Advanced Data Visualization

CS 6965

Fall 2019

Prof. Bei Wang Phillips

University of Utah

Lecture 21

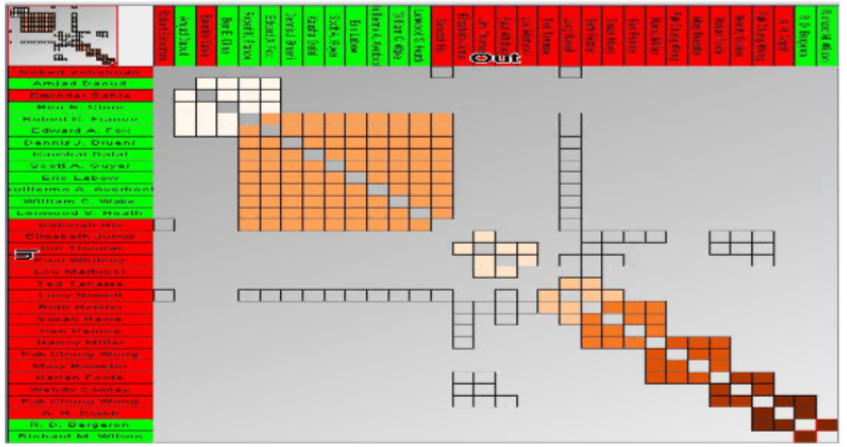
Graph Layout

NV

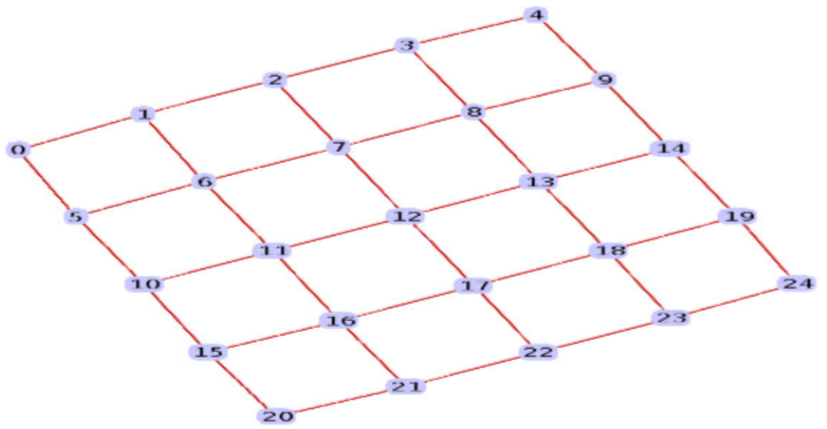
Graph Layout: A Brief Overview

TarawnehKellerEbert2011

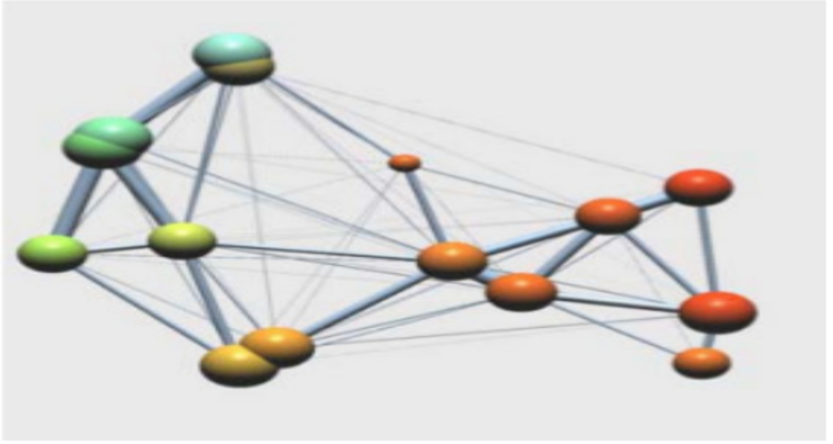
http://drops.dagstuhl.de/opus/volltexte/2012/3748/pdf/13.pdf



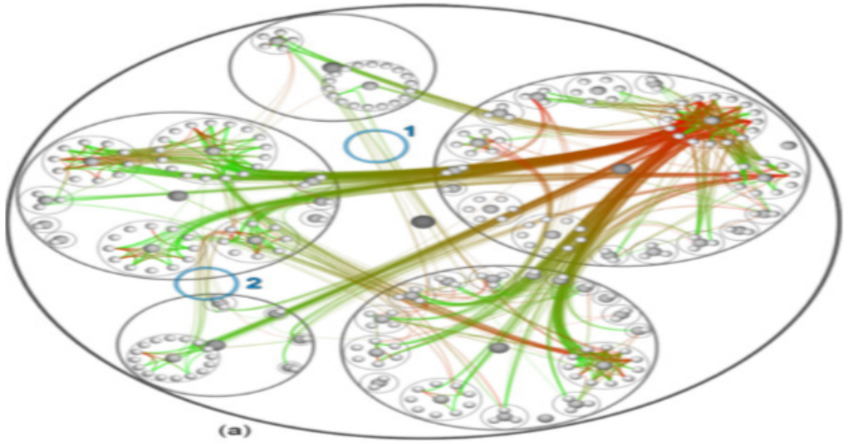
(a) Matrix visualization layout [21].



(b) Example of a force-directed layout.



(c) Clustering example by [54].



(d) Edge bundling example by[23].

TarawnehKellerEbert2011

Graph Vis: layouts + interactions

1. Node-Link Layouts
   1. The Spring Layout Algorithm: Force-directed layouts
   2. Topological Feature-Based Layout
   3. Planar Graphs
2. Tree Layout
   1. Node-Link Tree layout Algorithms
   2. Space-Filling Techniques
3. Matrix Visualization
4. 3D Layout
5. Nodes and Edges Clustering

- Interaction Techniques
  - Zooming and Panning
  - Focus+Context Techniques

TarawnehKellerEbert2011

1

Node-Link Layouts

Criteria and examples

Desirable criteria:

Computation of the coordinates of the nodes and the representation of the lines:

- Nodes and edges should be evenly distributed.
- Edge-crossings should be minimized.
- Depict symmetric sub-graphs in the same way.
- Minimize the edge bending ratio.
- Minimize the edge lengths, which helps readers detecting the relations among different nodes faster.
- In cases where the data is inherently structured, distribute the nodes into different layers. This increases the understandability of the underlying graph.

TarawnehKellerEbert2011

Topological Feature-Based Layout

Pipeline contains 4 phases:

1. Decomposition: graph is decomposed into many sub-graphs based on the topological features of each internal sub-graph.
   - E.g. if the nodes in one sub-graph are topologically connected among each other in form of a tree, then the set of nodes are grouped together representing a meta-node.
   - 7 topological features: trees, complete graphs, bi-connected components, clusters, etc.
2. Feature layout: meta-nodes (or grouped sub-graphs) are laid out.
3. Crossing reduction: eliminate the crossing ratio in the produced layout.
4. Overlap elimination: change the node sizes in the final layout to ensure that no nodes overlap each other.

ArchambaultMunznerAuber2007

Topological Feature-Based Layout

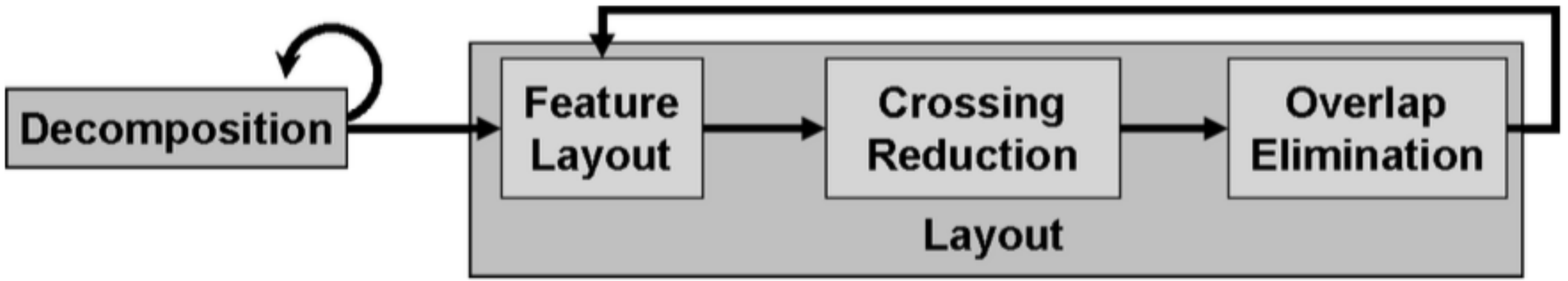


Fig. 1. TopoLayout algorithm phases.

ArchambaultMunznerAuber2007

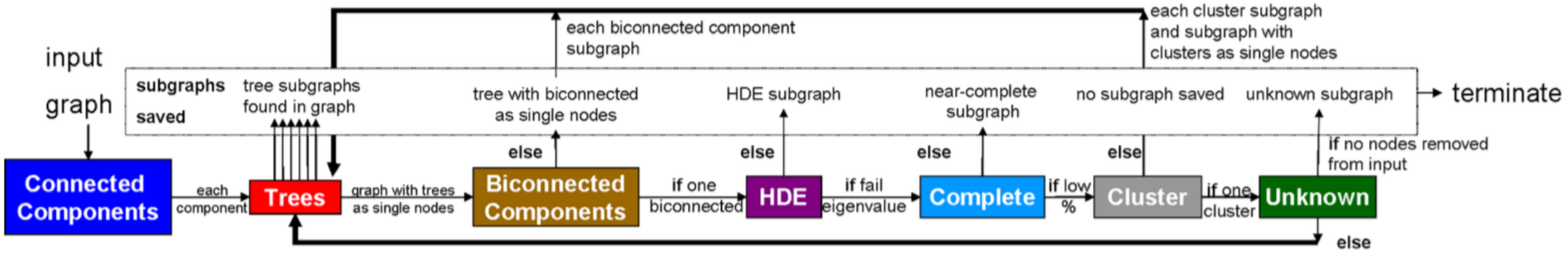


Fig. 3. Decomposition phase for TopoLayout. Detection algorithms in boxes coloured by feature type as in Figure 2. If a clause on a horizontal is true, we transition along the arrow. Otherwise, we follow the vertical arrow to save some subgraphs and recursively decompose others. Bold arrows indicate the recursive cases.

ArchambaultMunznerAuber2007

Feature Hierarchy

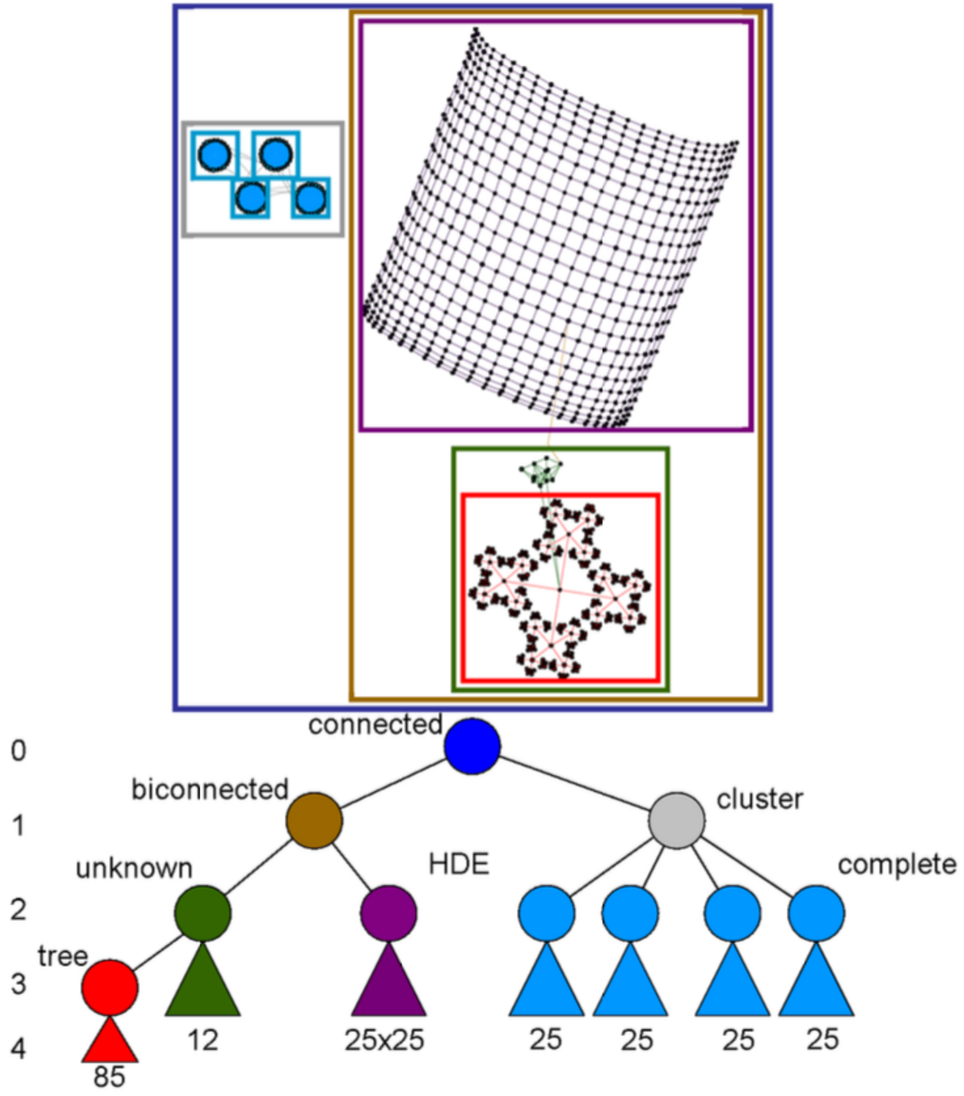
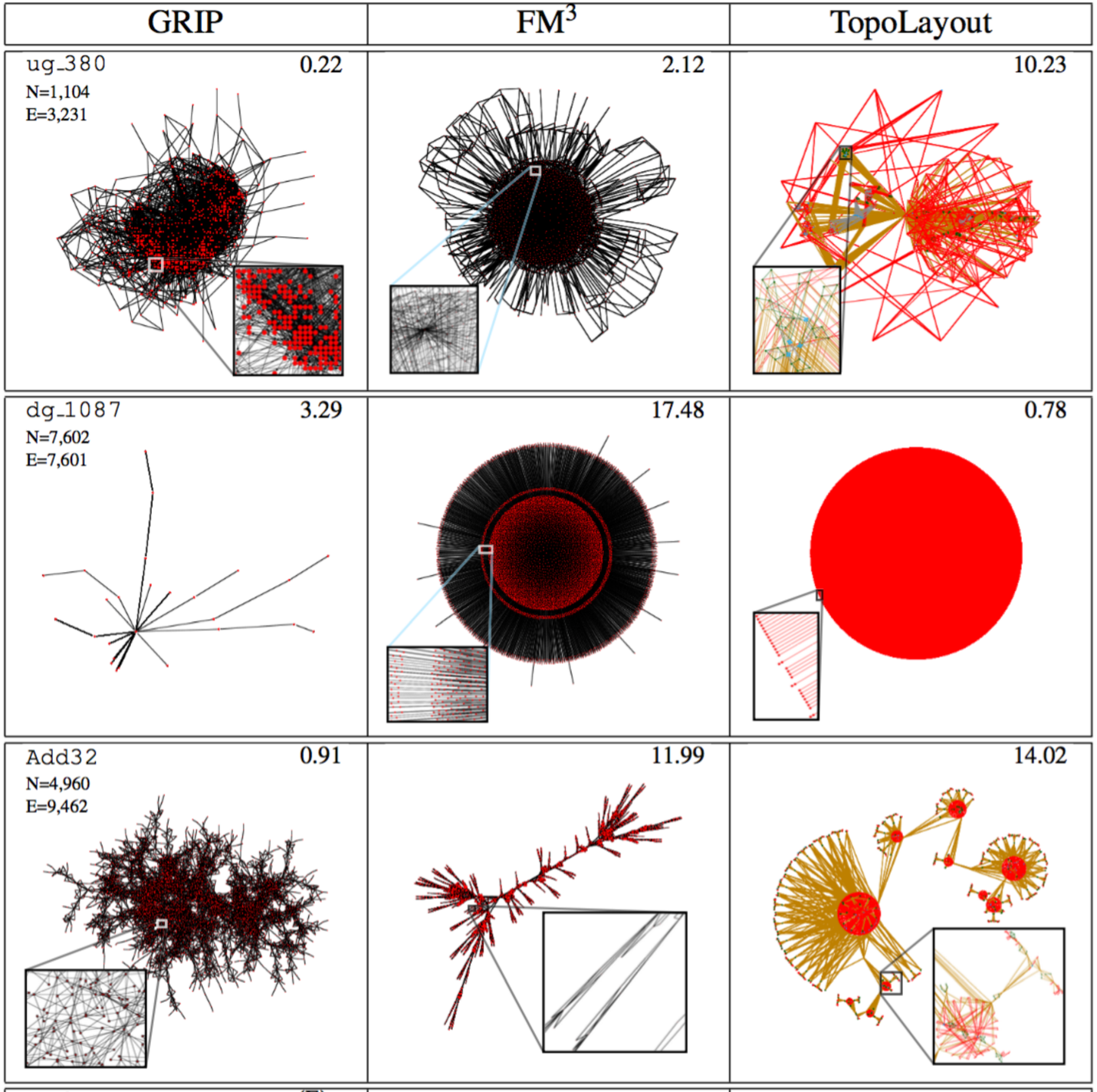


Fig. 2. Feature hierarchy after decomposition, with topology encoded by colour. **Top**: Layout annotated with bounding boxes to show hierarchy structure: meta-nodes encompass the subgraphs of their children. **Bottom**: Diagram of feature hierarchy, with levels enumerated and nodes labeled by feature type.

ArchambaultMunznerAuber2007



ArchambaultMunzner
Auber2007

Discussions

- TopoLayout Extensions?
- The power of flexibility
- Feature based graph analysis and visualization

Challenge: directed graph layout

Two step approach:

- First, layer the graph nodes: assign a layer for each node and placing the nodes into the corresponding layer.
- Second, reduce the edge-crossings and the node overlapping.

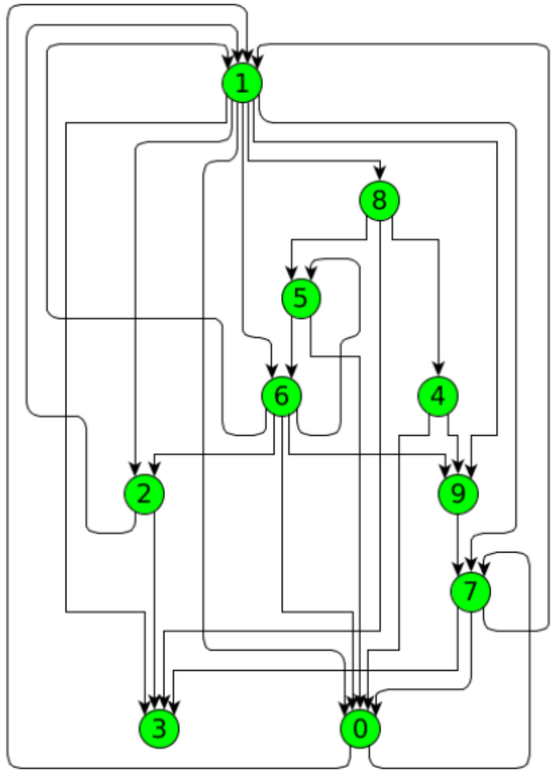
SugiyamaTagawaToda1981

L-Drawing of directed graphs

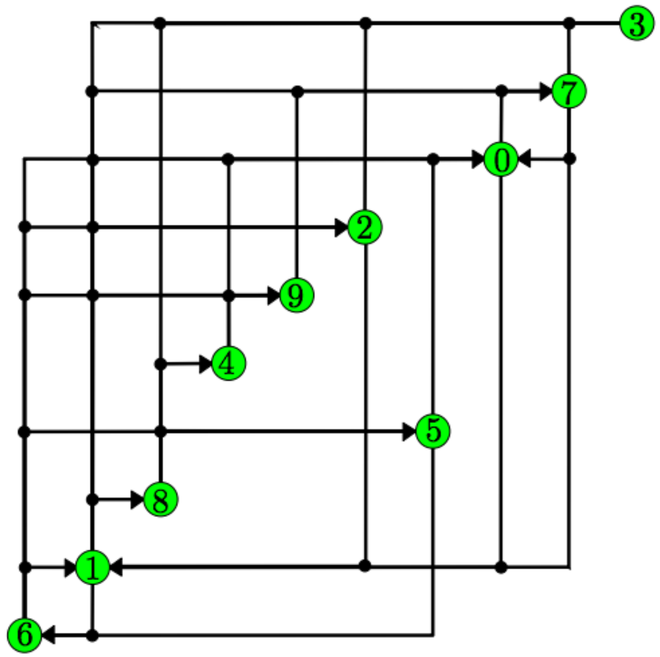
Combine Orthogonal Graph Drawing with matrix representation:

- Each vertex has exclusive x- and y-coordinates and each directed edge has two orthogonal segments, one leaving the source vertically and one entering the destination horizontally.
- Edges are allowed both to overlap and to intersect.
- Graphically, the joint between the horizontal and the vertical segment of an edge is drawn as a small circular arc, allowing the user to easily identify the edges even in the presence of overlaps and intersections
- The matrix is symbolically represented by the edges, that identify the portions of the rows and columns that have to be followed to connect adjacent vertices

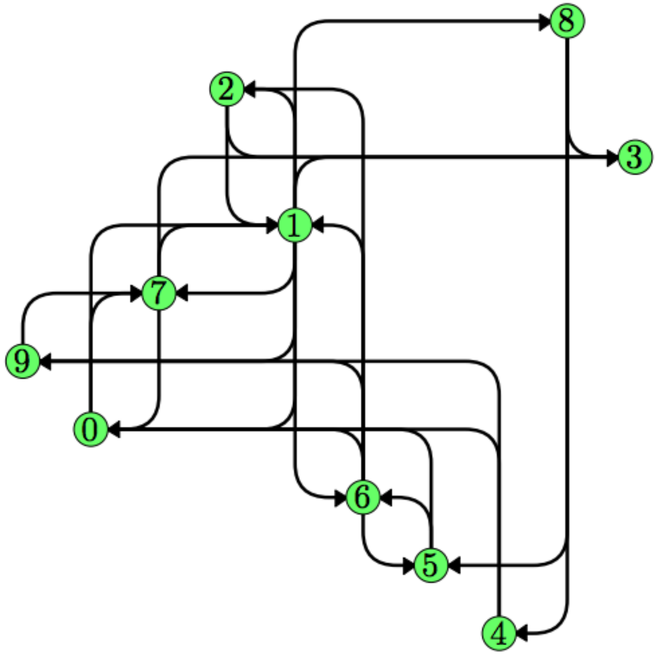
L-Drawing of directed graphs



(a)



(b)



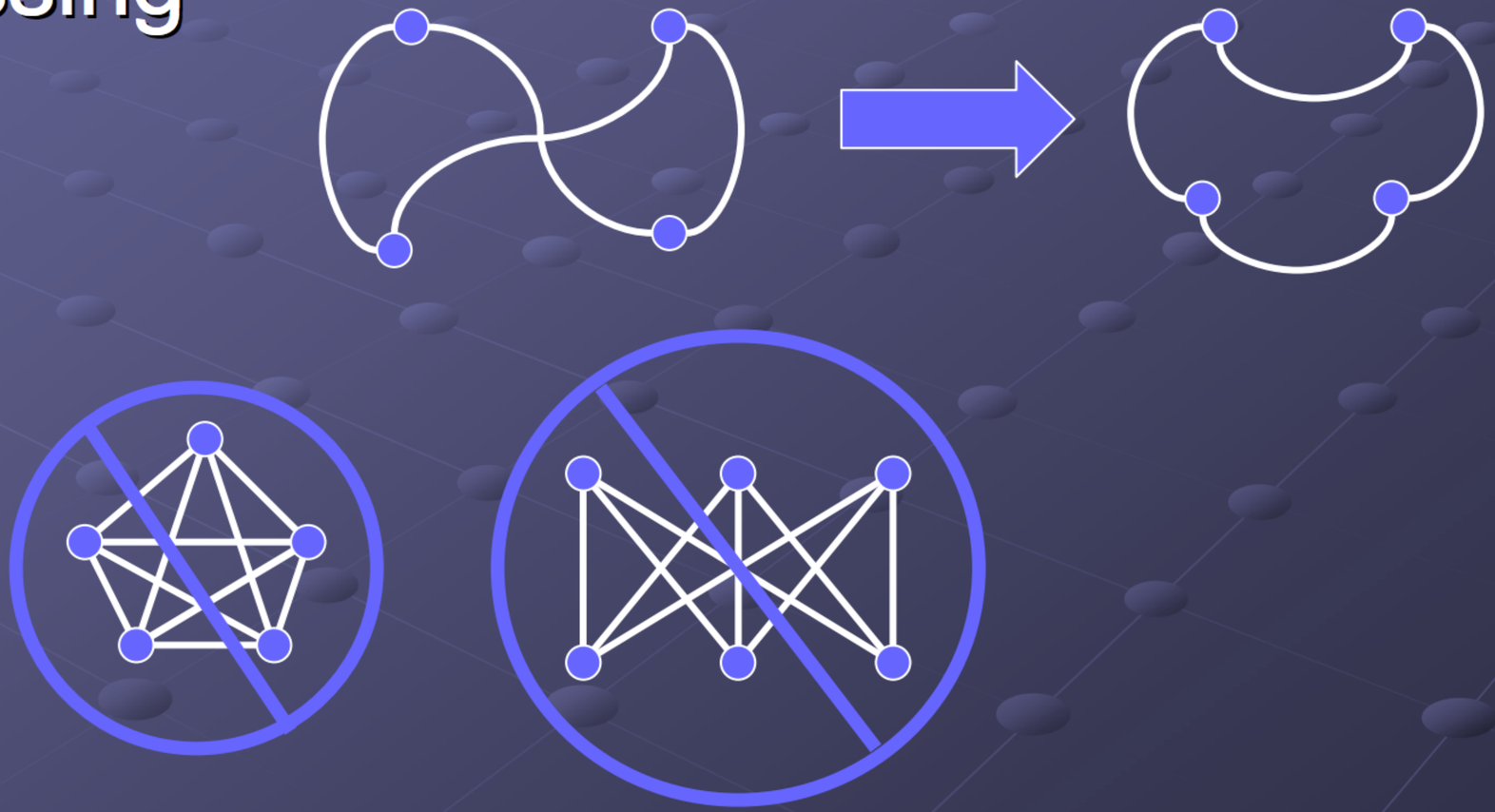
(c)

AngeliniLozzoBartolomeo2015

Planer Graph Layout

- Graphs that can be drawn without edge crossings
- Pre-requisites:
  - Testing whether it is possible to draw the given graph without edges crossings or not.
  - Finding a planar layout algorithm satisfying the required application constrains.
- Schnyder's Algorithm

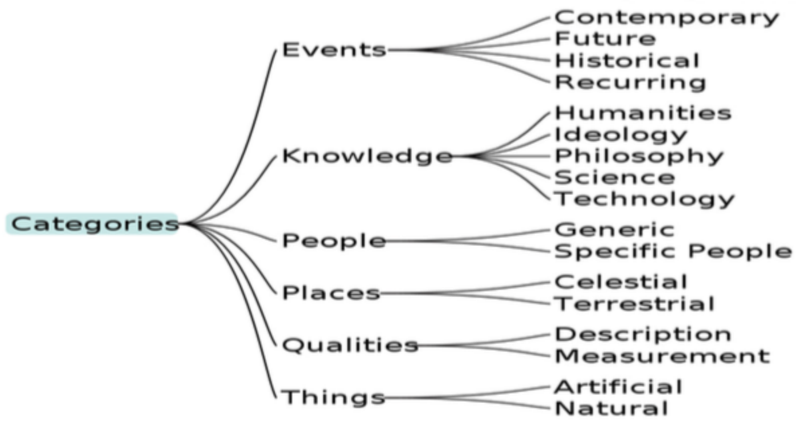
- A Graph G=(V,E) is a Planar Graph if it can be drawn in the plane with no edges crossing



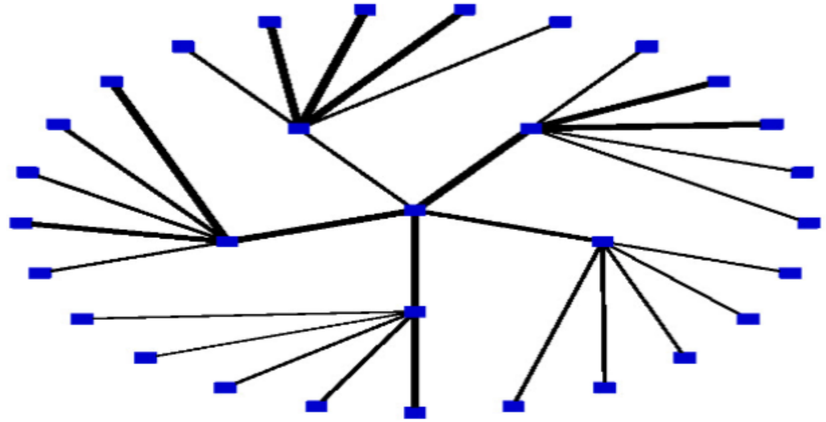
https://pdfs.semanticscholar.org/9c89/7e65499cc6caacabd8abab6071010b03248c.pdf
Slides by Jon Harris

2
Tree Layouts

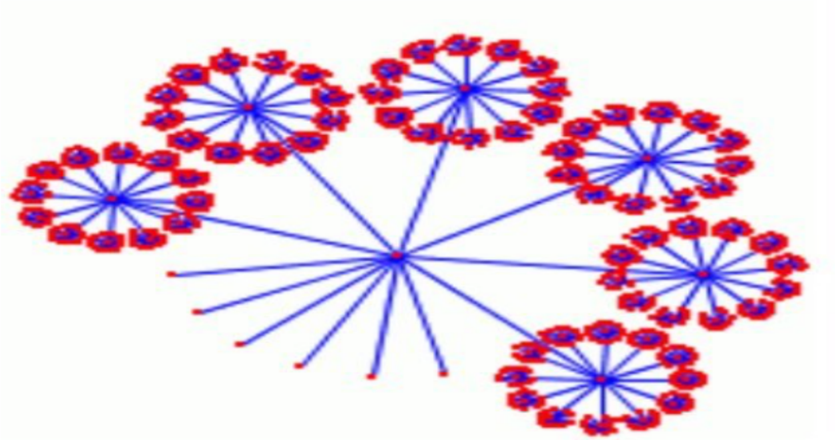
Node-Link Tree Layout



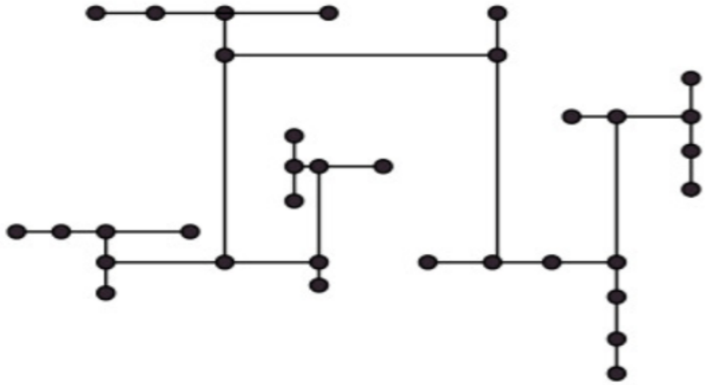
(a) Classical tree layout, produced with [19].



(b) Radial tree layout Example.



(c) Balloon tree layout: produced by [22].

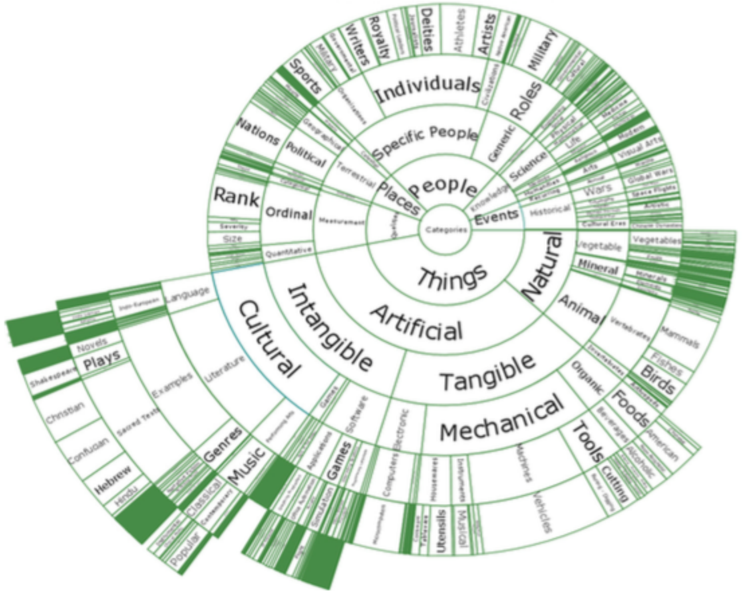


(d) H-Tree layout: produced by [22].

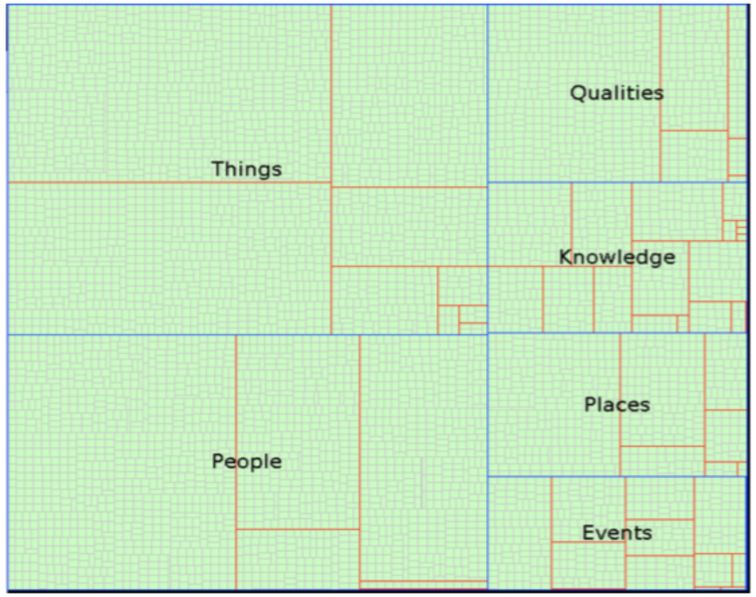
Figure 2 Tree Layout Examples.

TarawnehKellerEbert2011

Space-Filling



(a) SunBurst layout.



(b) TreeMap layout.

Figure 3 Examples of space-filling techniques [19].

TarawnehKellerEbert2011

3
Matrix Visualization

MatrixExplorer

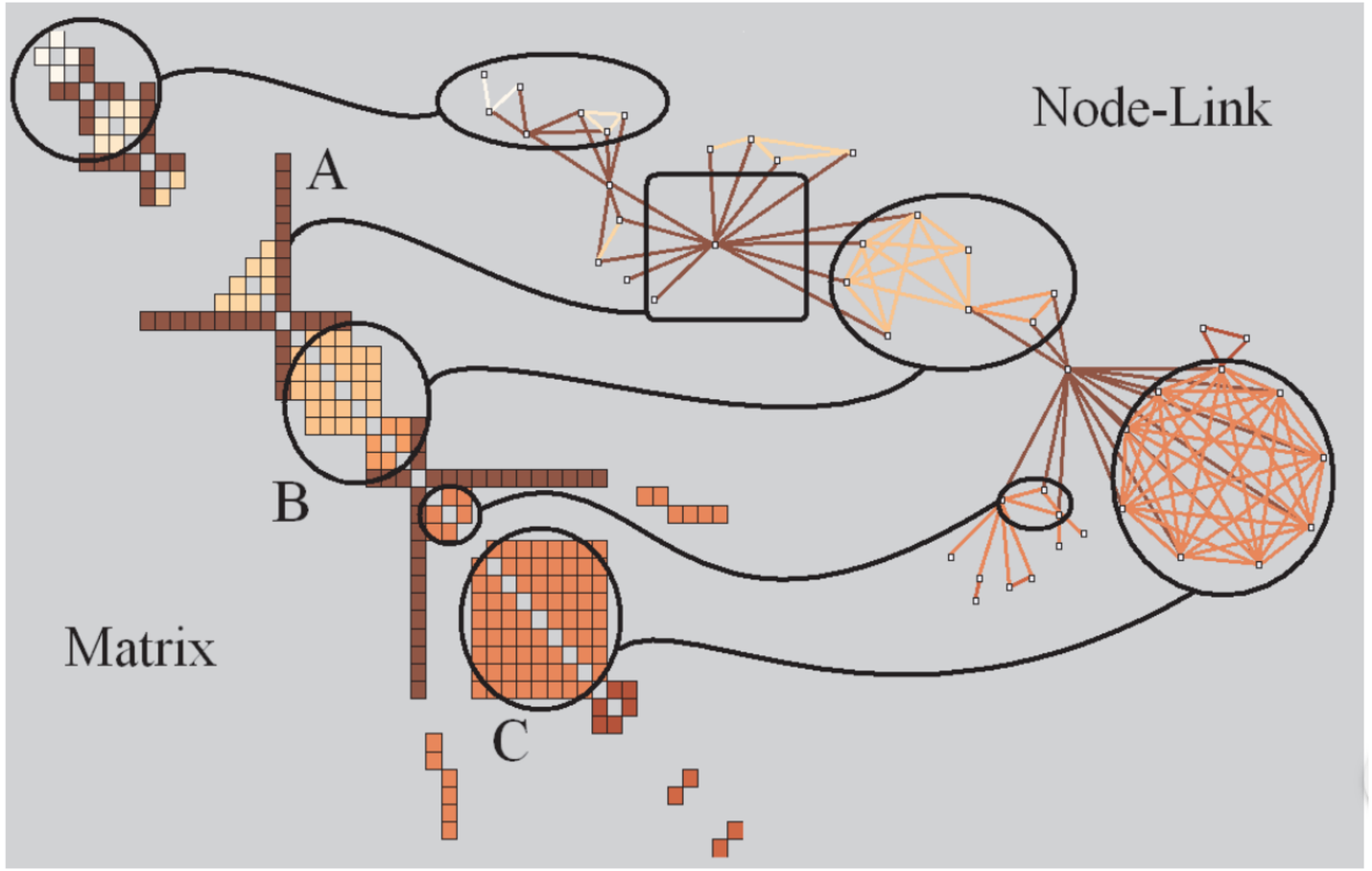


Fig. 3. Visual patterns in Matrix and Node-link representations of social networks. A represents an actor connecting several communities, B a community and C a clique (complete sub-graph).

HenryFekete2006

4

3D Layout

TreeCube

Extension from TreeGraph

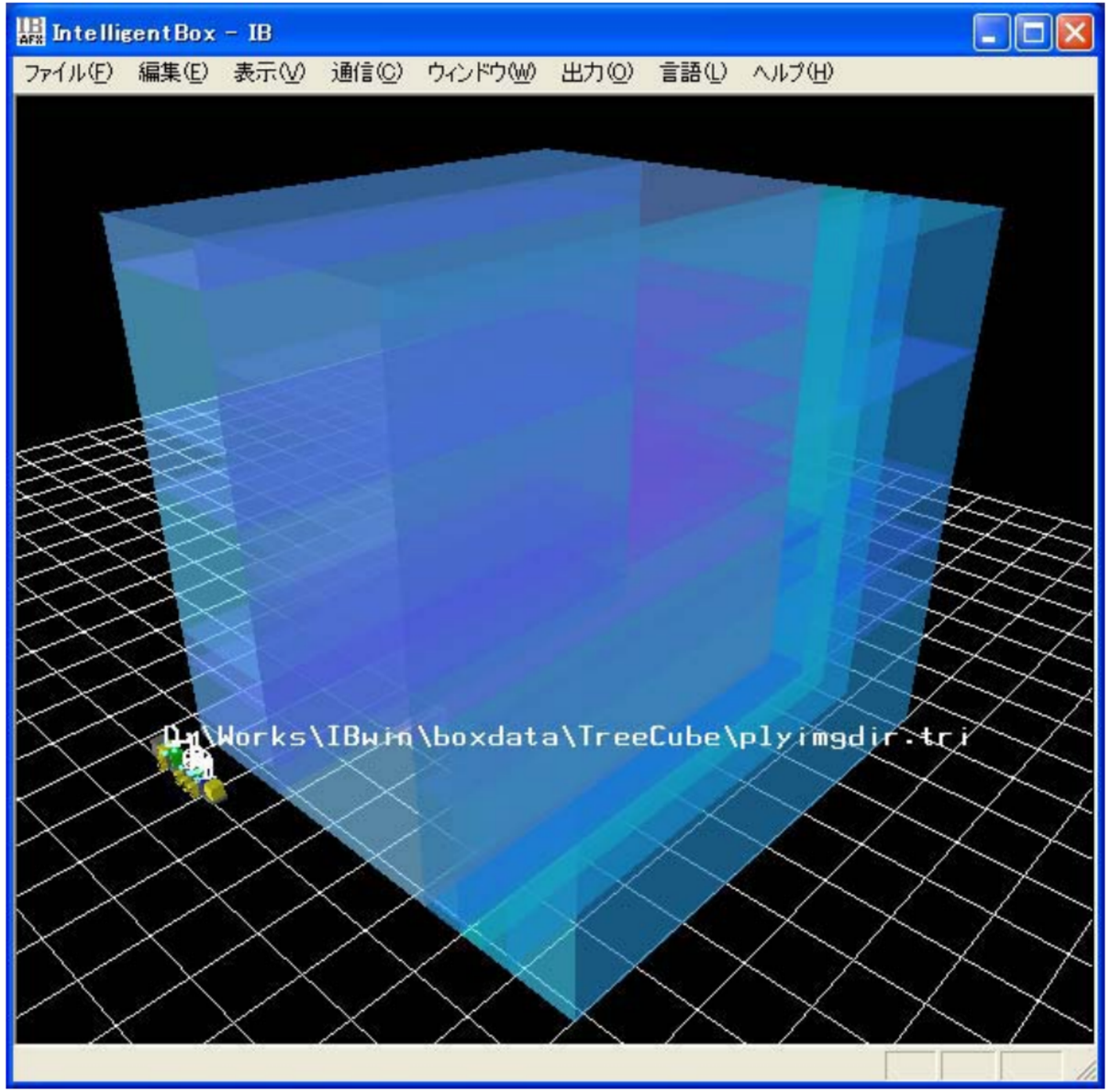


Fig. 2: Layout example of hierarchical information by the slice-and-dice treecube algorithm.

TanakaOkadaNiijima2003

True Cube

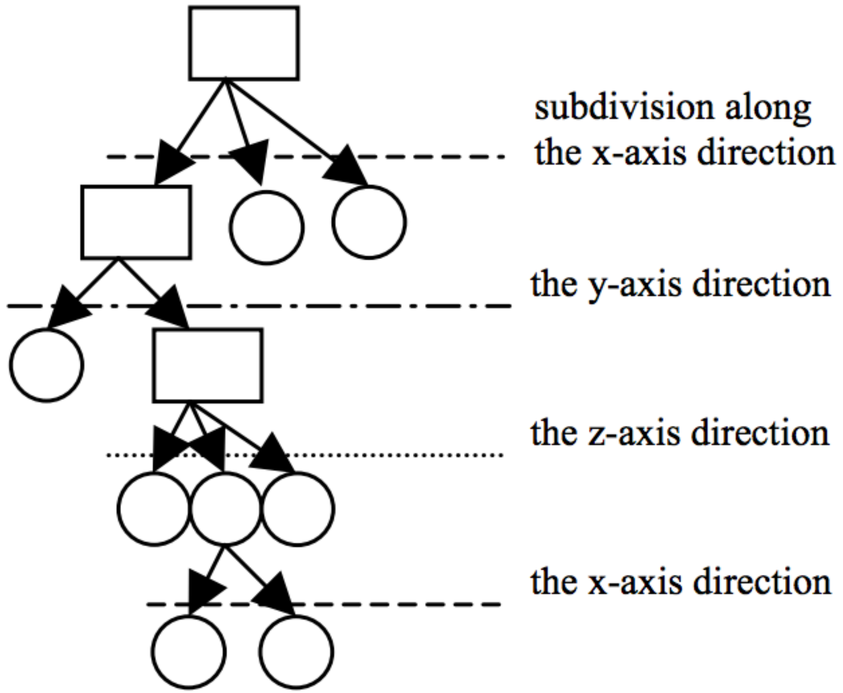


Fig. 1(a): Hierarchical information and the direction of the subdivision operation in its each level.

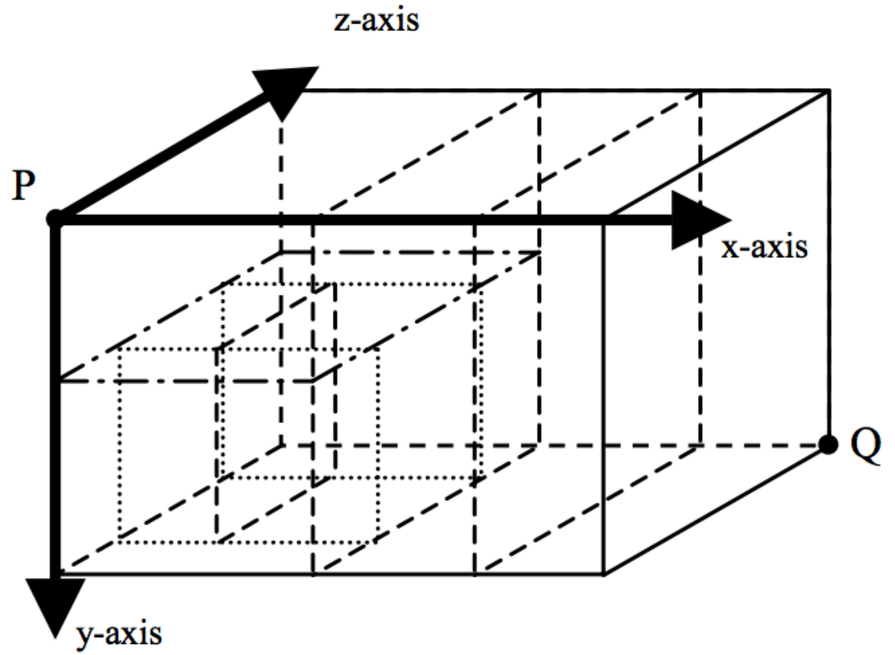
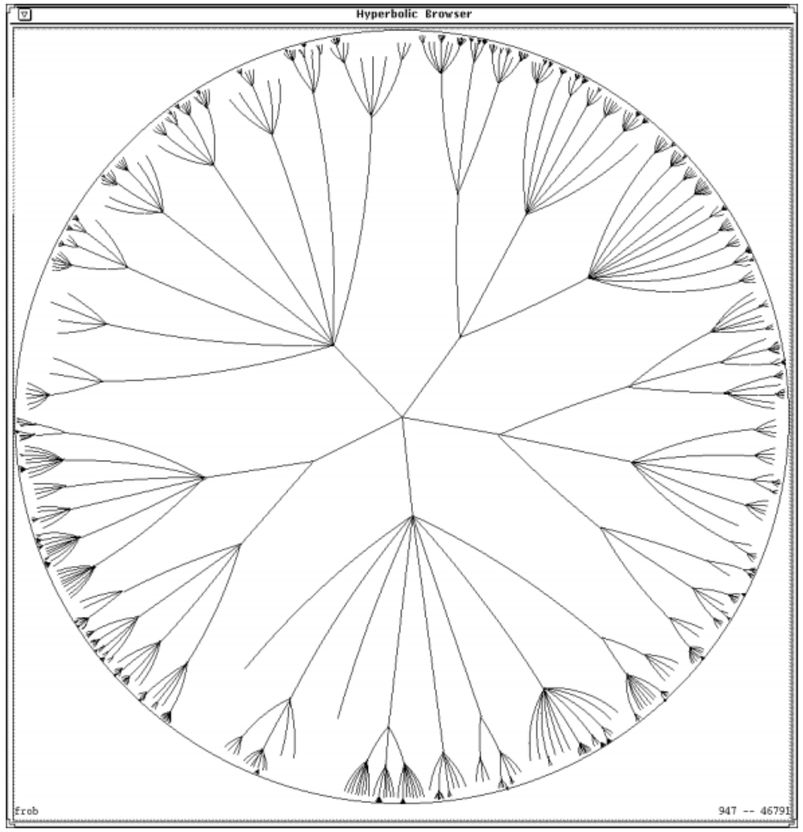
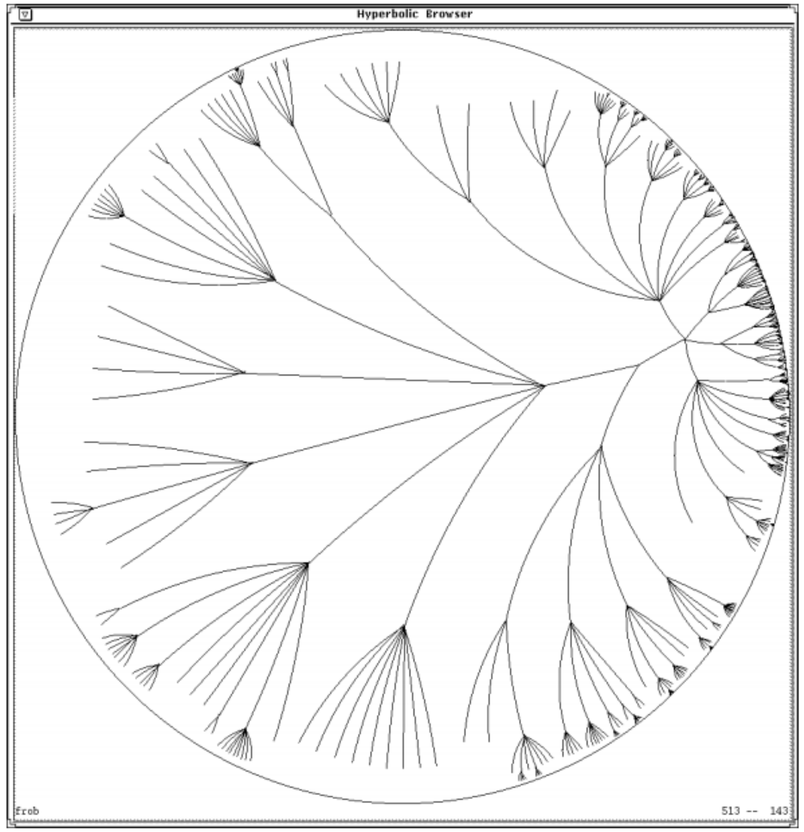


Fig. 1(b): Slice-and-dice treecube layout of the hierarchical information shown in Fig. 1(a).

TanakaOkadaNiijima2003

Hyperbolic graph layout

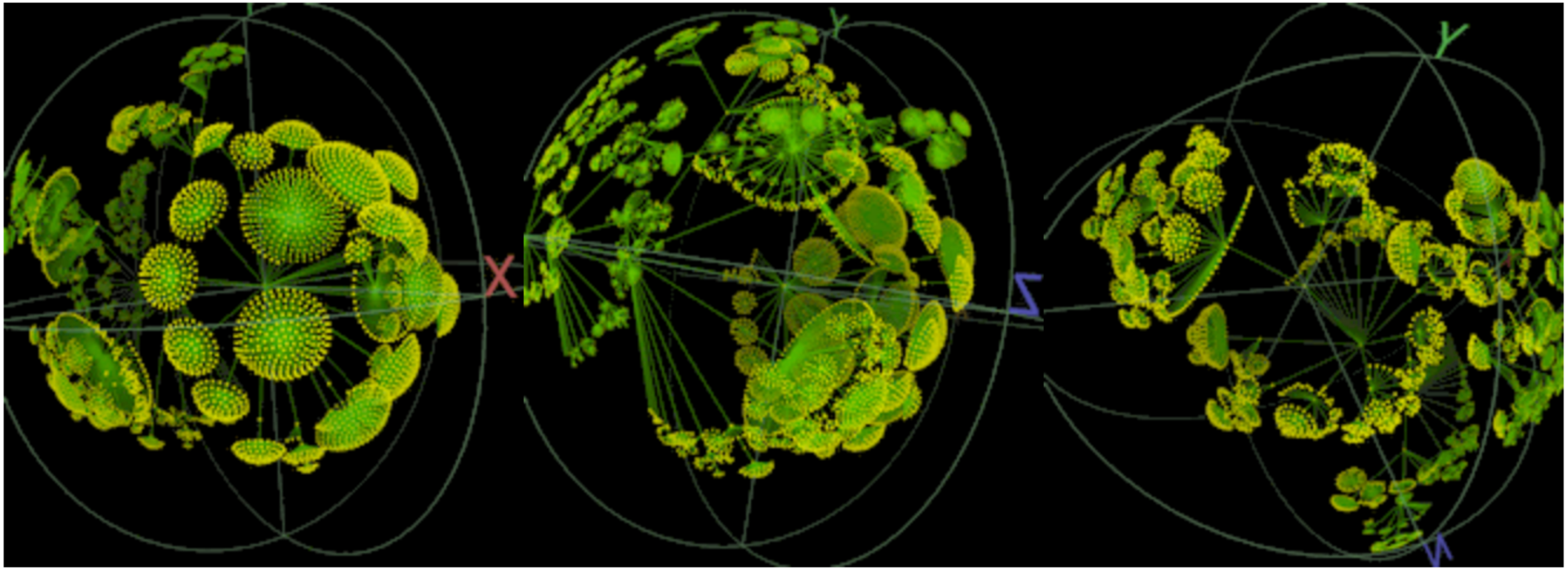




LampingRaoPirolli1995

(3D) Hyperbolic graph layout

CVS Repository (18,474 nodes and 18,473 links)



http://www.caida.org/tools/visualization/walrus/

Immersive Graph Visualization

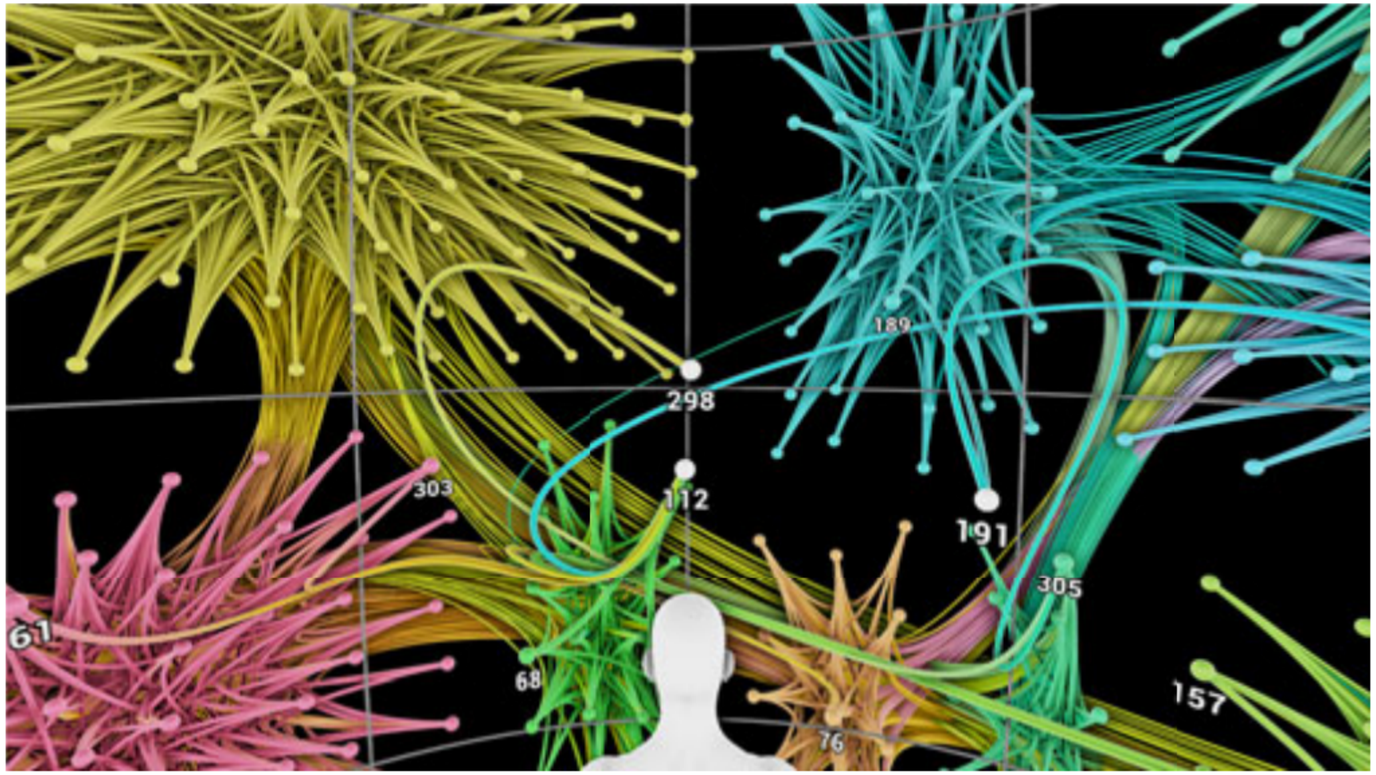
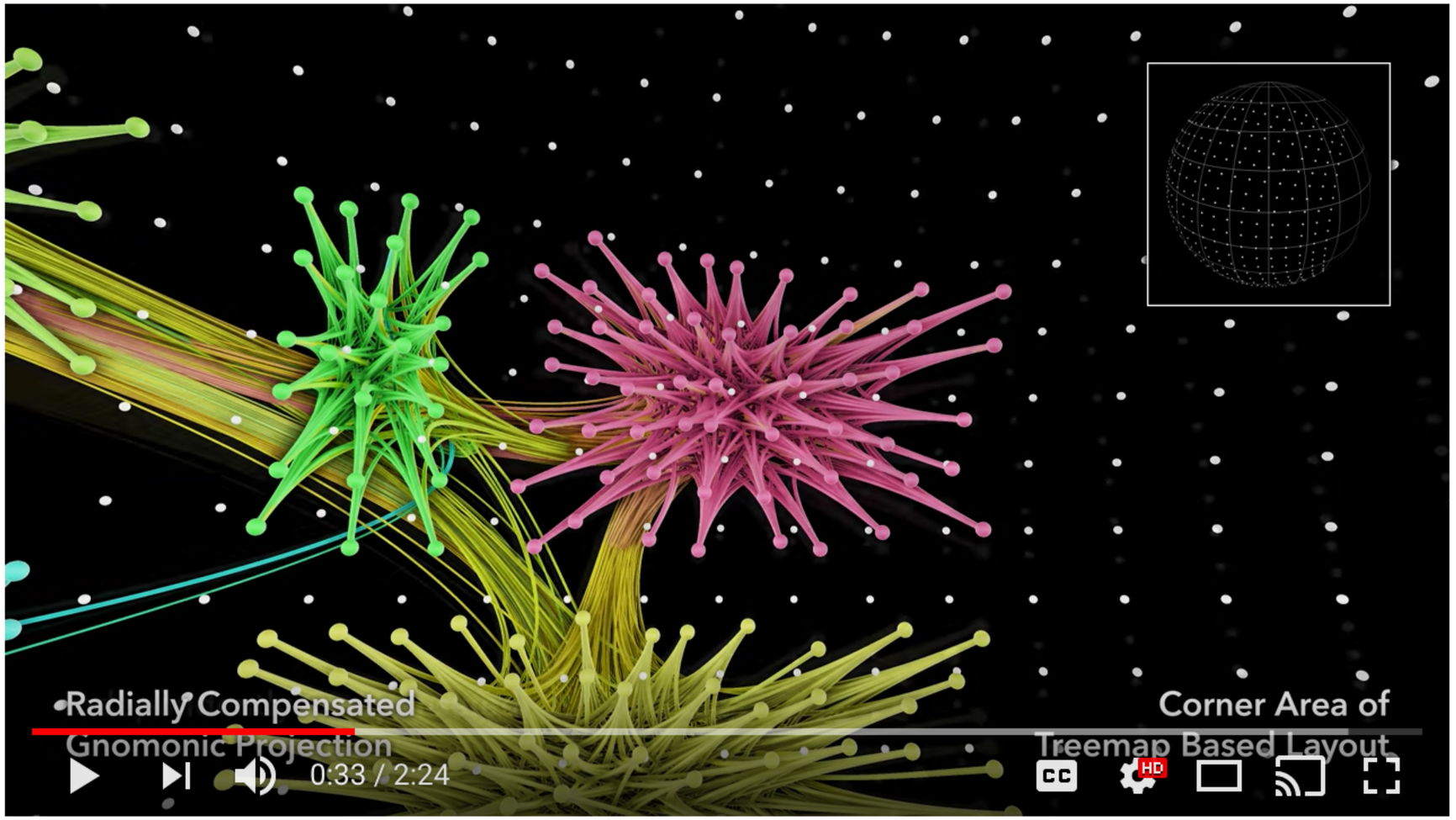


Fig. 1. The viewer is placed at the center of the sphere, on which the graph is laid out.

https://www.youtube.com/watch?v=LQYamaU8OvA

KwonMuelderLee2016

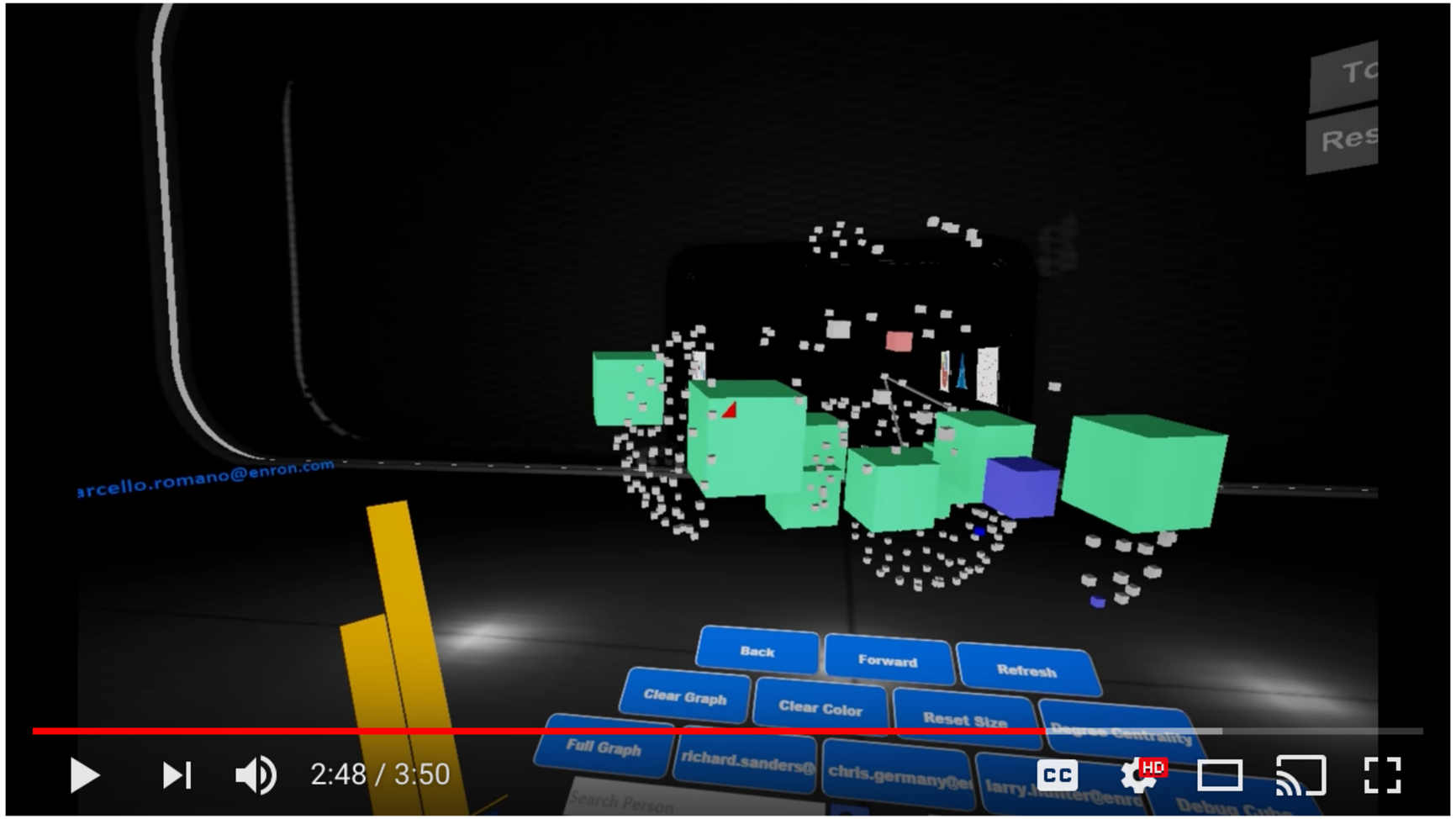
Immersive Graph Visualization



https://www.youtube.com/watch?v=LQYamaU8OvA

KwonMuelderLee2016

VR Graph Vis



https://www.youtube.com/watch?v=vsMTdd12NB0

5

Node and edge clustering (more on this later)

Interacting with Graph layout

General Interactions

- Selecting: highlight and process specific objects
- Abstracting/Elaborating: change the level of detail of the representation.
- Reconfiguring: change the layouts for the same representation.
- Encoding: switch between different layout methods.
- Exploring: change the view point of the graph layout, e.g. zooming and panning.
- Filtering: remove unnecessary detail, filter the nodes based on their attributes.
- Connecting: highlight the paths between relevant objects and the focus object.

TarawnehKellerEbert2011

Zooming and Panning

- Panning: move the camera across the scene
- Zooming: switch between abstract and detailed views
  - Geometric zooming: adjusts the screen transformation, different level of magnification.
  - Semantic zooming: size of objects or displayed information may change when approaching a particular area of the graph.
  - Example: google earth
- Scale space diagram: combine geometric and semantic zooming

Scale Space Diagram

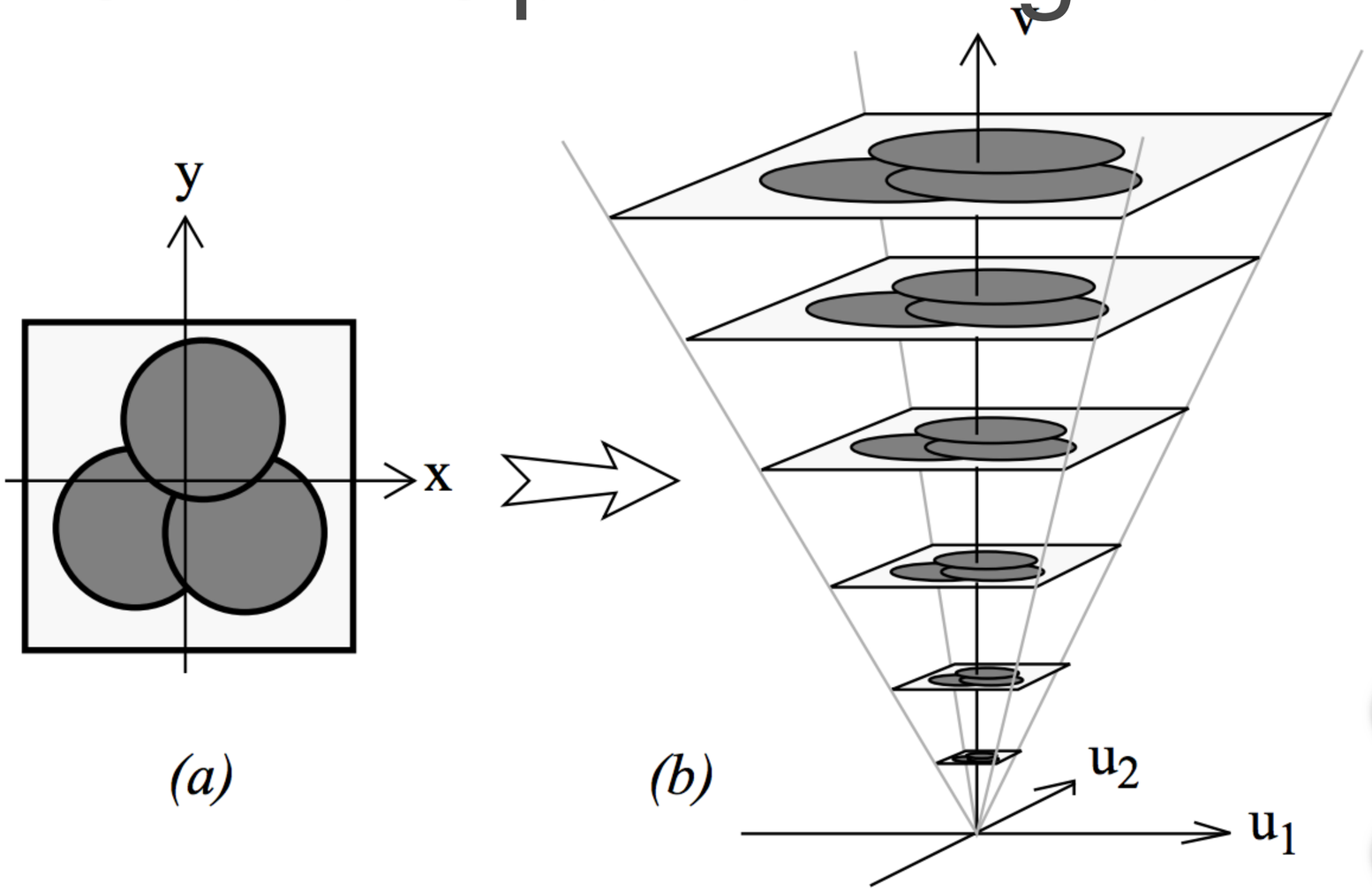


Figure 1. *The basic construction of a Space-Scale diagram from a 2D picture.*

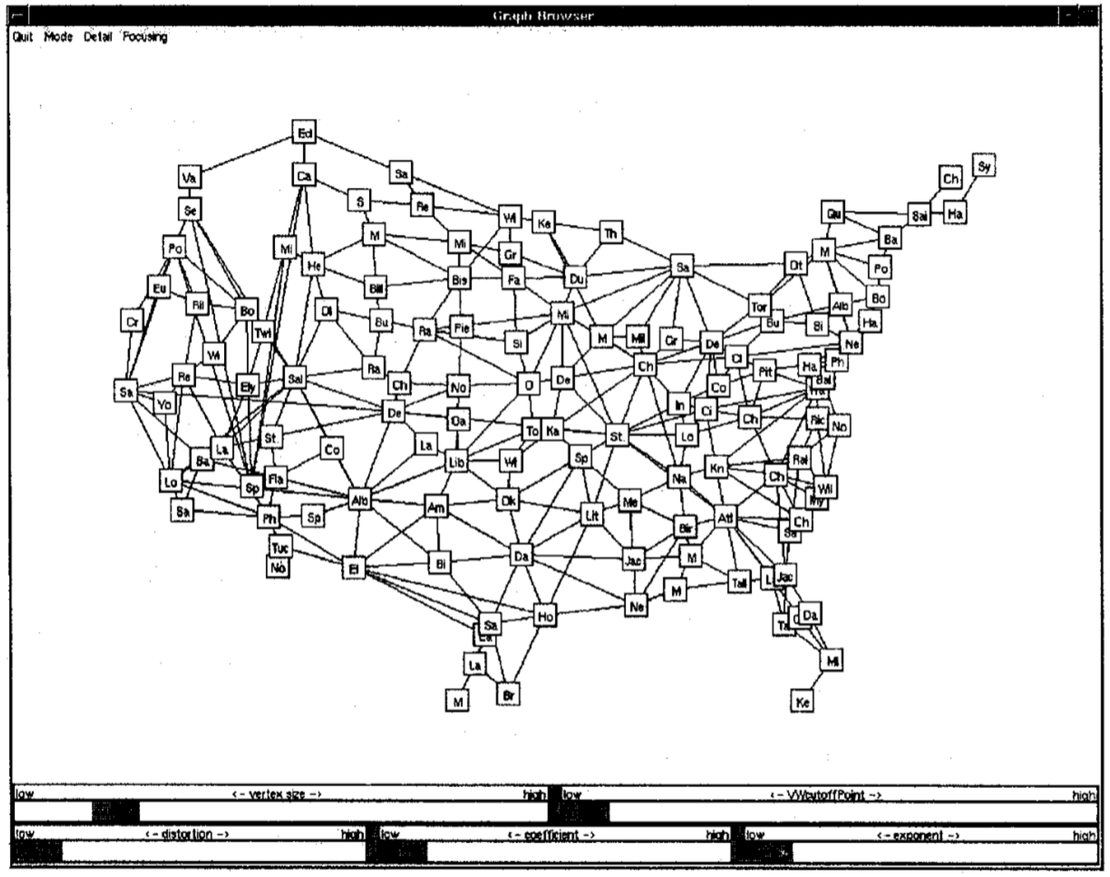
FurnasBederson1995

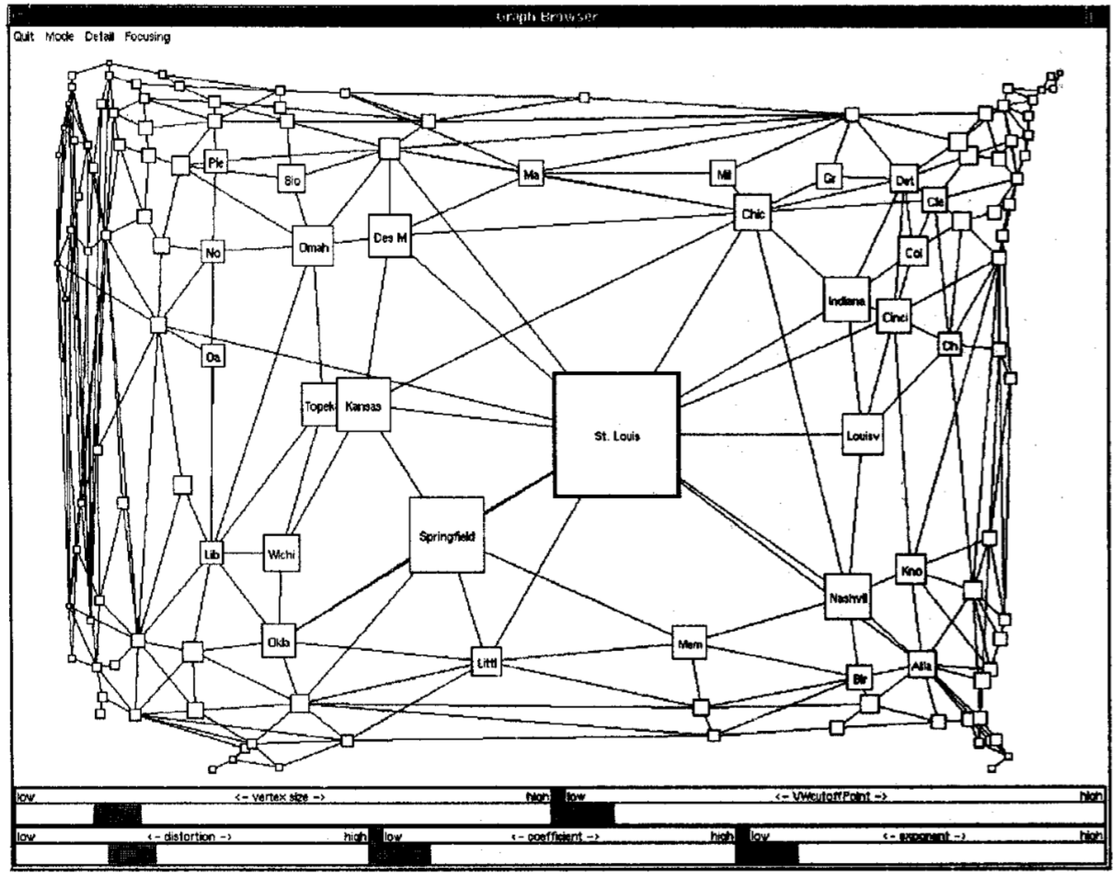
Focus+Context

- Avoid losing context when zooming into given data
- See the primary object in a detailed view (focus) together with an overview of all the surrounding information (context)
- For example: fish-eye
  - The area of interest becomes larger while at the same time the other regions of the layout are successively shown with less detail.

TarawnehKellerEbert2011

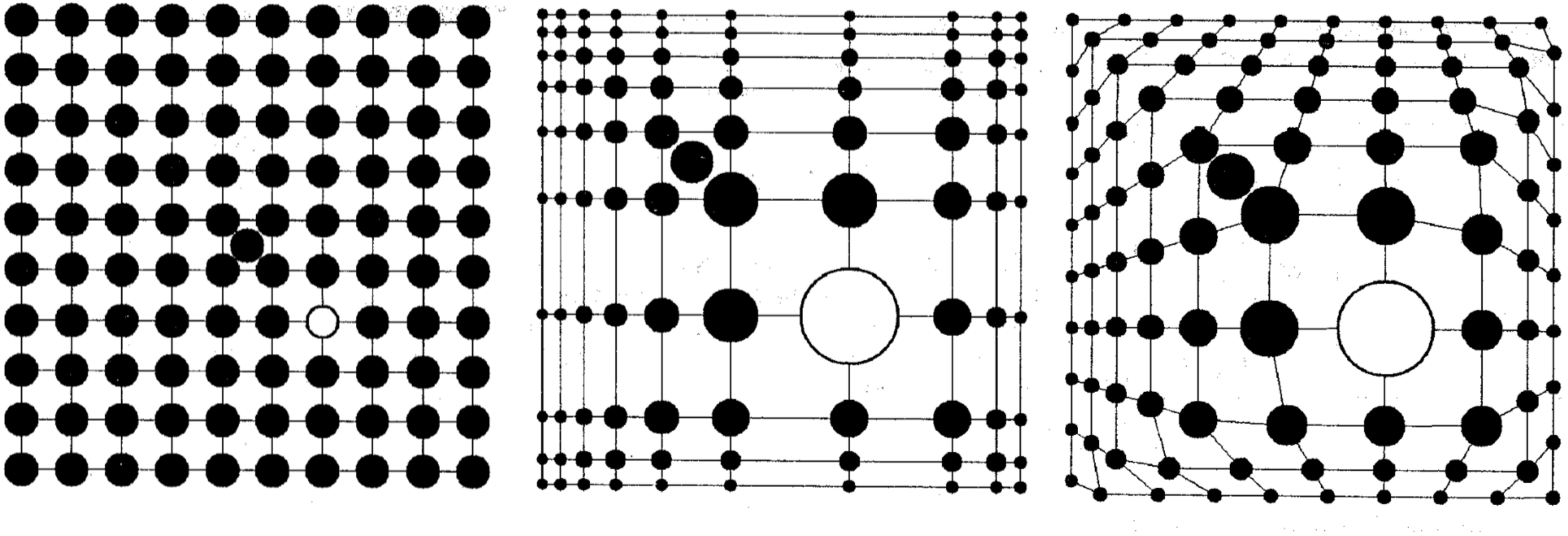
Fish-Eye





SarkarBrown1992

Fish-Eye



Introducing distortion to graphs

SarkarBrown1992

Fish-Eye

Figure 6: Outline of the United States

Figure 7: A cartesian transformation of Figure 6. The focus is at the point where Missouri, Kentucky, and Tennessee meet.

Figure 8: A polar transformation of Figure 6. The focus is at the point where Missouri, Kentucky, and Tennessee meet.

SarkarBrown1992

Topological Fish-Eye View

- Precompute a hierarchy of coarsened graphs that are combined on-the-fly into renderings
- Rendering's level of detail is dependent on distance from one or more foci.
- Geometric distortion yields constant information density displays from these renderings

GansnerKorenNorth2005

Topological Fish-Eye View

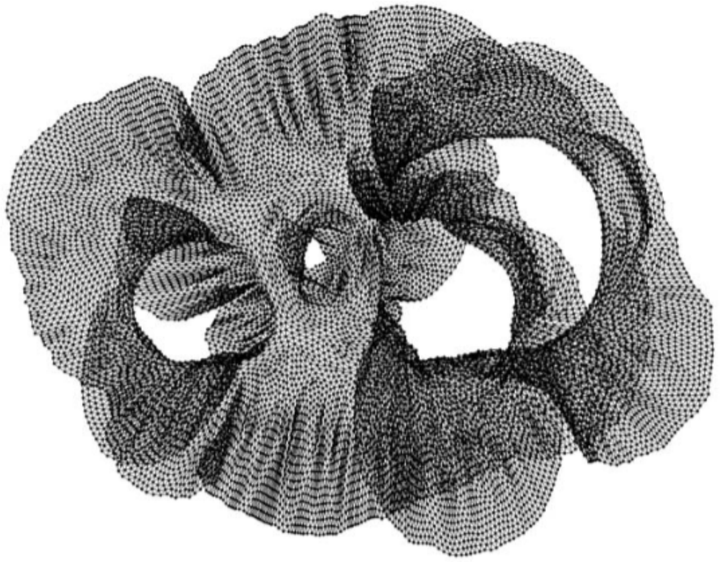


Fig. 1. The 4elt graph, $|\mathrm{V}| = 15,606$, $|\mathrm{E}| = 45,878$.

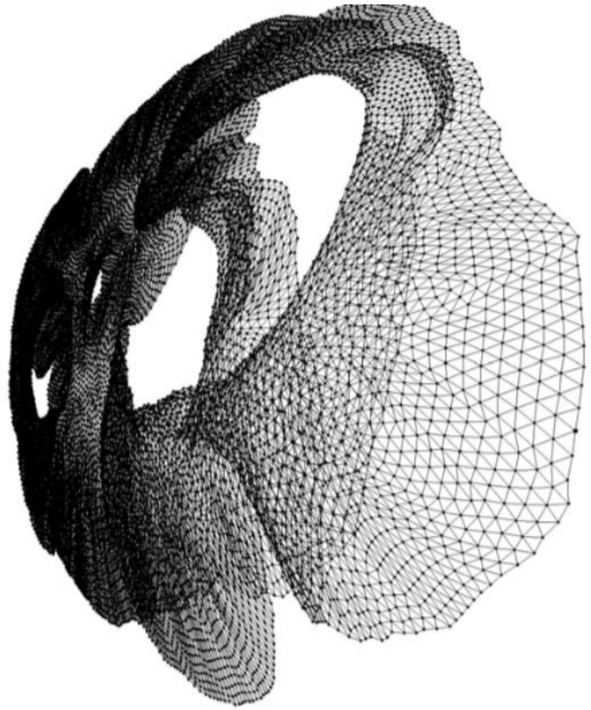
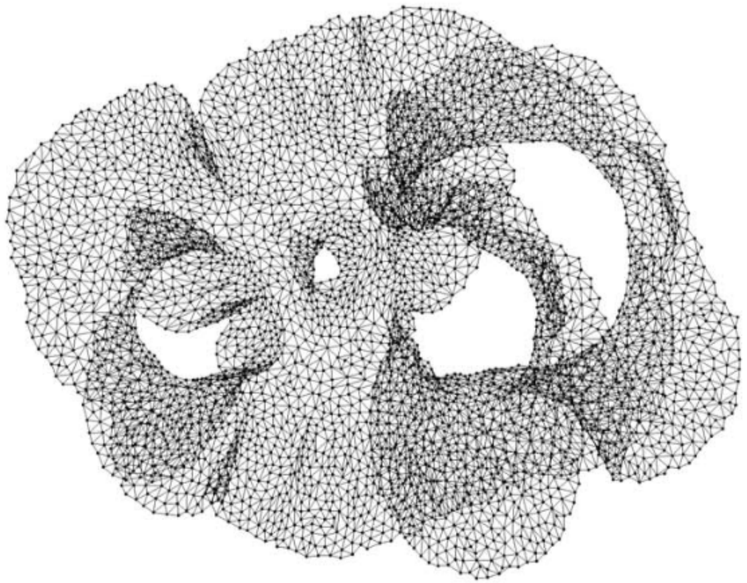


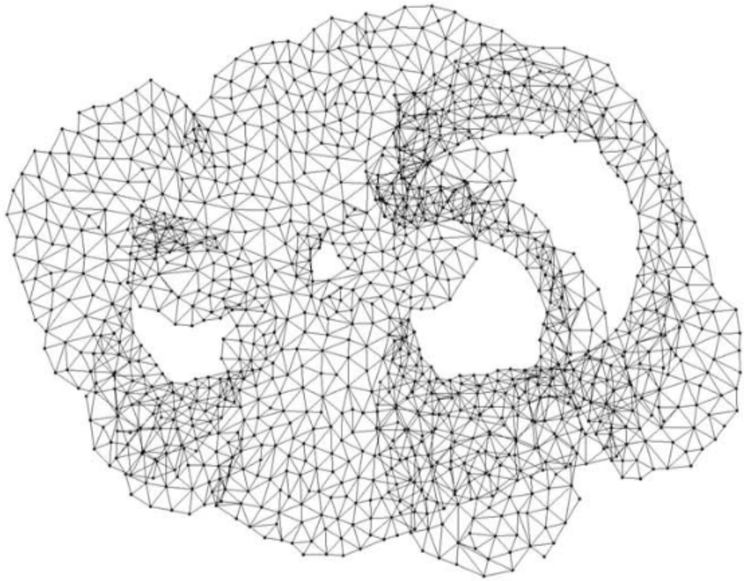
Fig. 2. A fisheye view of the 4elt graph focused on the right-hand portion.

GansnerKorenNorth2005

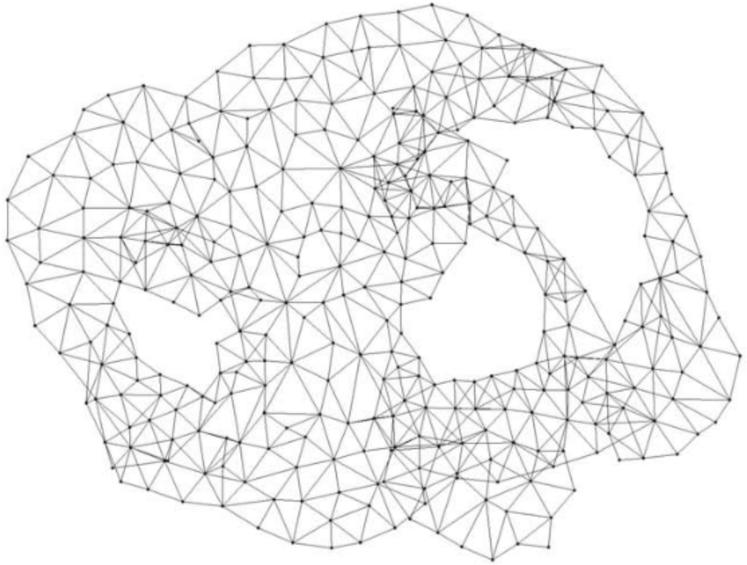
Topological Fish-Eye View



4394-node approximation



1223-node approximation



341-node approximation

Fig. 3. Approximating the 4elt graph at three different scales of decreasing size and accuracy.

GansnerKorenNorth2005

Topological Fish-Eye View

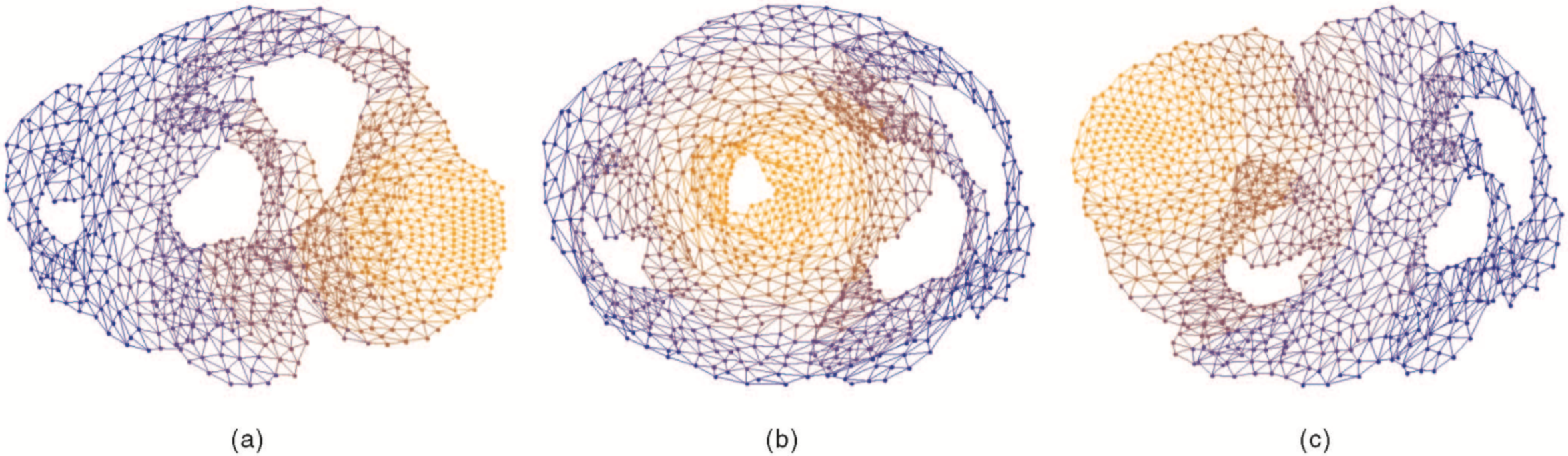
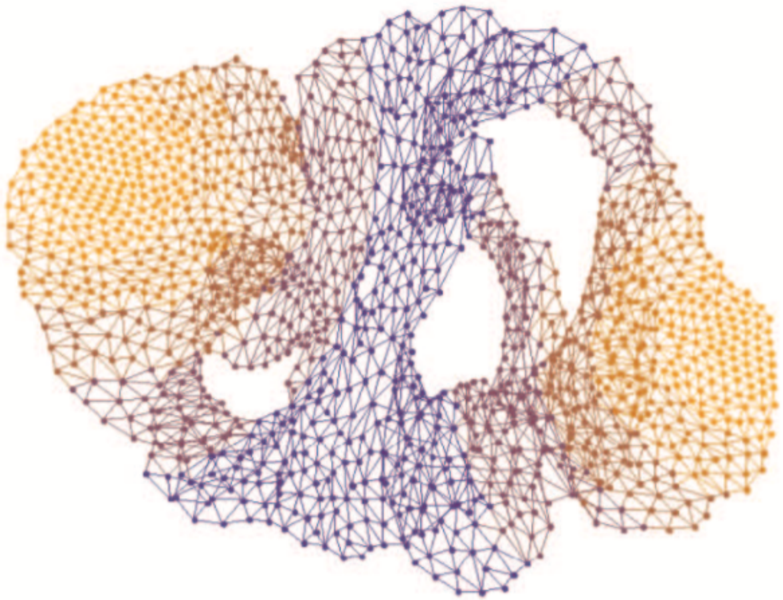


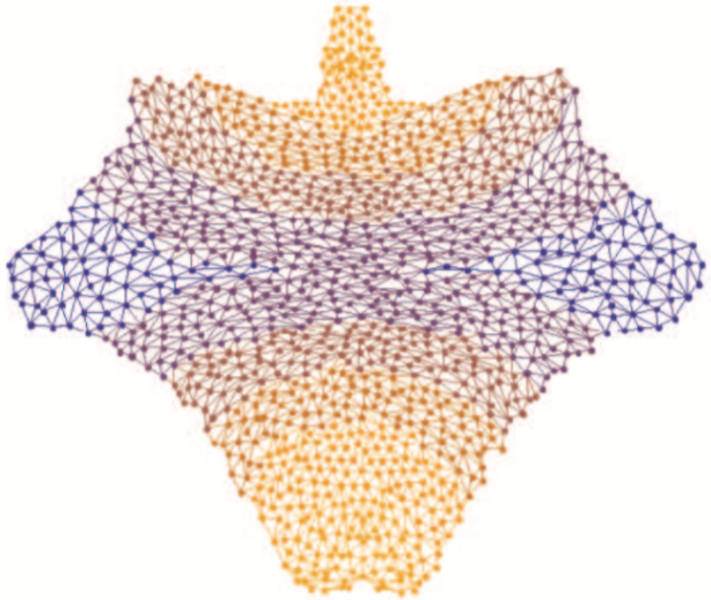
Fig. 4. Topological fisheye views of the 4elt graph. Views are based on “hybrid graphs” formed by superposition of several approximations of the graph. Levels are colored orange-to-blue, where the focus area from the finest graph is in orange. The figure shows three examples, focusing on (a) the right-hand side, (b) the small central hole, and (c) the left-hand side.

GansnerKorenNorth2005

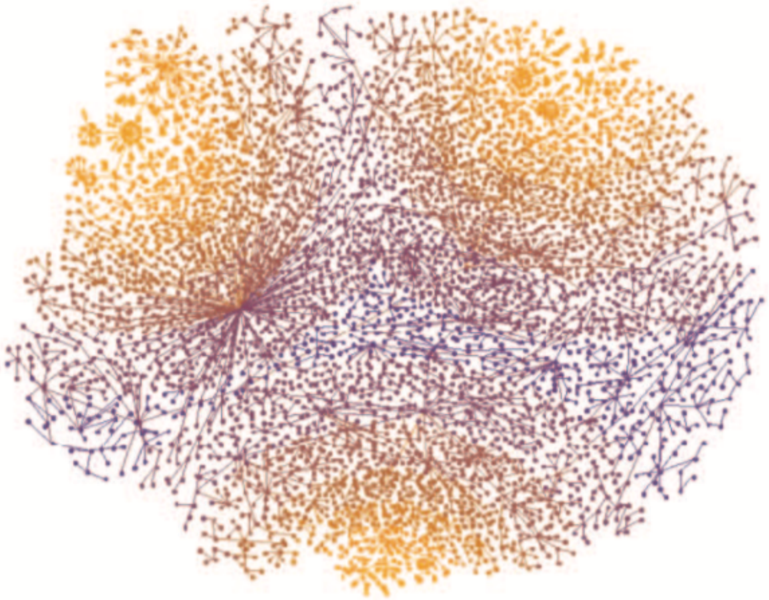
Topological Fish-Eye View



4elt graph (2 foci)



Crack graph (2 foci)



Internet graph (3 foci)

Fig. 17. Viewing graphs with multiple foci.

GansnerKorenNorth2005

Thanks!

Any questions?

You can find me at: beiwang@sci.utah.edu

CREDITS

Special thanks to all people who made and share these awesome resources for free:

- Presentation template designed by Slidesmash
- Photographs by unsplash.com and pexels.com
- Vector Icons by Matthew Skiles

Presentation Design

This presentation uses the following typographies and colors:

Free Fonts used:

http://www.1001fonts.com/oswald-font.html

https://www.fontsquirrel.com/fonts/open-sans

Colors used