Advanced Data Visualization

CS 6965

Spring 2018

Prof. Bei Wang Phillips

University of Utah

Lecture 23

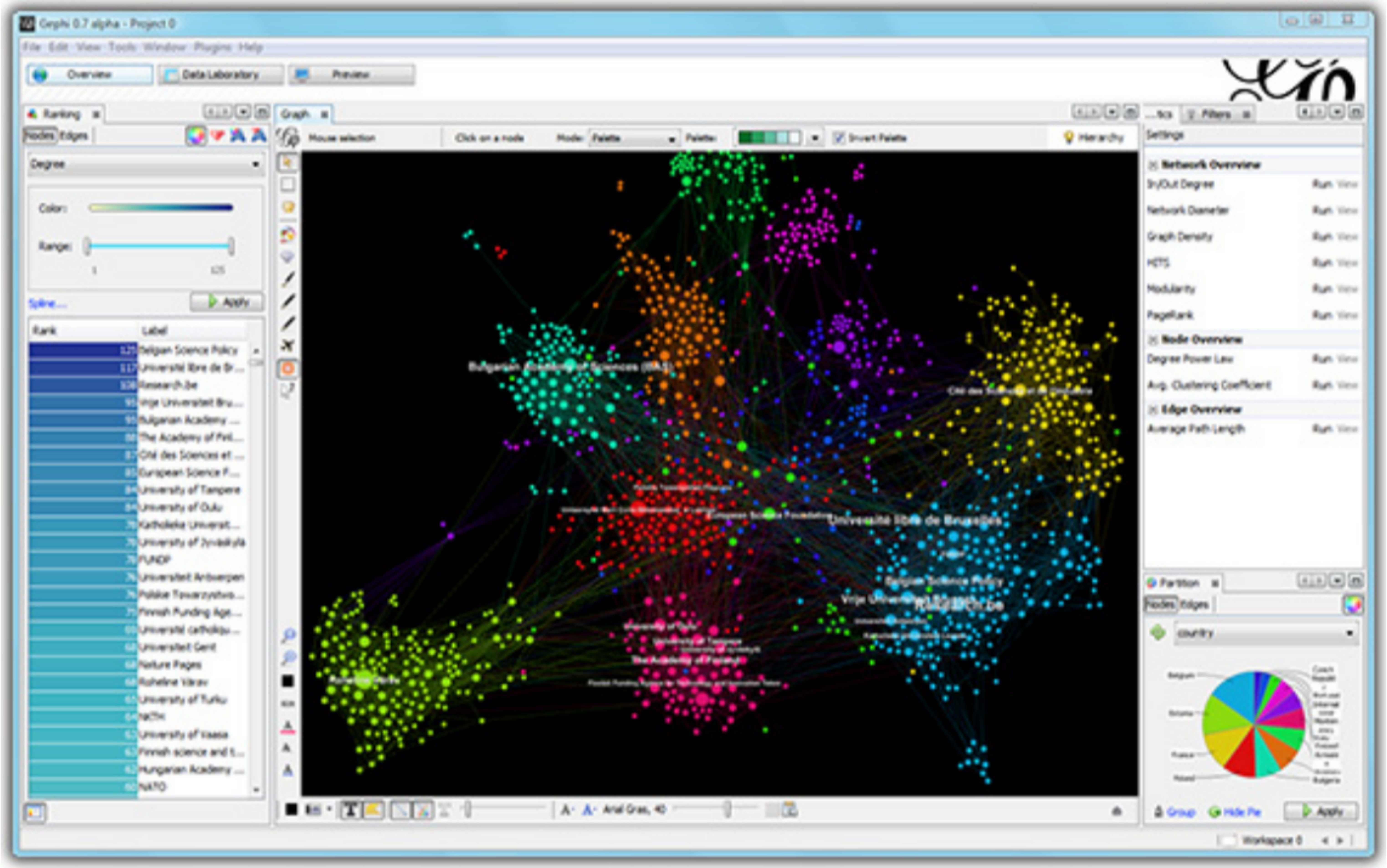
Beyond Force-Directed Layout

NV

Open source network visualization tools

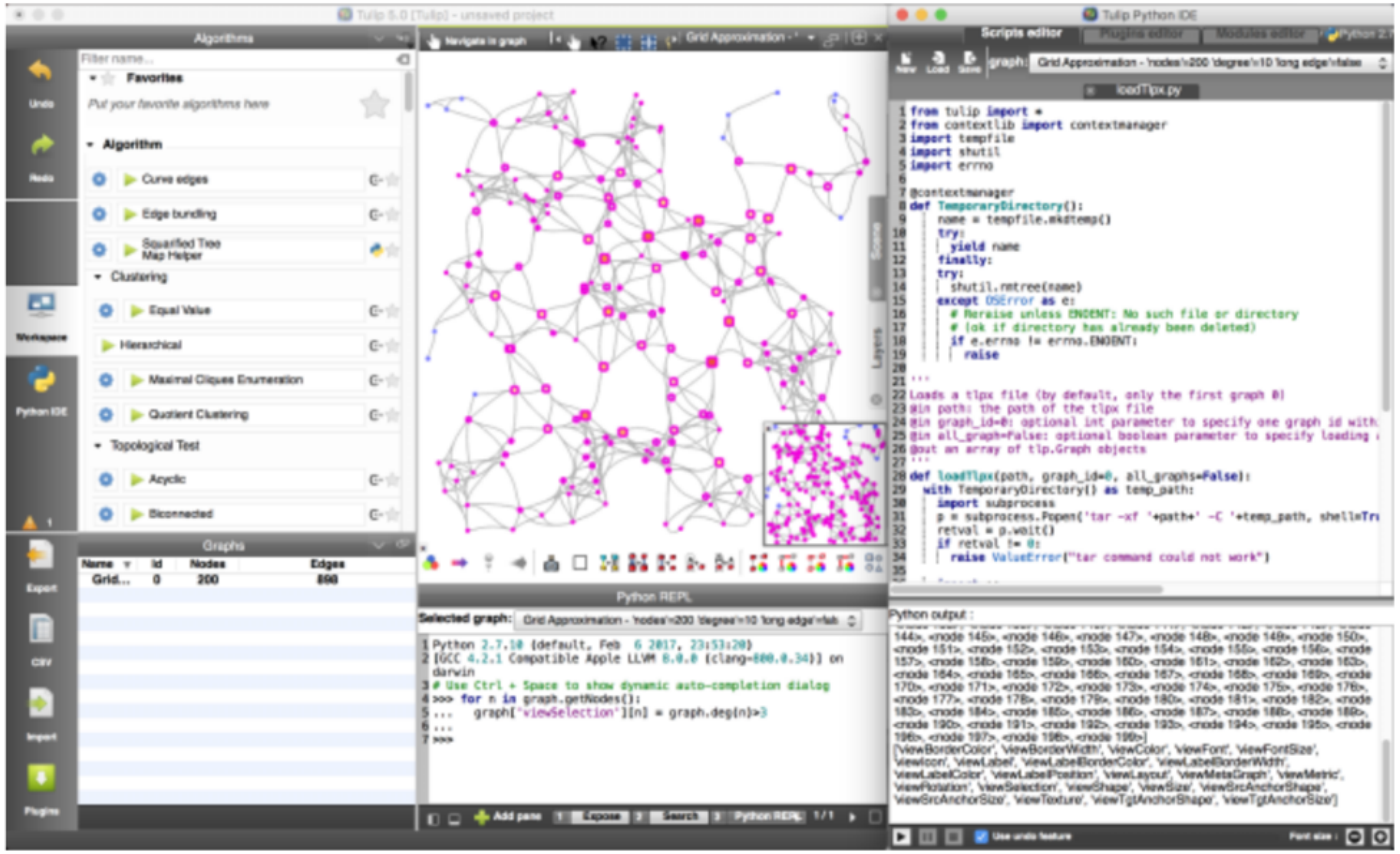
Gephi

- Java
- https://gephi.org/
- Open Source
- User Manual: https://gephi.org/users/



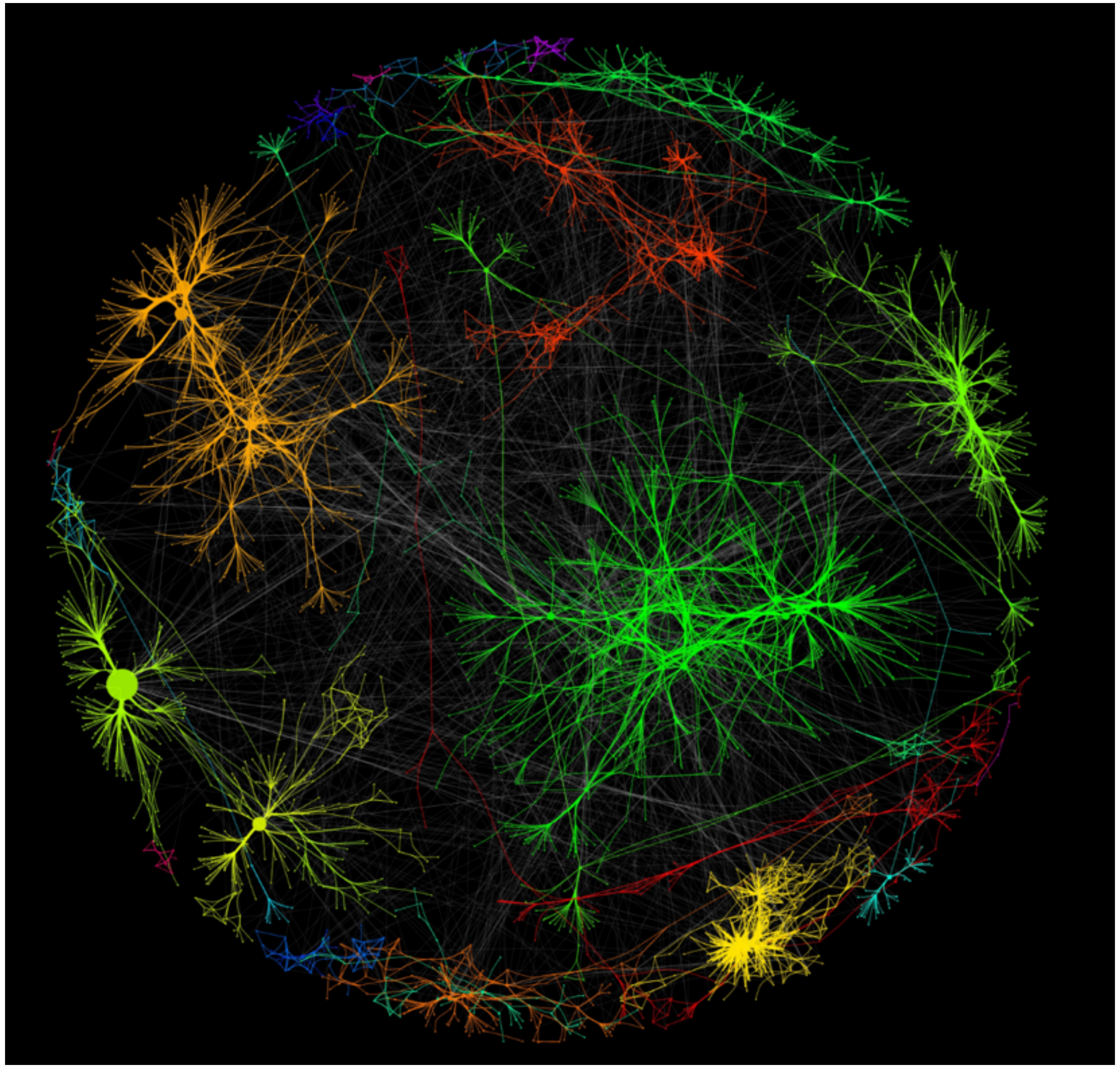
Tulip

- C++ / Python
- http://tulip.labri.fr/TulipDrupal/
- Open Source
- User Manual: http://tulip.labri.fr/Documentation/3_5/userHandbook/html/index.html



Cytoscape

- Java
- http://www.cytoscape.org/
- Open Source
- User Manual: http://manual.cytoscape.org/en/3.4.0/index.html



GraphViz

- C
- http://www.graphviz.org/
- open source

Others

- NodeXL
  - http://nodexl.codeplex.com/
  - Basic Version is free but Pro Version must be paid for

Datasets of large graphs

Stanford SNAP

Google web graph

- https://snap.stanford.edu/data/

Dataset information

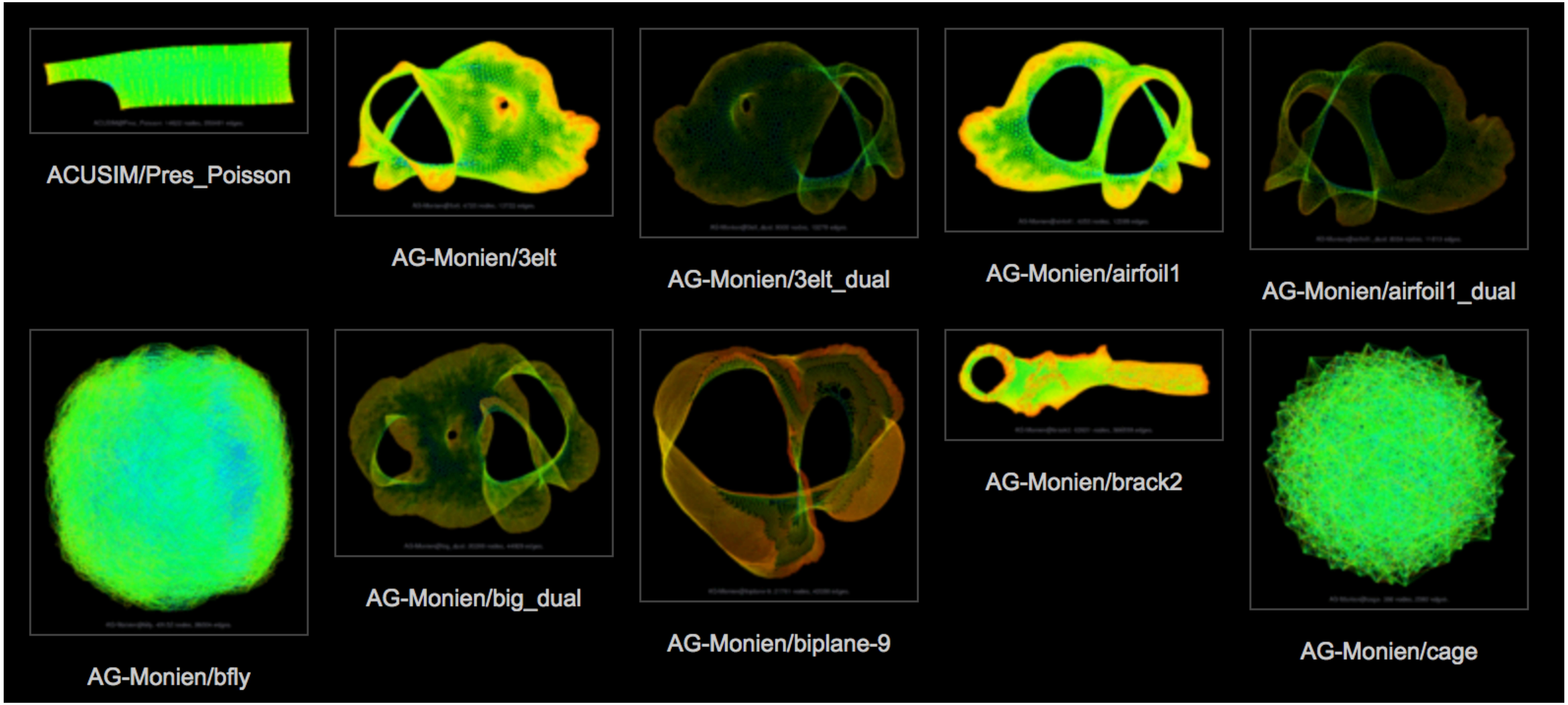
Nodes represent web pages and directed edges represent hyperlinks between them. The data was released in 2002 by Google as a part of Google Programming Contest.

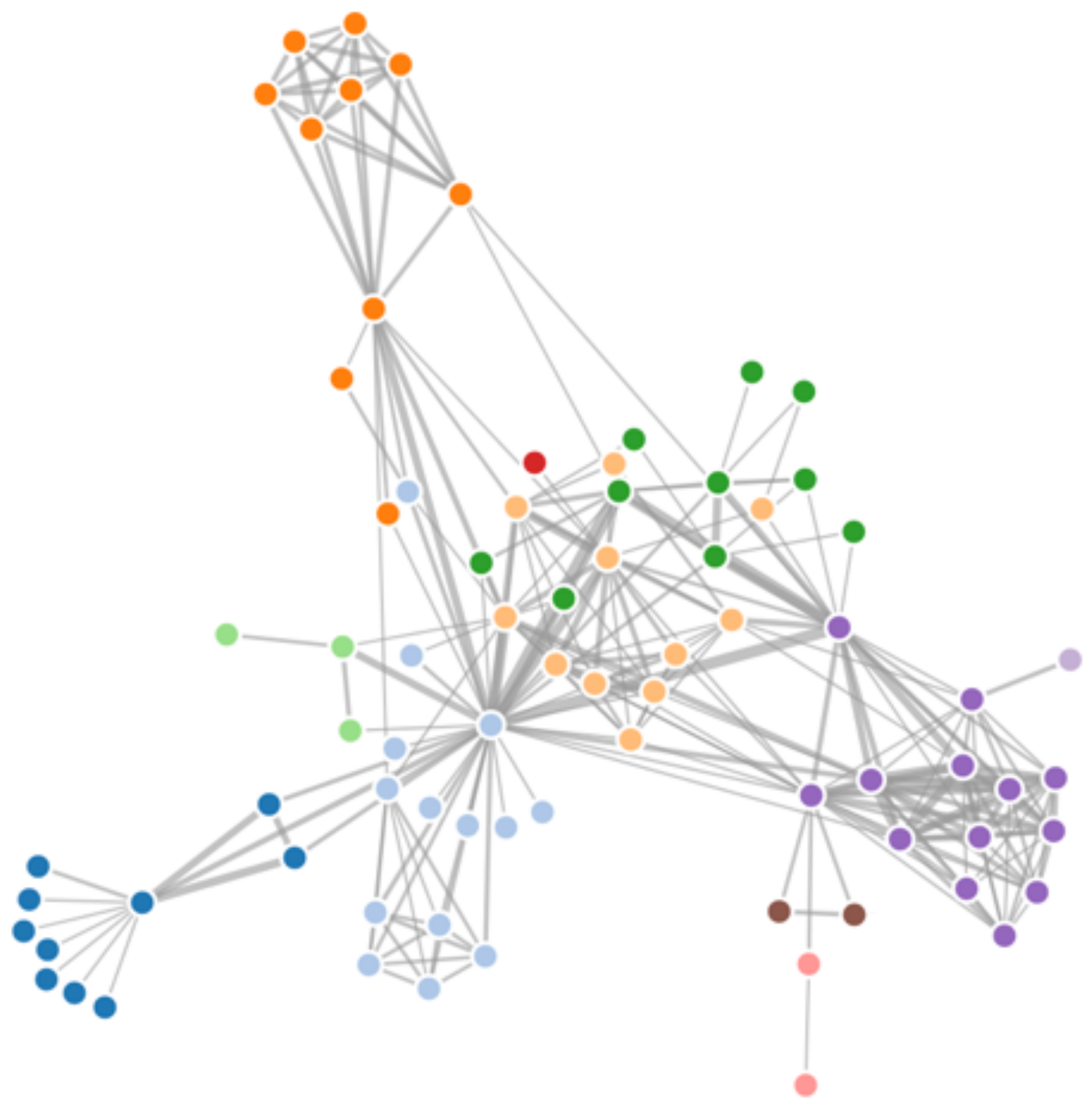
Dataset statistics	
Nodes	875713
Edges	5105039
Nodes in largest WCC	855802 (0.977)
Edges in largest WCC	5066842 (0.993)
Nodes in largest SCC	434818 (0.497)
Edges in largest SCC	3419124 (0.670)
Average clustering coefficient	0.5143
Number of triangles	13391903
Fraction of closed triangles	0.01911
Diameter (longest shortest path)	21
90-percentile effective diameter	8.1

SNAP

Large Matrices

- http://yifanhu.net/GALLERY/GRAPHS/
- https://sparse.tamu.edu/ (UF Sparse Matrix collection)

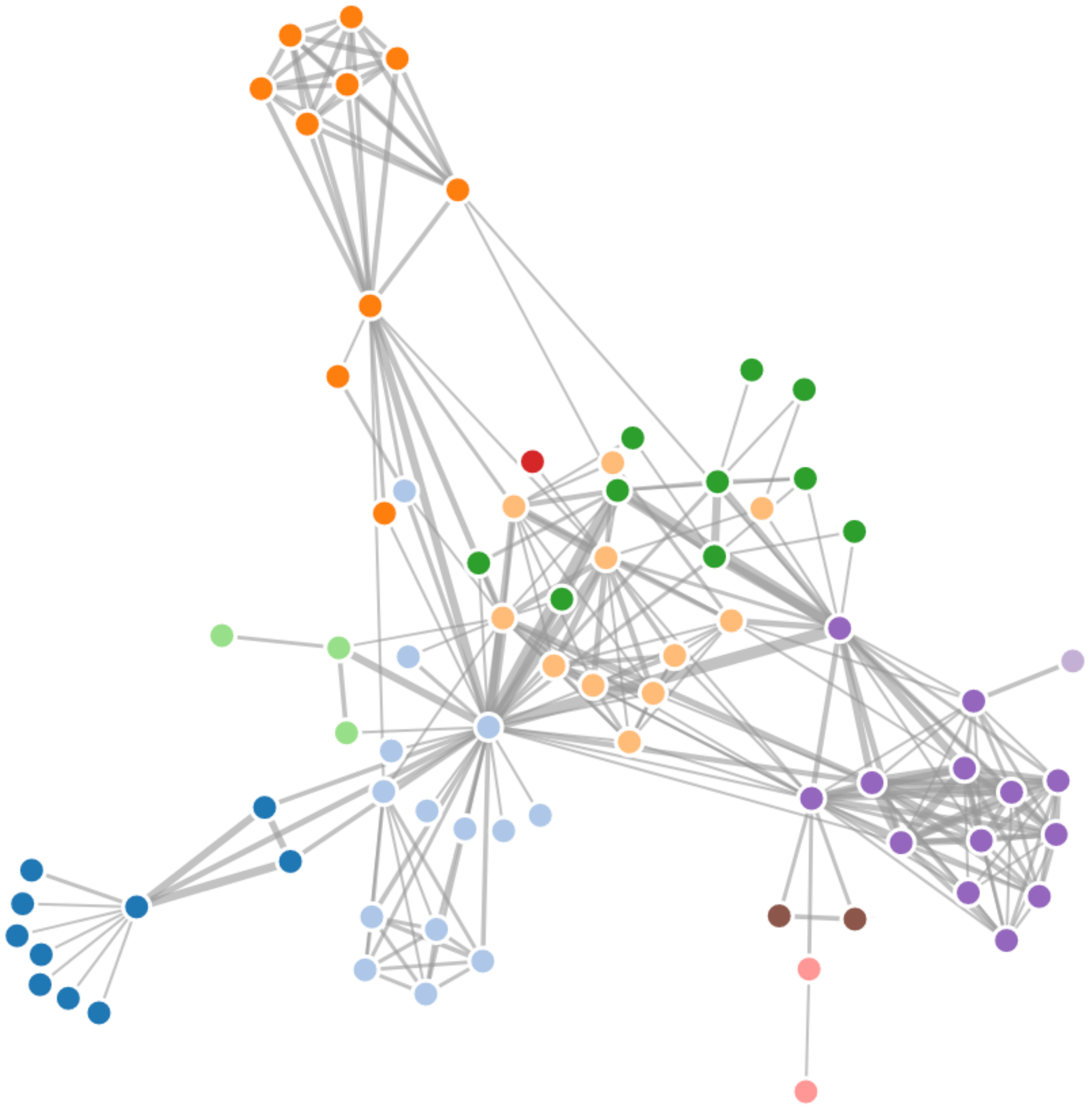




Force-directed Layout

https://tenor.com/view/may-the-force-be-with-you-gif-10808590

Force-directed Layout



https://bl.ocks.org/mbostock/4062045

Force directed layout

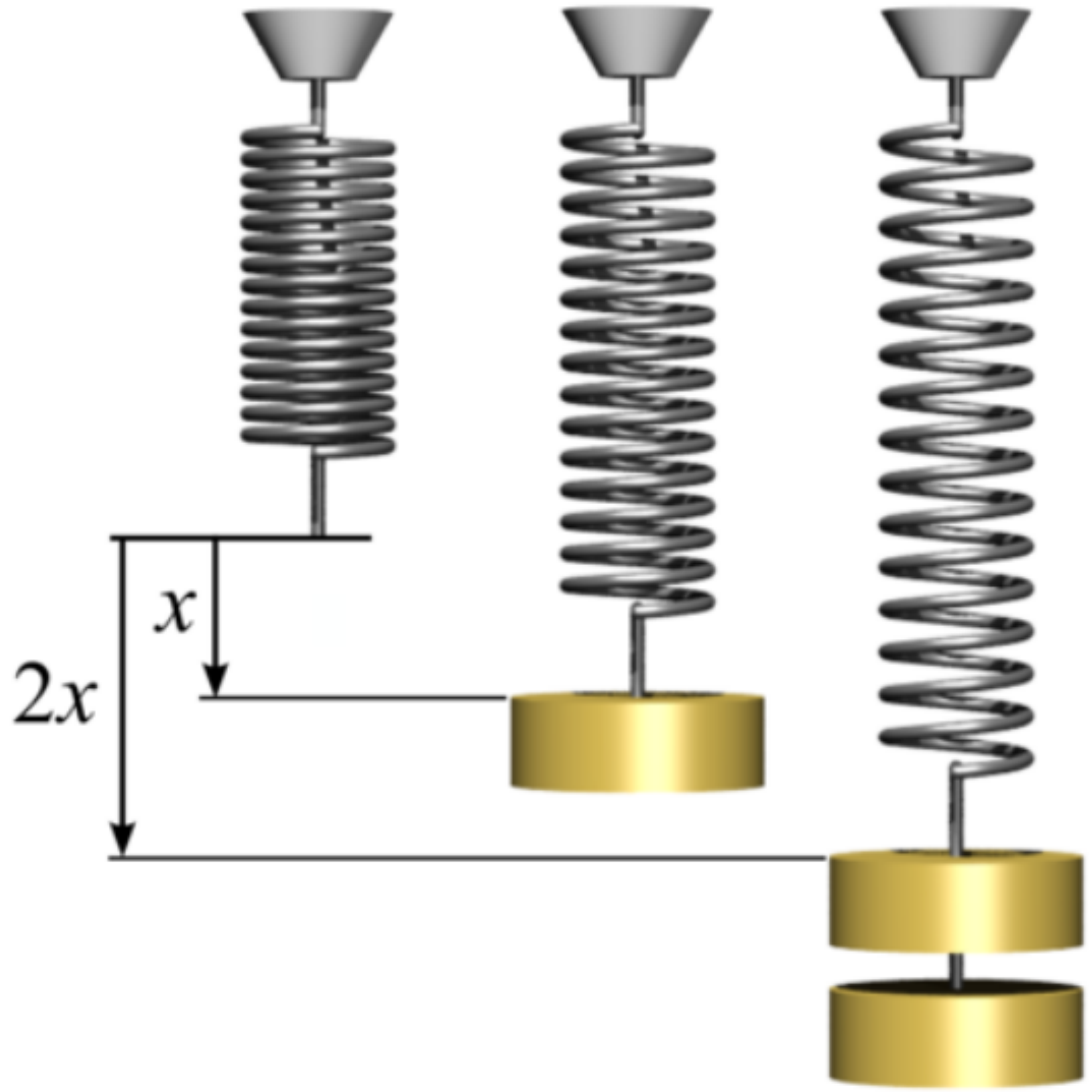
- A class of graph drawing algorithms
- Aesthetically-pleasing drawings
- Assign forces among edges and nodes
- Attractive forces are used to attract endpoints of edges towards each other (spring-like, Hooke's law)
- Repulsive forces are used to separate all pairs of nodes (electrical repulsion, Coulomb's law)

Variation: using only spring forces between all pairs of vertices, with ideal spring lengths equal to the vertices' graph-theoretic distance.

https://en.wikipedia.org/wiki/Force-directed_graph_drawing

Attraction: Hooke's Law

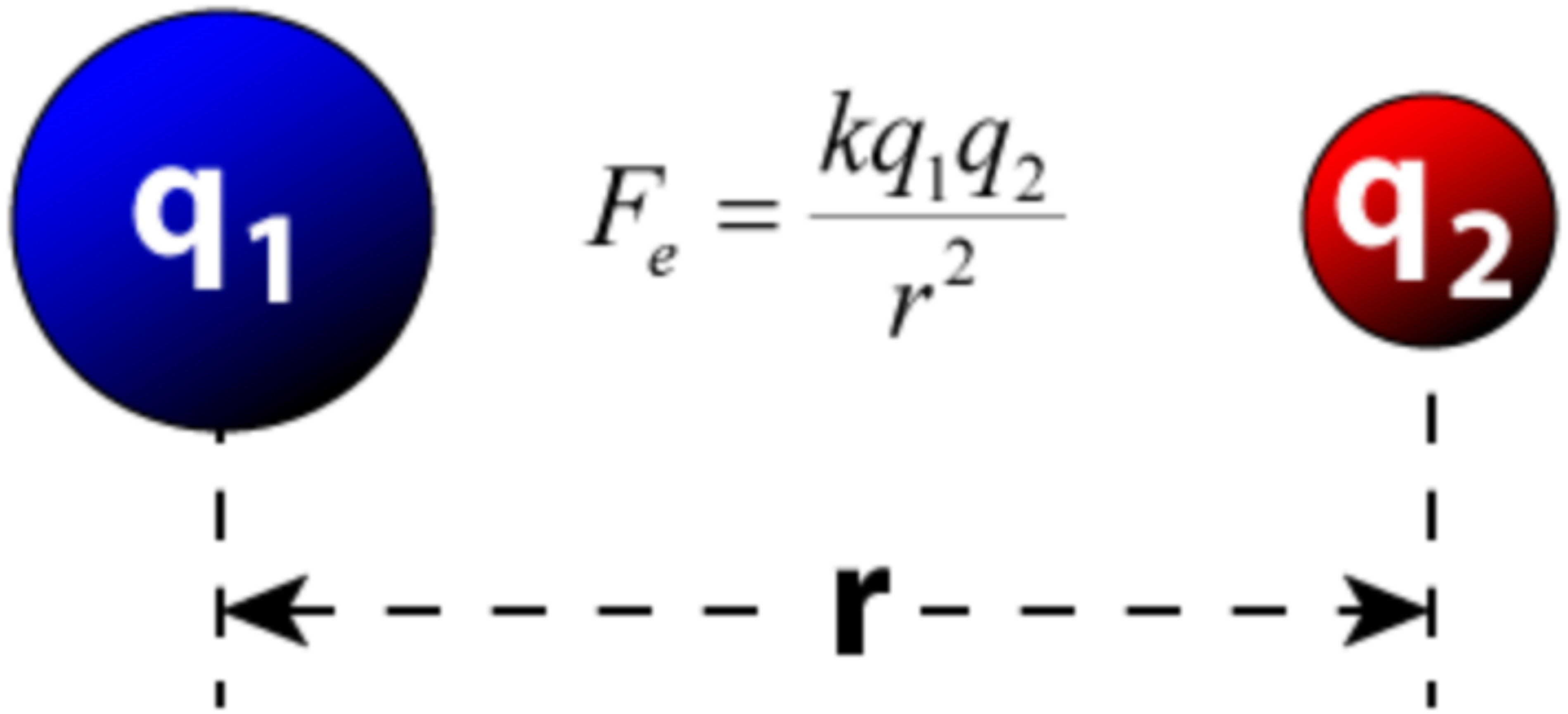
- The force is proportional to the length of the spring
- F = kX
  - k: stiffness parameter
  - X: distance to compress or extend a spring



https://en.wikipedia.org/wiki/Hooke%27s_law

Repulsion: Coulomb's law

- Describes force interacting between static electrically charged particles



- k_e: Coulomb's constant (k_e = 8.9875×109 N m^2 C^{−2})
- q1, q2: signed magnitudes of the charges
- r: distance between the charges.

https://en.wikipedia.org/wiki/Coulomb%27s_law

http://www.aplusphysics.com/courses/honors/estat/Coulomb.html

Force directed layout: equilibrium

At equilibrium:

- Edges tend to have uniform length (due to the spring forces)
- Nodes that are not connected by an edge tend to be drawn further apart (due to electrical repulsion).
- Other attraction/repulsion definitions are not possible, not necessarily based on physical behaviors.

Pros

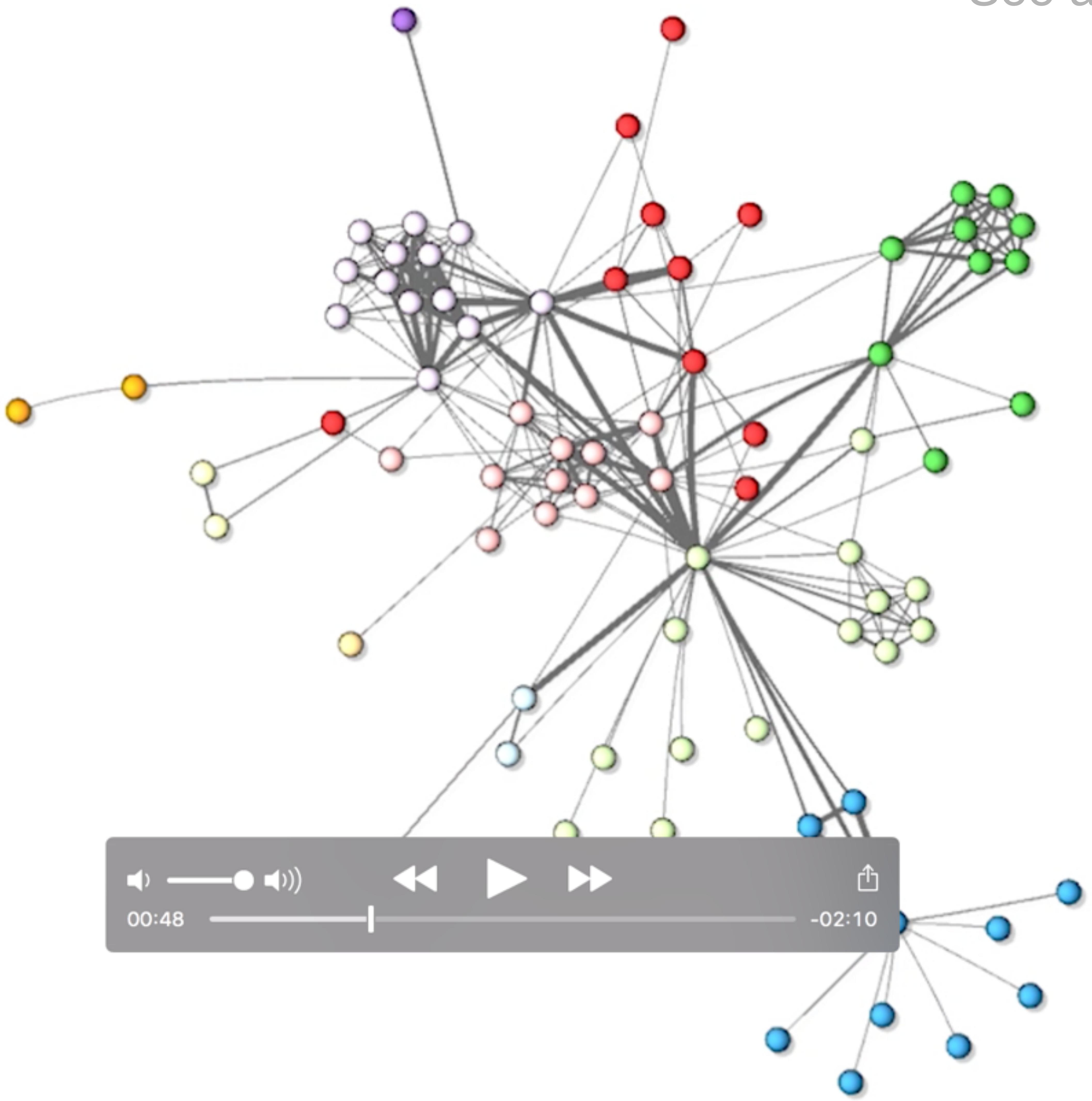
- Reasonable quality for graphs up to a few hundred node: uniform edge length, uniform vertex distribution, symmetry.
- Flexibility: easily extendable to other aesthetic criteria
- Easy to understand (e.g. physical springs)
- Easy to implement
- Dynamic and online drawing: interactivity; move nodes, converges
- Stress majorization: theoretical properties

Cons

- Local minima is not necessarily near optimal
- Takes a long time to converge O(n^3)
- Scalability
- Lack predictability: running the algorithm twice, produces different results.

A Recent Example

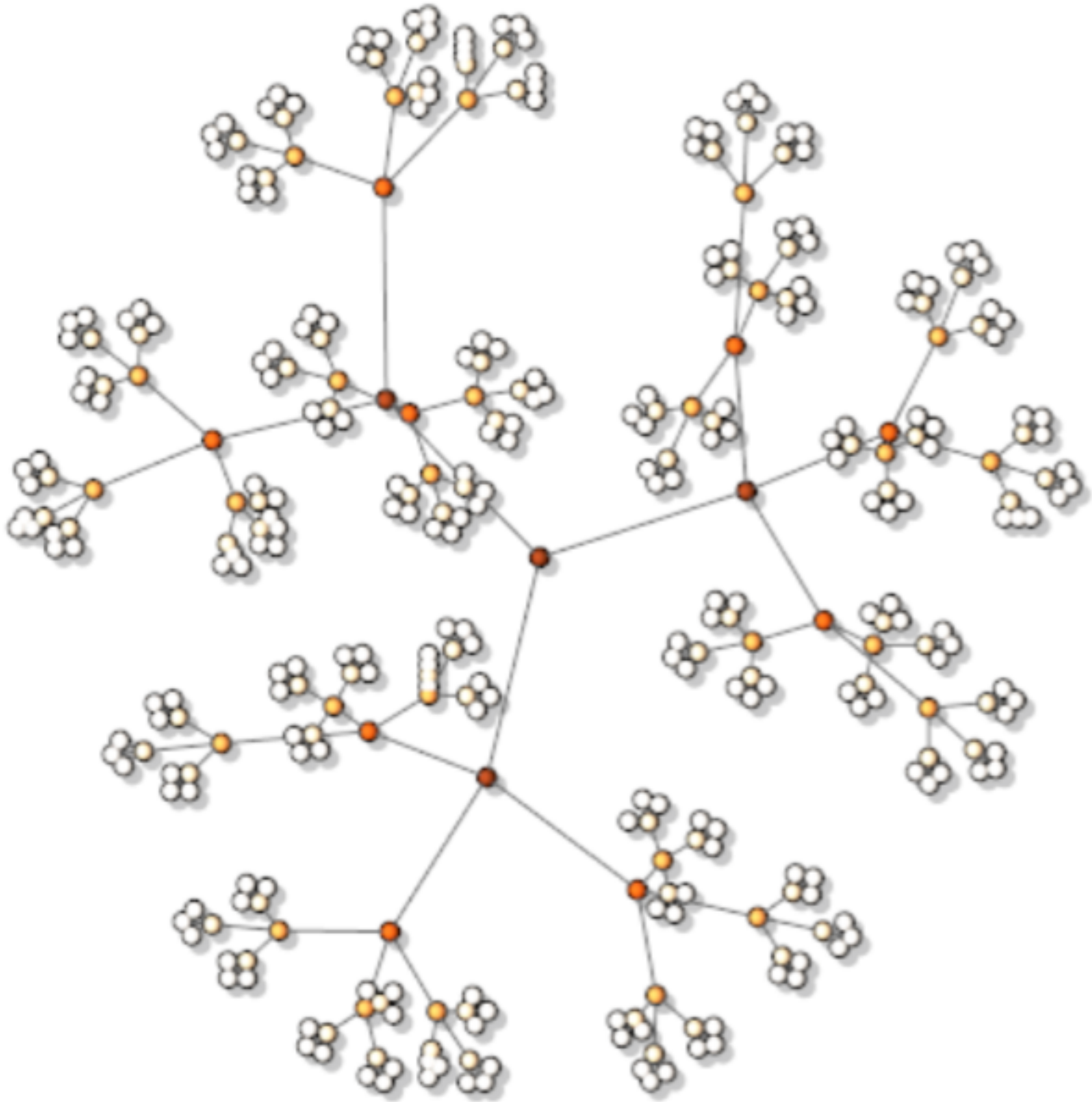
See a separate video

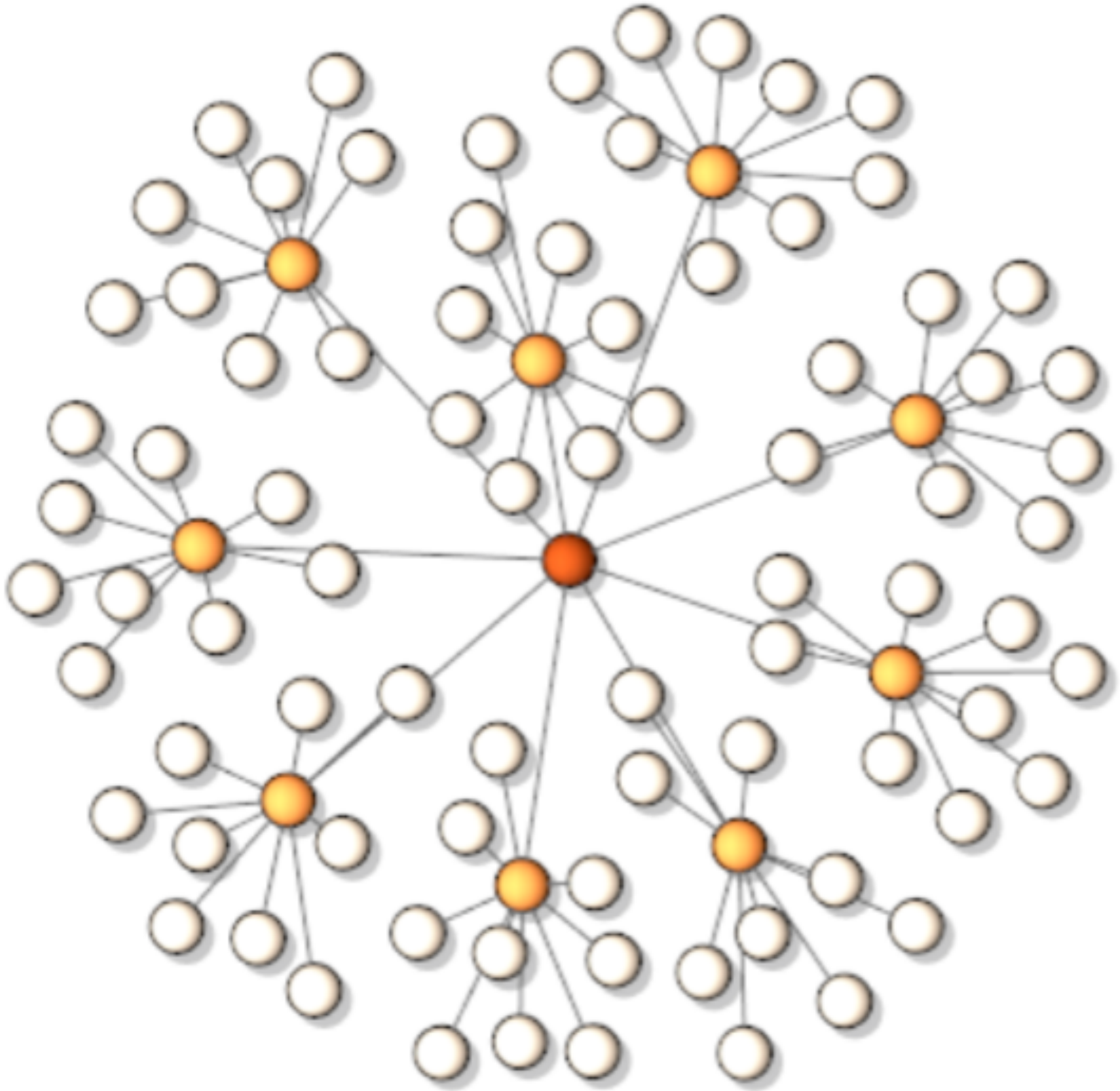


Dealing with Scalability

- Clustering/aggregation
- Collapsible Layout
- Multiple layers / multi-dimensional approaches

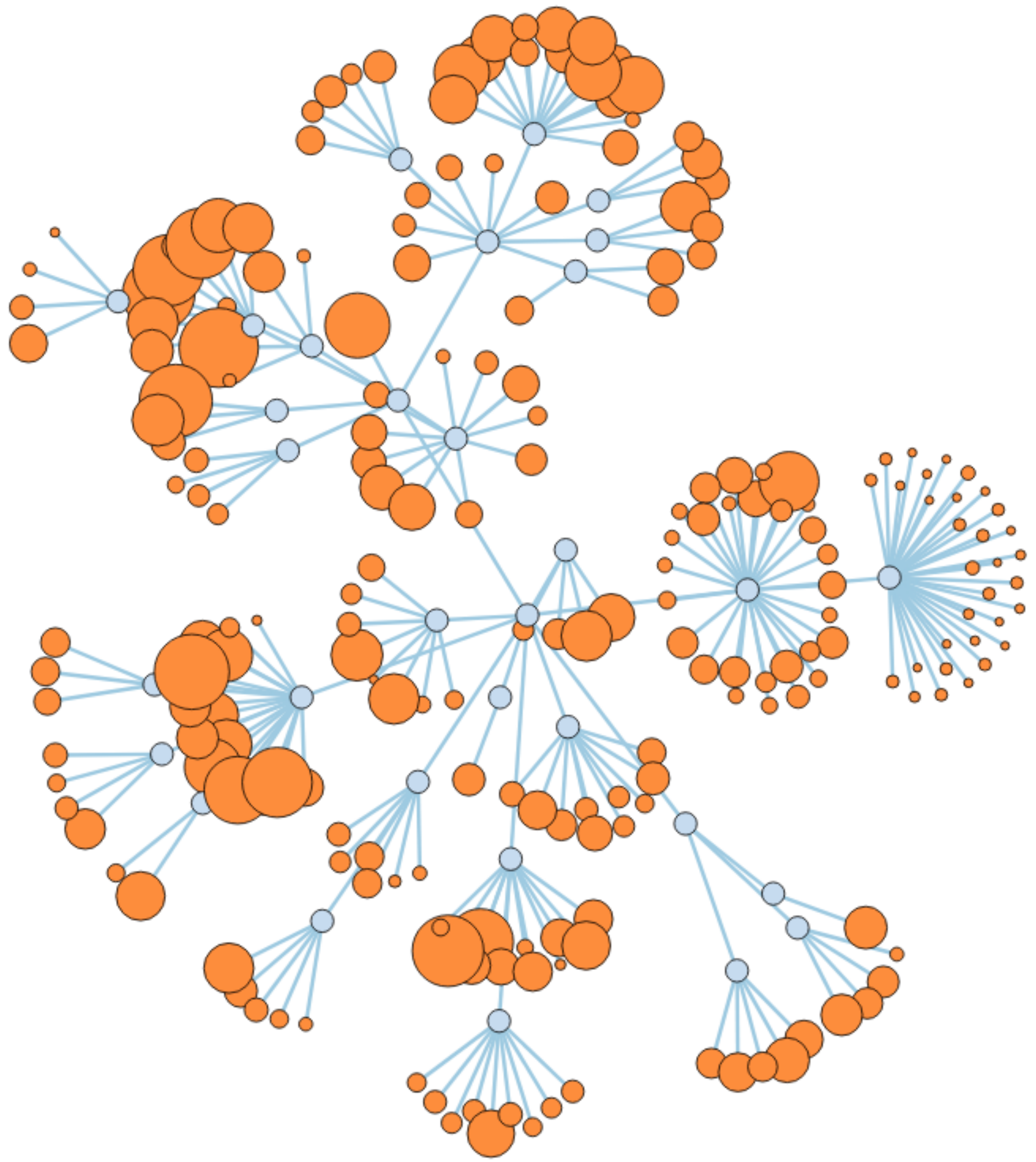
Clustering (more on this later)





Collapsible layout

- A force layout of a hierarchy whose internal nodes are collapsible
- Rely on clustering



https://mbostock.github.io/d3/talk/20111116/force-collapsible.html

Multidimensional approaches

- Maximal independent set (MIS): S ⊂ V is an independent set of a graph G = (V , E) if no two elements of S are connected by an edge of G. E
- Equivalently, S is an independent set of G if the graph distance between any two elements of S is at least two.
- Multi-dimensional drawing: drawing graphs in high dimensions and then project

GajerGoodrichKobourov2004

Multidimensional approaches

- Intelligent initial placement of vertices
- Multi-dimensional drawing
- Simple recursive coarsening scheme
- Fast energy function minimization
- Space and time efficiency

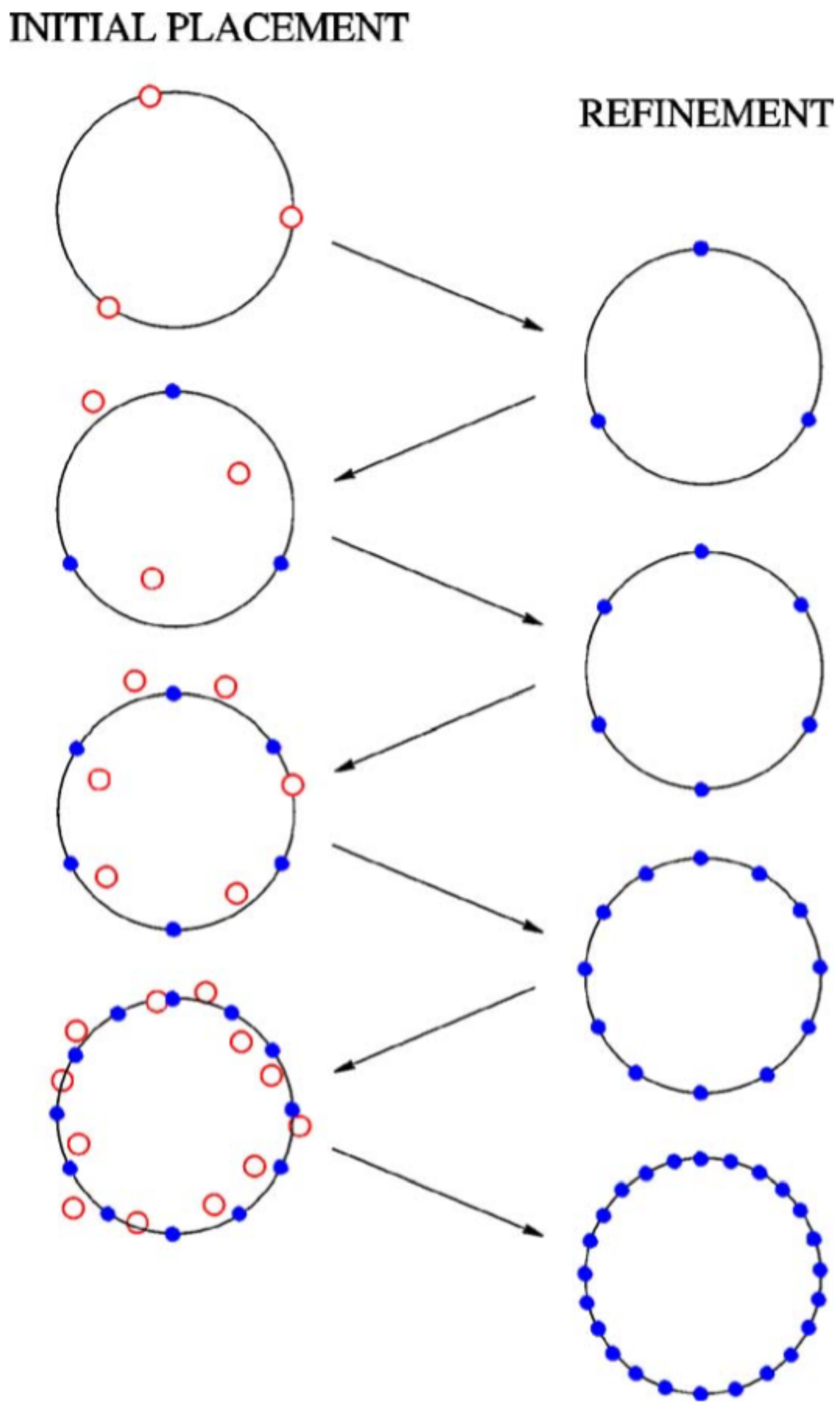
GajerGoodrichKobourov2004

Multi-dim. approaches

Main Algorithm

create a filtration $\mathcal{V}$: $V_0 \supset V_1 \supset \cdots \supset V_k \supset \emptyset$
for $i = k$ **to** 0 **do**
  for each $v \in V_i - V_{i+1}$ **do**
    find vertex neighborhood $N_i(v), N_{i-1}(v), \ldots, N_0(v)$
    find initial position `pos`$[v]$ of v
  repeat `rounds` times
    for each $v \in V_i$ **do**
      compute local temperature `heat`$[v]$
      `disp`$[v] \leftarrow$ `heat`$[v] \cdot \vec{F}_{N_i}(v)$
    for each $v \in V_i$ **do**
      `pos`$[v] \leftarrow$ `pos`$[v] +$ `disp`$[v]$
add all edges $e \in E$

Initial layout: using graph distances among initial set as Euclidean distance for layout.
Local force: computed over neighbors of a node (not all nodes)



GajerGoodrichKobourov2004

Multidimensional approaches

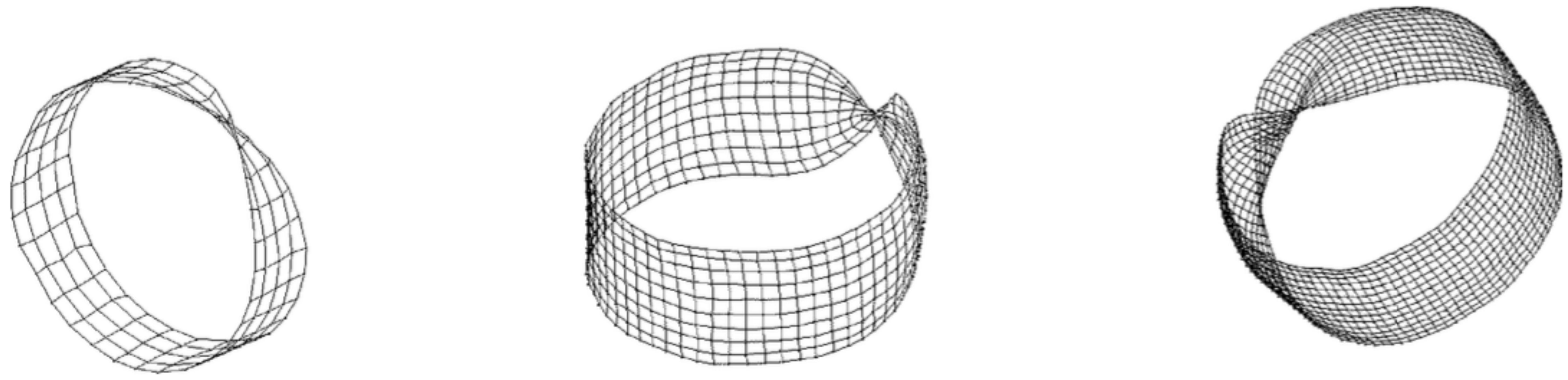


Fig. 5. Moebius strips on 150, 300 and 1500 vertices drawn in directly 3D. Note the rough “twists”.

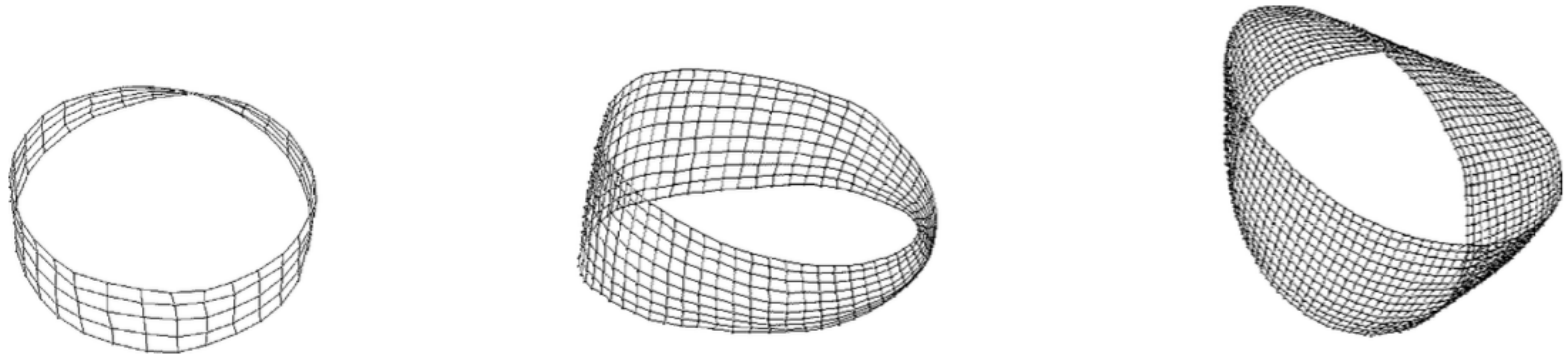


Fig. 6. The same Moebius strips as in Fig. 5 but drawn in 4D and projected in 3D. Note the smooth twists.

GajerGoodrichKobourov2004

Distributed Multilevel Force-directed layout

- 10 millions edges in an hour
- FM3 (Fast Multipole Multilevel Method)

https://arxiv.org/abs/1606.02162
https://arxiv.org/pdf/1608.08522.pdf

FM3: Solar Merger algorithm

- FM3 (Fast Multipole Multilevel Method)
- Solar Merger algorithm

1. Vertices of G are partitioned into subgraphs called solar systems with diameter at most 4.
2. Within each solar system S: a vertex s is classified as a sun.
3. Vertex v of S at distance 1 from s: a planet.
4. Distance 2 from s: a moon.
5. There is an inter-system link between two solar systems S1 and S2, if there is at least an edge of G between a vertex of S1 and a vertex of S2.
6. The coarser graph G1 is obtained by collapsing each solar system into the corresponding sun, and the inter-system links are transformed into edges connecting the corresponding pairs of suns

https://arxiv.org/abs/1606.02162 https://arxiv.org/pdf/1608.08522.pdf

FM3 vs MULTI_GILA

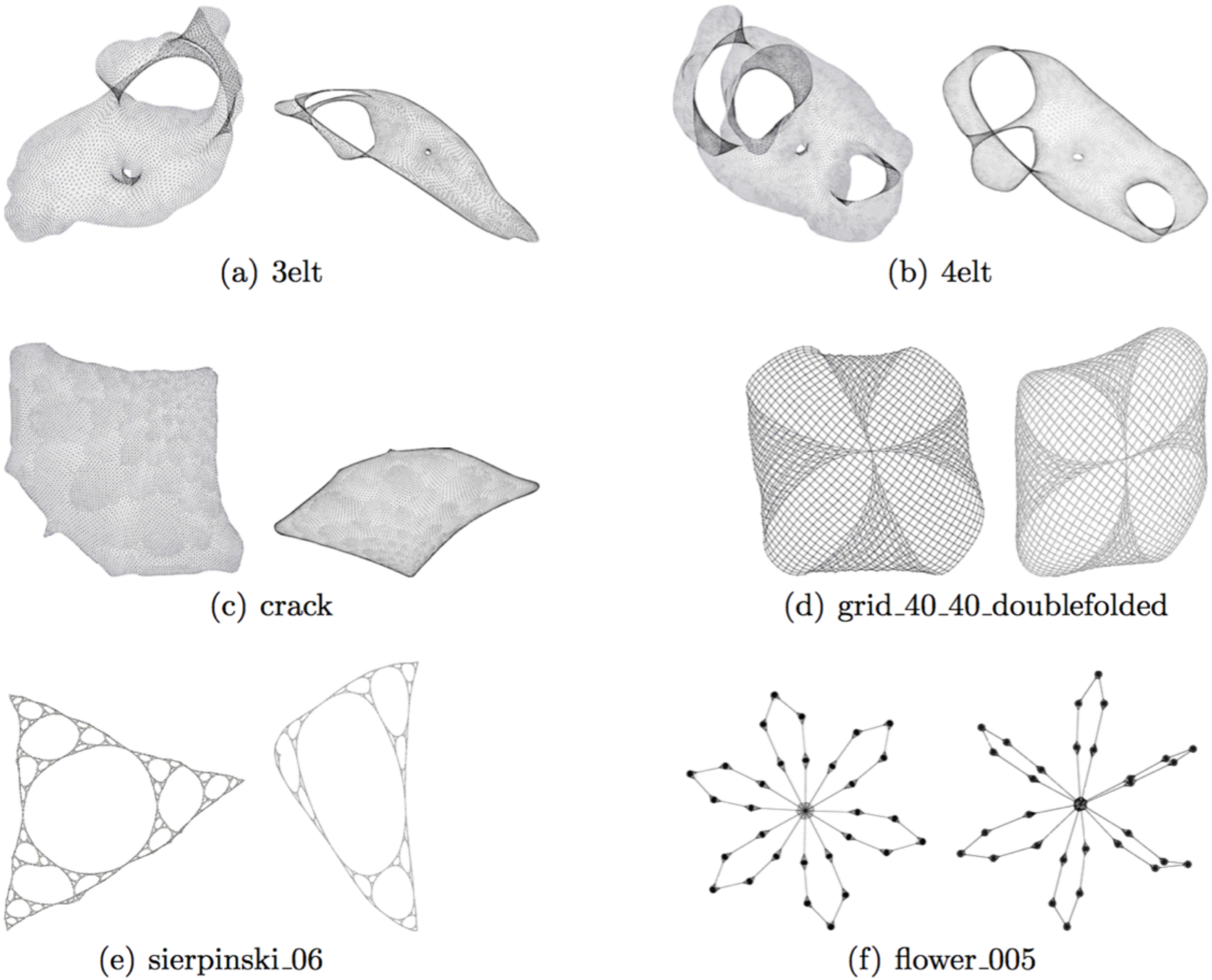


Fig. 2. Layouts of some REGULARGRAPHS instances. For each graph, the drawing computed by FM3 (MULTI-GILA) is on the left (right).

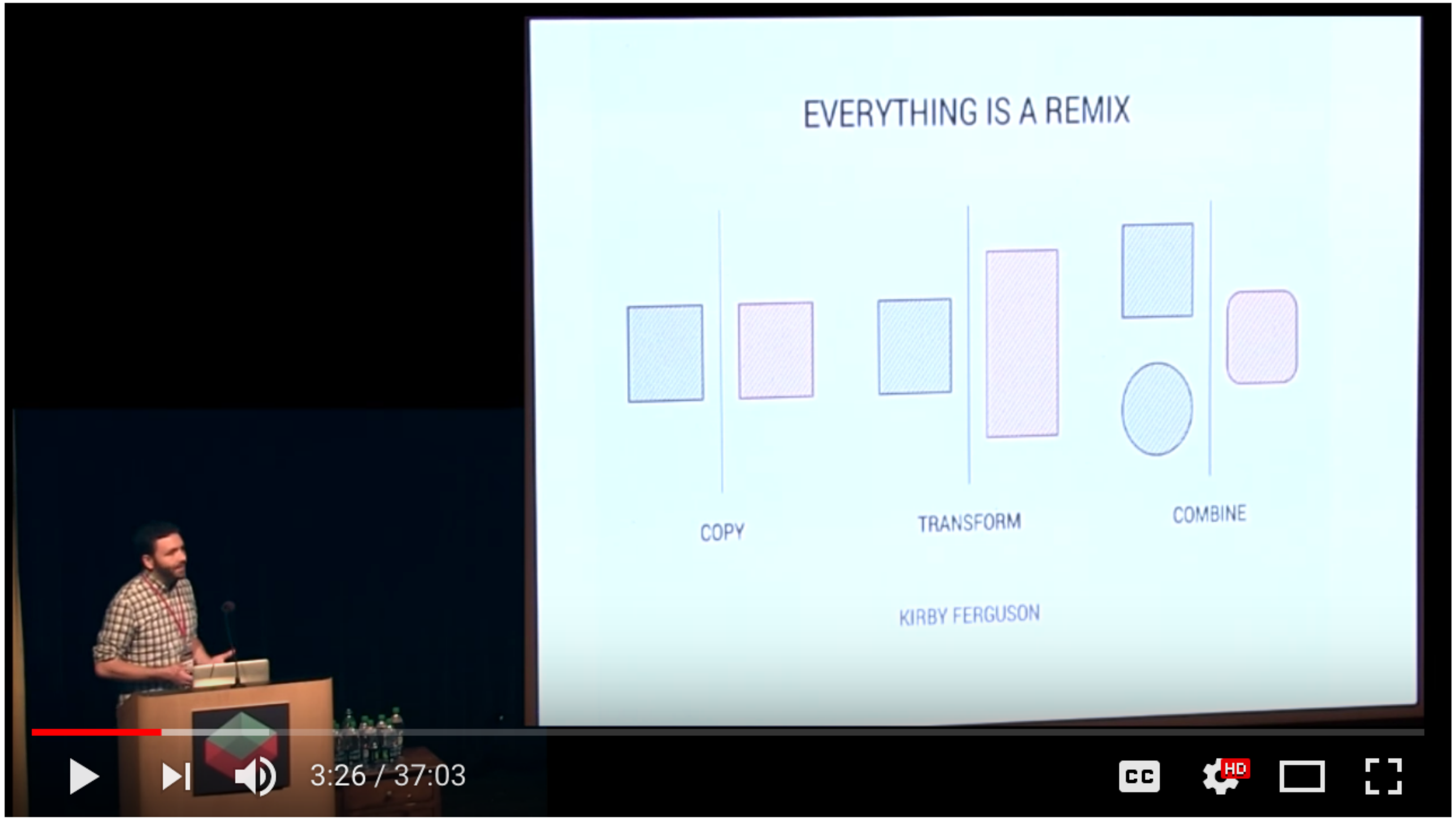
https://arxiv.org/pdf/1608.08522.pdf

https://arxiv.org/abs/1606.02162

Questions to ask

- If we can draw a large graph, what about interpretability?
- Is there any room for growth?

Using and Abusing the Forces



Starting at around 3:30 minutes

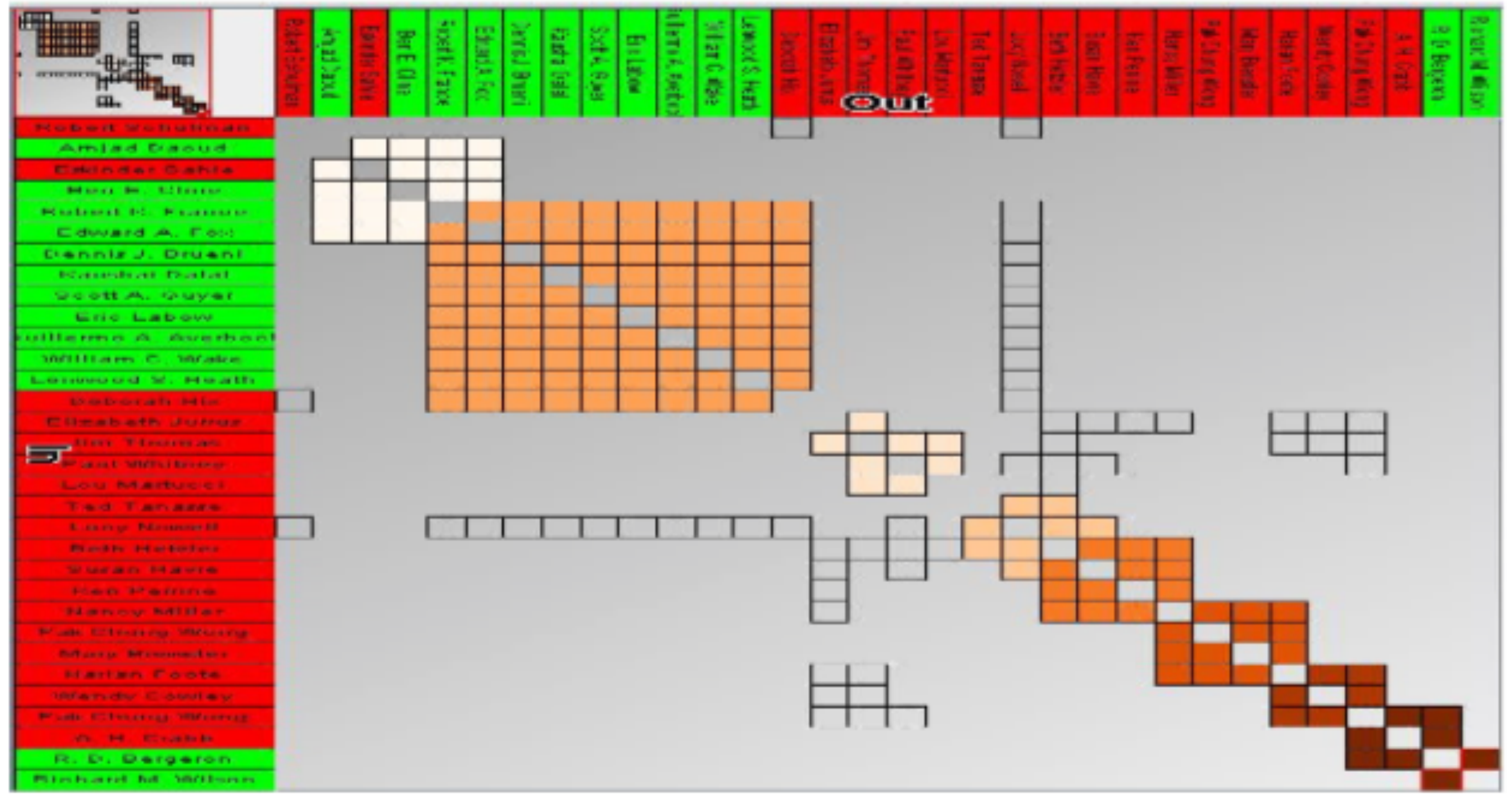
https://www.youtube.com/watch?v=Mucmb33711A&feature=youtu.be

- https://medium.com/@sxywu/understanding-the-force-ef1237017d5

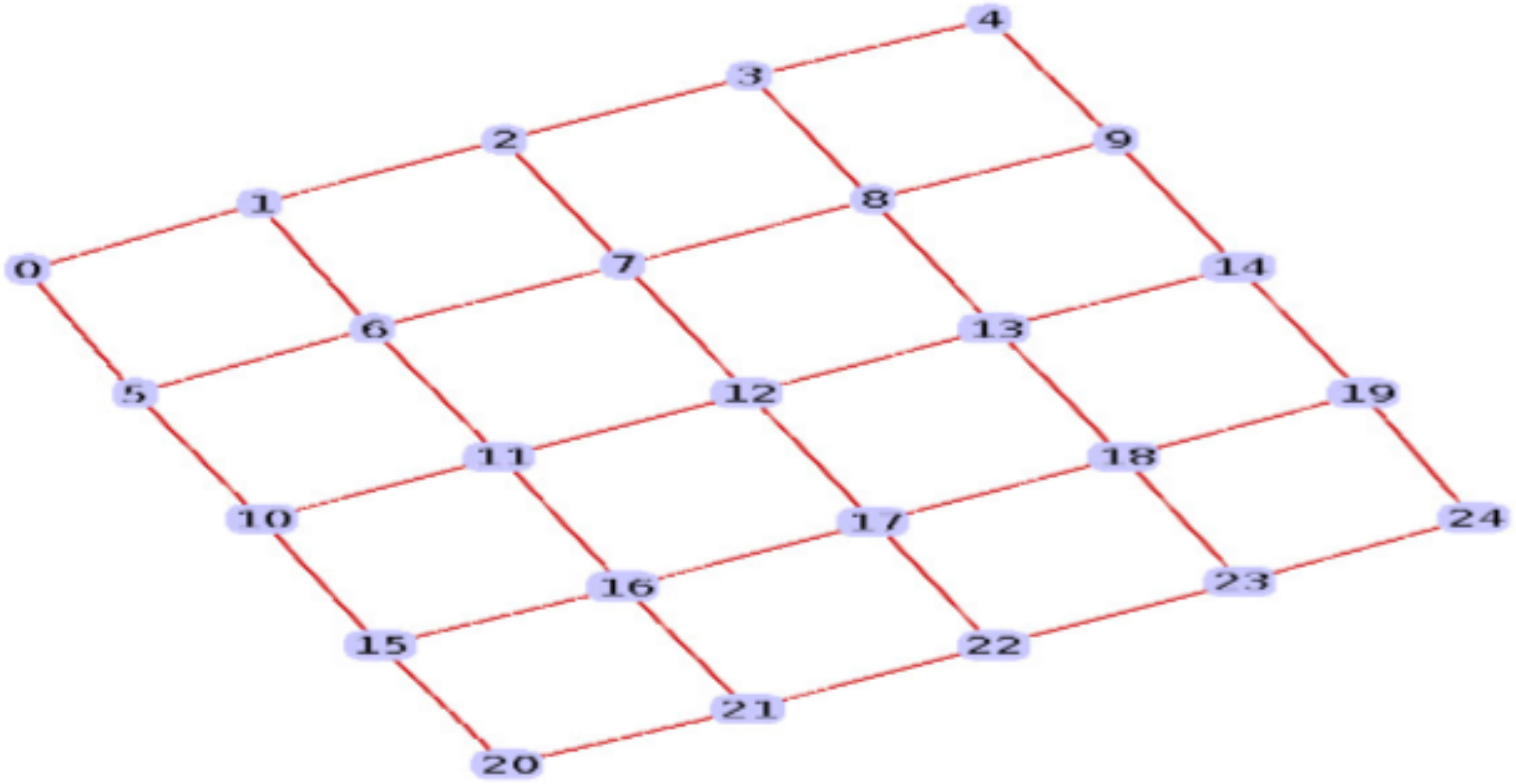
Graph Layout: A Brief Overview

TarawnehKellerEbert2011

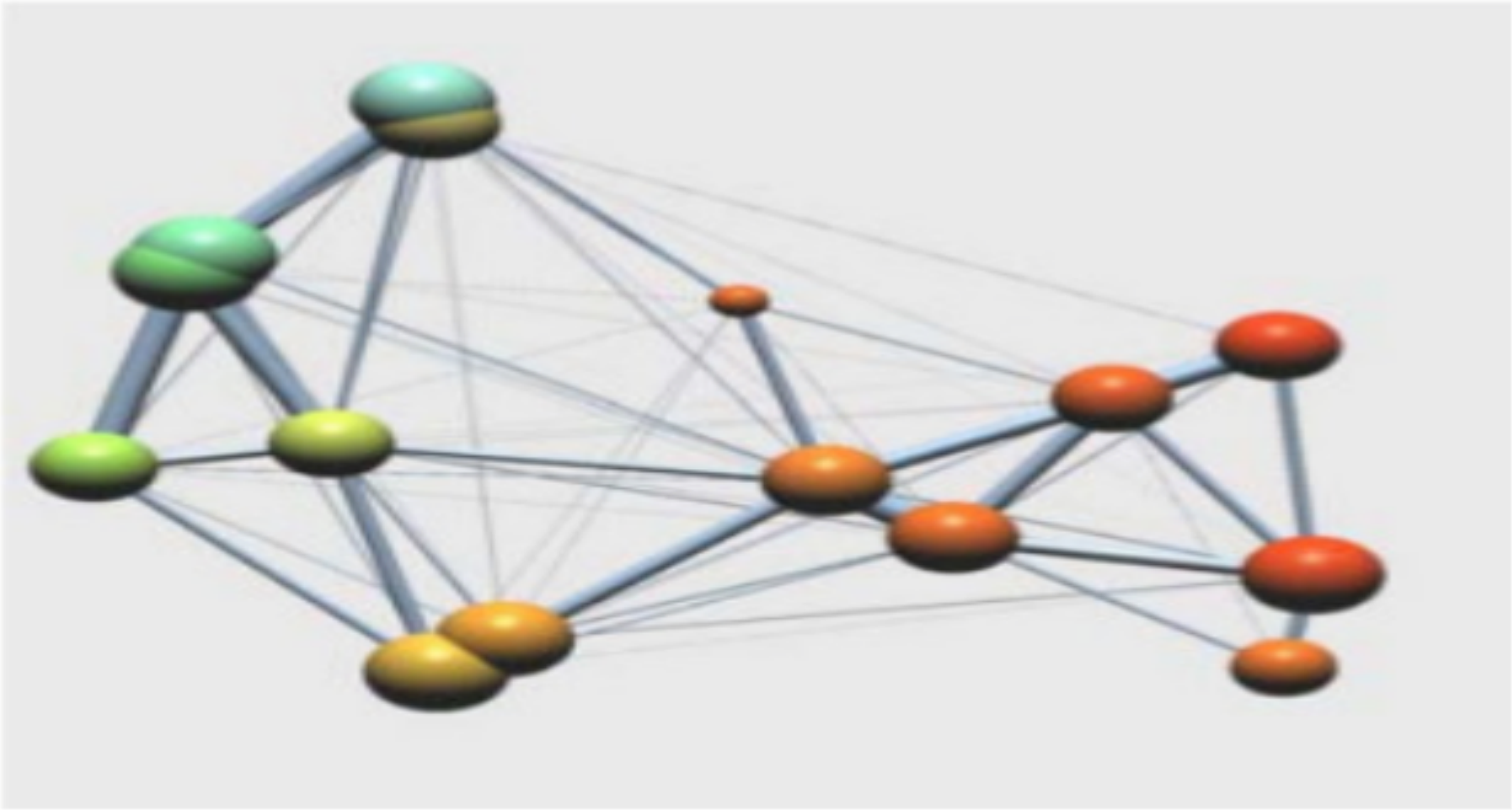
http://drops.dagstuhl.de/opus/volltexte/2012/3748/pdf/13.pdf



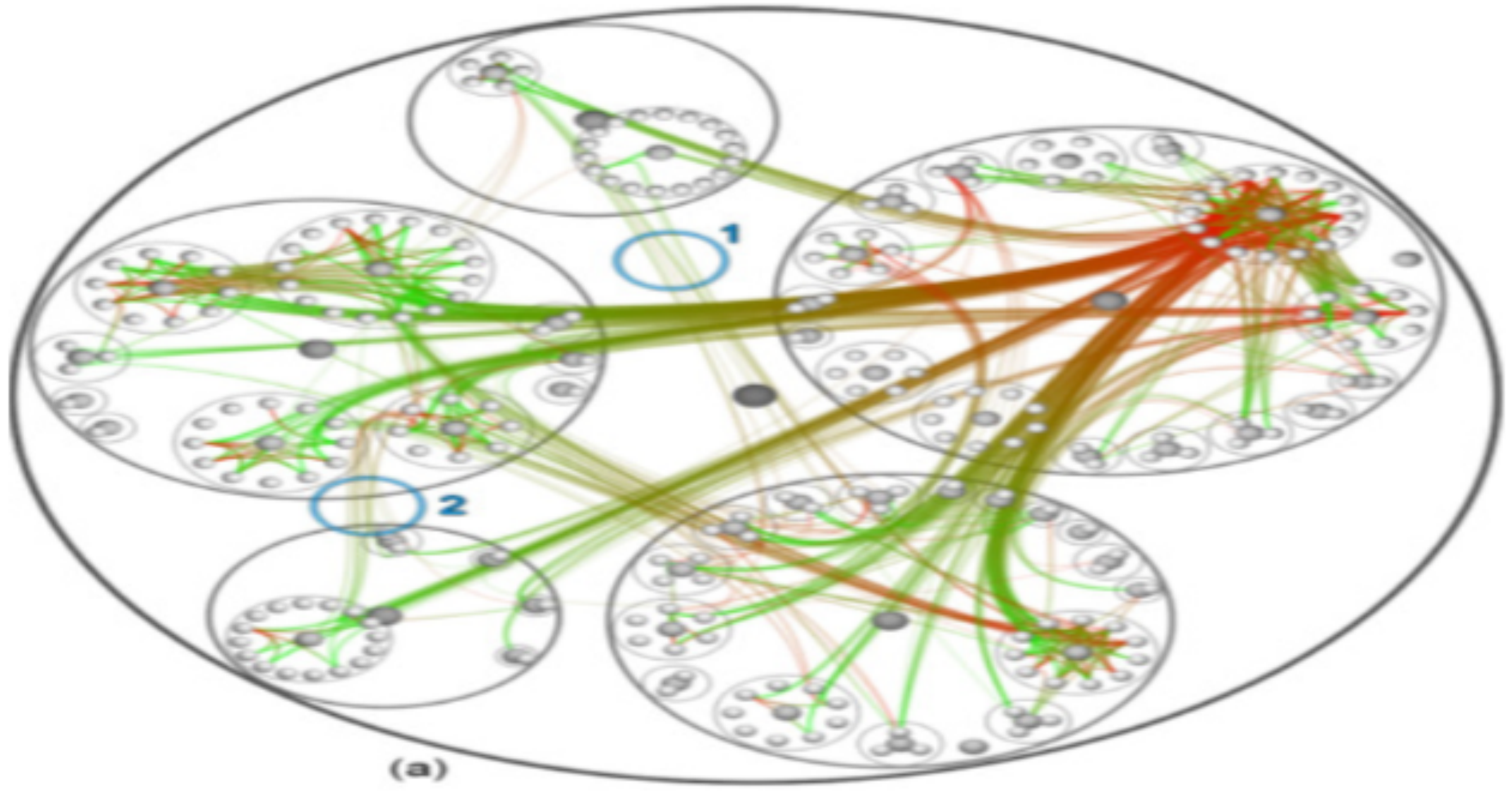
(a) Matrix visualization layout [21].



(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.

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.

Topological Feature-Based Layout

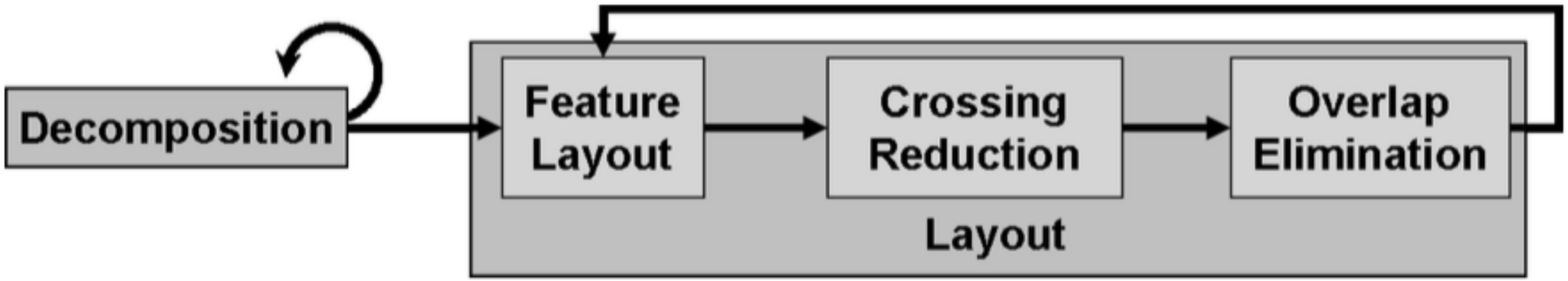


Fig. 1. TopoLayout algorithm phases.

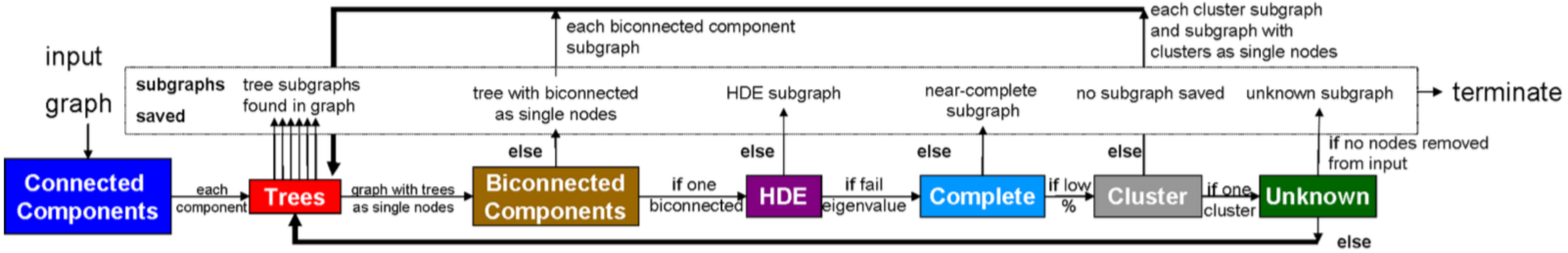


Fig. 3. Decomposition phase for TopoLayout. Detection algorithms in boxes coloured by feature type as in Figure 2. If a clause on a horizontal is true, we transition along the arrow. Otherwise, we follow the vertical arrow to save some subgraphs and recursively decompose others. Bold arrows indicate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

ArchambaultMunznerAuber2007

Feature Hierarchy

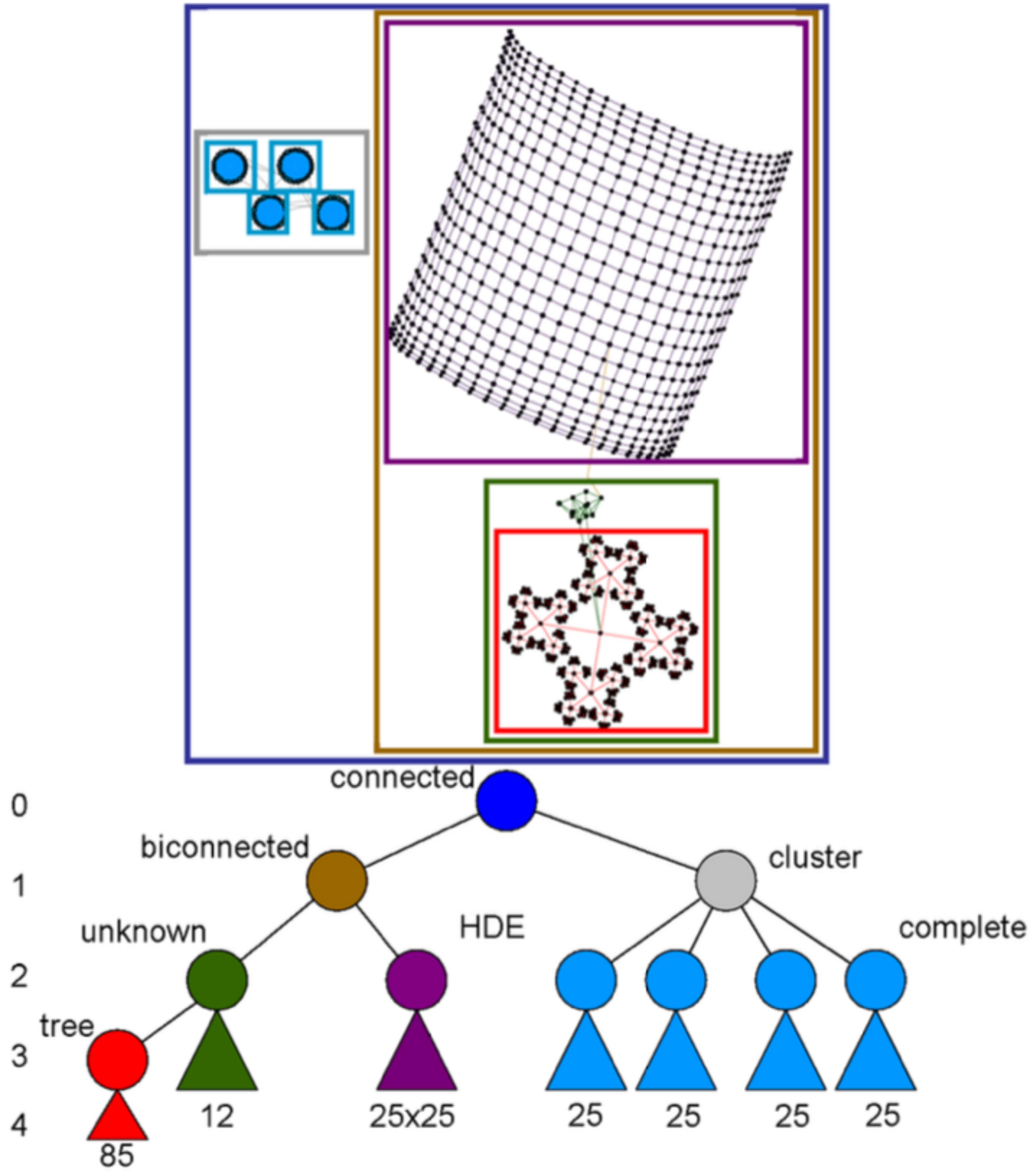
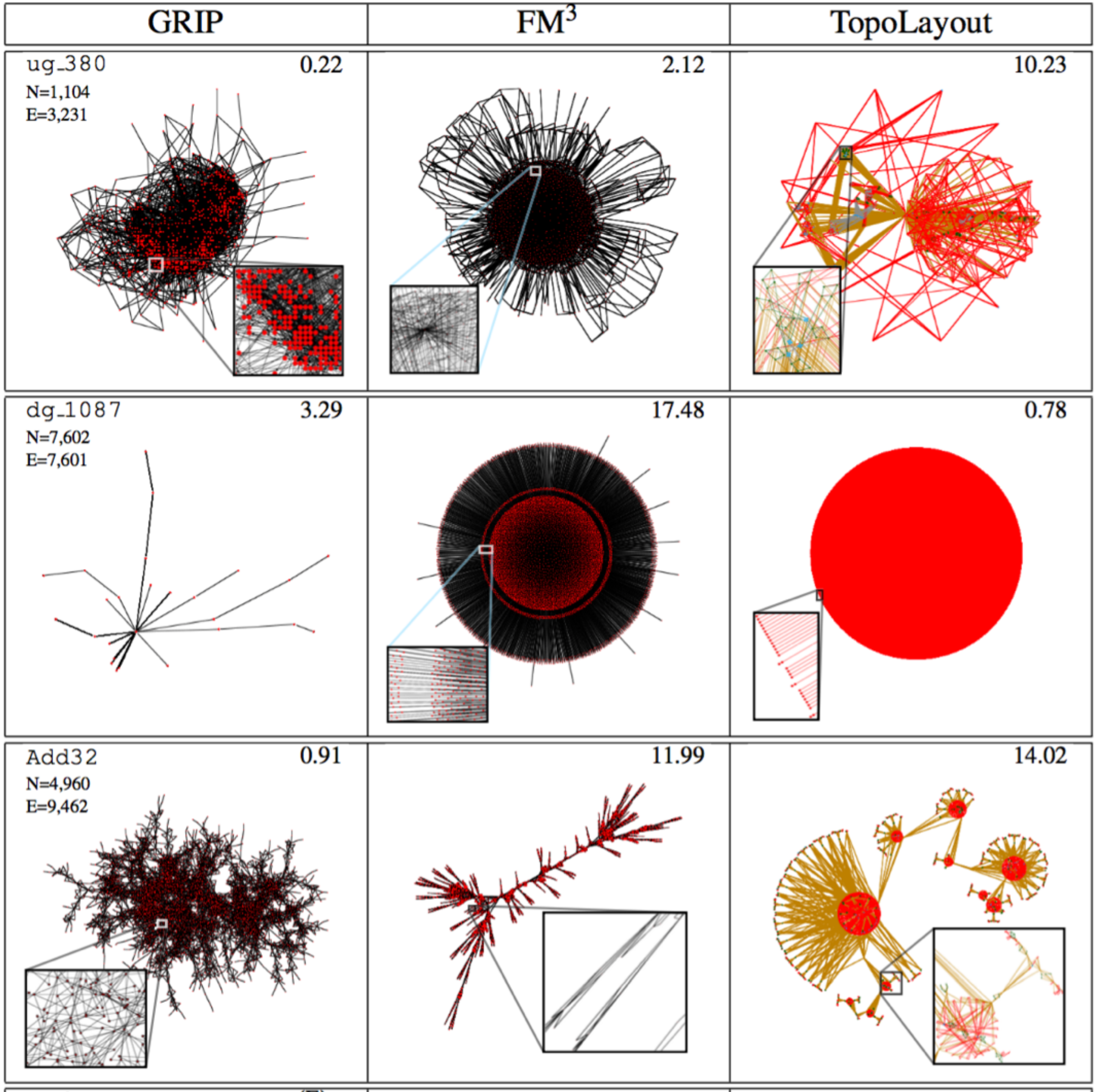


Fig. 2. Feature hierarchy after decomposition, with topology encoded by colour. **Top**: Layout annotated with bounding boxes to show hierarchy structure: meta-nodes encompass the subgraphs of their children. **Bottom**: Diagram of feature hierarchy, with levels enumerated and nodes labeled by feature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

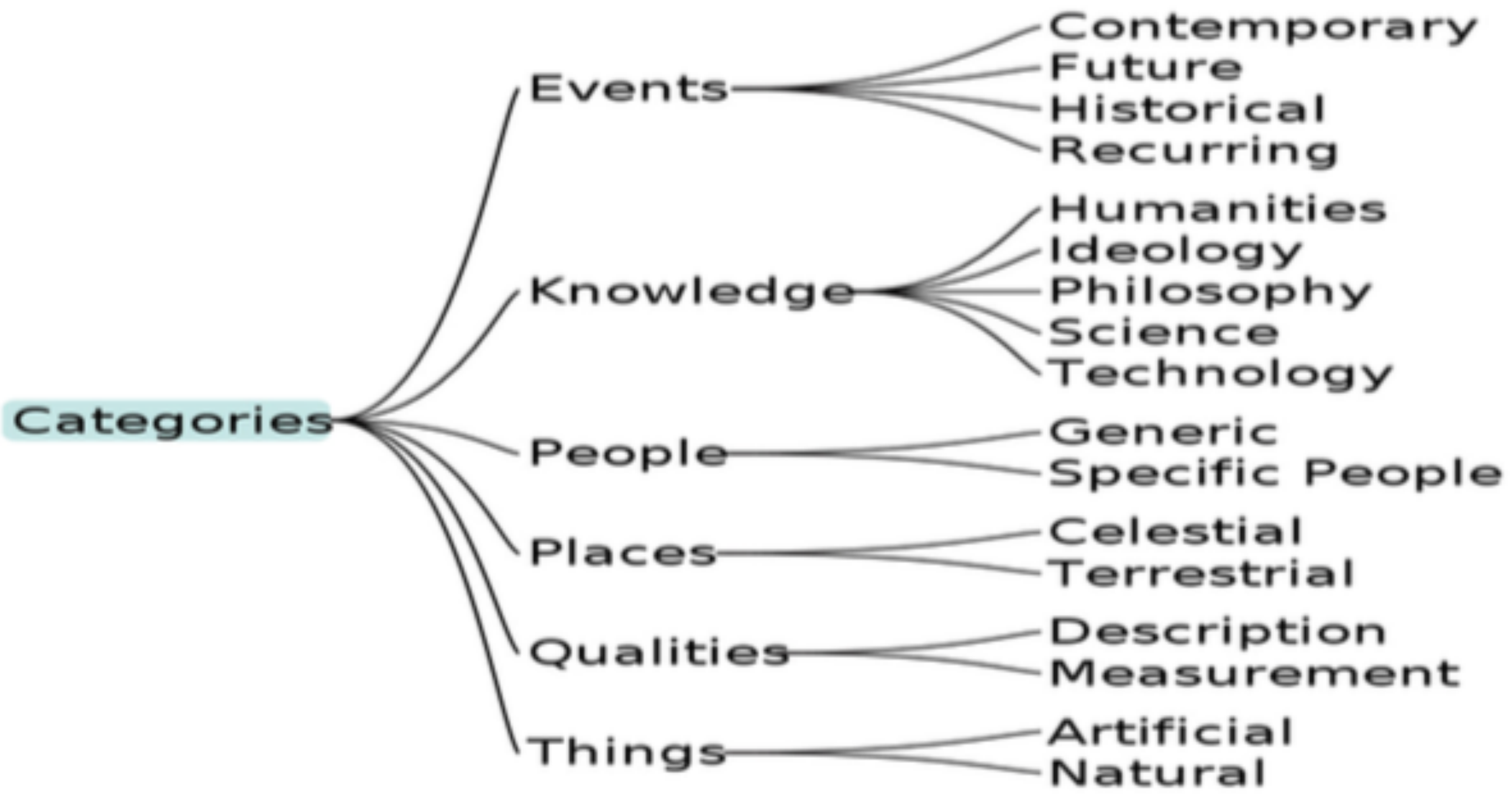
ArchambaultMunznerAuber2007



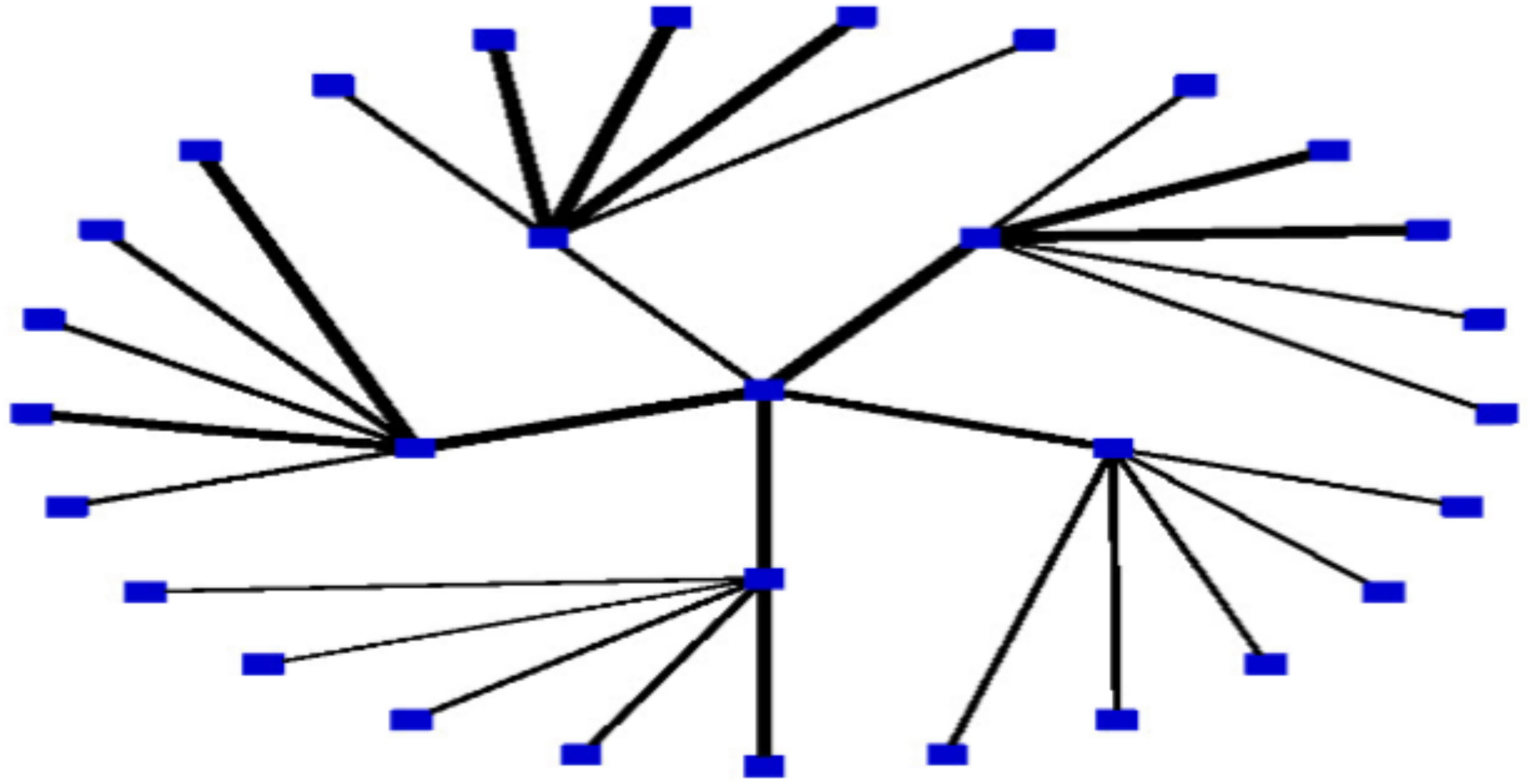
ArchambaultMunzner
Auber2007

2
Tree Layouts

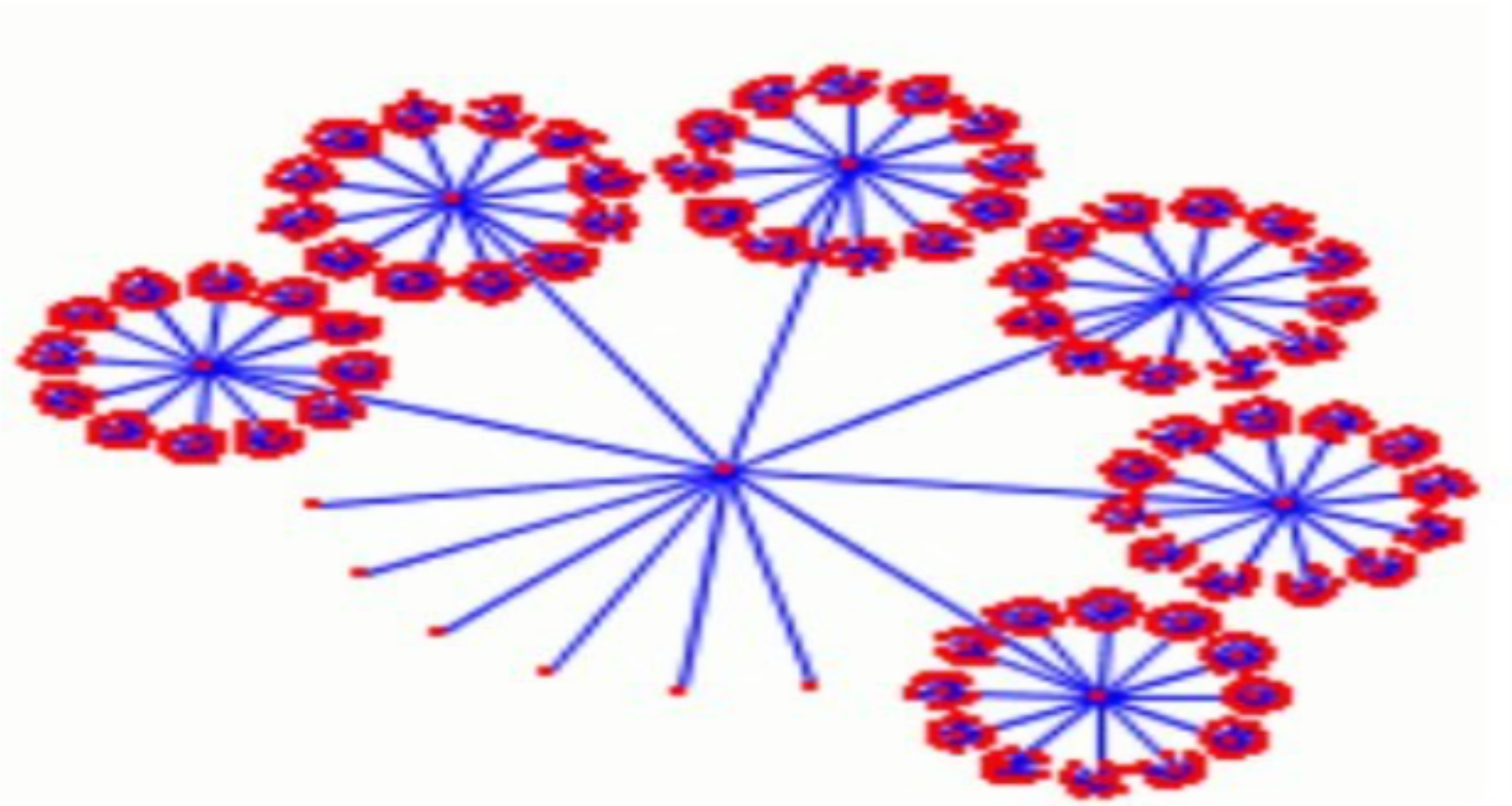
Node-Link Tree Layout



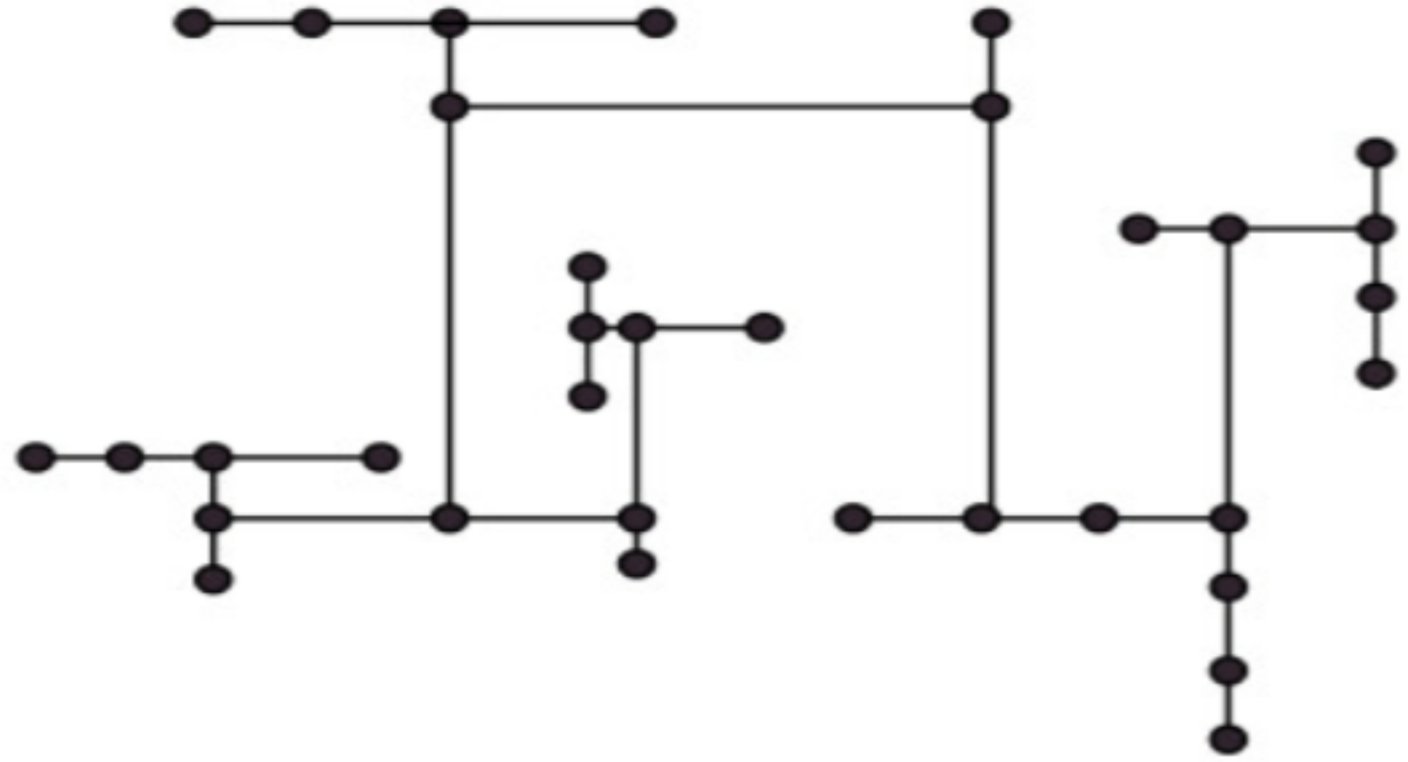
(a) Classical tree layout, produced with [19].



(b) Radial tree layout Example.



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

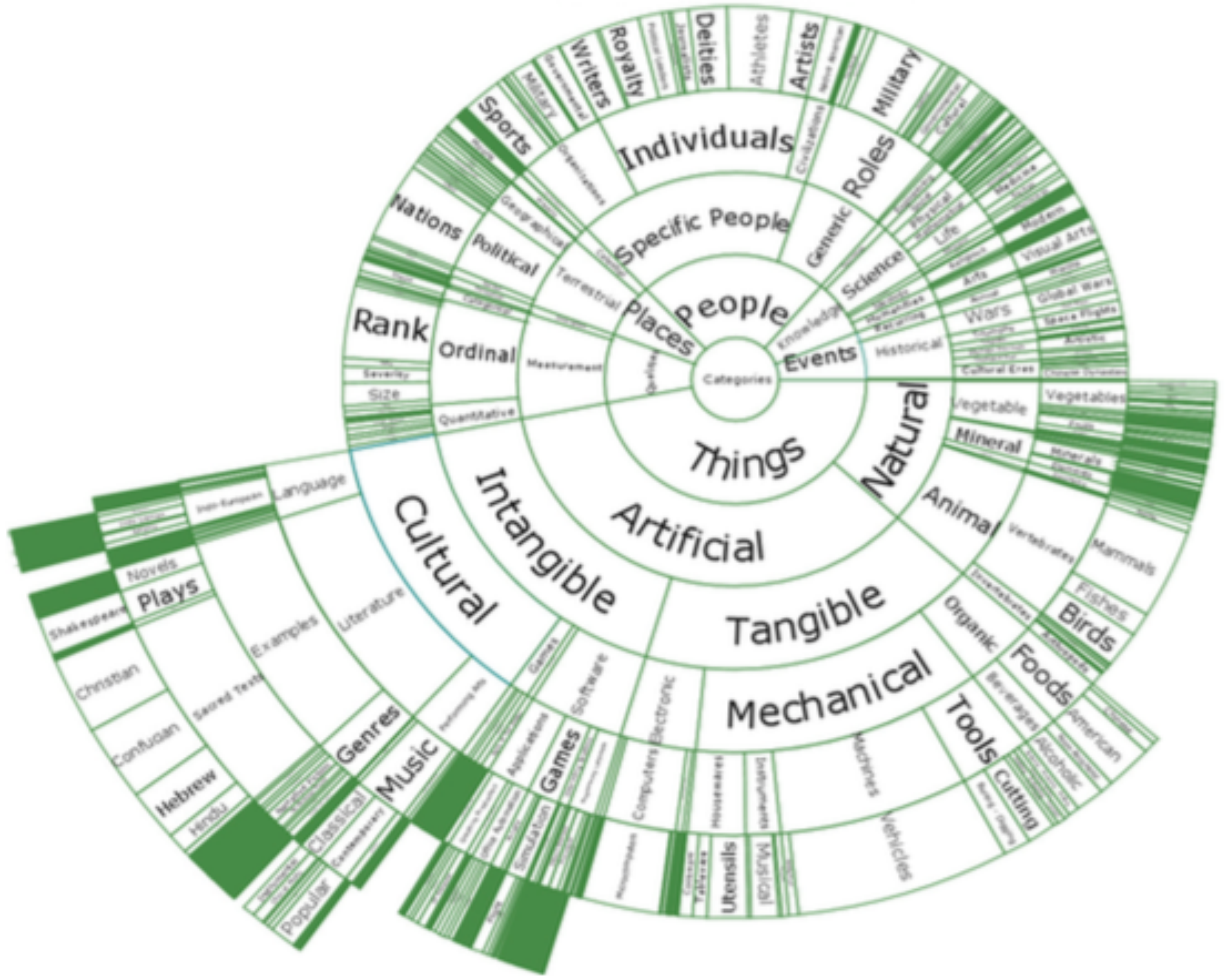


(d) H-Tree layout: produced by [22].

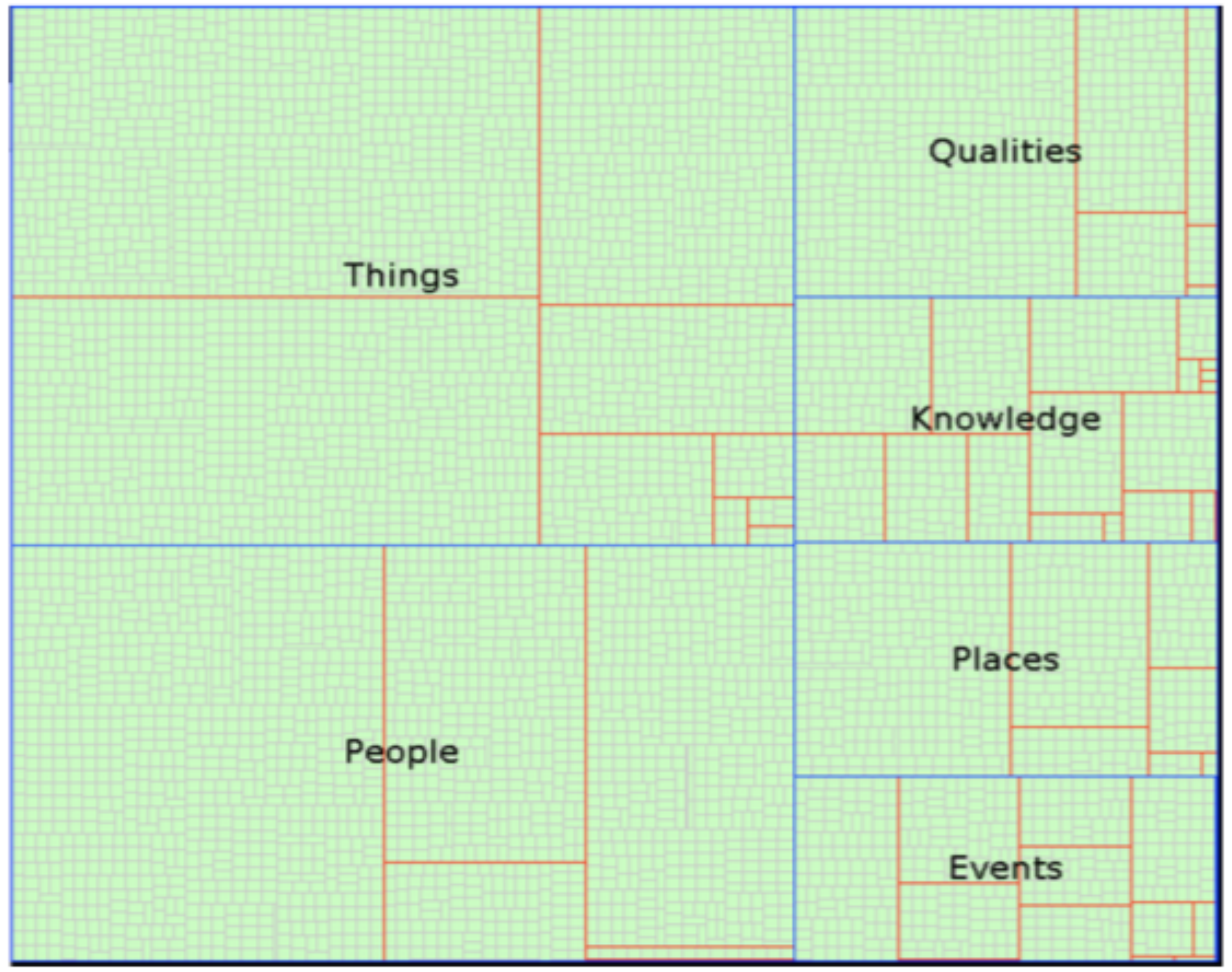
Figure 2 Tree Layout Examples.

GajerGoodrichKobourov2004

Space-Filling



(a) SunBurst layout.



(b) TreeMap layout.

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

GajerGoodrichKobourov2004

Coming up…

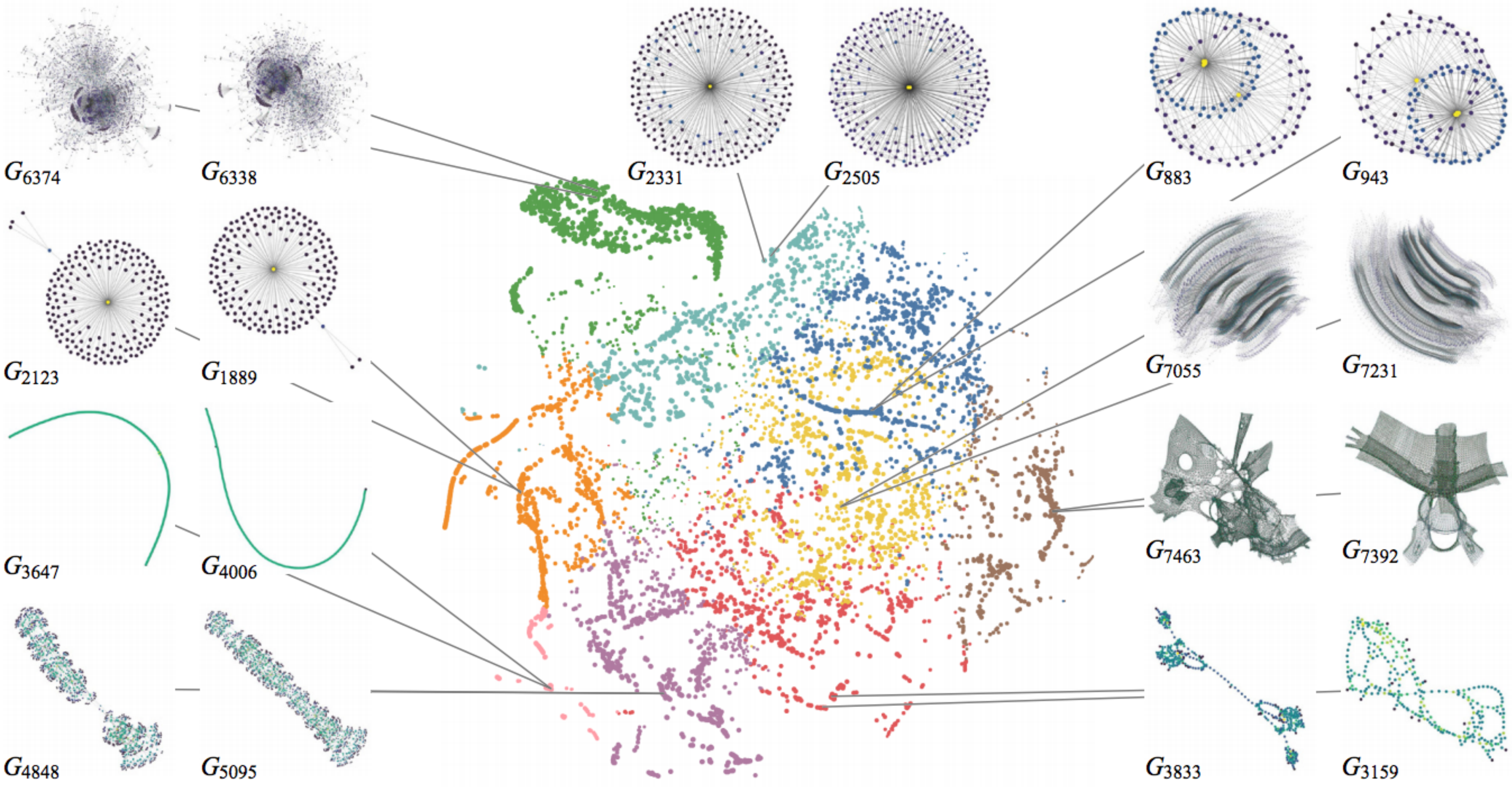
3
Matrix Visualization

4
3D Layout

5

Node and edge clustering

Machine learning approach to large graph visualization



KwonCrnovrsaninMa2018

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