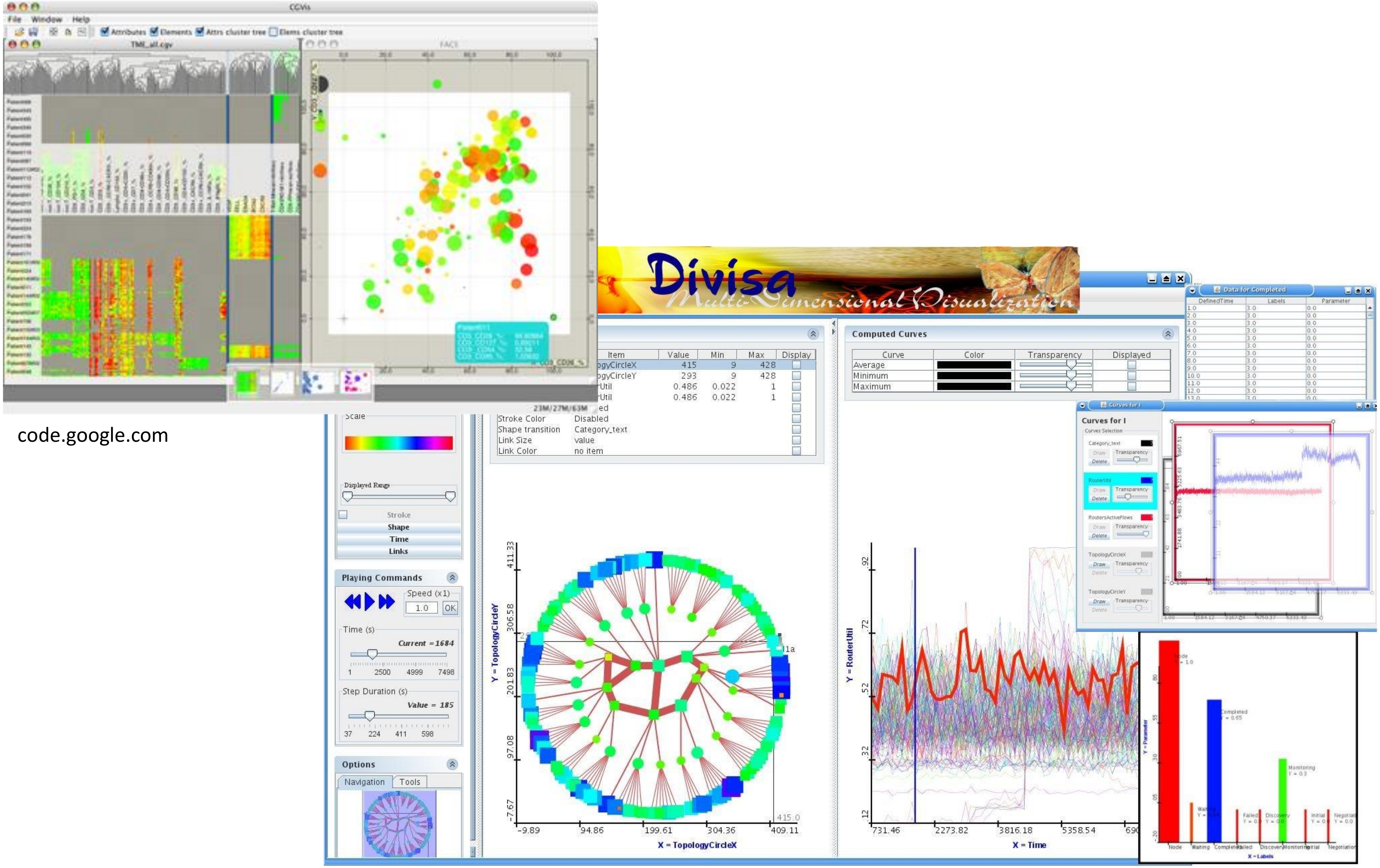
Reconfiguration operators

- Revealing data properties, handling the data complexity and scale
- Providing different views onto data
- Popular methods
  - PCA (principal component analysis)
  - MDS (multidimensional scaling)
  - Trying to transfer the relationships between data from the high-dimensional space to the reduced projected space

Encoding operators

- Data properties, which are invisible in a particular visualization, can be displayed using other visualization type
- Currently visualizations commonly support several different views onto data
- Mapping, different views onto data, modification of the color map, size of graphic entities, their shape, transparency, texturing, line style, dynamic attributes – loss of intensity, flickering, …
- Using different variations, we can overcome several limitations of a given visualization technique (e.g., overlaps of points in scatterplots)

Encoding operators



Item	Value	Min	Max	Display
ogyCircleX	415	9	428	□
ogyCircleY	293	9	428	□
rUtil	0.486	0.022	1	□
rUtil	0.486	0.022	1	□
ed				□
Stroke Color	Disabled			□
Shape transition	Category_text			□
Link Size	value			□
Link Color	no item			□

Computed Curves

Curve	Color	Transparency	Displayed
Average	■		□
Minimum	■		□
Maximum	■		□

Speed (x1) 1.0 OK

Time (s) Current = 1684

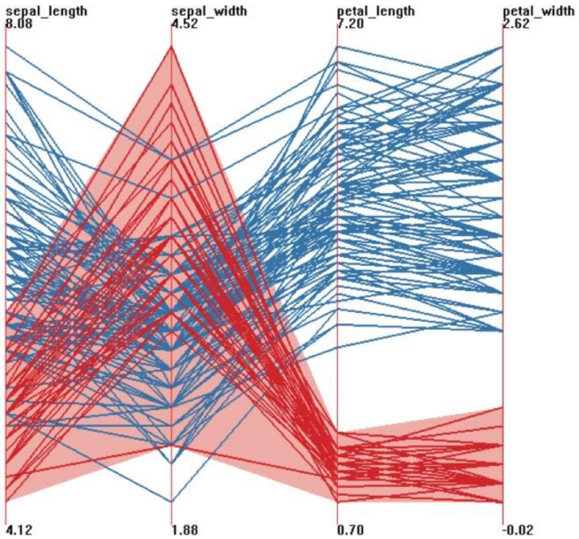
Step Duration (s) Value = 185

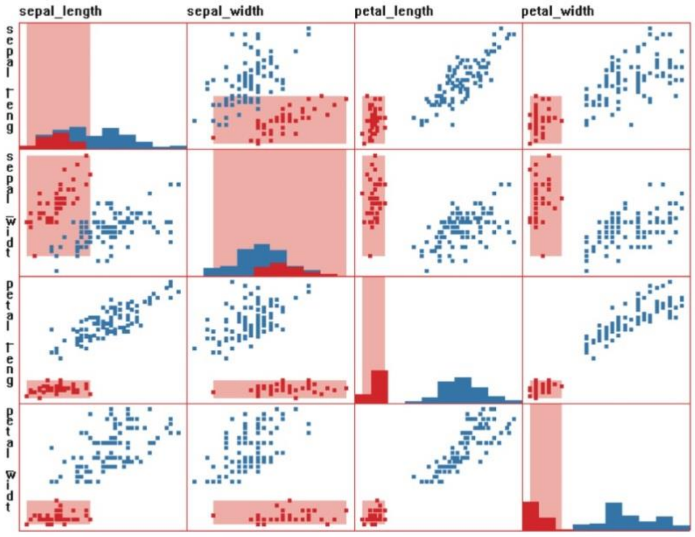
code.google.com

www.nist.gov

Aggregation operators

- Connecting selected data in one view with the corresponding data in other views
- Most popular are *linked selections* – each view can reveal interesting data properties





Aggregation operators

- Interactive change of selected data – *brushing* – change of selection in one view highlights the corresponding data in other views
- Possibility to specify complex limitations for a given selection
- Possibility to *unlink* some visualizations (we can specify if the information should be transferred to other views)
- Local interactions (zoom) vs. interactions shared between all views (dimensional stacking)

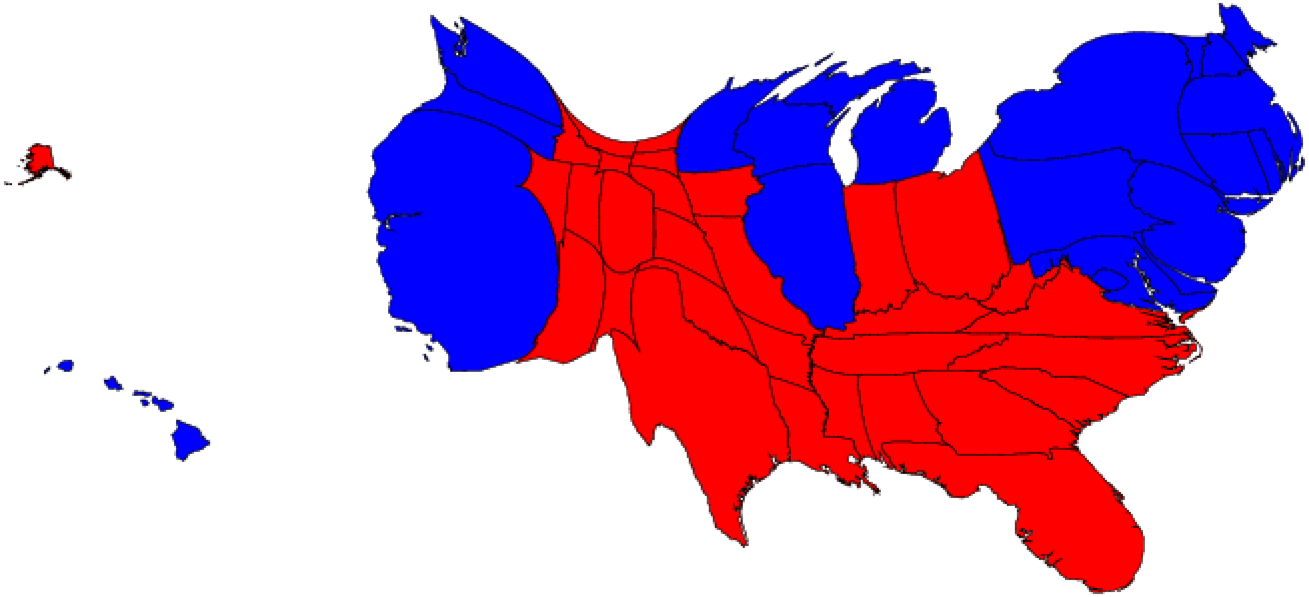
Abstraction/Specification operators

- Displaying large amount of data – better to focus only on a given subset, where we show details (concretization) and in the other parts we reduce the LOD
- **Distortion operators** (functions) – transformation which can be applied to an arbitrary visualization
- Distortion can be part of the visualization or is displayed in the separate window

Distortion operators

- Linear, non-linear; with C_0, C_1, or C_2 continuity (also non-continuous)
- Can be applied to structures instead of continuous spaces – specific for a given operand type (see later)
- Operators have different „footprints“ – shape (rectangular, circular footprints) or range of space influenced by the transformation (defined by the distance function)

Distortion



http://www.humantransit.org/marketing/

Interaction operands and spaces

- **Interaction operand** is a part of the space onto which we apply an interaction operator
- In order to be able to determine the result of the interaction operation, we need to know the space where the operation will be applied
- We will mention several different classes of interaction spaces, including examples of existing interaction techniques for each class

Interaction operands and spaces

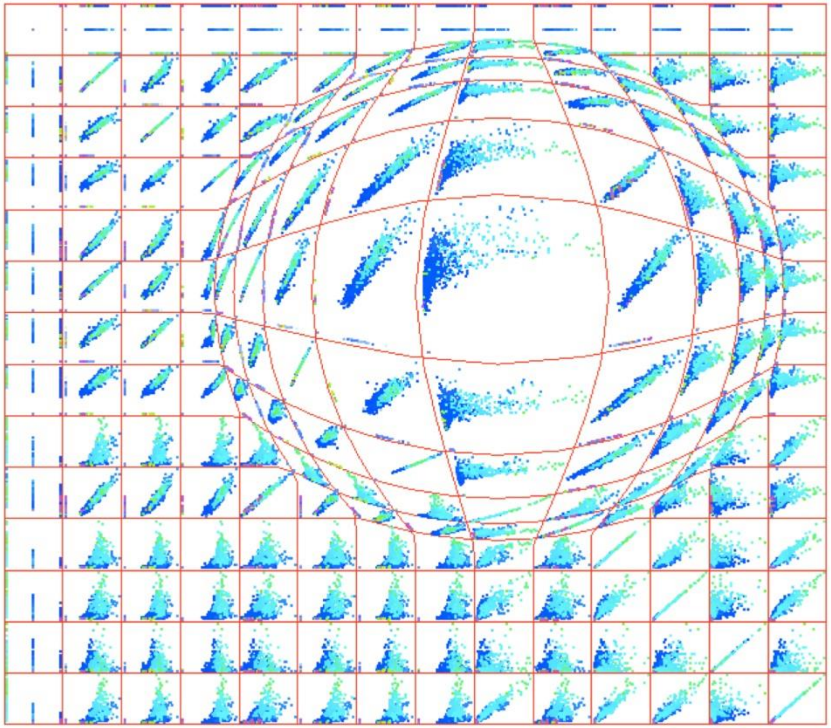
- Screen space (Pixels)
- Space of data values (Multivariate data values)
- Space of data structures (Components of data organization)
- Space of attributes (Components of graphical entities)
- Space of objects (3D surfaces)

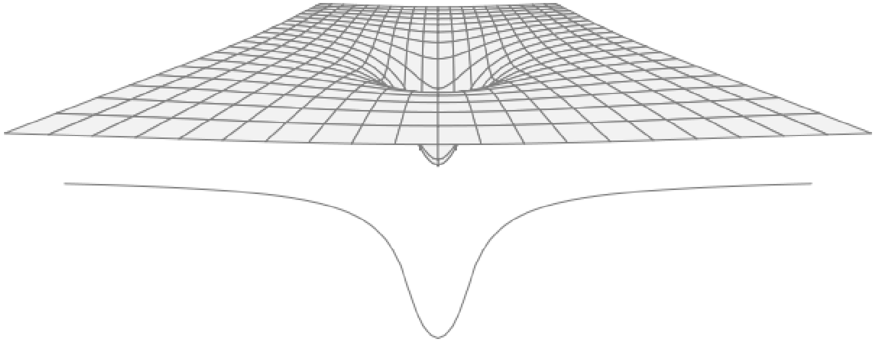
Screen space (Pixels)

- Selection of pixels = each pixel is classified as selected or non-selected
- We can select individual pixels, rectangular or circular pixel area, areas of arbitrary (user defined) shape
- Distortion = transformation on pixels: **(x’, y’) = f(x, y)**

Screen space (Pixels)

- Magnification **m(x, y)** in a given point is a derivation of this transformation
- Fisheye, rubber sheet, …





www.ilovephilosophy.com

Fisheye view

- We have to specify:
  - central point of transformation - (c_x, c_y)
  - Radius of the magnifying lens – r_l
  - Size of distortion (deflection) – d

where

$$r_{new} = s\log(1 + d(r_{old}))$$

$$s = \frac{r_l}{\log(1 + d * r_l)}$$

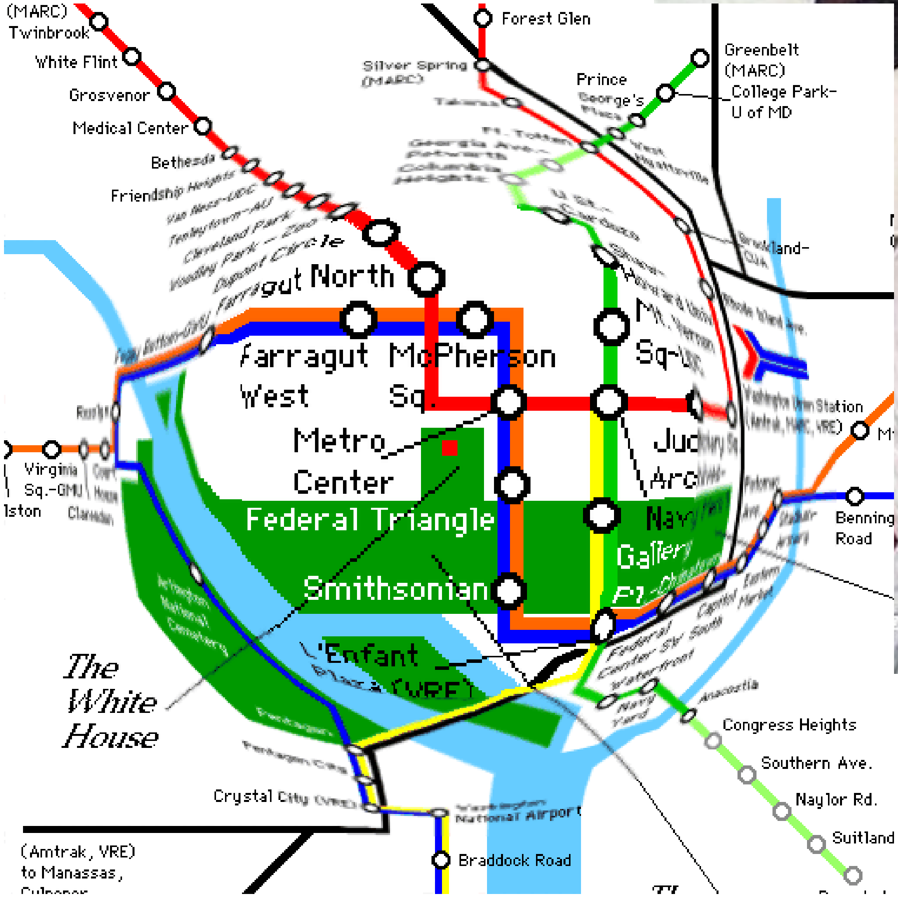
Fisheye view – pseudocode

```
1. Clear the output image.
2. For each pixel of the input image:
    a) Calculate corresponding polar coordinates.
    b) If the radius is smaller than the radius of the magnifying lens:
        i.   Calculate new radius $r_{new}$.
        ii.  Get color in this pixel from the original image.
        iii. Set this color to the pixel in the output image.
    c) Else set the resulting pixel in the output image to the same value as it has in the original image.
```

Screen space

- Distortion causes pixel overlaps or holes
  - Overlaps are solved by averaging
  - Holes have to be fixed using interpolation
- Type of interpolation depends on the type of the magnifying lens
  - E.g., in text visualization the central part of the lens cannot be distorted (for better readability)

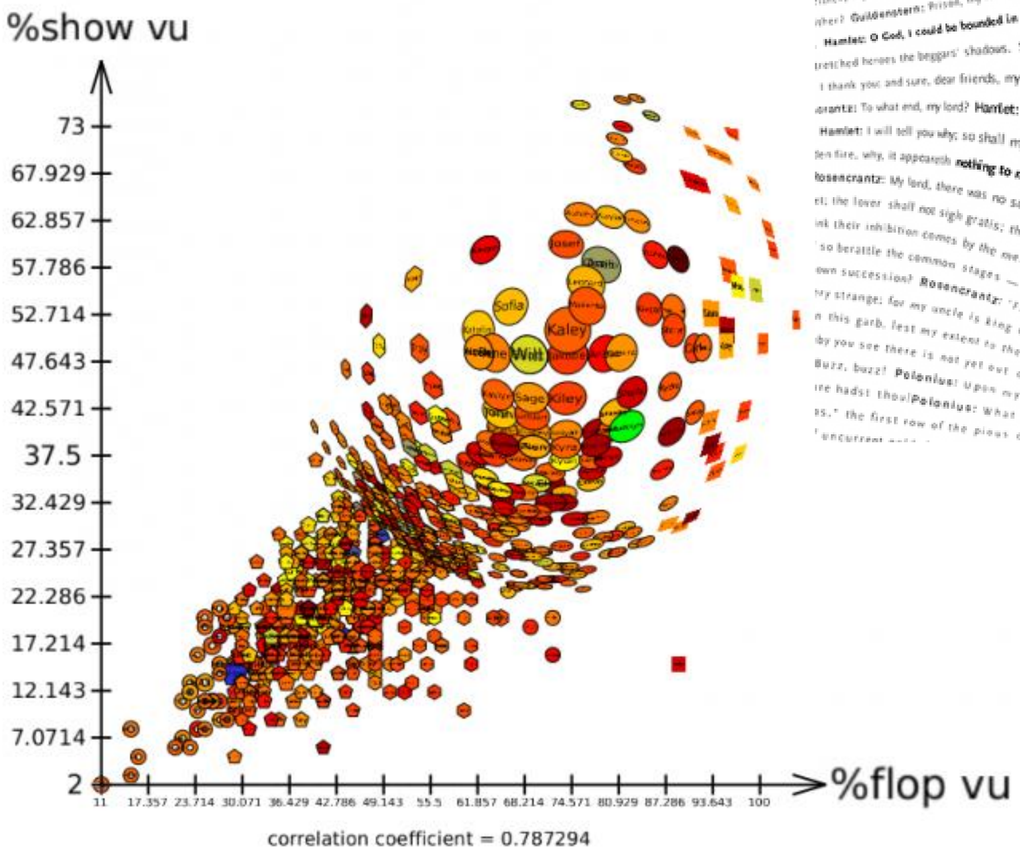
Fisheye view



www.datavis.ca

kizziecat.blogspot.com

Fisheye view



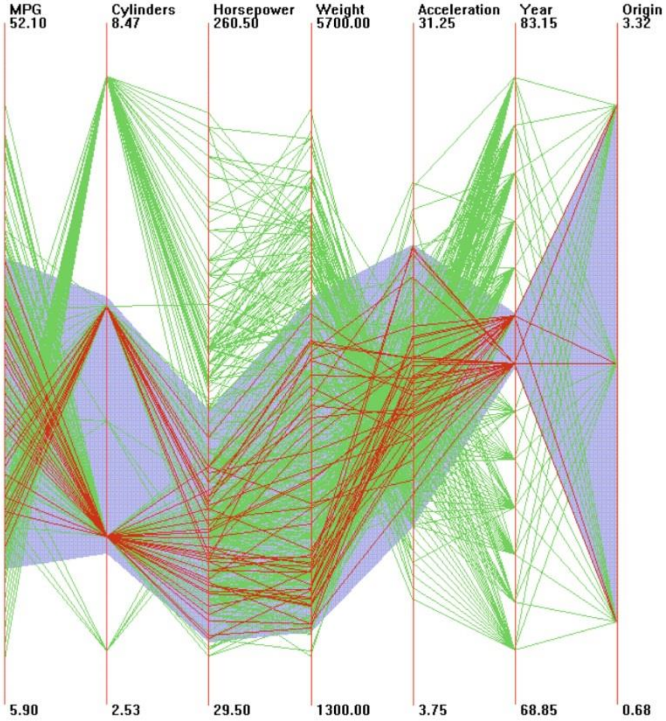
tulip.labri.fr

412,000 results
154,000 results
5,000 results
10,000 results

flowingdata.com

Space of data values (Multivariate data values)

- Specification of viewpoint
- Change of displayed values – similar to database queries
- E.g., data-driven brushing



Space of data values

- Intuitive space for applying filtration – data and/or dimension reduction
- Space distortion using transformation:

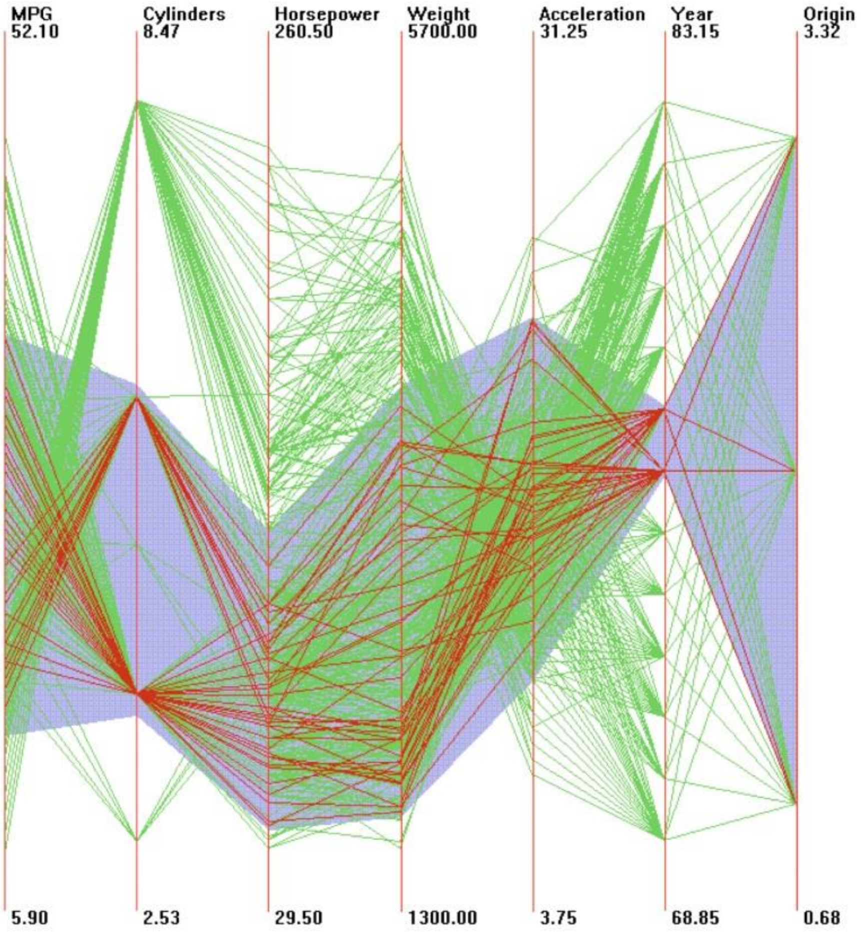
$$(d'_0, d'_1, \ldots, d'_n) = j(d_0, d_1, \ldots, d_n)$$

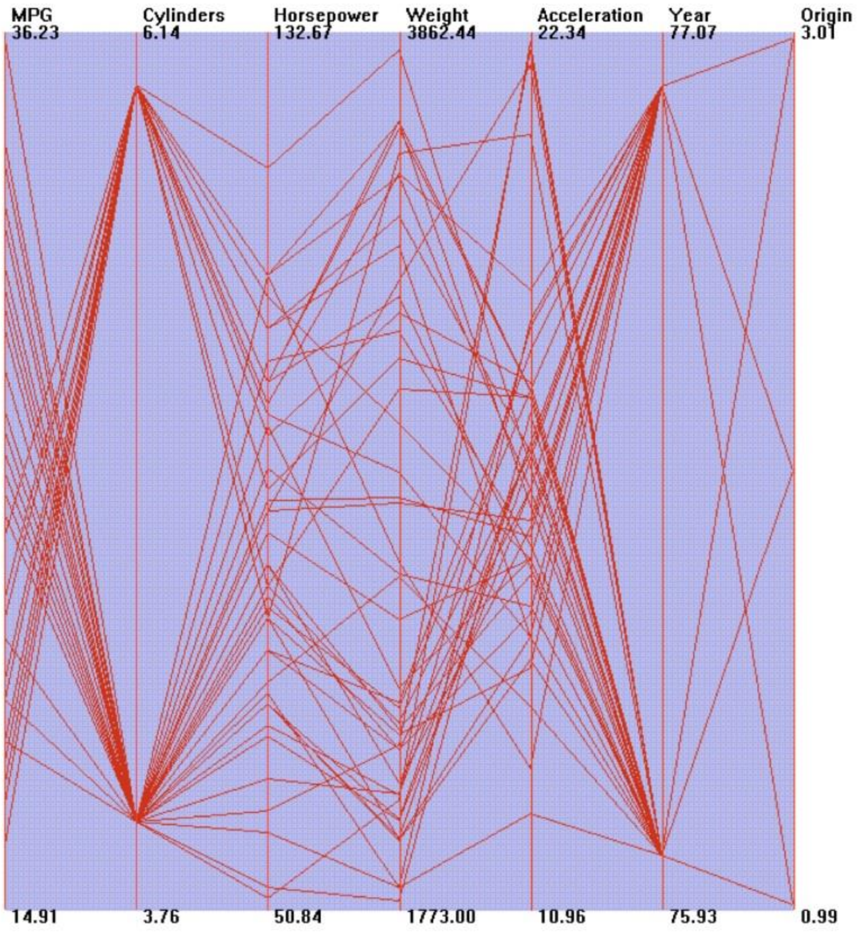
- In fact, each dimension can have its own transformation function:

$$j_i : d'_i = j_i(d_i)$$

- The most common case: j_i depends on an arbitrary number of dimensions

Space of data values



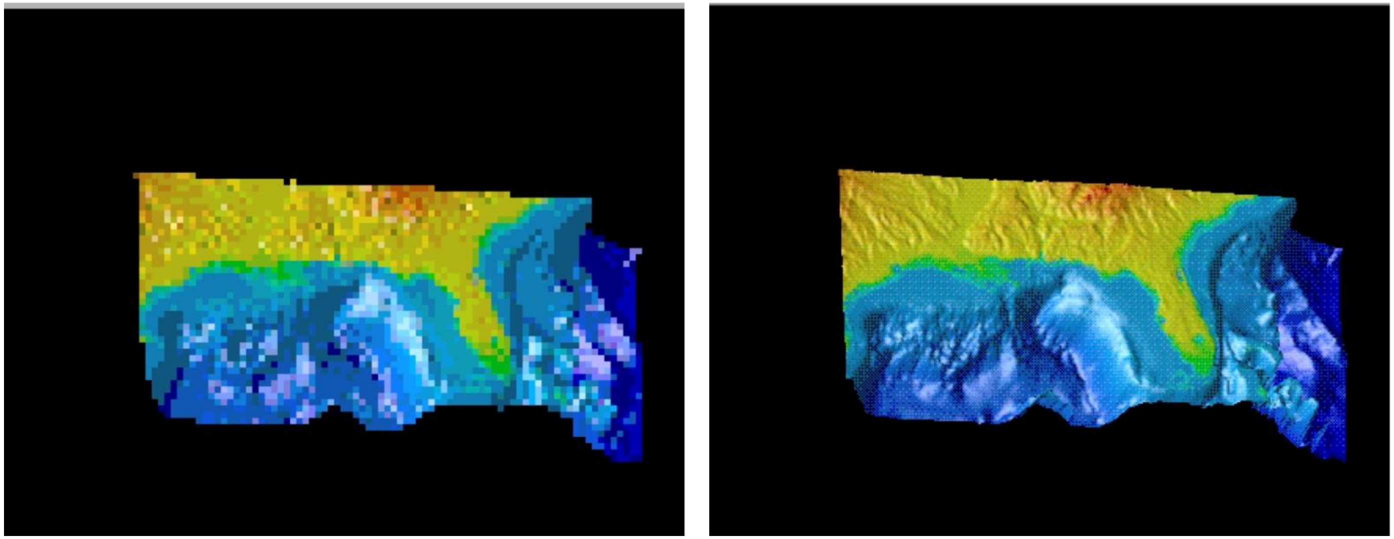


Space of data values

- The user has to be informed about the transformation applied to data
- Often we have to apply transformations of the range of data values, so they fit to the range of graphical entities
- Incorrect mapping = values are mapped to the space out of screen, etc.

Space of data structures (Components of data organization)

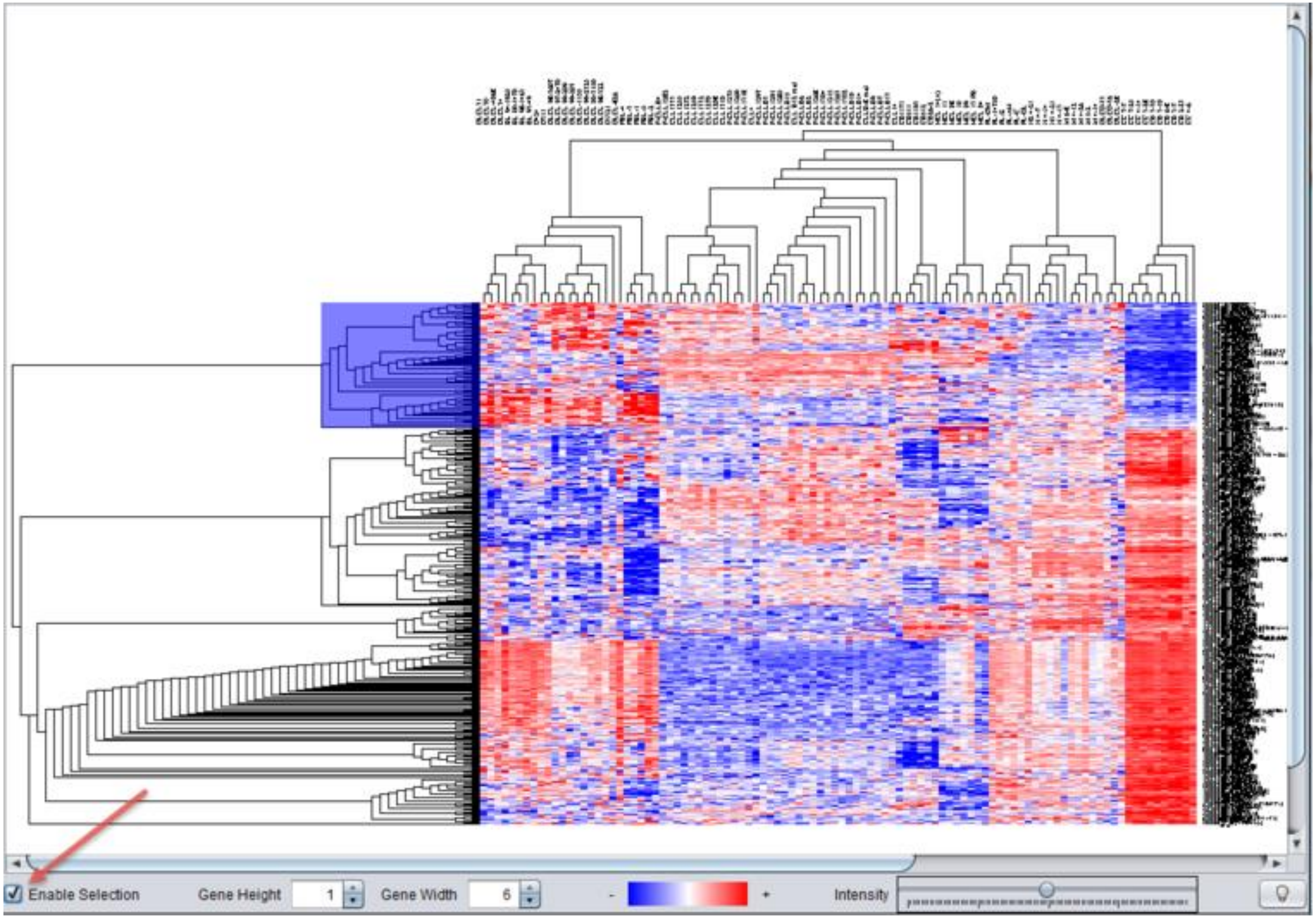
- Data are structured into lists, tables, grids, hierarchies, graphs
- Each of these structures can have its own special interaction mechanism for selection of a subset of data
- Zooming in screen space vs. in data structure



Space of data structures

- **Filtration** is often used for reducing the amount of displayed information:
  - Time-dependent visualization – we define the temporal range
  - Graph visualization – filtration of nodes and edges (define the number of “hops”)
  - Hierarchical visualization – filtration based on the level of hierarchy

Hierarchical filtration – example



wiki.c2b2.columbia.edu

Space of data structures

- When designing interactions in the space of data structures we have to define the level of automatization and how we define the interactions (directly in the visualization window or in a separate dialog window)
- Automatic techniques:
  - Thorough and time-consuming techniques vs. fast and imprecise techniques

Space of data structures

- We need to consider the ordering of dimensions for visualization of multivariate data
- Fully manual approaches or automatic techniques for reordering of dimensions
- **Manual approach** – manipulating with items in textual lists (shifting items up and down, drag-and-drop), manipulation with axes in parallel coordinates and scatterplot matrices

Space of data structures

- **Automatic approach** – we need to know at least two basic decisions influencing the design:

1. How to measure the quality of ordering of dimensions
2. Which strategy to follow when searching for these high-quality ordering

- We can use different metrics

Measuring the quality of ordering

- **Correlation coefficient** between two dimensions is defined as:

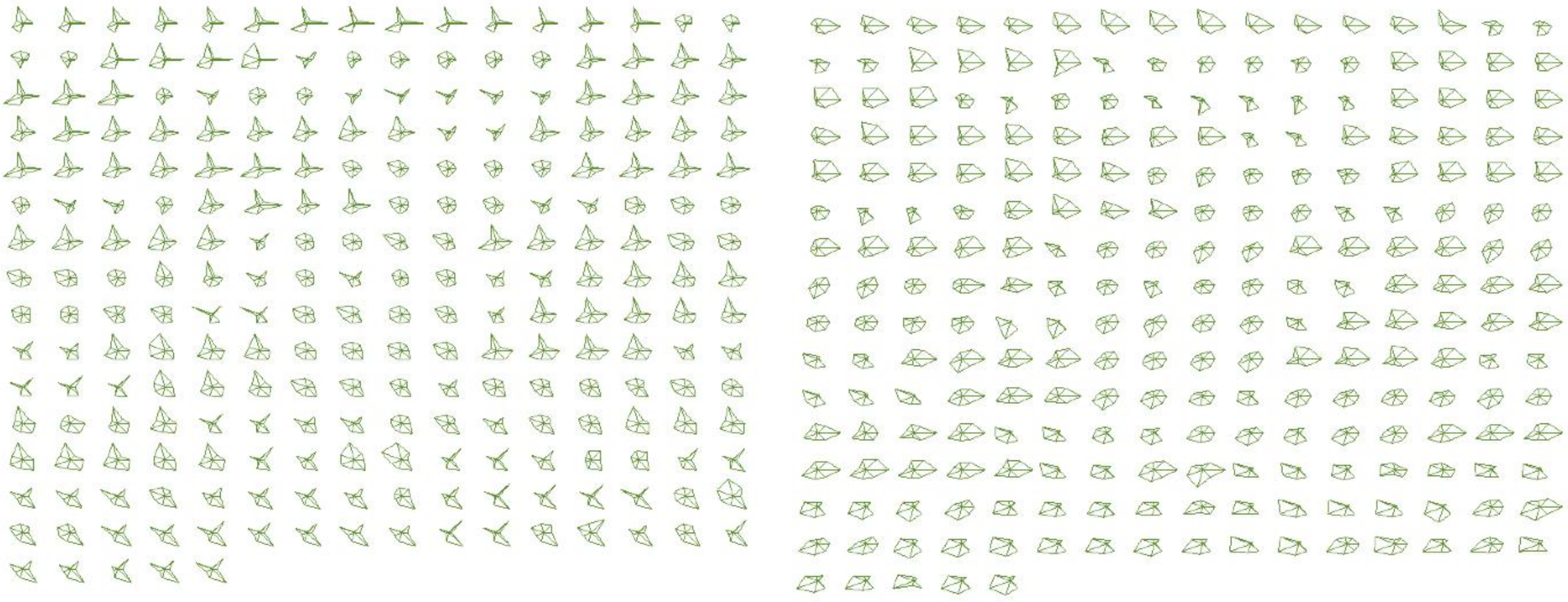
$$\rho_{X,Y} = \frac{\sum (x_i y_i - n\mu_X \mu_Y)}{(n-1)\sigma_X \sigma_Y}$$

where *n* is the number of data points, X and Y are two dimensions, x_i and y_i are the values for i-th data point, μ_X is the mean value in X and σ_X is the standard deviation for X

Measuring the quality of ordering

- Another approach to quality measurement – **simplicity of interpretation**
  - Different dimension stackings lead to visualizations containing bigger or smaller visual clusters
  - It is easier to interpret simple glyph shapes than the complex ones
  - If we are able to measure the average or cumulative complexity of the glyph shape (e.g., by computing the number of hollows or vertices), we can compare the visual complexity of different orderings

Measuring the quality of ordering



Original ordering vs. results after dimension reordering – the goal is to reduce the concave areas and increase the number of symmetrical shapes

Searching for the best searching strategy

- Searching for high-quality ordering of dimensions
- Evaluating of all possible orderings = N! options
- Utilizing different optimization techniques
- Similar to travelling salesman problem

Searching for the best searching strategy

- Simple algorithm:

1. Select two arbitrary dimensions
2. Swapping their position and calculating the quality of new ordering
3. When the quality is lower than before, swap back
4. Repeat steps 1-3 *n* times (*n* is user-defined) or until a given number of tests does not lead to quality improvement

Searching for the best searching strategy

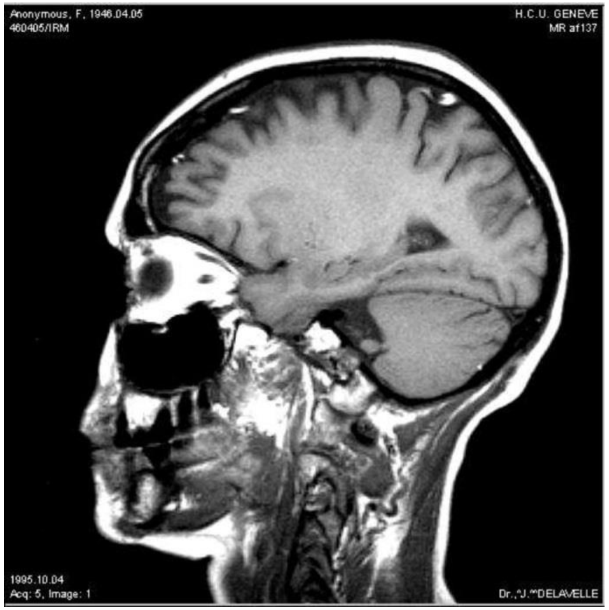
- Heuristic approaches are not optimal but often lead to an acceptable solution
- Possibility to combine with manual approach – the user can set some ordering based on their experience, the rest is calculated automatically

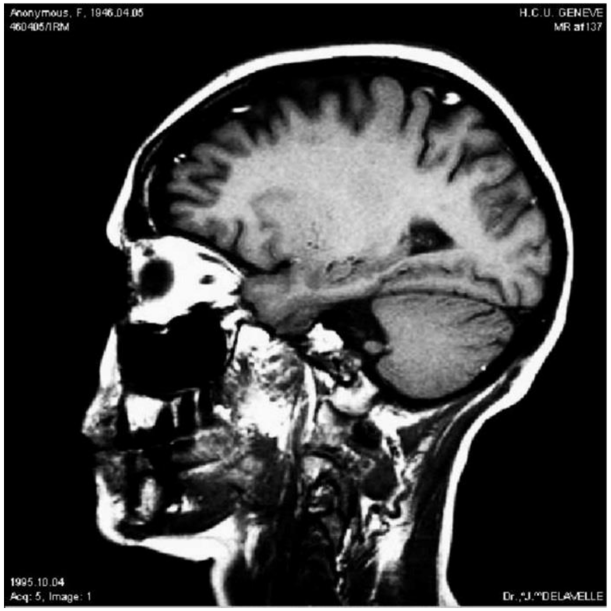
Space of attributes (Components of graphical entities)

- Navigation similar as in the space of data values – panning, zooming (by scaling of attributes or increasing the range of values of interest)
- Filtration based on attributes
- Remapping in the attribute space – by selecting different ranges of attributes or selection of different attributes for given input dataset

Space of attributes

- The most used interactions – color and transparency attributes
- Change of contrast and brightness in order to highlight specific properties:



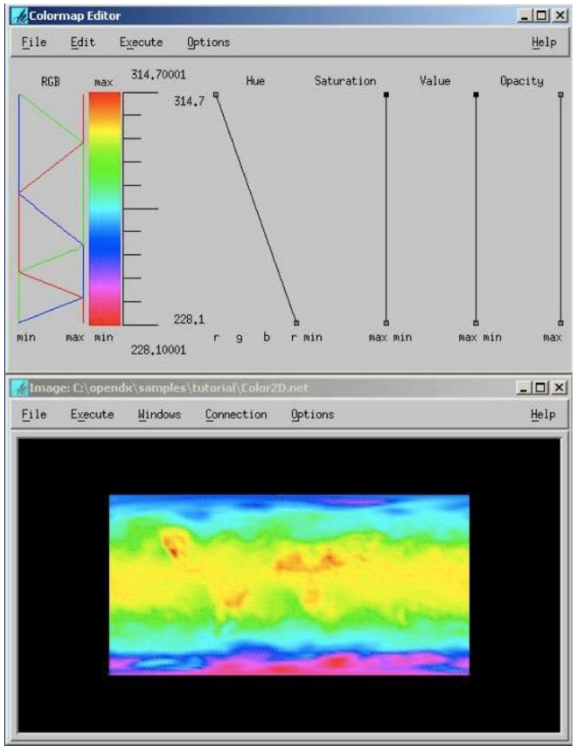


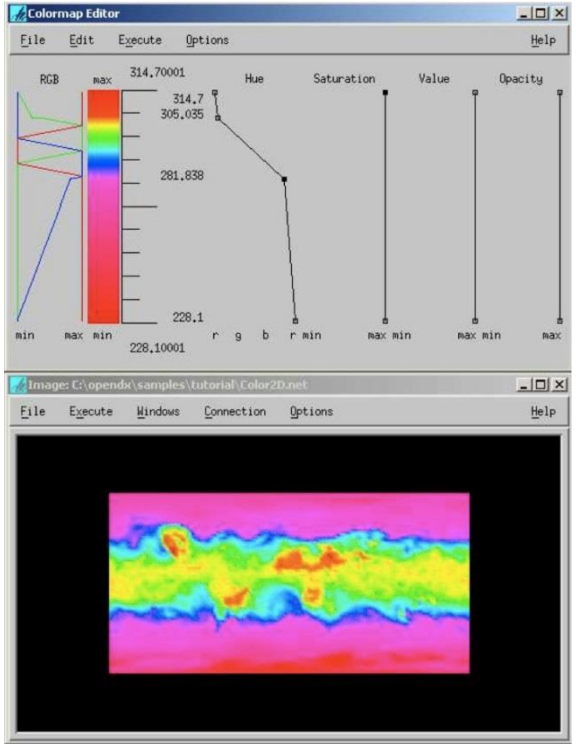
Space of attributes

- Interactive tools for specifying and modification of the transfer function in volume rendering (controlling color and transparency) (https://www.youtube.com/watch?v=UHOUFJmj_fM (23:01))
- The simplest form – data values on the horizontal axis + transparency or color component

Space of attributes – example

- A is the attribute of the graphic entity. We can apply distortion k:a’ = k(a).





Space of attributes

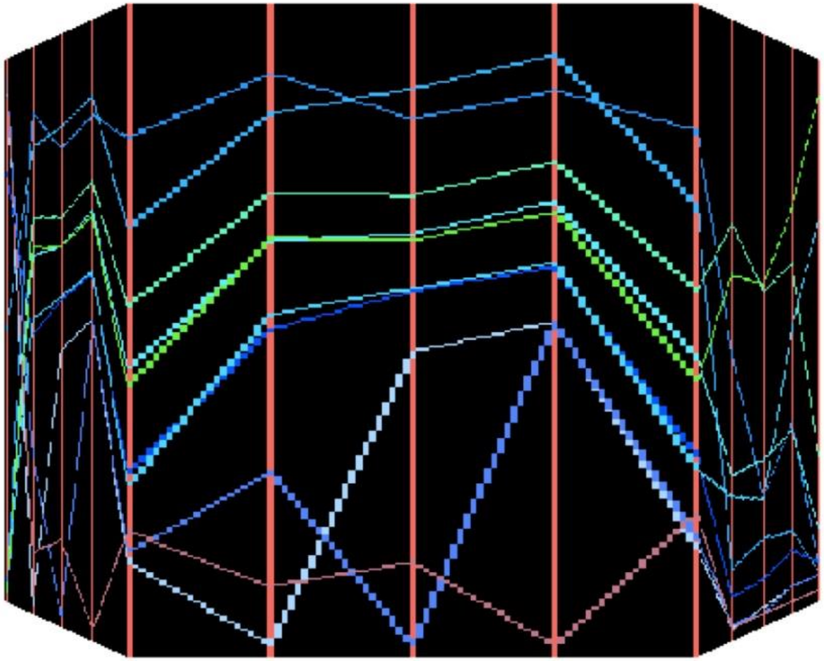
- Deriving color or transparency only from data values can lead to visual artifacts caused by noise or variability in the data
- Possible solution is to use also other data characteristics than only their values (first, second derivation, …)

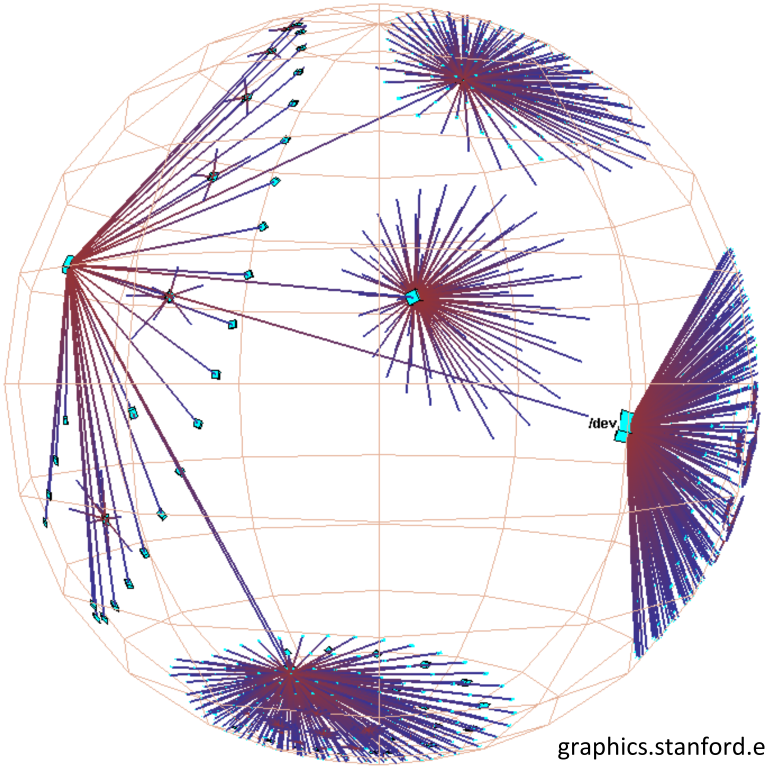
Space of objects (3D surfaces)

- Data is mapped to geometric objects which are subsequently transformed and changed by interactions
- Navigation – flying around objects and observing their surface (global and detailed views)
- Selection – clicking on objects of interest or selecting these objects from a list
- Remapping – change of object used for data mapping

Space of objects

- Distortion examples – perspective walls and hyperbolic projections

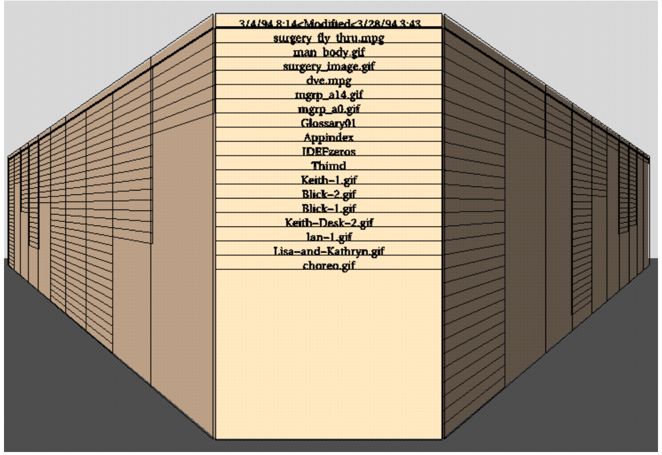




graphics.stanford.edu

Space of objects

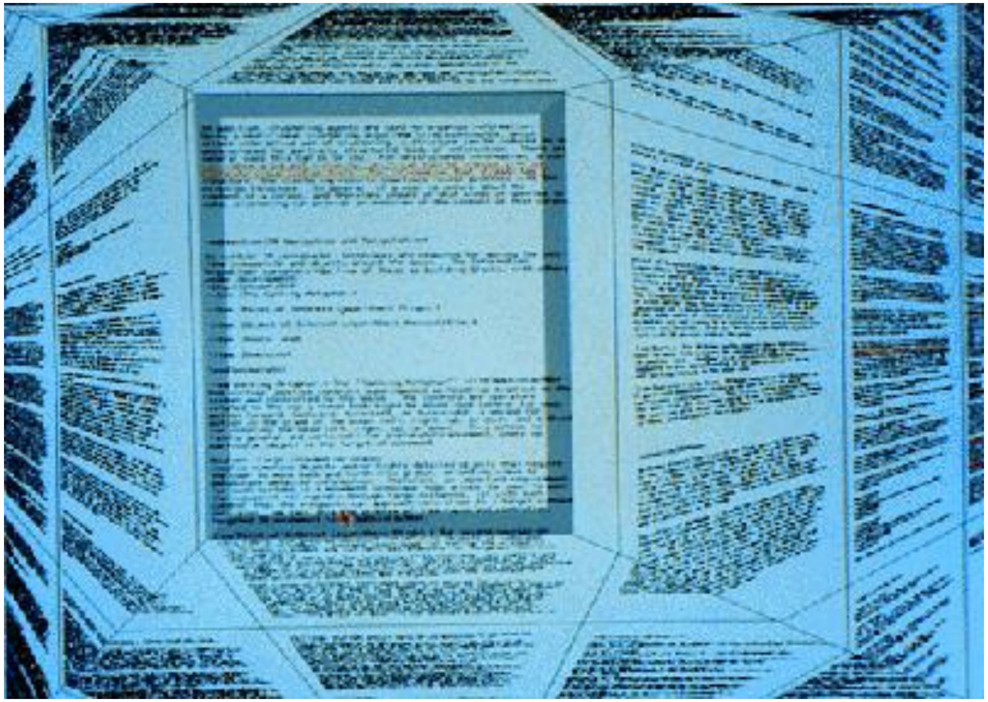
- **Perspective walls** are method for navigation in the large set of visualized documents and data
- Display one panel positioned orthogonally to the viewpoint and the other panels are distorted based on the distance from the central panel – using perspective deformation



www.cc.gatech.edu

Perspective wall

- Simplified version – front panel is scaled horizontally, neighboring segments are scaled horizontally and vertically + segments are distorted (shear)



www.pcmag.com

Perspective wall

- If the left, middle, and right part of the original image is bounded by (x_0, x_{left}, x_{right}, x_1) and the left, middle, and right panel of the resulting image is determined using (X_0, X_{left}, X_{right}, X_1), then the transformation is defined as:
  - for $x < x_{left}$:

$$x' = X_0 + (x - x_0) * \frac{(X_{left} - X_0)}{(x_{left} - x_0)}$$

$$y' = (X_{left} - x') + y\left(1 - \frac{(X_{left} - x')}{(X_{left} - X_0)}\right)$$

Perspective wall

– for $_{\text{xleft}} \leq x < x_{\text{right}}$:

$$x' = X_{left} + (x - x_{left}) * \frac{(X_{right} - X_{left})}{(x_{right} - x_{left})}$$

$$y' = y$$

– for $x \geq x_{\text{right}}$:

$$x' = X_{right} + (x - x_{right}) * \frac{(X_1 - X_{right})}{(x_1 - x_{right})}$$

$$y' = (x' - X_{right}) + y\left(1 - \frac{(x' - X_{right})}{(X_1 - X_{right})}\right)$$

Perspective wall

- The user can sequentially traverse through panels, can use jumps to the regions of interest (often implemented as a bookmark on the top of the beginning of each section)
- http://www.youtube.com/watch?feature=player_embedded&v=hYUZbrWtCZg

Space of structure visualization

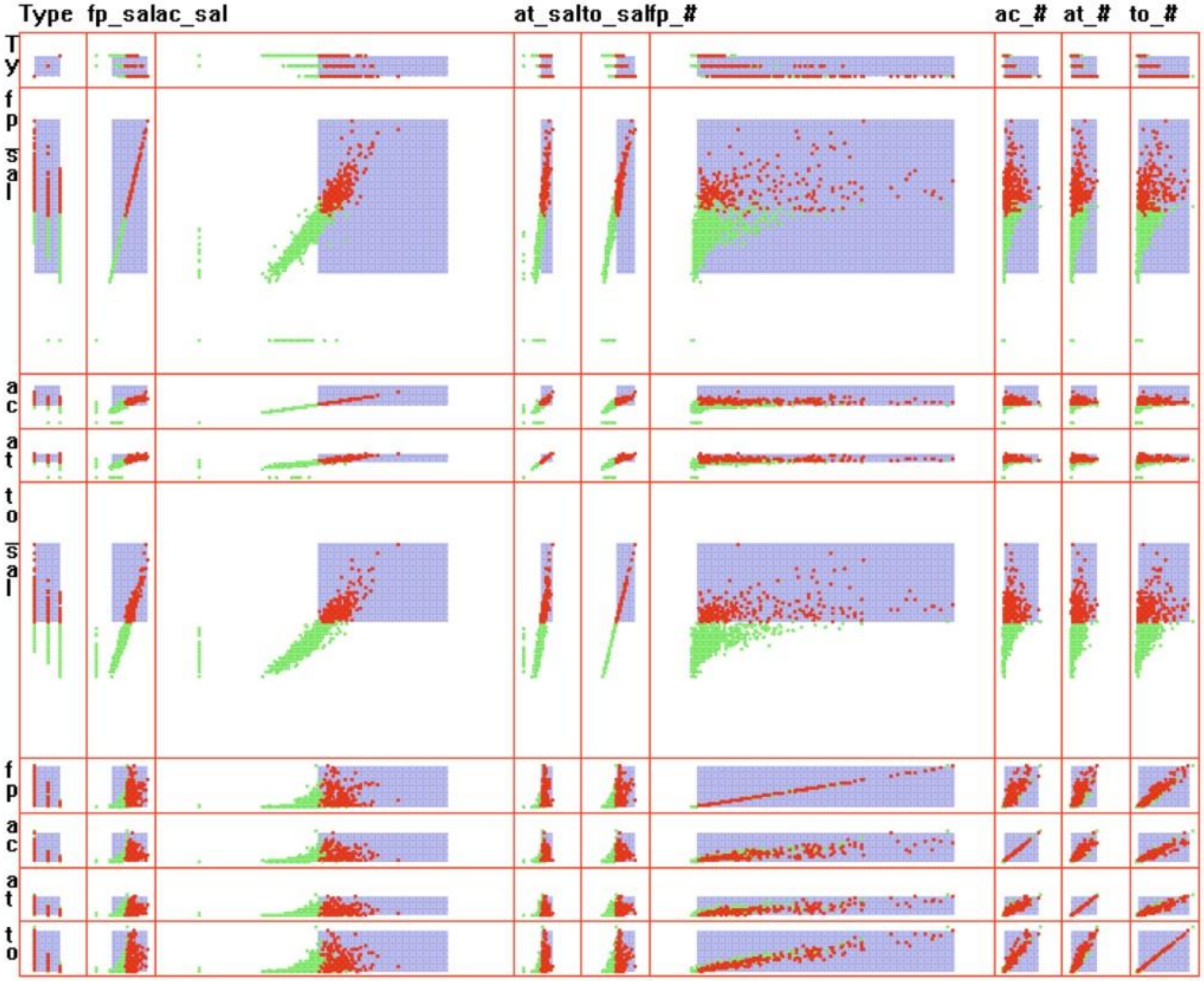
- Visualization focuses on structure relatively dependent on values, attributes, and data structure – e.g., grid containing a scatterplot matrix
- Navigation – shifting pages in table-based visualization, zooming to individual graphs in a scatterplot matrix

Space of structure visualization

- Selection – selecting components which should be hidden, moved, or shuffled
- Distortion – e.g., table lens technique – transformation of rows and/or columns in order to reach multiple LODs

- Smooth transition between visualizations is crucial

Space of structure visualization – distortion



Animation transformations

- All interactions lead to changes in the visualized image
- Changes can be significant (opening new dataset) or small (change of some view aspects)
- It is desirable to create a smooth transition between the starting and end position (e.g., when rotating with a 3D object). Linear interpolation is often sufficient.
- More appealing result can be reached using acceleration

Animation transformations

- First step is to get a uniform parametrization of a variable or variables which should be controlled in the animation
- For changing position along a straight line or scaling, **linear interpolation** is sufficient
- For calculating uniform distribution of positions along a curved path we need to introduce a new parameter

Animation transformations

- Let’s assume that the original parameter is a function of *t* (with values between 0 and 1)
- For calculation of positions, we can use a cubic polynome (same for y axis):

$$x(t) = At^3 + Bt^2 + Ct + D$$

- For $0 \leq i \leq n$ (n is the number of steps between the starting and end position) we can create a list of positions p_i

Animation transformations

- The length of arc A can be assessed as a sum of distances between two consecutive points:

$$A = \sum_{i=1}^{i=n} dist(p_{i-1}, p_i)$$

- However, for most curves this distance is different for each pair. So the described approach would lead to uneven speed along the curve.

Animation transformations

- For each point p_i we can calculate distance d_i from the curve starting point to this point p_i
- We calculate function A(i) which represents a percentage ratio of distance of the point in the i-th time step

$$A(i) = d_i / A$$

- For simplification, we can use variable *t* (0.0 ≤ t ≤ 1.0) instead of variable *i*. We also define parameter s = A(t).

Animation transformations

- Results are stored in table where for each *t* we know the corresponding s = A(t)
- Value *s* is then used for determining the uniform speed – using linear interpolation

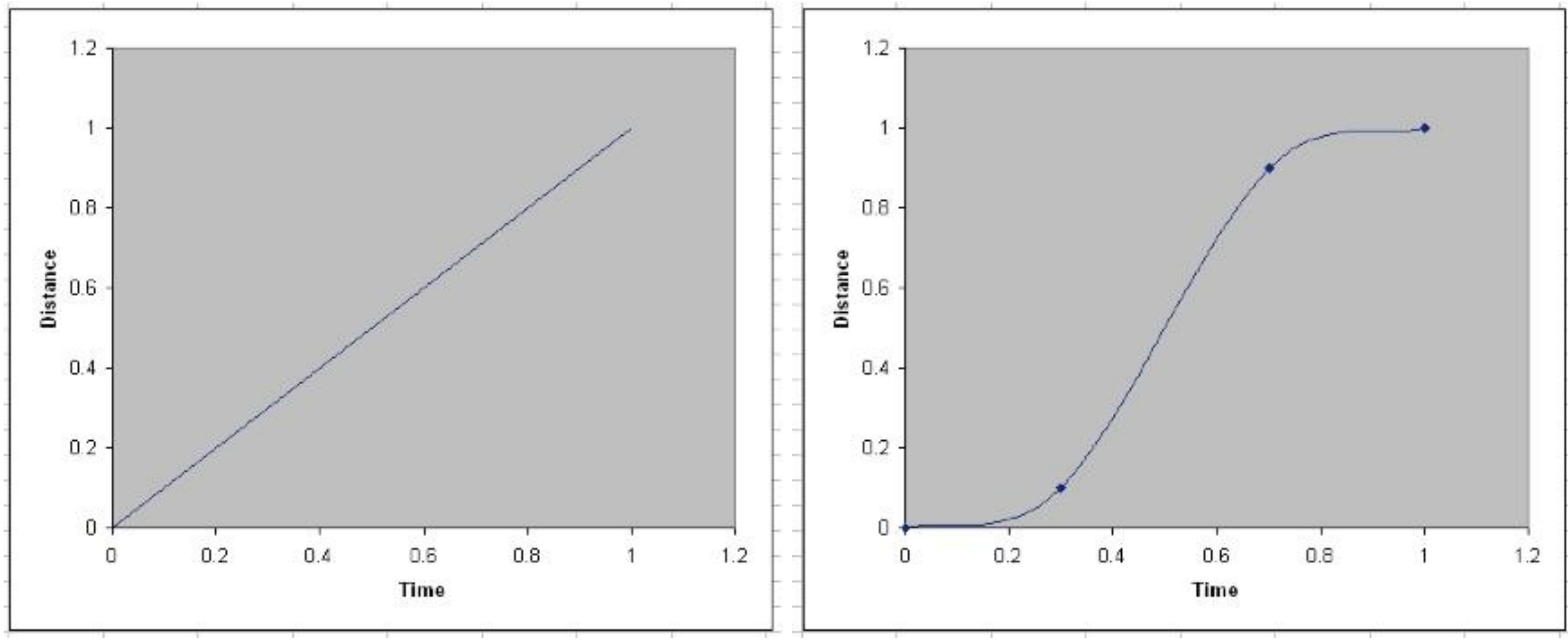
Animation transformations

- The described approach is called **reparametrization**
- The *s* parameter controls the speed – the speed corresponds to the curve inclination
- Curve does not have to be straight – parts with small inclination correspond to slow animation, high inclination means high speed
- Starting and end points are fixed

Animation transformations

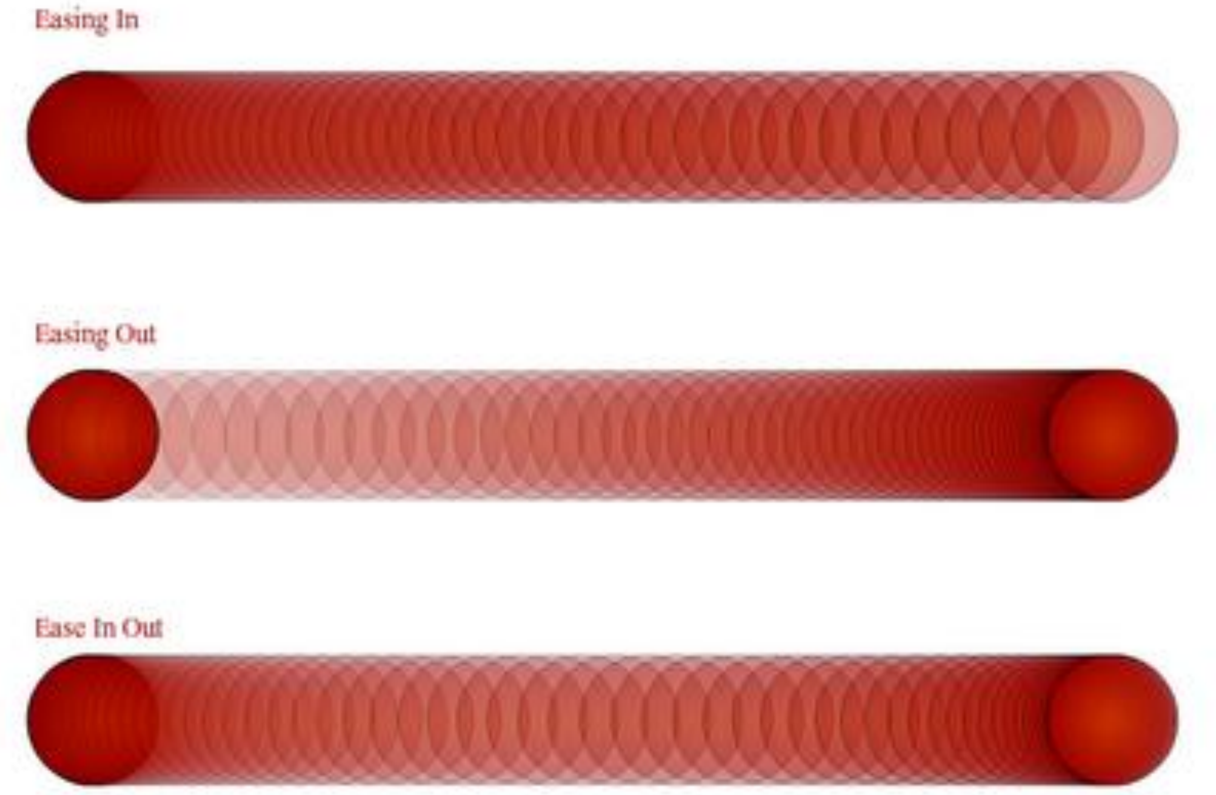
- Infinite number of possible animation settings between the starting and end point (the animation can be also paused anytime)
- We assume that the curve increases and cannot return back
- Commonly used curve is the sine curve – it corresponds to gradual increase of speed, from zero at the starting point to the desired speed and decrease of speed at the end

Animation transformations



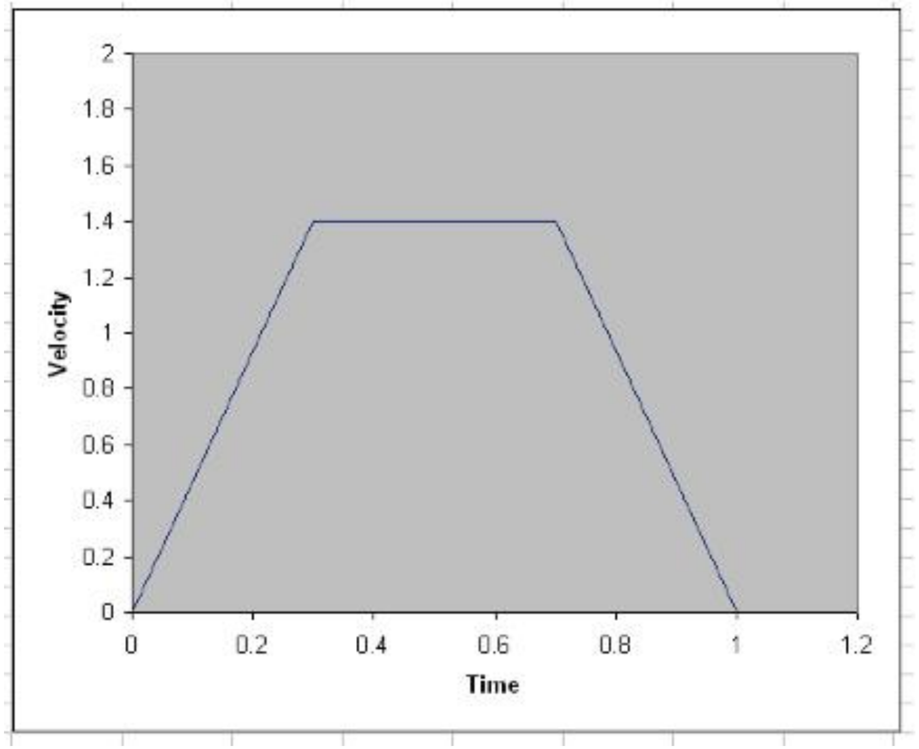
- Constant speed vs. sine curve for gradual increasing and decreasing of speed

Gradual increasing and decreasing of speed



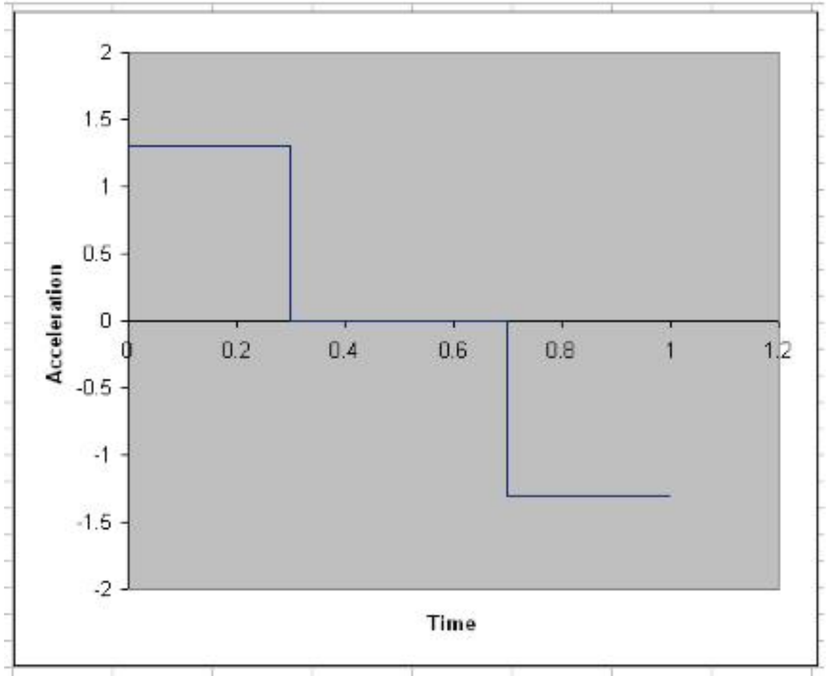
Animation transformations

- Specification of the movement using the **speed curve**
- Speed is the first derivation of the curve for positions
- Curve for continuous increasing and decreasing of speed:



Animation transformations

- Third type of curve is the **acceleration curve** – it corresponds to the second derivation of the curve for positions or to the first derivation of the curve for speed
- Curve consists of three horizontal line segments:



Virtual reality

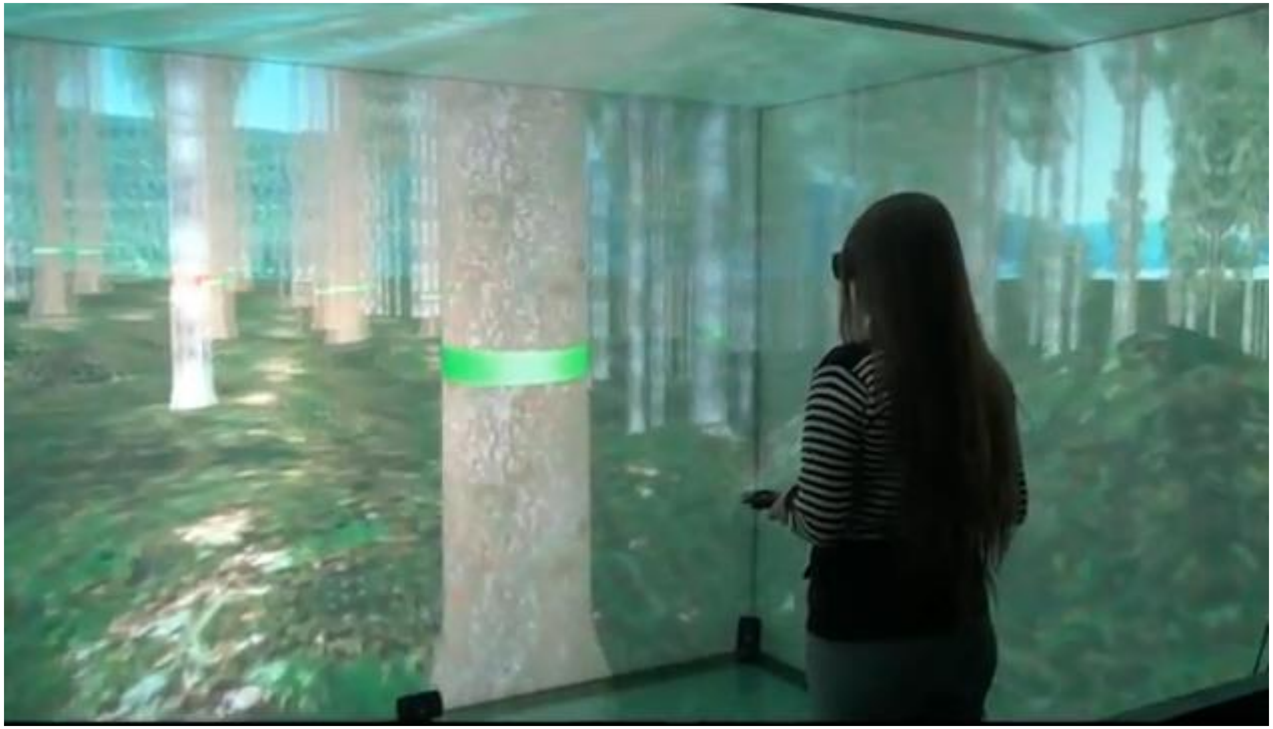
- Interaction in 3D is more complex, problem with depth perception
- Navigation has to handle six degrees of freedom
- We need to visualize not only the virtual environment but also the position of the user and the view direction
- Selection in the virtual environment vs. 3D menu

Virtual reality

- Unique benefits:
  - Navigation – movement can be influenced by head movements
  - Interaction – data gloves, optical tracking, …
  - Stereoscopic projection and depth perception – polarized glasses, active glasses, HMD…
  - Immersion – user is surrounded by the virtual world (glasses, specialized rooms)

CAVE

- http://www.youtube.com/watch?v=j59JxfbvxGg



World builder

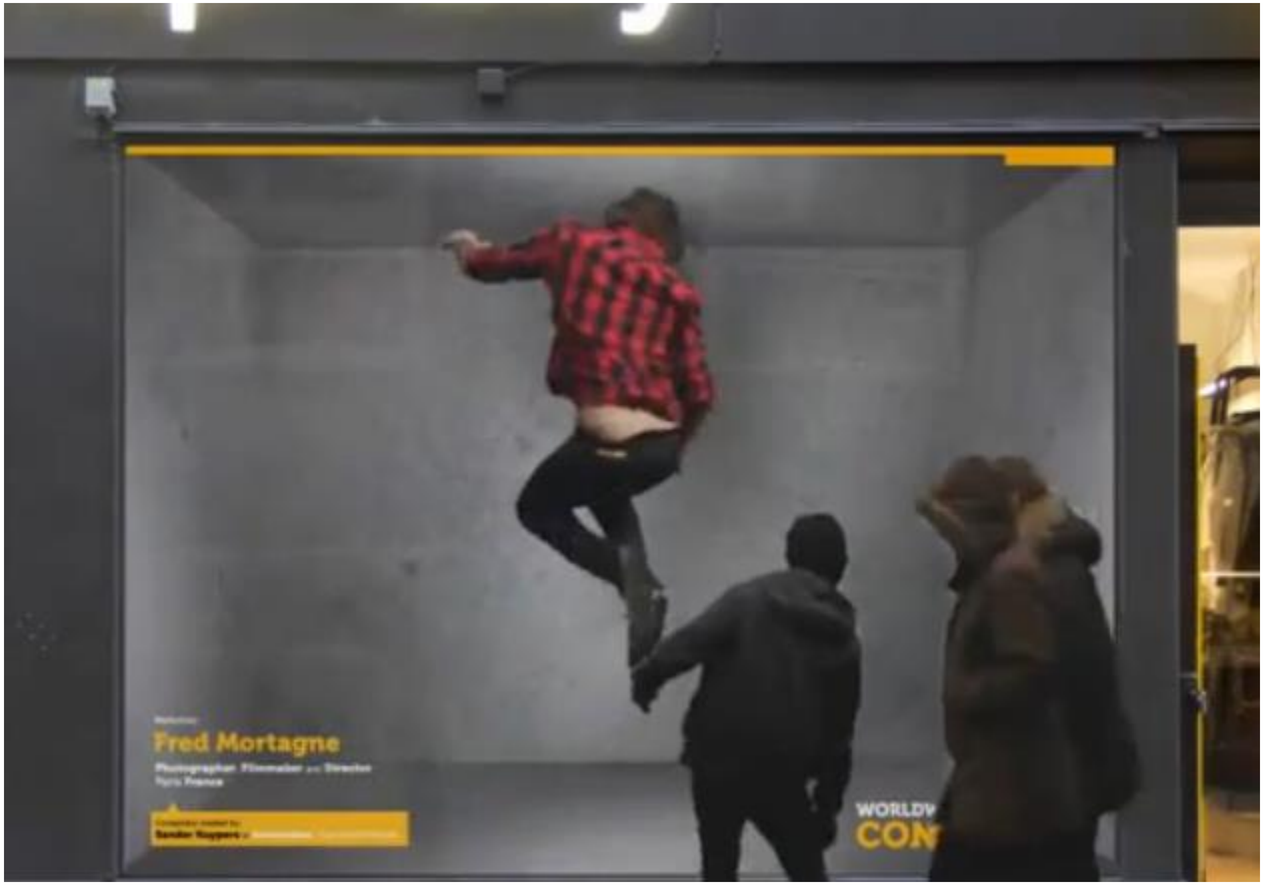
- http://www.youtube.com/watch?v=VzFpg271sm8

Microsoft’s concept of 2019

- http://www.youtube.com/watch?v=bwj2s_5e12U

Interactive display window

- http://www.youtube.com/watch?v=xFgvNMN2DiQ



Interactive table prototype

- http://www.youtube.com/watch?v=1T2veycjpTI

Interactive table

- http://www.youtube.com/watch?v=j9Pl-Nmp9nw