Graph Drawing

Yuntao Jia
yjia3@illinois.edu
CS598 Information Visualization
02/03/2010

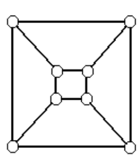
Reference

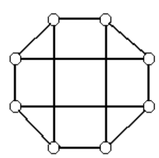
- Isabel F. Cruz, Roberto Tamassia. Graph Drawing Tutorial.
- Jesper Nederlof, Roeland Luitwieler. Introduction to Graph Drawing.
- Ioannis G. Tollis, Giuseppe Di Battista, Peter Eades, Roberto Tamassia. Graph Drawing: Algorithms for the Visualization of Graphs
- Graph Drawing, Tom Germano
- Tamassia, R. 1998. Constraints in Graph Drawing Algorithms. Constraints 3, 1 (Apr. 1998), 87-120.
- Eades, P. (1984). A heuristic for graph drawing. Congressus Numerantium, 42:149–160.
- K. Sugiyama & K. Misue. (1995). Graph drawing by magnetic-spring model. J. Visual Lang. Comput. 6(3).
- T. Kamada & S. Kawai. (1989). An algorithm for drawing general undirected graphs. Inform. Process. Lett. 31: 7–15.
- T. Fruchterman & E. Reingold. (1991). Graph drawing by force-directed placement. Softw. – Pract. Exp. 21(11): 1129–1164.
- R. Davidson & D. Harel. (1996). Drawing graphics nicely using simulated annealing. ACM Trans. Graph. 15(4): 301–331.
- M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. Czechoslovak Mathematical Journal, 25(100):619–633, 1975.
- Frishman, Y. and Tal, A. 2007. Multi-Level Graph Layout on the GPU. IEEE Transactions on Visualization and Computer Graphics 13, 6 (Nov. 2007), 1310-1319.
- Auber, Y. Chiricota, F. Jourdan, G. Melancon. Multiscale visualization of small world networks. INFOVIS 2003. pp. 75-81.

- ***Many slides and content are borrowed from those references***

Graphs

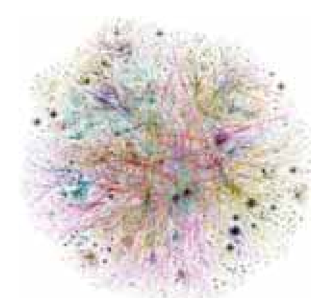
- G = {V, E}, nodes, edges
- Undirected graph, di-graph, weighted graph
- Planar graph drawing





Graph Drawings

- Visualization of *graphs/networks*
  - Models, algorithms, systems
- Applications
  - *Software engineering*
  - *Database systems*
  - *Project management*
  - *Knowledge representation*
  - *Telecommunications*
  - *WWW*



Internet map from the *Opte Project*

Graph Drawings

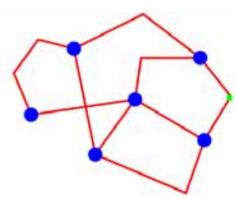
- Readable to users
- Follow drawing conventions
- Satisfy as many aesthetic rules as possible
- Efficient running time

Outline

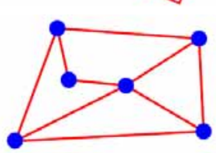
- Drawing Conventions
- Aesthetic Criteria
- Forced-directed method
- Multiscale method

Drawing Conventions

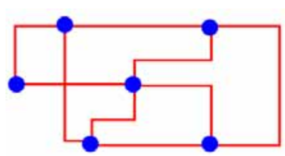
- Polyline drawings



- Straight-line drawings

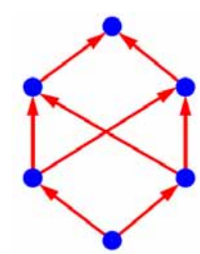


- Orthogonal drawings



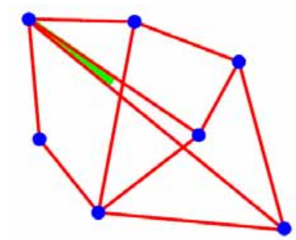
Drawing Conventions

- Planar drawings
  - No crossings allowed
- Upward drawings
  - Drawn as nondecreasing arcs
  - For hierarchical relationships
- Downward drawings
  - Drawn as nonincreasing arcs



Drawing Conventions

- Resolution
  - Smallest distance between vertices
  - Smallest distance between vertices and nonincident edges



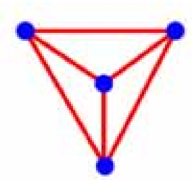
- Angular resolution
  - Smallest angle formed by two incident edges at a vertex

Aesthetic Criteria

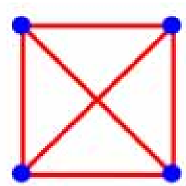
- Minimize edge crossings
- Minimize area
- Minimize bends (orthogonal drawing)
- Maximize angular resolution
- Symmetry
- Min. sum / maximum / variance of edge lengths

Aesthetic Criteria

- "In general, one cannot simultaneously optimize two aesthetic criteria"



min # crossings

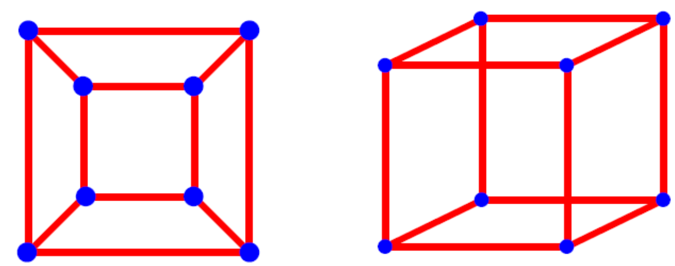


max symmetries

- Complexity
  - Minimizing edge crossing is NP-hard
  - Computing optimal angular resolution is NP-hard

Aesthetic Criteria

- Beyond aesthetic criteria



General Undirected Graph

- Planar straight-line drawings
  - Generates as few edge crossings as possible
- Forced directed method
- Multiscale method

Forced Directed Method

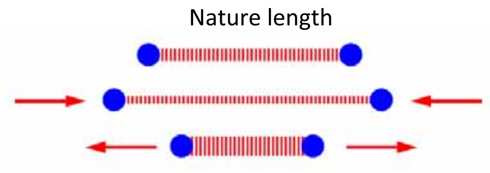
- Define a system of forces acting on the vertices and edges
- Find a minimum energy state by
  - Solving differential equations or
  - Simulating the evolution of the system

Forced Directed Method

- Spring embedder [Eades 1984]
- [Kamada and Kawai 89]
- [Fruchterman and Reingold 90]
- [Davidson and Harel 96]

Spring Embedder

- Replace an edge with a spring of unit length



- Connect nonadjacent nodes with springs of infinite nature length

Spring Embedder

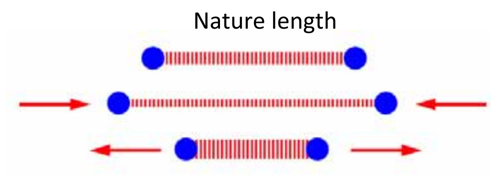
- Hooke's law

$$f = -\mathrm{k}(\mathrm{x} - \mathrm{x}_0)$$

  - f is the force
  - k is the factor
  - x_0 is the nature length
- Force model deviates from Hooke's law

Spring Embedder

- Replace an edge with a spring of unit length



$$f_a = c_a \log(\mathrm{r})$$

  - f_a is the attraction force
  - c_a is the attraction factor
  - r is the distance between nodes

Spring Embedder

- Connect nonadjacent nodes with springs of infinite nature length

$$f_r = \frac{c_r}{\mathrm{r}^2}$$

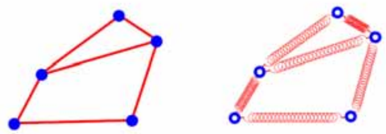
  - f_r is the repulsive force
  - c_r is the repulsive factor
  - r is the distance between nodes

Spring Embedder

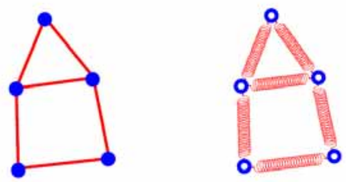
- Initial graph layout randomly
- Update layout iteratively
  - Apply spring forces to connected node pairs
  - Apply spring forces to unconnected node pairs
  - Update layout
  - Until the movements are small enough

Spring Embedder

- Initial layout and springs



- Final configuration



Forced Directed Method

- [Kamada and Kawai 89]
  - For a pair of nodes (u, v), the spring nature length is proportional to d(u, v) which is the shortest distance from u to v in the graph
  - Define energy of the system

$$\sum k_{u,v}(|p_u - p_v| - d(u,v))\text{\textasciicircum}2$$

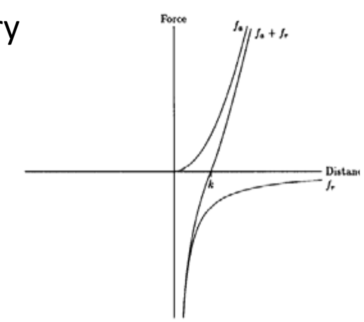
  - Reduce energy iteratively by solving PDE equations

Forced Directed Method

- [Fruchterman and Reingold 90]
  - Control drawings within a boundary
  - A complex system of forces

$$f_a = \frac{r^2}{k} \qquad f_r = -\frac{k^2}{r}$$

Where $k = C\sqrt{\frac{area}{\#nodes}}$



  - A “temperate” controlling scheme
    - Temperature from hot to cold and bounds vertex movement

Forced Directed Method

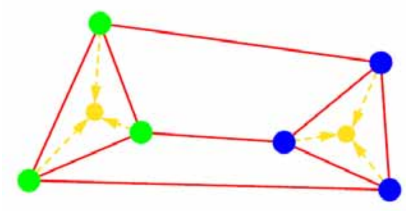
- [Davidson and Harel 96]
  - Consider vertex distributions, edge lengths, crossing into a energy function instead of forces
  - Simulated annealing to find solutions
    - Computation costly

Forced Directed Method

- Advantages
  - Simple implementations
  - Easy to add new heuristics, constrains
  - Smooth evolution of layout
  - Supports 3D
  - Often detects and shows symmetries
  - Works well with small graphs

Forced Directed Method

- Add new constrains through means of forces
  - Positions constrains with fixed positions or prescribed regions
  - Orientation constrains with "magnetic field"
    - [Sugiyama Misue 95]
  - Group constrains by adding dummy "attractors"



Forced Directed Method

- Disadvantages
  - Slow, running time, convergence
  - Few theoretical supports of drawing quality
  - Difficult to support orthogonal and polyline drawings

Multiscale Method

- Motivation
  - Force directed method is slow on large graphs
  - The result is sensitive to the initial layout when dealing with large graphs

- Multi-Level Graph Layout on the GPU
  - [Frishman and Tel 07]

Multiscale Method

- Generate graphs at different spatial scales
  - Partitioning, clustering, coarsening
- Start from the coarsest scale and work back to finest scale
  - How to propagate results from one level to another

Multiscale Method

- Advantages
  - Reduce computation cost
  - Maintain good quality
  - Less sensitive to initial configurations
- Disadvantages
  - Growing inaccuracy when going from one level to another

Multiscale Method

- Graph spectral partitioning
- Multilevel scheme for graph layout
- Accelerate force-directed method

Multiscale Method

- Graph spectral partition
  - Partition a graph to parts with two properties
    - Similar size <-> balance in layout
    - Minimum cut <-> weakly coupled, independent
- An eigenvector problem [Fiedler 75]
  - Given a graph G, its Laplacian L is defined as

$$\ell_{i,j} := \begin{cases} \deg(v_i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise.} \end{cases}$$

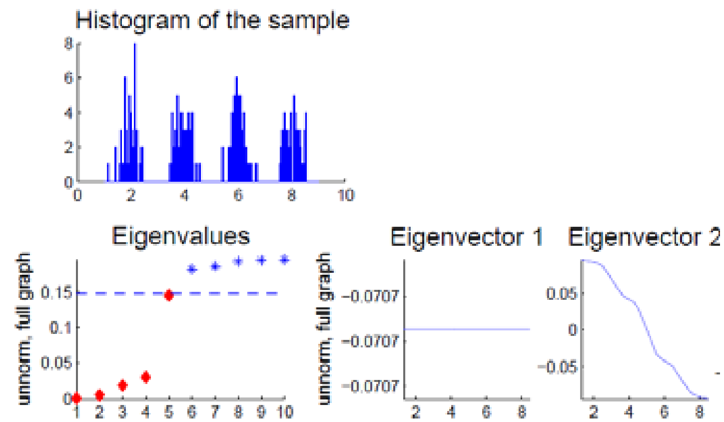
Multiscale Method

- Graph Laplacian L with eigenvalues $\lambda_0 \le \lambda_1 \le \cdots \le \lambda_{n-1}$
  - Positive semi-definite $\forall i, \lambda_i \ge 0, \quad \lambda_0 = 0$
  - # zero-eigenvalue = # connected component
  - Smallest non-zero eigenvalue => spectral gap, its eigenvector can be used to cluster the graph, namely “spectral clustering”

http://en.wikipedia.org/wiki/Spectral_clustering#Spectral_clustering

Multiscale Method

- Spectral clustering, a simple example



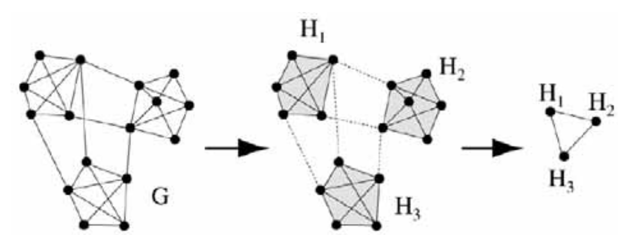
Ulrike von Luxburg, A Tutorial on Spectral Clustering

Multiscale Method

- To compute the eigenvector
  - Power iterations
    - Efficient in computation and memory
    - Slow convergence

Multiscale Method

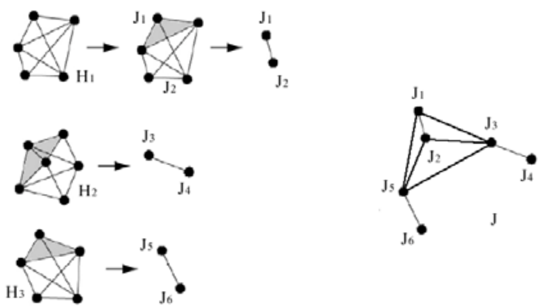
- Multilevel representation
  - First level partitioning



David Auber et al, Multiscale Visualization of Small World Networks

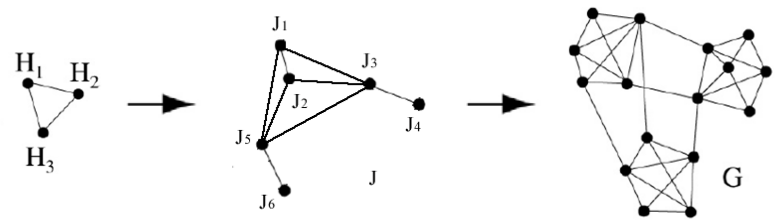
Multiscale Method

- Multilevel representation
  - Second level partitioning



Multiscale Method

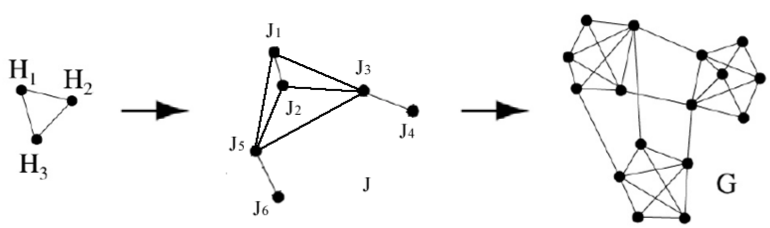
- Multilevel representation
  - Combined



First level Second level Entire graph

Multiscale Method

- Compute layout
  - Start from the coarsest level
  - Propagate to finer level, then refine



First level Second level Entire graph

Multiscale Method

- Layout propagation
  - Each node in current level is placed at its parent's position in the coarser level
  - Scale positions

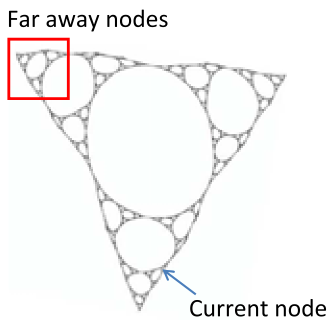
$$p_i(x,y) = \sqrt{\frac{|V(L^l)|}{|V(L^{l-1})|}} \cdot p_i(x,y).$$

  - Improve positions

$$p_i = \frac{1}{2}\Big(p_i + \frac{1}{degree(i)} \sum_{j \in N(i)} p_j\Big).$$

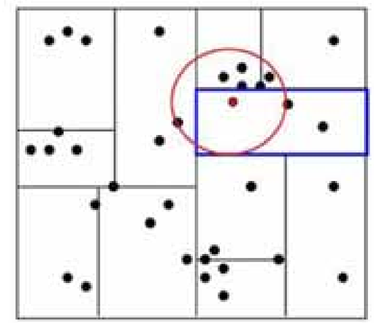
Accelerate Force-directed Method

- Bottlenecks
  – Repulsive forces between all pairs of nodes
  – N-body problem, O(|V|^2)
- Solutions
  – For nodes that are far away, use approximations



Accelerate Force-directed Method

- Kd-tree
  – Spatial partition of nodes
  – Give a particular node
    - Nodes out of a range or its cell are far away nodes
    - Use cell centers to approximate repulsive forces



Region query on a kdtree

Accelerate Force-directed Method

- Parallel implementation

Pseudo code:

```
For  all nodes in parallel
    For all other nodes
        compute and accumulate repulsive forces
```

- "Embarrassingly parallel"
- Suits GPU and multi-core architectures

Summary

- Graph drawing
  – Conventions
  – Aesthetic criteria
- Drawing algorithm for undirected graphs
  – Force directed methods
  – Multiscale methods

Summary

- Graph drawing
  - Conventions
  - Aesthetic criteria
- Drawing algorithm for undirected graphs
  - Force directed methods
  - Multiscale methods

- Questions?