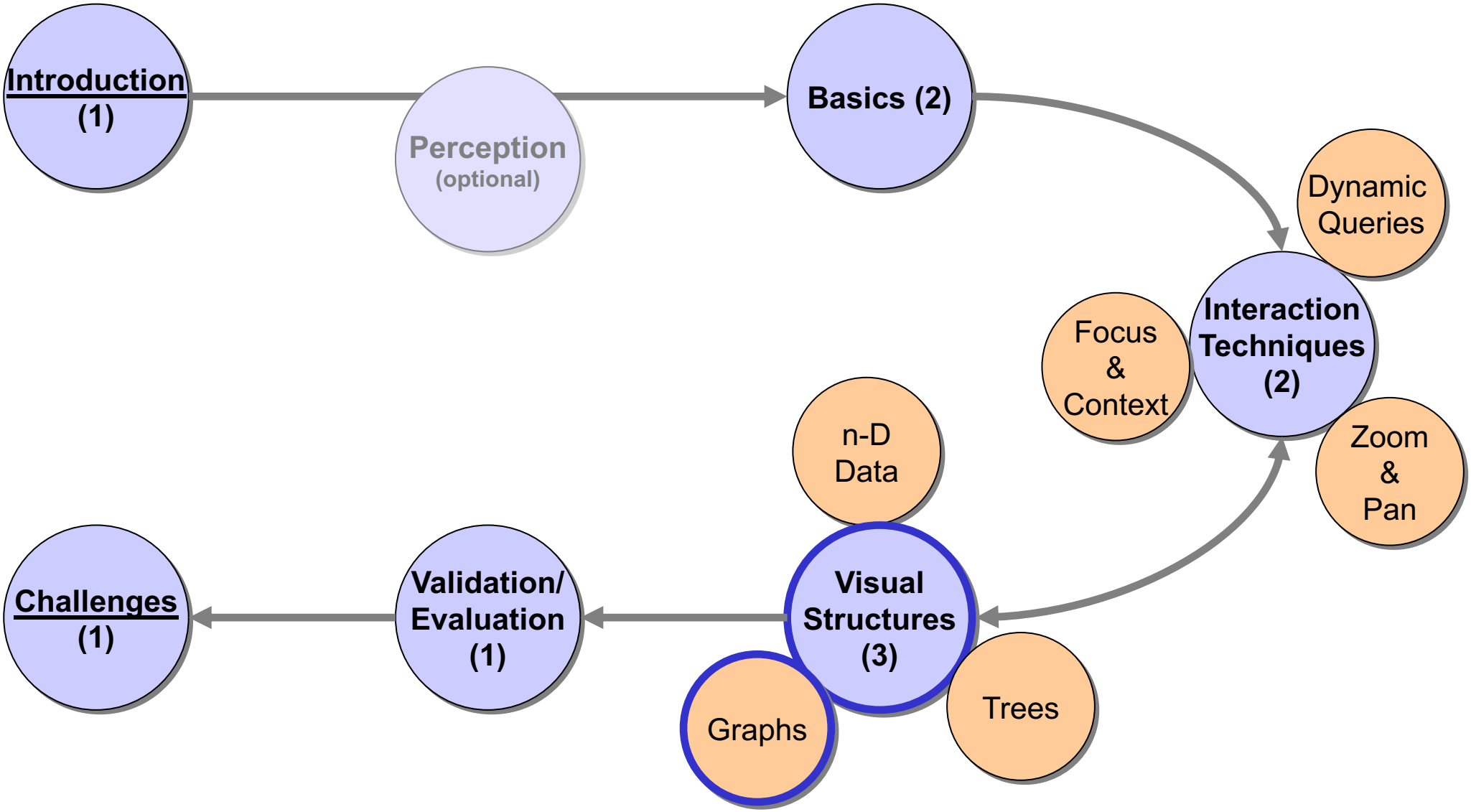
Information Visualization

6. Networks (Graphs)


Introduction (1)
Perception (optional)
Basics (2)
Dynamic Queries
Focus & Context
Interaction Techniques (2)
Zoom & Pan
n-D Data
Challenges (1)
Validation/ Evaluation (1)
Visual Structures (3)
Graphs
Trees

6.1 Motivation

- Examples for networks and graph related data
  - Molecular and genetic maps, biochemical pathways
  - Object-oriented systems and data structures, scene graphs (VRML)
  - Real-time systems (state diagrams)
  - Semantic networks and knowledge representation diagrams
  - Project management (PERT diagrams) or documentation management
  - …

6.2 Definitions

- Graphs are abstract structures, that can be used for modeling relational information
- Graph $G = (V, E)$
  - V: Set of nodes (objects)
  - E: Set of edges connecting nodes (relation)
  - Data structures:

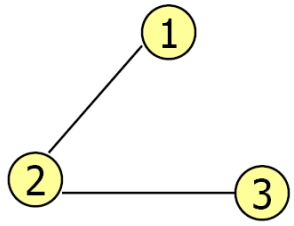
	1	2	3
1	0	1	0
2	1	0	1
3	0	1	0

Adjacency matrix

Adjacency list

1: 2
2: 1, 3
3: 2

- **Graph Drawing:** automatic drawing of graphs in 2D and 3D



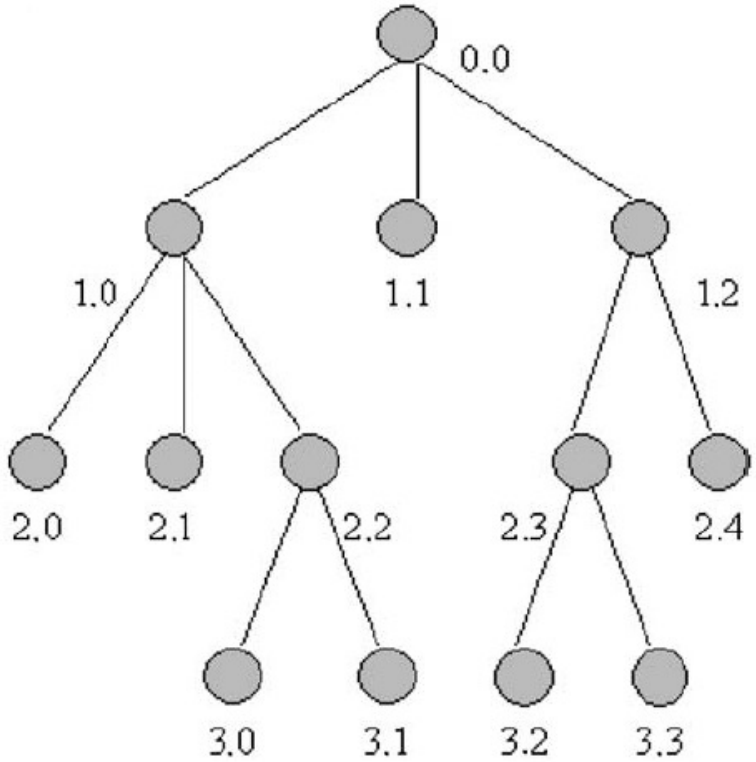
Drawing

6.2 Definitions

- Terminology
  - Graphs can have *cycles*
  - Edges can be *directed* or *undirected*
  - The *degree* of a node is the number of edges that are connected with this node
    - At directed graphs
      - *In-degree* is the number of the incoming edges
      - *Out-degree* is the number of the outgoing edges
  - Edges can have *values* (edge weights) with different types

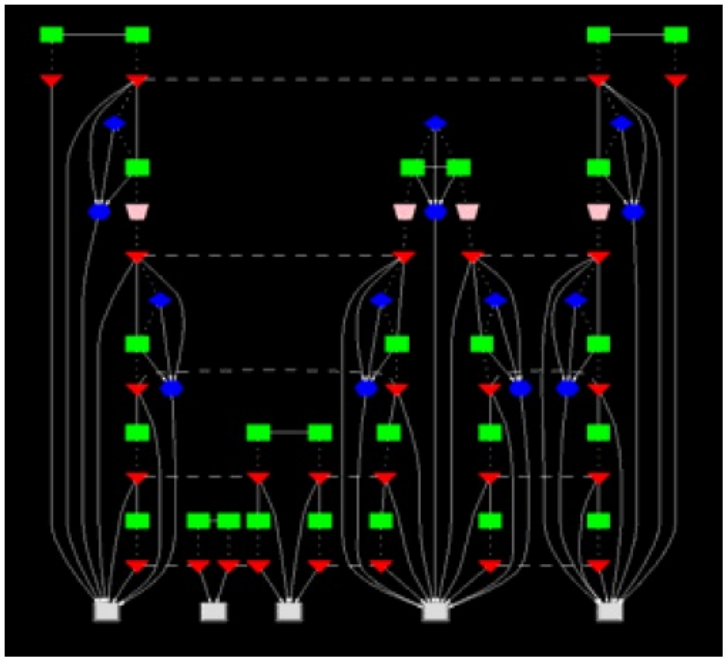
6.2 Definitions

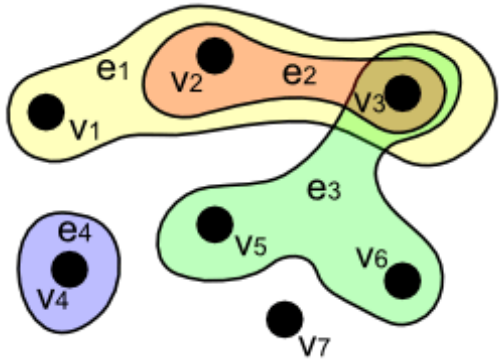
- Types of graphs
  - Trees
    - Properties
      - Special case of a graph
      - No cycles
      - Special root node
    - Free trees
    - Binary trees
    - Root trees
    - Ordered trees
  - Planar graphs



6.2 Definitions

- Types of graphs (cont.)
  - Directed/Undirected graphs
  - Extended graph models
    - Hierarchical graphs
    - Clustered graphs
    - Hypergraphs
    - …





6.3 Graph Drawing

- Own research community

  ⇒ very large field!

  - We can only give an overview
  - A good starting point for literature search and further information are the annual *Graph Drawing* conferences (*GD*) or the IEEE *InfoVis* conferences

- Major research areas of GD

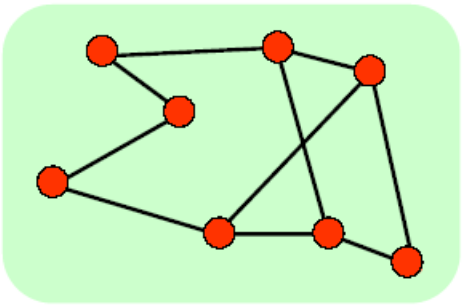
  - Graph layout and positioning of nodes
  - Scalability, dynamic graphs, compound graphs, …

6.3 Graph Drawing

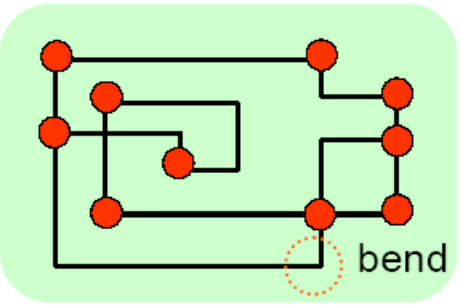
- Independent from layout and interaction techniques, there are many different possibilities to draw nodes and edges
  - Nodes
    - Shape, color, size, position, label, …
  - Edges
    - Color, size, thickness, direction, label, …
    - Shape
      - straight, curved, planar, orthogonal, …

6.3 Graph Drawing

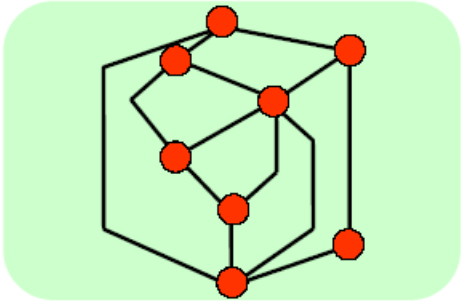
- Drawing Conventions
  - Polyline Drawing
  - Straight-line Drawing
  - Orthogonal Drawing
  - Grid Drawing
  - Planar Drawing
  - Upward Drawing
  - Circular Drawing
  - …



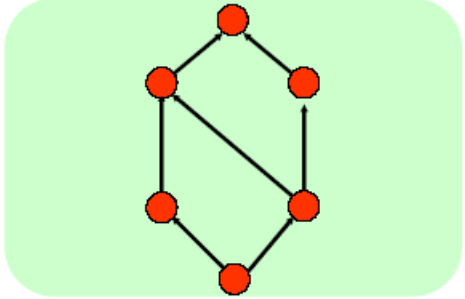
Straight-line drawing



Orthogonal drawing



Polyline drawing

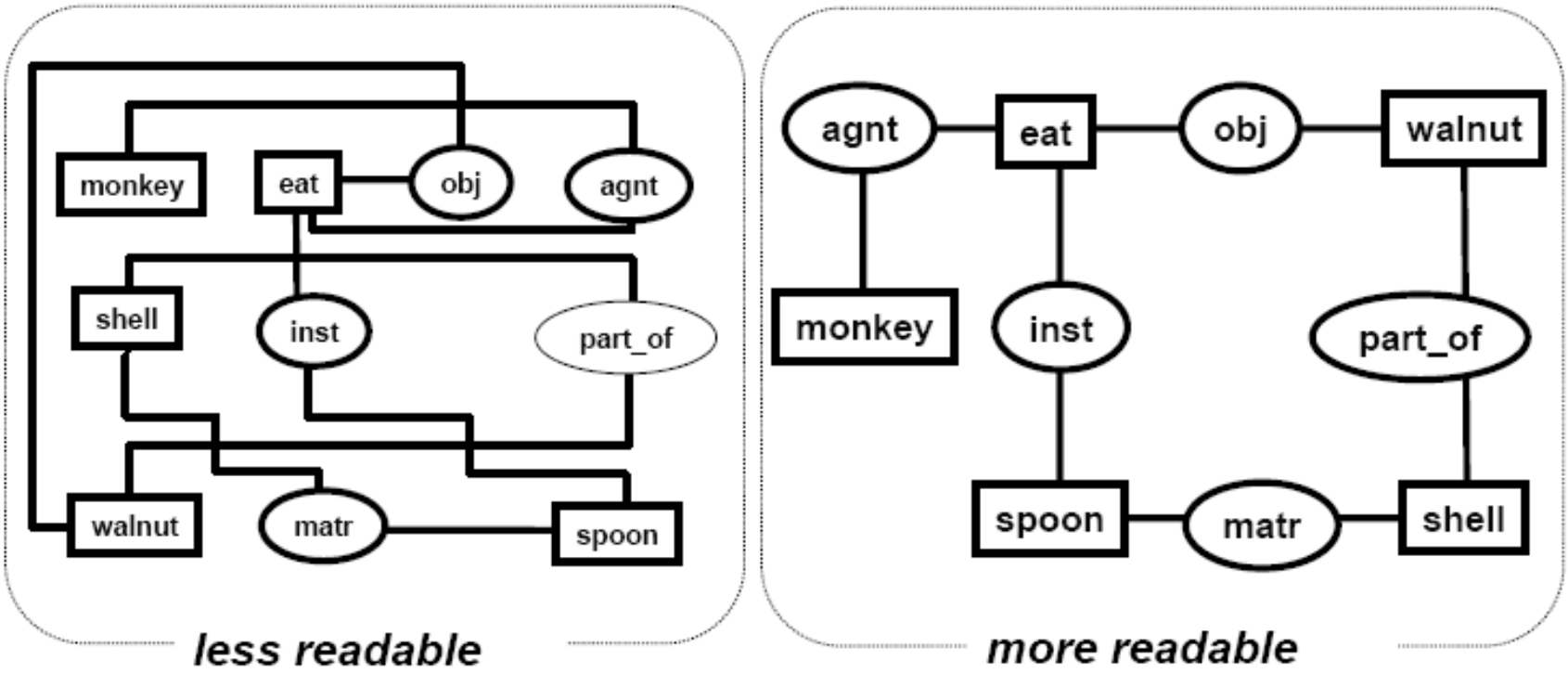


Upward drawing

[Inspired by S. Hong und P. Eades' course]

6.3.1 Aesthetics

- A graph layout should be easy to read and to understand, easy to remember, as well as have a certain aesthetics



[taken from S. Hong und P. Eades' course]

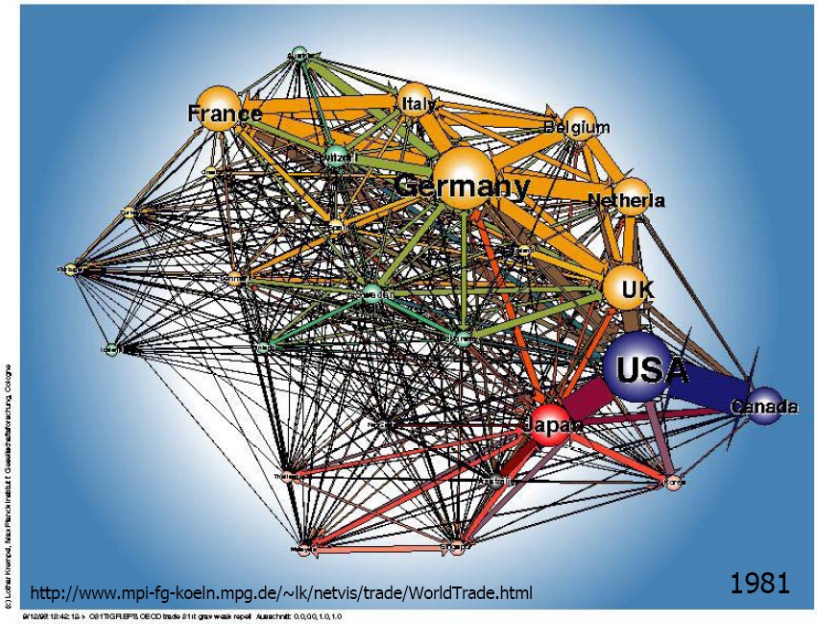
6.3.1 Aesthetics

- All layout algorithms fulfill more or less a set of *aesthetics criteria*
- Furthermore, the layout itself affects the perception of graphs
- Problem: These aesthetics criteria are sometimes contradictory are their computation mostly NP hard!
- Thus, the most GD algorithms are heuristics

[taken from S. Hong und P. Eades‘ course]

6.3.1 Aesthetics

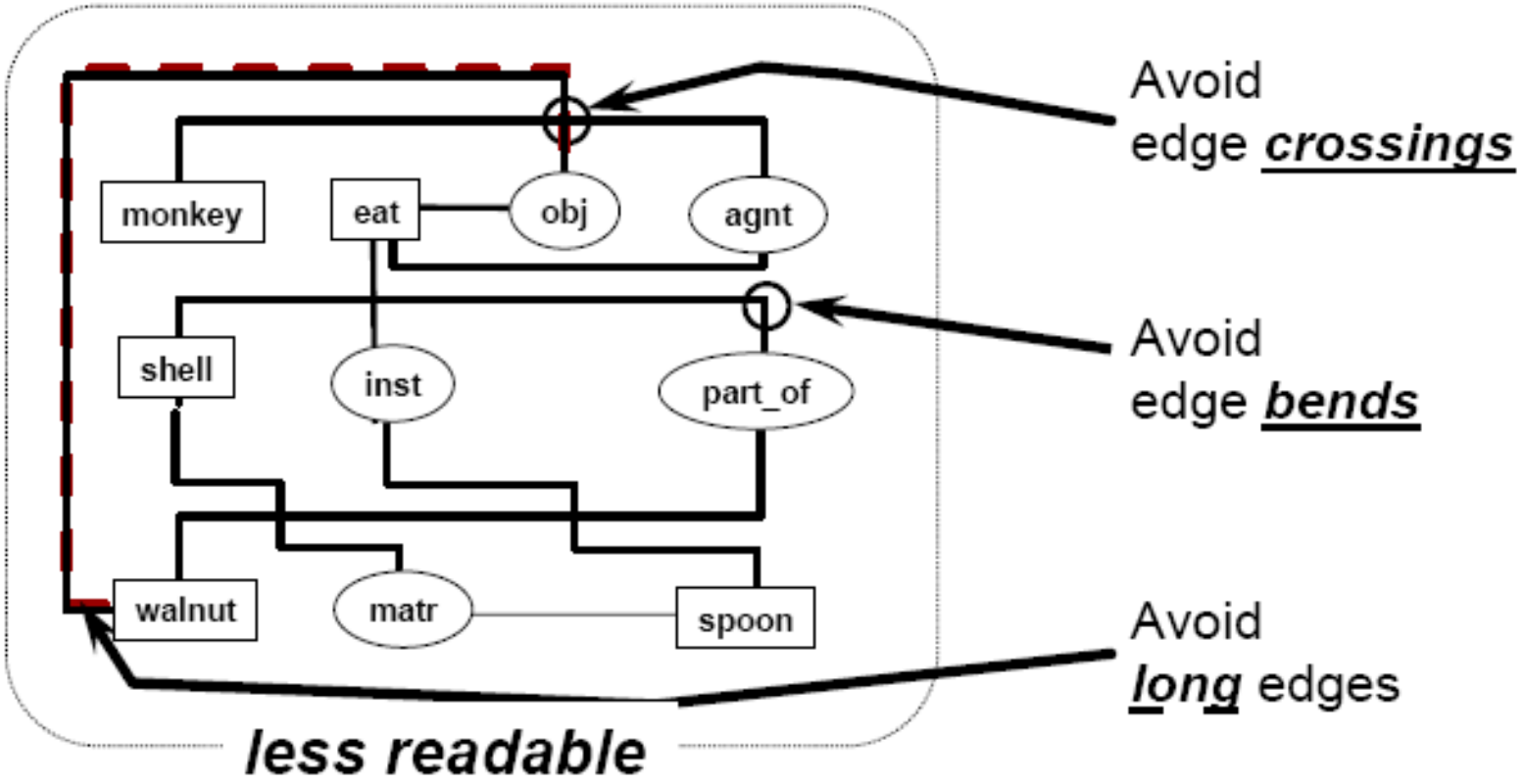
Aesthetics Criteria



- Edge crossings ↓
- Area ↓
- Symmetry ↑
- Edge length ↓
  - Maximal edge length, uniform edge length, total edge length
- Bends of edges ↓
  - Maximal bends, uniform bends, total bends
- Resolution ↑

6.3.1 Aesthetics

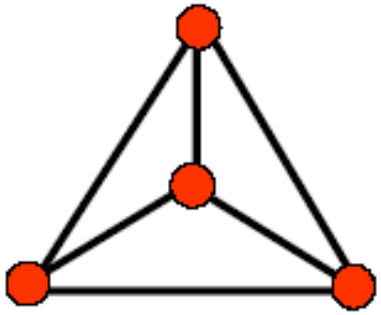
- Example: crossings and bends



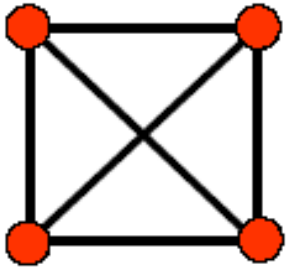
[taken from S. Hong und P. Eades' course]

6.3.1 Aesthetics

- Example: Conflict between two criteria



Minimize edge crossings



Maximize symmetry

[taken from S. Hong und P. Eades‘ course]

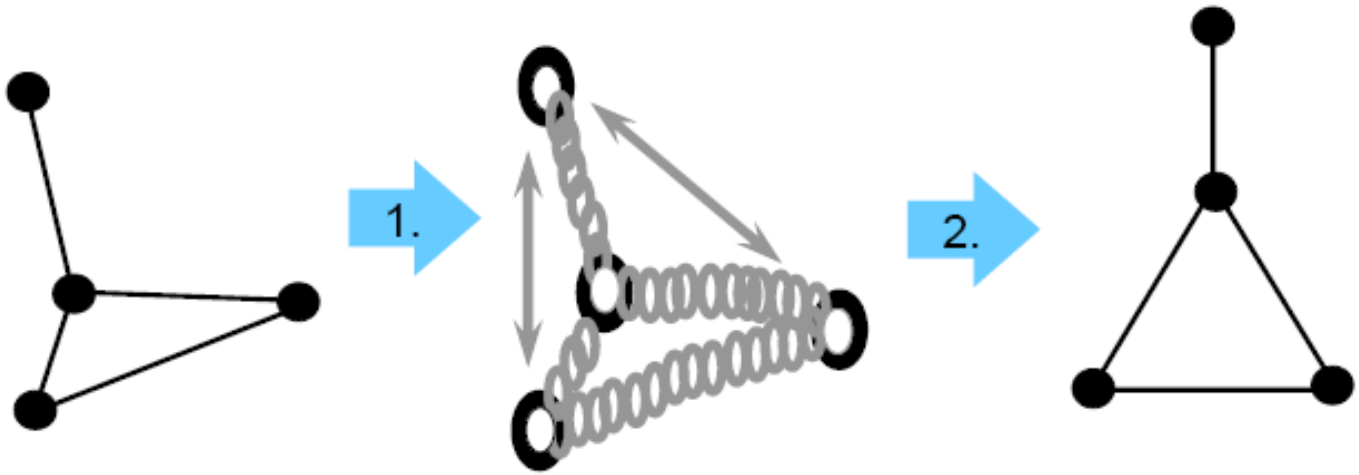
6.3.2 Force-directed GD

- Force-directed methods model nodes and edges as physical objects
- Examples
  - Spring forces for the edges
  - Gravitation forces for the nodes
- Aim is to find a stable configuration, that gets by with as few energy as possible
- We have here also optimization problems, that are solved locally

6.3.2 Force-directed GD

Spring Embedder

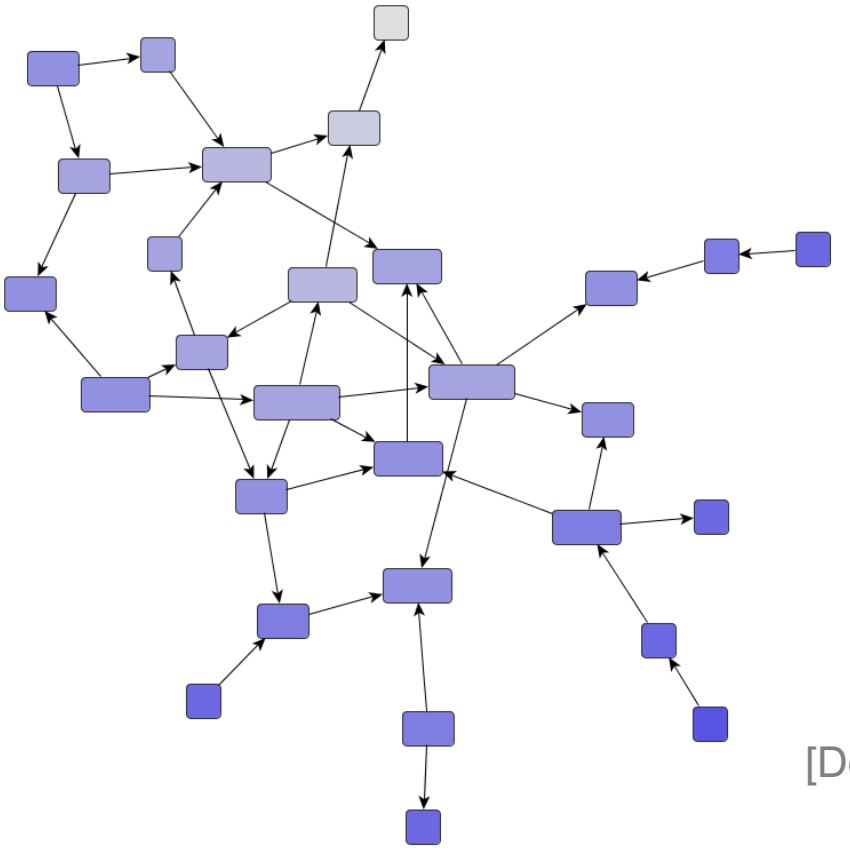
- Firstly presented by P. Eades, 1984
- Approach realizes two criteria
  - Symmetry
  - Uniform edge lengths



[taken from S. Hong und P. Eades‘ course]

6.3.2 Force-directed GD

- Problems of the classic Spring Embedder algorithm is the high runtime and its (possibly) breakdown with very large graphs
- Layout example:



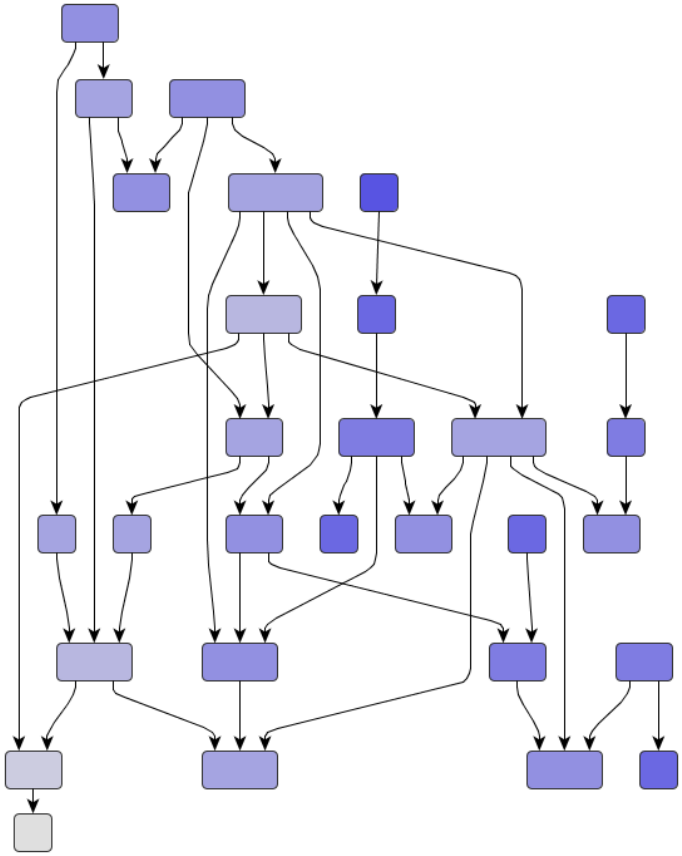
[Demo]

6.3.3 Layered GD

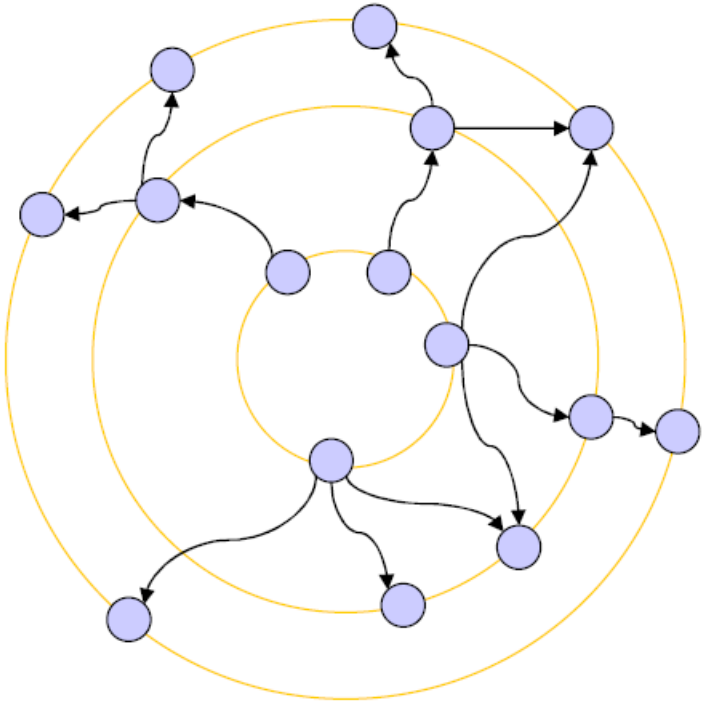
- In another approach, called *Layered GD*, the layout method firstly looks for a suitable layering that assigns each node an integer number
- Most methods computes on an extracted, acyclic subgraph that contains all nodes
  - A layer number is assigned to all nodes. Thus, the nodes are arranged top-down in rows, i.e., all nodes of an acyclic graph direct down
  - The placement (order) within the rows is used for the minimization of the number of computing steps; mostly only until the next layer is reached
  - Even this problem is NP hard, i.e., one tries to find heuristics

6.3.3 Layered GD

- Parallel layers



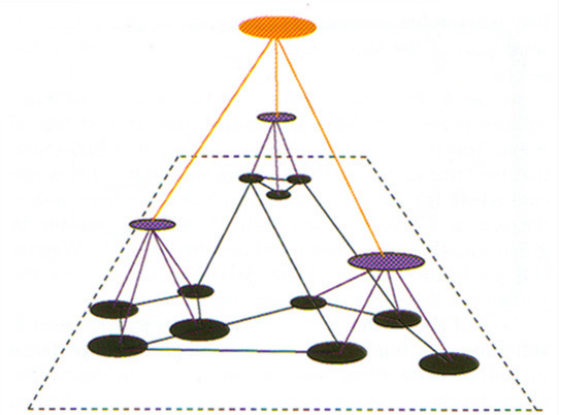
- Radial layers



[taken from S. Hong und P. Eades‘ course]

6.3.4 Clustered Graphs

- In many application areas, cluster of subgraphs play an important role
  - Example
    - Metabolic pathways
    - Social networks
    - …

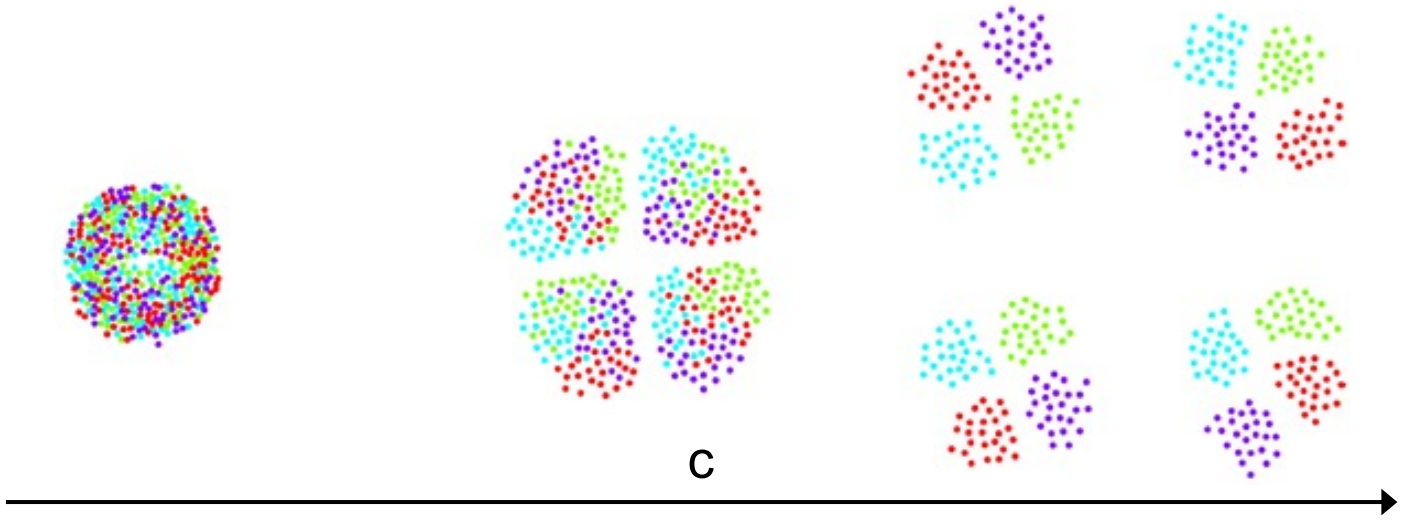


A structure induced by hierarchical clustering. (Adapted from Eades and Feng [37].)

- Cluster can result from the application itself, but also be predefined (e.g., in case of hierarchical graphs, see below)
- The graph layout should visualize the clusters adequately

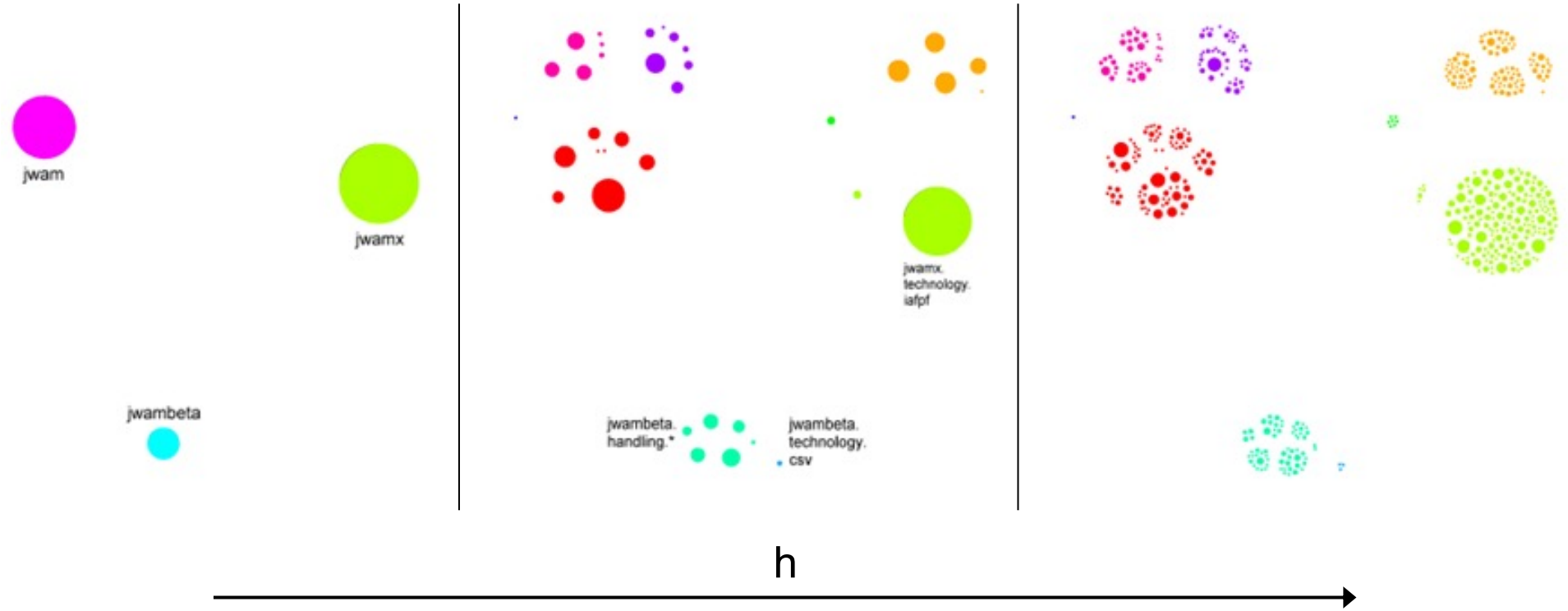
6.3.4 Clustered Graphs

- Approach of Noack and Lewerentz, 2005
  - [A. Noack and C. Lewerentz. „A Space of Layout Styles for Hierarchical Graph Models of Software Systems". ACM SoftVis `05, 2005]
- Layout criteria for clustering and hierarchy (among others)
- Modeling as energy model (not a force-based model)
- Clustering with c $\in$ (0, 1] a parameter for the cluster creation



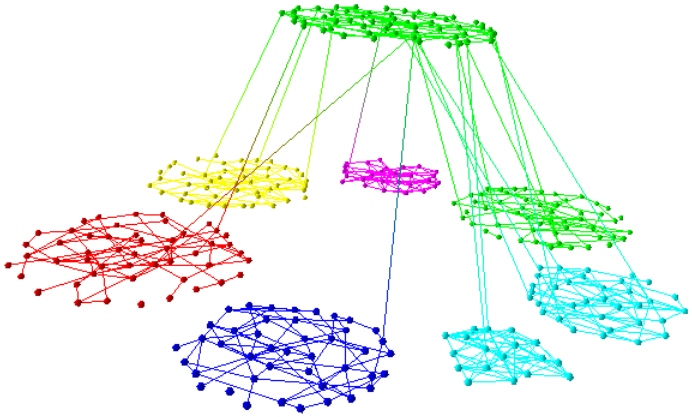
6.3.4 Clustered Graphs

- Hierarchy with c as described before and h ∈ [0, 1] a parameter for the hierarchy creation



6.3.5 Graph Drawing in 3D

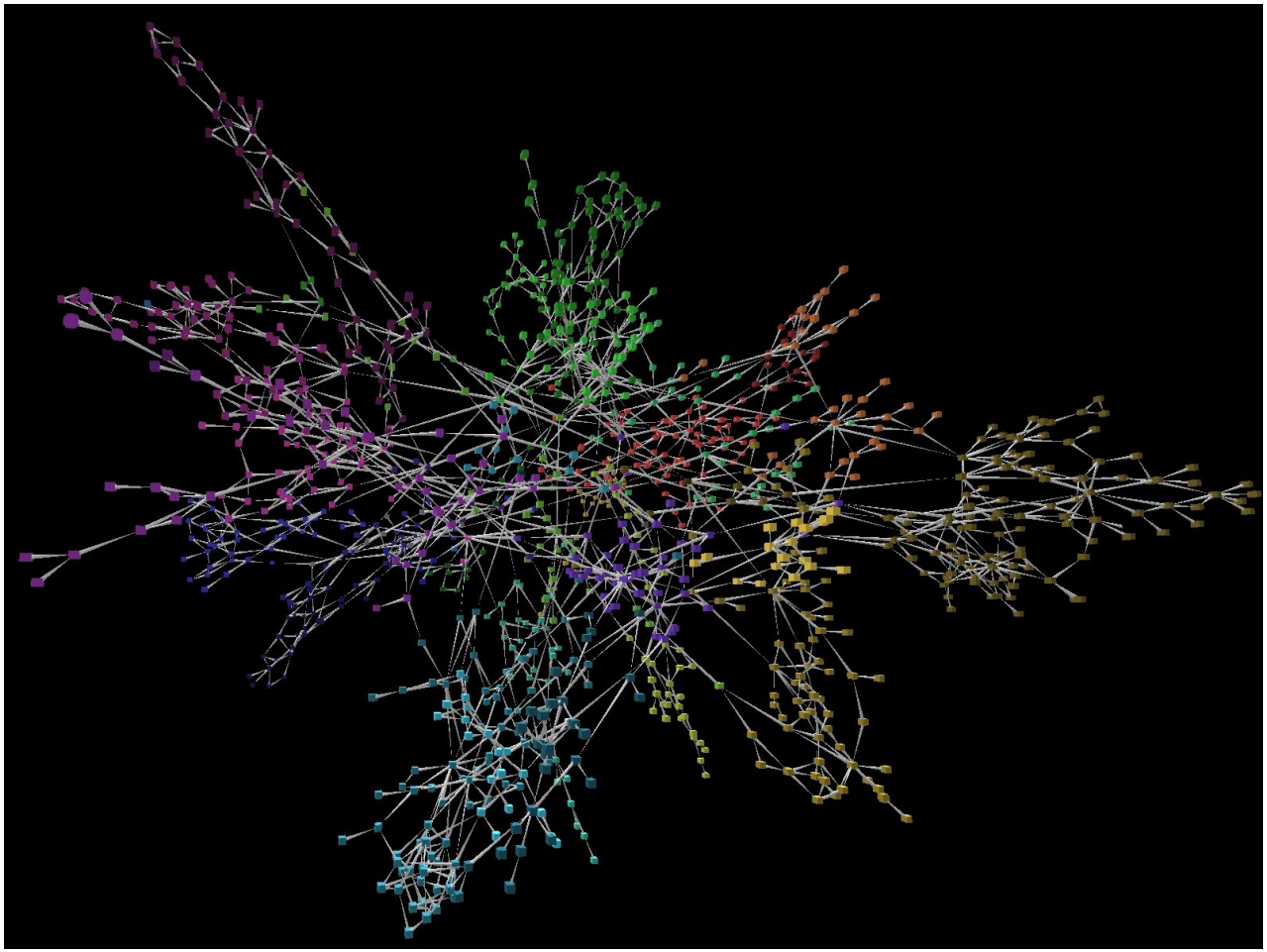
- Challenge because of the growing size of real world networks: **Scalability**
- Solutions
  - Clustering
    - Collapse strong connected nodes to super nodes (see Sect. 6.3.4)



  - 3D (more “space”)
    - Classic 2D algorithms are extended to 3D (two examples follow)
    - Problems
      - Navigation, massive overlaps, mental map, …

6.3.5 Graph Drawing in 3D

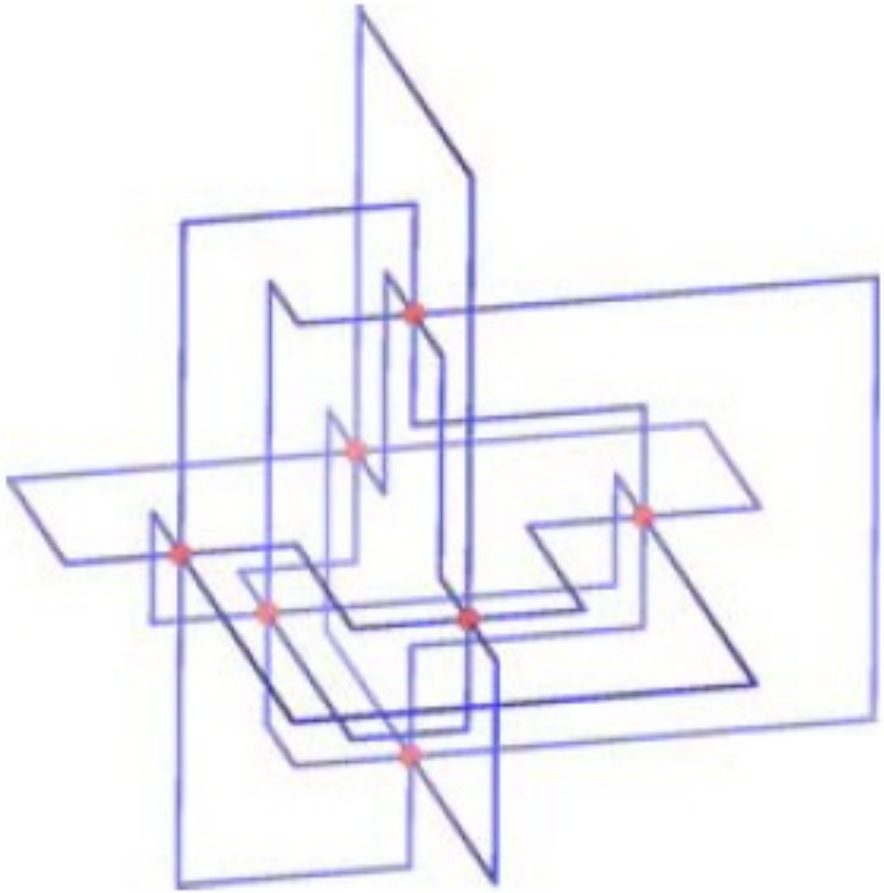
- Example: 3D force-based layout



https://vis.lbl.gov/archive/Events/SC04/ImmersiveInfoVisSC04/

6.3.5 Graph Drawing in 3D

- Example: 3D orthogonal layout



by D. Wood et al., GD ‘99

https://doi.org/10.1007/3-540-37623-2_25

6.4 Network Visualization

- Aim of *Information Visualization*
  - InfoVis focuses on the use of visualization techniques to help people understand and analyze *abstract* data
- Comparing to Graph Drawing, the focus is not on the pure layout of a graph
- More important are for example
  - Interacting with the graph visualization
  - Exploring the possibly huge graph topology
  - Adding of additional information (attributes) into the drawing

→ **Network Visualization**

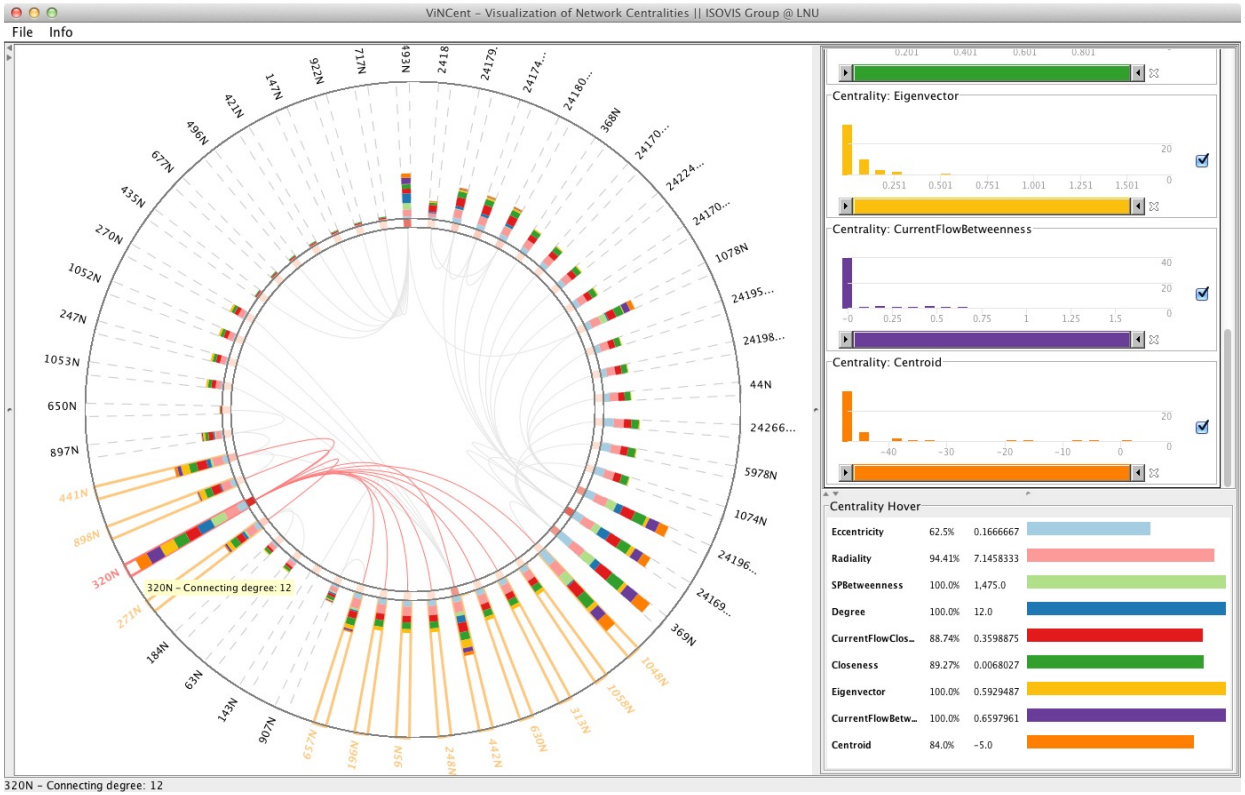
6.4 Network Visualization

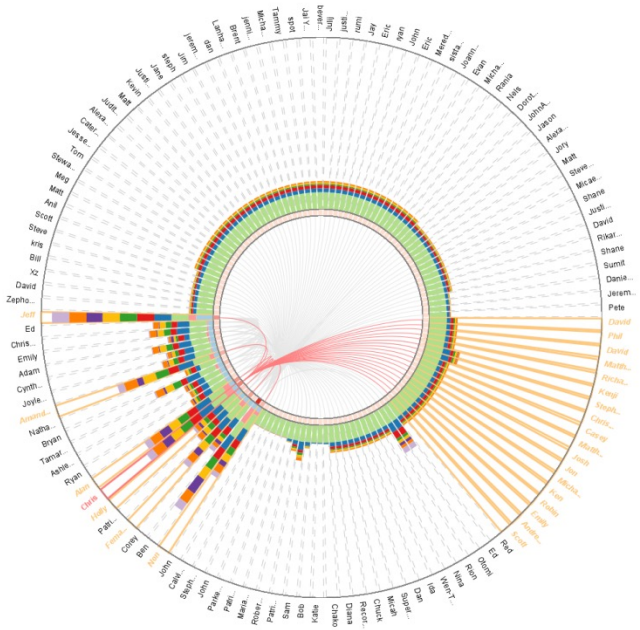
- What is a network compared to a graph?
  - Network = graph + attributed information to nodes and edges (also called *multivariate network*)
- Just to give you some impressions, we will investigate four specific aspects
  - Interactive **exploration** of (multivariate) networks and (node/edge) **clustering**
  - **Dynamic** graph layout and visual analysis
  - **Domain-specific** network visualization
  - **Attribute-based** network visualization

6.5 Exploration & Clustering

- Typical InfoVis approaches that go beyond traditional GD approaches
  - Interactive exploration of large networks that may even contain additional data attached to nodes/edges
    - Additional data of (multivariate) networks can come directly with the input data or be derived from computational analyses
  - Automatic clustering of nodes and/or edges to improve the scalability of the visual display
    - Usually, the clustering can be controlled by the users
    - The cluster results should be visualized as well

- Visualization of Network Centralities



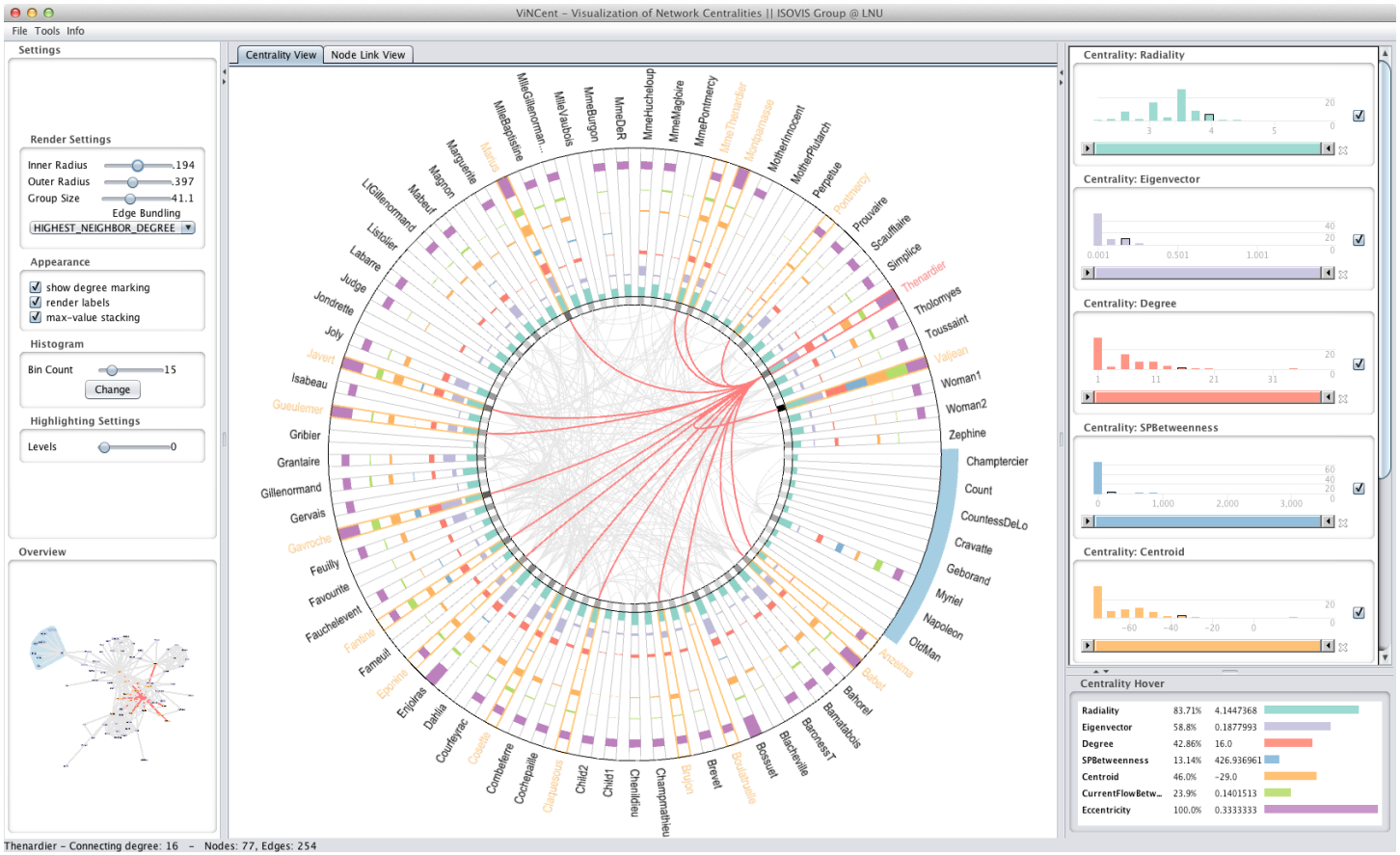


[A. Kerren, H. Köstinger, and B. Zimmer. ViNCent – Visualization of Network Centralities. In Proceedings of the International Conference on Information Visualization Theory and Applications (IVAPP '12), pages 703-712, Rome, Italy, 2012. INSTICC.]

[Video: https://vimeo.com/128267706]

6.5.2 Node Clustering

- ViNCent 2.0



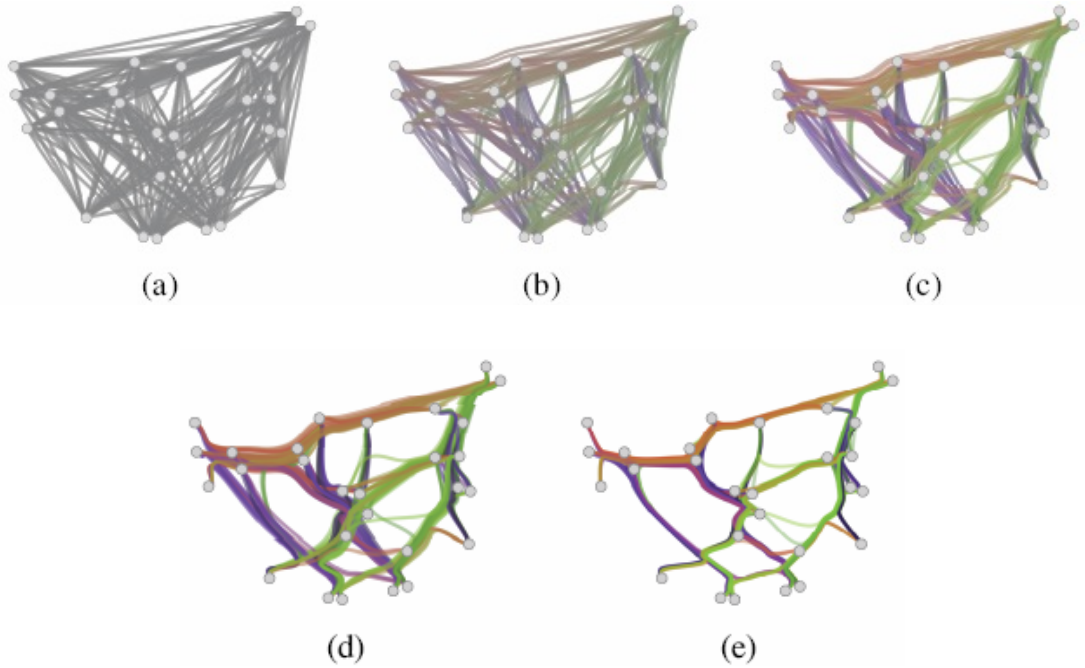
[B. Zimmer, I. Jusufi, and A. Kerren. Analyzing Multiple Network Centralities with ViNCent. In Proceedings of SIGRAD 2012 – Interactive Visual Analysis of Data (SIGRAD 2012), Växjö, Sweden, 2012.]

[Video: https://vimeo.com/128268456]

6.5.3 Edge Clustering

- Edge Bundling
  - [Weiwei Cui et al. „Geometry-Based Edge Clustering for Graph Visualization“. In Proceedings of Information Visualization 2008.]

- Idea
  - Avoid clutter of edges
  - Compute edge bundles
  - Uses a control mesh for controlling purposes



[Video: https://ivis.itn.liu.se/courses/resources/edgeBundlesInGraphs.mov]

6.6 Dynamic GD

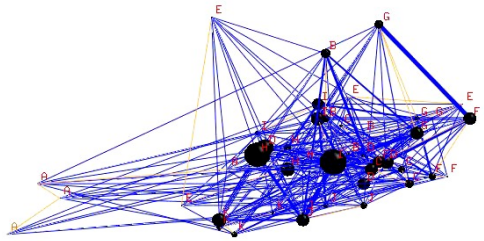
- Over the past decades, networks became more important that change with time, e.g.
  - biochemical networks have to be modified because of new discovered paths
  - social networks change through new contacts between people
  - …
- Visualizations must preserve the „Mental Map“
  - „Old structures“ should be recognized again
  - [K. Misue, P. Eades, W. Lai, and K. Sugiyama, "Layout Adjustment and the Mental Map", Journal of Visual Languages and Computing 6 (1995), 183-210.]

6.6.1 Morphing

- There are several approaches to address this problem
- One of them is the so-called **Morphing**
- Idea
  - Visualize the transitions between two layouts using smooth animations
- Advantages
  - Looks very good (good aesthetics)
- Disadvantages
  - Nodes usually change their position
  - Eventually, new added nodes or deleted nodes are not correctly identified

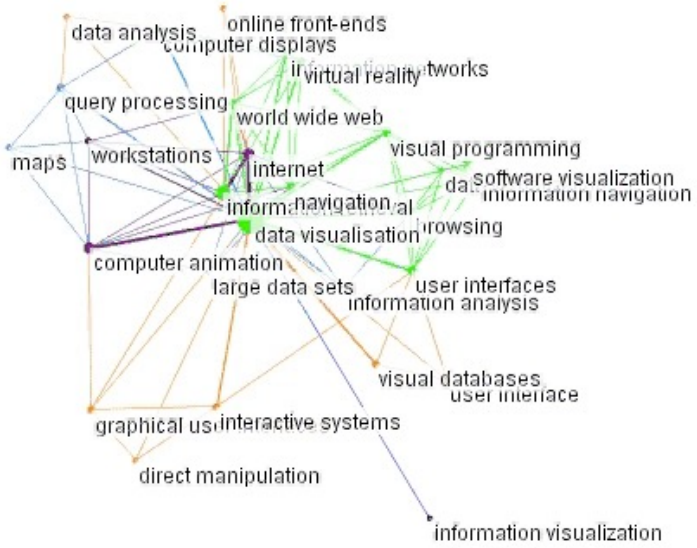
6.6.1 Morphing

- Morphing can be applied to each 2D/3D layout algorithm:
  - If a node is changing its position in the new layout then compute an animation path between the old and the new position with the help of interpolation



- Example system: *GraphAEL*
  - [C. Erten, P. J. Harding, S. G. Kobourov, K. Wampler, and G. Yee, "GraphAEL: Graph Animations with Evolving Layouts", 11th Symposium on Graph Drawing (GD), p. 98-110, 2003.]
  - Here, mainly force-based methods are used

6.6.1 Morphing

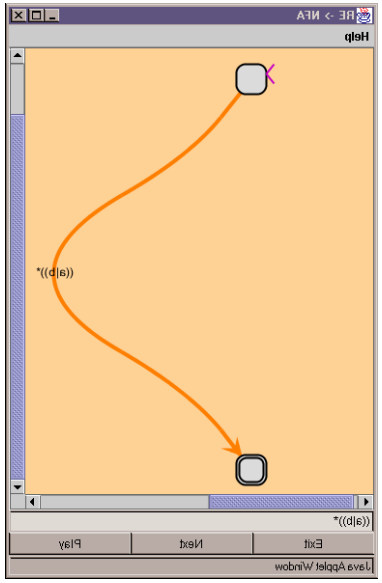


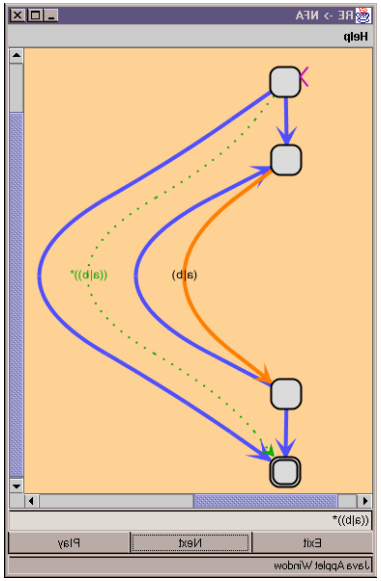
[Video: http://cs.lnu.se/isovis/courses/resources/iv04_top50keyword_labels.mov]

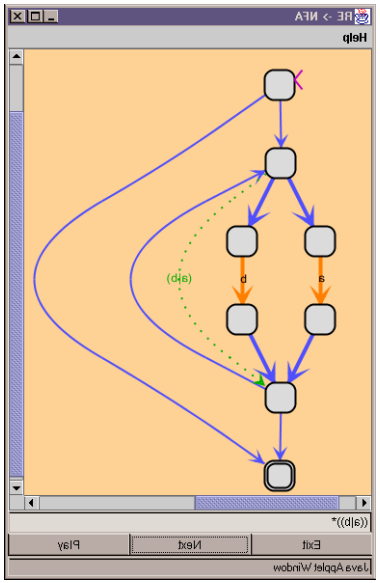
6.6.2 Foresighted GD

- If we know a sequence of graph in advance or if it is possible to precalculate it, then there is another method:
- **Foresighted Graphlayout**
  - [S. Diehl, C. Görg, and A. Kerren. „Preserving the Mental Map using Foresighted Layout“. In Proceedings of Joint Eurographics - IEEE TCVG Symposium on Visualization, VisSym ’01, Springer, 2001.]
- Idea
  - Compute a supergraph based on the sequence of graphs
  - Position the nodes at the beginning in such a way that they don‘t change their positions later

6.6.2 Foresighted GD



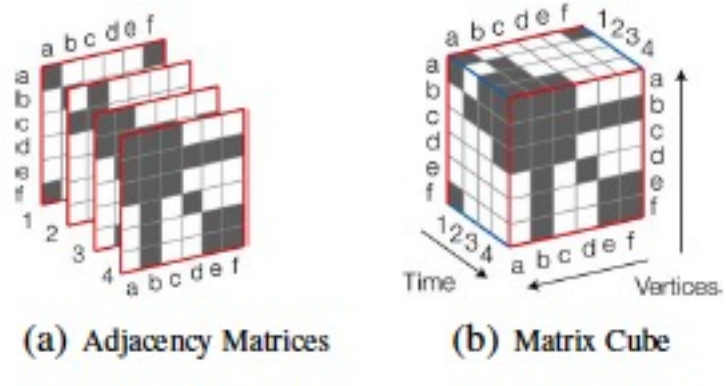




- Advantages
  - Preserving the mental map
  - Independent of the used graph layout algorithm
- Disadvantages
  - Sequence of graphs is often unknown
  - Partly bad aesthetical results (gaps at the beginning, etc.)

6.6.3 Matrix Cubes

- If the graphs are represented as matrices, then we can stack the individual matrices to display a dynamic graph
- The *Matrix Cube* approach realizes the so-called space-time-cube metaphor
  - [Benjamin Bach, Emmanuel Pietriega, Jean-Daniel Fekete: Visualizing Dynamic Networks with Matrix Cubes, Conference on Human Factors in Computing Systems (CHI), Toronto, Canada, 2014, Pages 877-886.]
- Idea
  - Matrix stack builds a cube (3D)
  - Time is ordered along the slices (2D)
  - Many interaction features support the analysis of dynamic graphs



(a) Adjacency Matrices (b) Matrix Cube

6.6.3 Matrix Cubes

- Design space

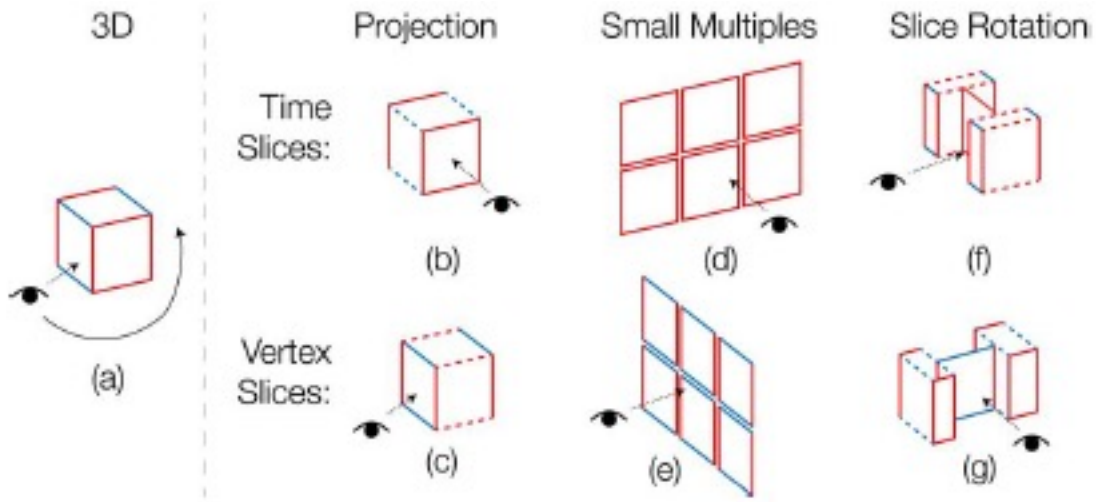
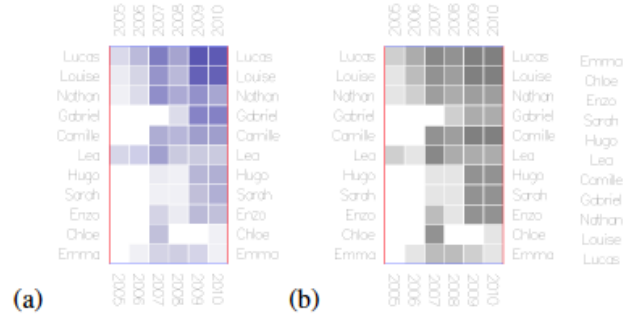


Figure 3. Cubix View design space. Columns indicate operations applied to the cube. Rows indicate operations applied to time (red × red) and vertex slices (blue × red), respectively. (a) 3D view, (b) time-projection view, (c) vertex-projection view, (d) time small multiples, (e) vertex small multiples, (f) time-slice-rotation, and (g) vertex-slice-rotation.

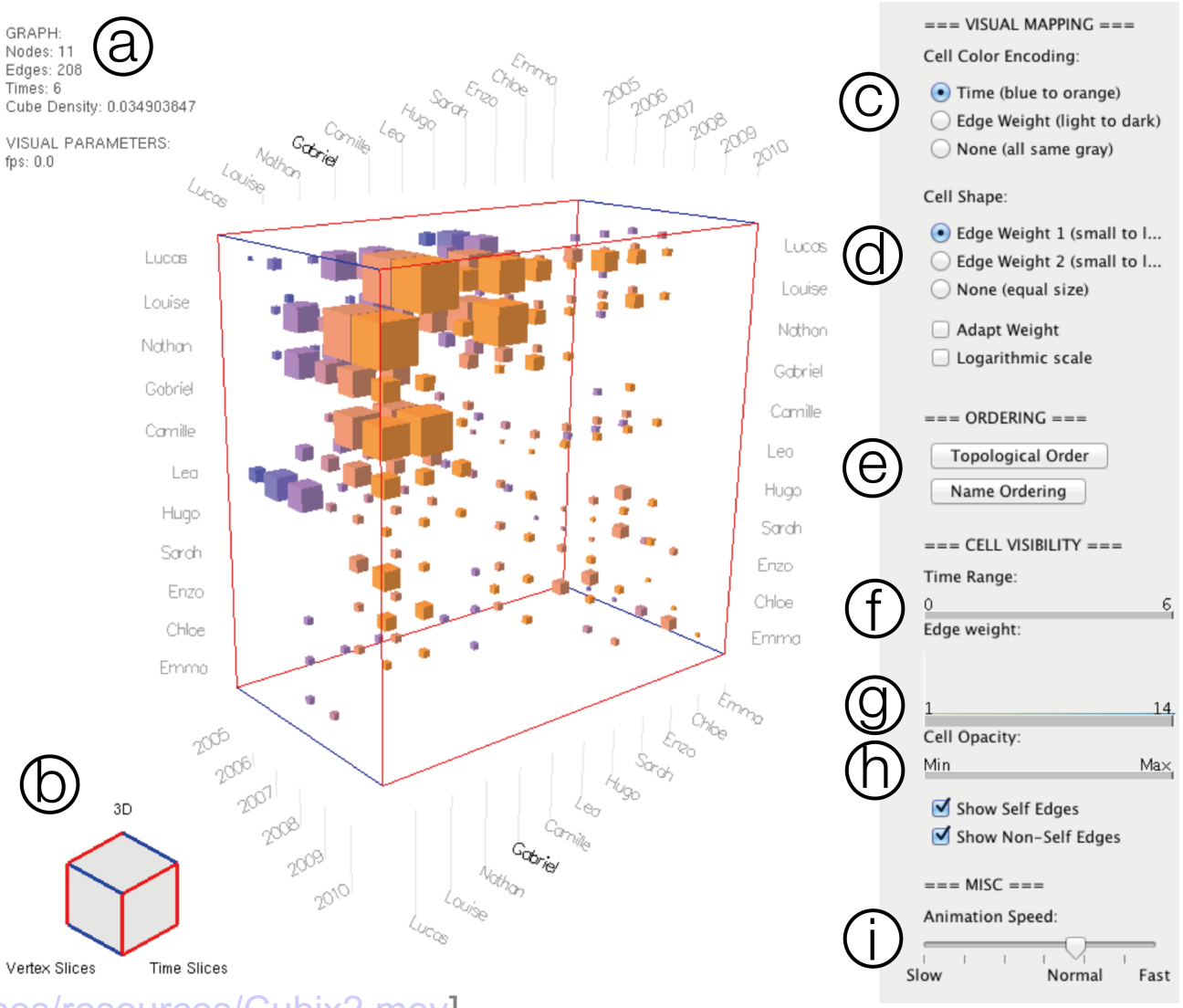
- Example: trend analysis between vertex slices



http://www.aviz.fr/Research/Cubix

6.6.3 Matrix Cubes

- Screenshot



[Video: https://ivis.itn.liu.se/courses/resources/Cubix2.mov]

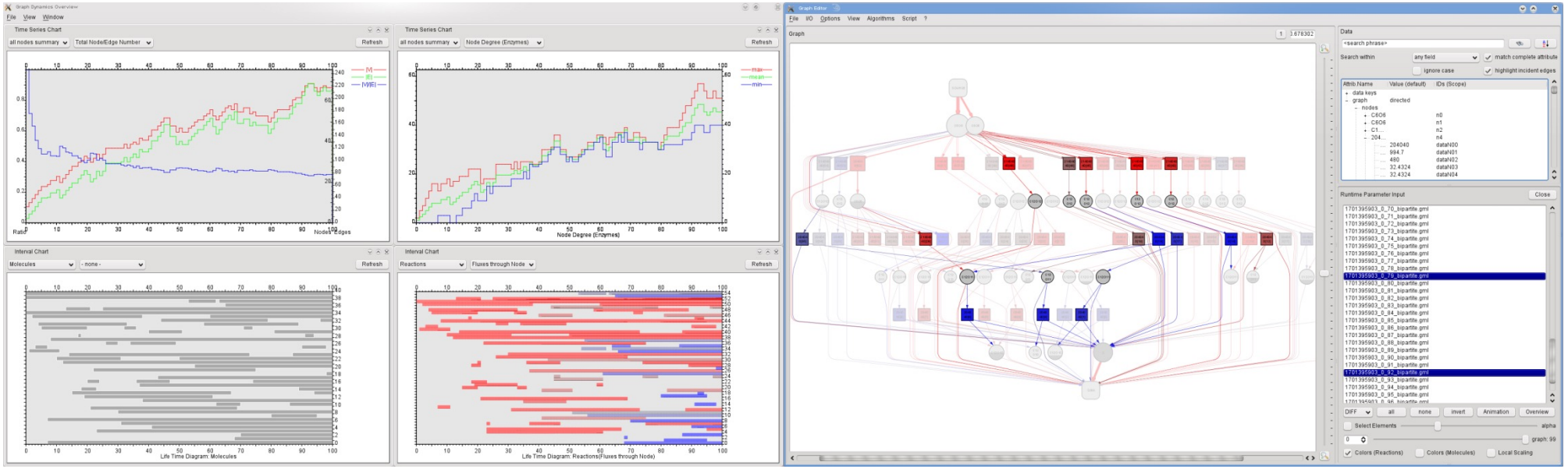
http://www.aviz.fr/Research/Cubix

6.7 Domain-specific Networks

- Visual network analysis is in the center of several application fields, most notably
  - Systems Biology
    - Metabolic networks, regulatory networks, protein-protein interaction networks, …
  - Social Sciences
    - Social networks, publication networks, food networks, …
  - Software Engineering
    - Call graphs, dependency graphs, packages hierarchies, …
- They all come with specific needs, and the input networks are often time-dependent or are attached with additional attributes

6.7.1 Biology

- Dynamic metabolic networks using Foresighted Layout

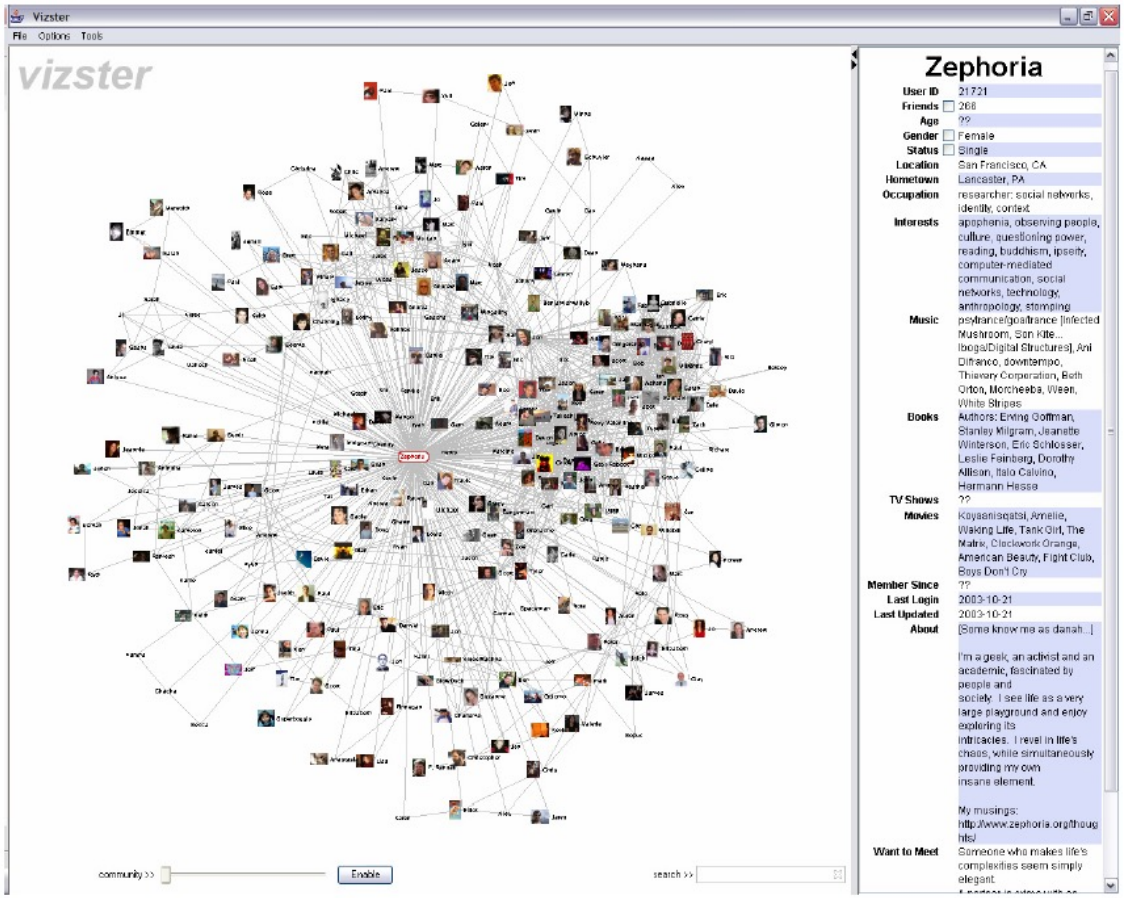


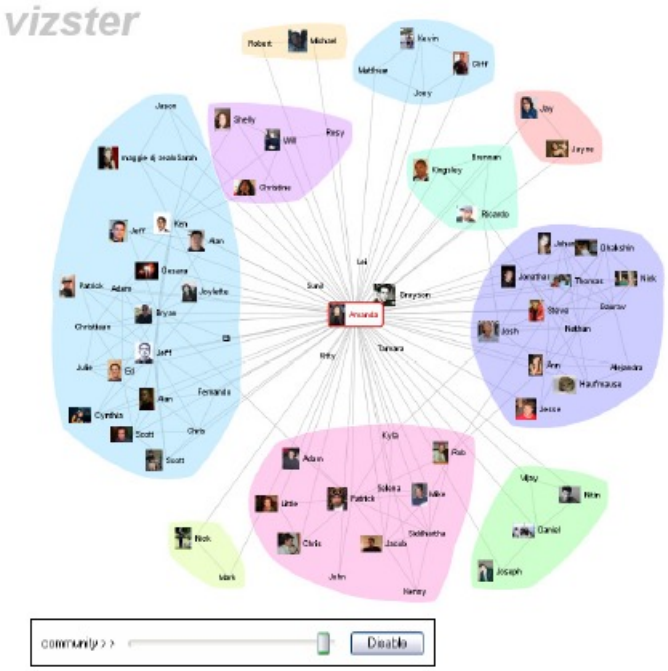
[M. Rohrschneider, A. Ullrich, A. Kerren, P. F. Stadler, and G. Scheuermann. Visual Network Analysis of Dynamic Metabolic Pathways. In Proceedings of the International Symposium on Visual Computing (ISVC '10), pages 316-327, volume 6453 of LNCS, Las Vegas, Nevada, USA, 2010. Springer.]

[Video: https://ivis.itn.liu.se/courses/resources/anim_1701395903_0_reactionview.mp4]

6.7.2 Social Sciences

- Visualizing online social networks



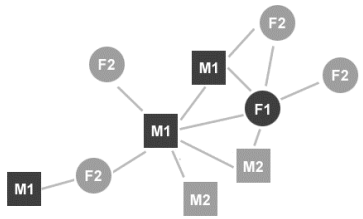


J. Heer and D. Boyd, InfoVis `05

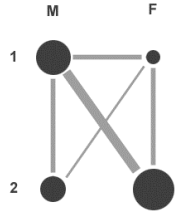
[Video: https://ivis.itn.liu.se/courses/resources/vizster.mp4]

6.8 Attribute-based Visualization

- Here, the attributes are the first-class citizen for the visual network analysis and design, and not the topology of the network, i.e., the nodes and edges
- There are several types, such as
  - **Semantic Substrates** (shown in more detail next)
    - Network nodes are placed in specific regions, see example on the next slide
  - Attribute-driven Layouts
    - Control the placement of a node in the graph layout by considering the node's attributes (not necessarily in specific regions)
    - PivotGraph, e.g., uses a grid layout for showing interactions between attributes →



Node and Link Diagram



PivotGraph Roll-up

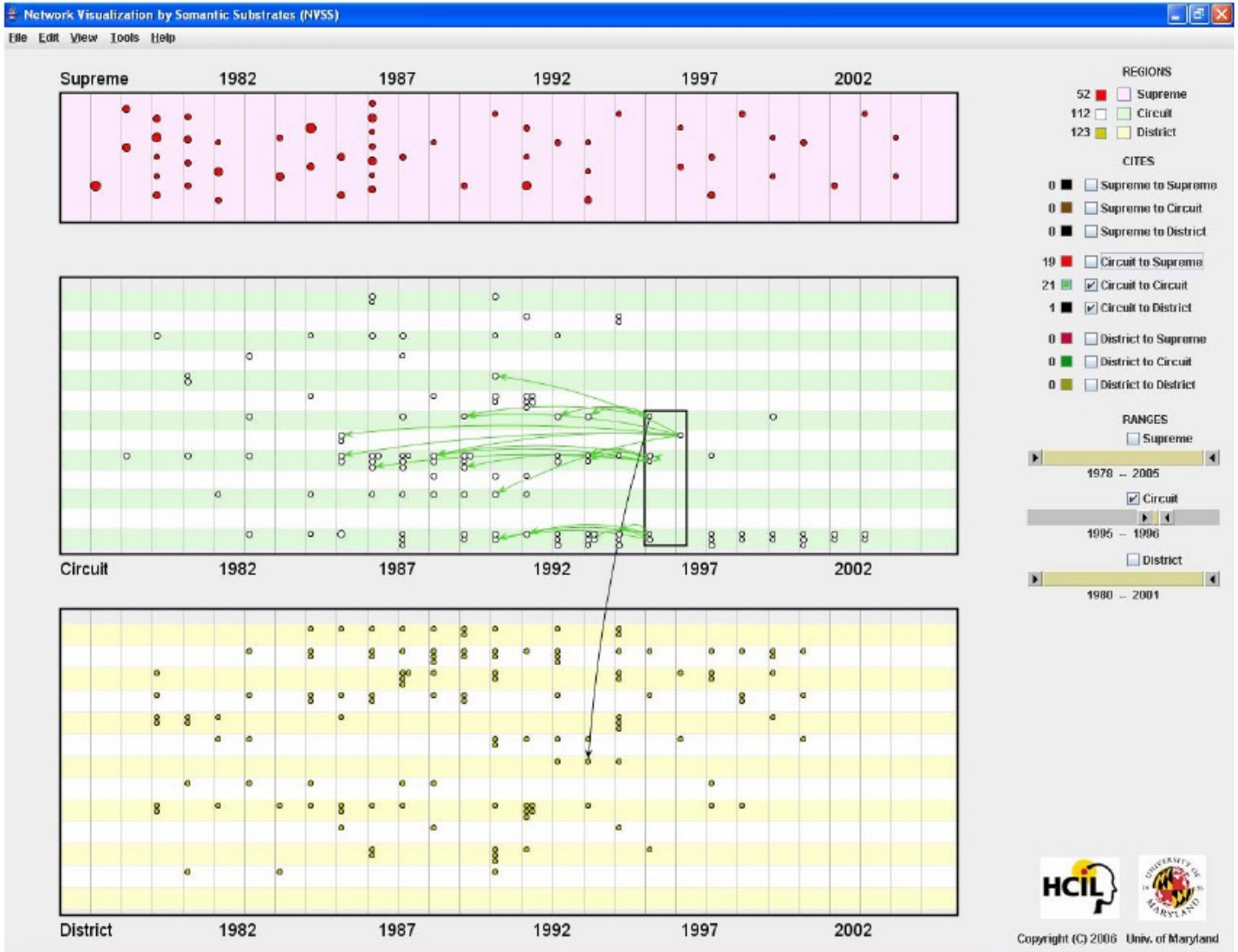
https://doi.org/10.1145/1124772.1124891

6.8.1 Semantic Substrates

- Network visualization through so-called *Semantic Substrates*
  - [B. Shneiderman and A. Aris. Network Visualization by Semantic Substrates. IEEE Transactions on Visualization and Computer Graphics, 12(5), pp. 733-740, 2006]
- Idea
  - Layout is based on user-defined semantic substrates
    - Non-overlapping regions for nodes
    - Node positioning is dependent on the attributes
  - Slider in order to control the visibility of the edges. Thus, it is possible to simplify the edge clutter

6.8.1 Semantic Substrates

- Each region corresponds to a level of jurisdiction in the legal system of the US
- Nodes corresponds to the different cases (1978-2005); Node size corresponds to the number of references on that case
- Edges corresponds to the single references



[Video: https://ivis.itn.liu.se/courses/resources/semanticSubstrates.mpg]