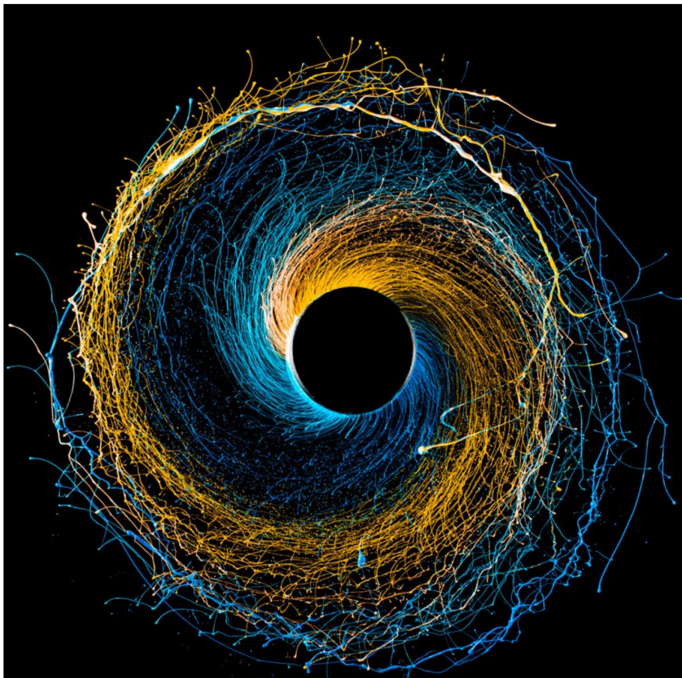
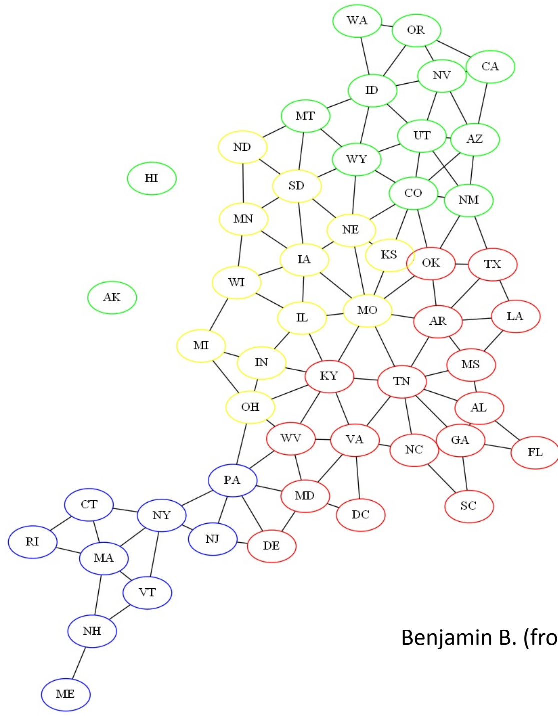
Graph Visualization I

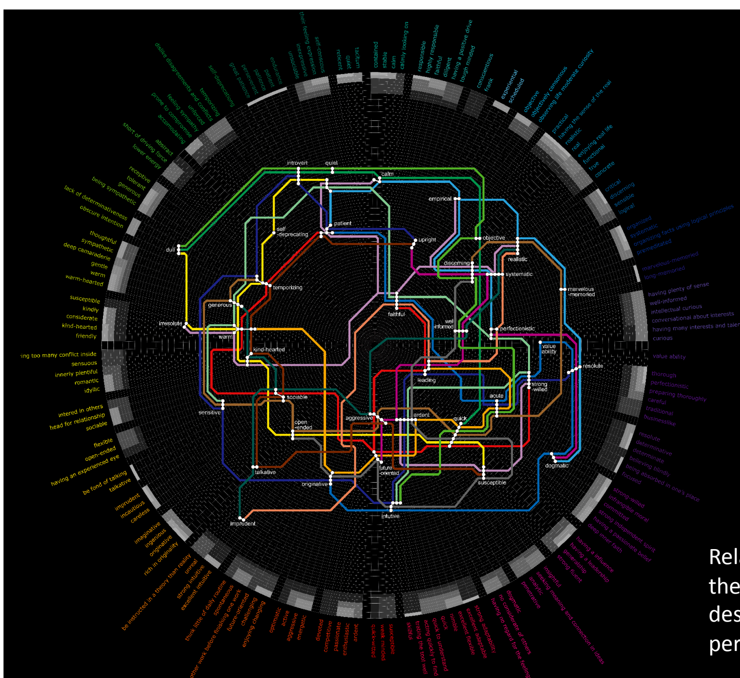


"Various shades of acrylic paint are dripped onto a metallic rod, which is connected to a drill. When switched on, the paint starts to move away from the rod."

http://fabianoefner.com/?portfolio=black-hole-2



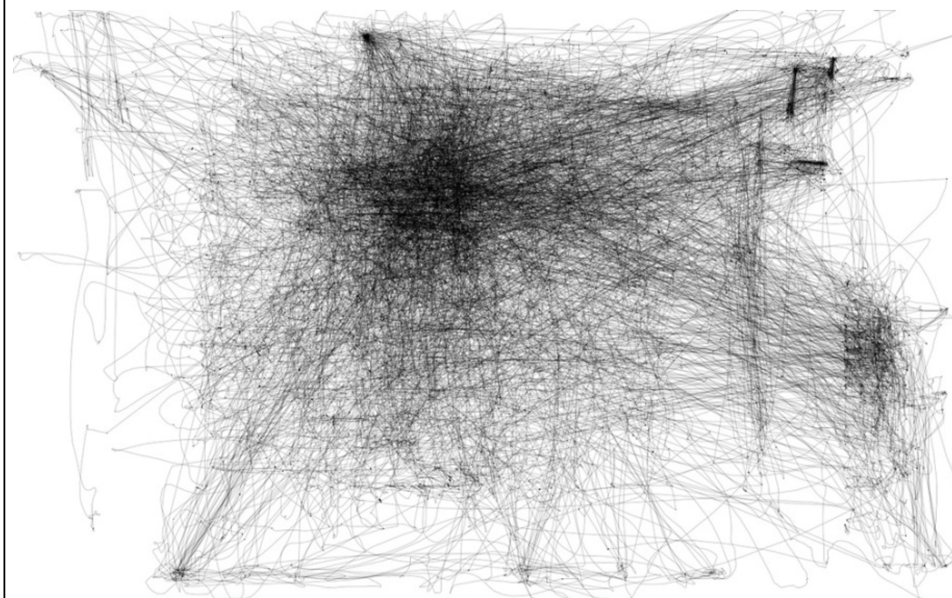
Benjamin B. (from Spring '12)



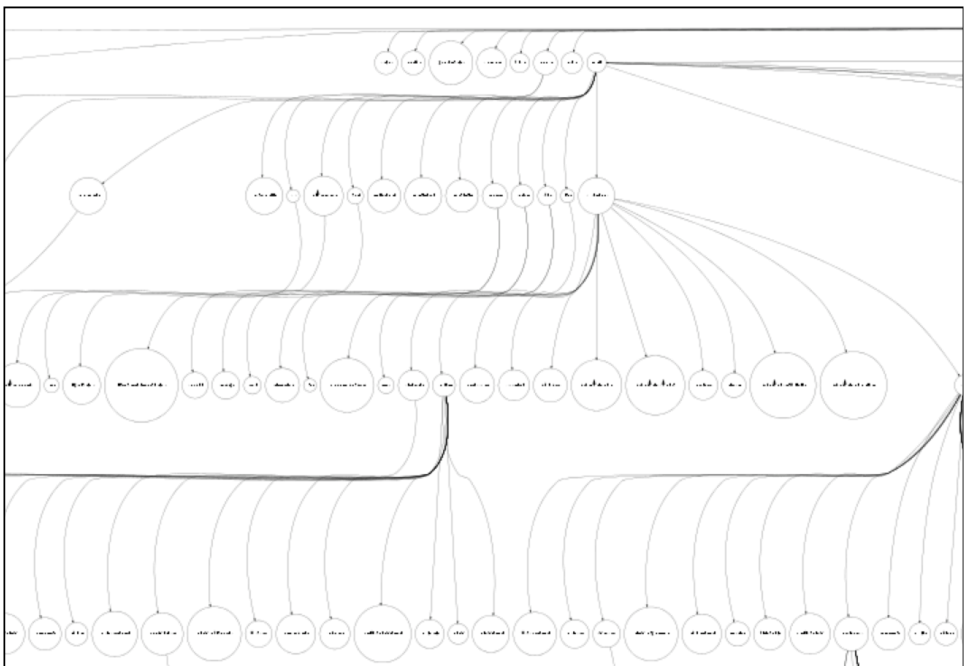
Many datasets don't (initially) have coordinate systems!

Relationship among the words that describe people's personalities

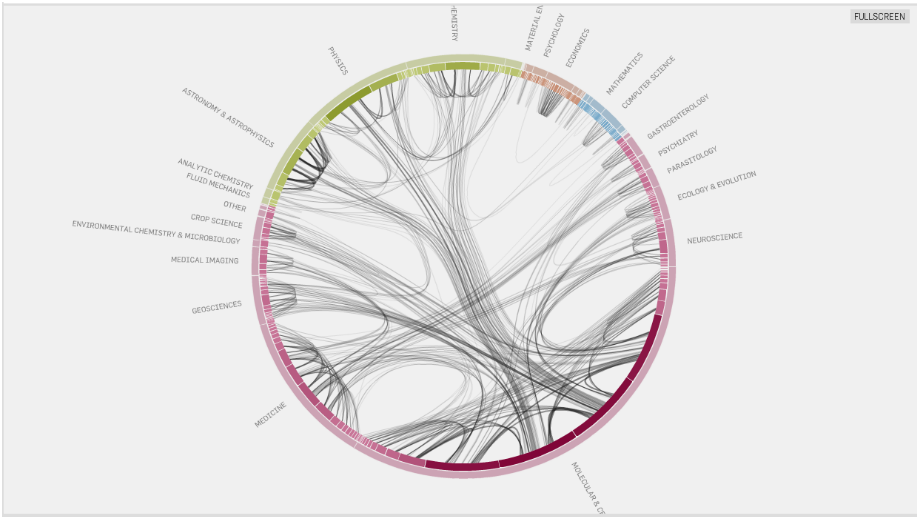
http://madang.ajou.ac.kr/~kwlee/images/theMBTImap_actual%20poster%20image.pdf



IOGraphica http://iographica.com



Justin J. (from Spring '12)



Stefaner, Moritz. "Citation Patterns." *Wellformed Eigenfactor*. N.p., 2008. Web.

Today

- Questions about Homework 2?
- Some Graph Terminology
- Graph Drawing Goals, Questions, & Challenges
- Some Related Terms/Algorithms (mentioned indirectly in the reading)
- Reading for Today: "Force Directed Graph Drawing"
- Reading for Friday

Homework Assignment 2: Graph Visualization w/ GraphViz

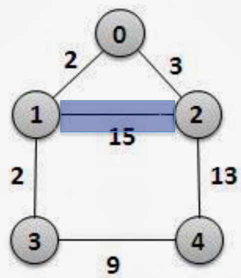
due Thursday @ 11:59pm

- Download GraphViz and explore the examples
- Using your favorite programming language, write code to create a range of synthetic input files
  - a tree, a clique, a planar graph, bipartite graph, etc.
  - medium size (a "good, ~optimal" layout *could* be done by hand)
- Experiment with the visual options (layout, color, line style, label font, shapes, etc.)
- Create a graph (or multiple) of our social network, using the data we gathered in Lecture 1.
- Analyze the quality of the results. Note strengths & weaknesses of GraphViz.

Testing our Hypotheses

- Please enter you data (if you haven't already)
- What (if anything) surprised you about the data?

knew before RPI	share(d) dormroom or apartment	met in CSCI 1200 Data Structures	met at RPI	met today!		total
0	0	11	7	22		40
0	1	2	3	34		40
0	0	2	1	37		40
0	1	3	7	29		40
0	0	2	5	33		40
0	0	2	3	35		40
0	2	1	4	33		40
0	0	4	5	31		40
0	0	1	5	34	ka	40
0	0	1	6	33	lar	40
0	4	2	3	31		40
0	0	0	4	36		40
0	0	1	8	31		40
0	0	0	2	38		40
0	1	1	1	37	XD	40
0	0	8	4	28		40
0	1	2	9	28		40
0	0	1	5	34		40
0	0	4	1	35		40
0	0	3	6	31		40
0	0	1	4	35		40
0	0	1	0	39		40
0	0	0	3	37		40
0	0	0	5	35		40
1	0	2	1	36		40
						0
0.04	0.4	2.2	4.08	33.28		



	0	1	2	3	4
0	0	2	3	0	0
1	2	0	15	2	0
2	3	15	0	0	13
3	0	2	0	0	9
4	0	0	13	9	0

Adjacency Matrix Representation of Weighted Graph

http://www.thecrazyprogrammer.com/2014/03/representation-of-graphs-adjacency-matrix-and-adjacency-list.html

Today

- Questions about Homework 2?
- Some Graph Terminology
- Graph Drawing Goals, Questions, & Challenges
- Some Related Terms/Algorithms (mentioned indirectly in the reading)
- Reading for Today: "Force Directed Graph Drawing"
- Reading for Friday

Graph Terminology I

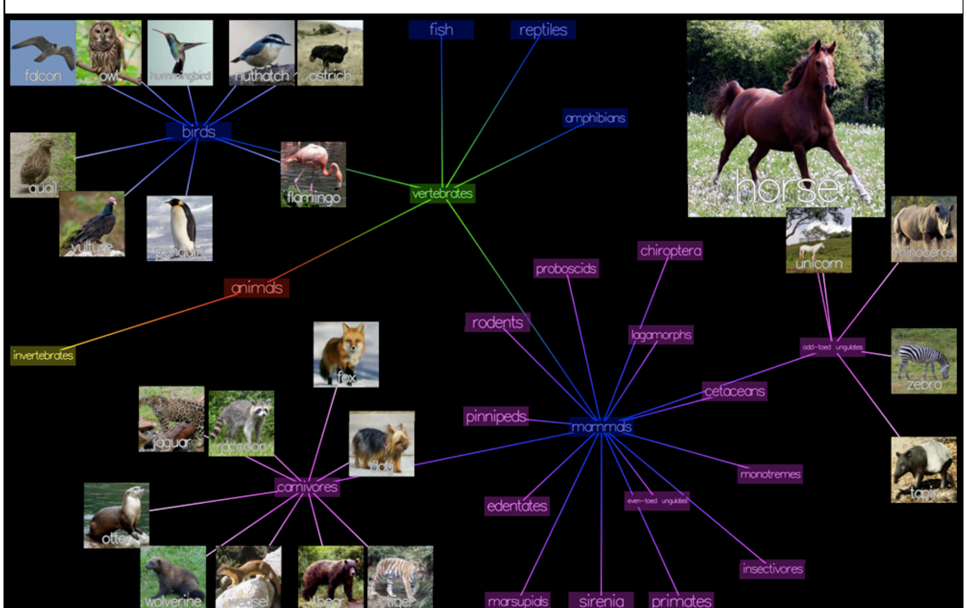
- Directed / Undirected Edges
- Tree – no cycles
- Cycle – A path along edges through the graph starting & ending at the same vertex. Variants: closed walk, simple cycle, directed cycle, …
- Valence (a.k.a. Degree) of a Vertex - # of edges incident on the vertex
- Regular – Each vertex has same valence, a 3-regular graph is also called cubic
- Polygon – 2D flat or on a sphere, with straight or great circle edges
  Polyhedron – 3D solid formed by flat faces
  Polytope – flat sides in any dimension
- Bipartite – vertices can be split into two groups, A & B. No edge connects a vertex in A to another vertex in A. Same for B.

Graph Terminology II

- Clique – subset of vertices in an undirected graph with an edge connecting every pair of vertices in the subset.
- Upward Drawing (of a tree) – no child is drawn with vertically above (higher y value than) its parent.
- Plane Graph – A 2D drawing of the graph where no edges cross (touching at the endpoint vertices they share is ok)
- Planar Graph – A graph for which a Plane Graph exists.
- Euler's Theorem for planar graphs. For a plane graph with n vertices, m edges and f faces, we have $n - m + f = 2$.

Today

- Questions about Homework 2?
- Some Graph Terminology
- Graph Drawing Goals, Questions, & Challenges
- Some Related Terms/Algorithms (mentioned indirectly in the reading)
- Reading for Today: "Force Directed Graph Drawing"
- Reading for Friday



Graph Drawing Goals

- Automated!
- Can read all of the labels
- Can follow the line and see exactly which 2 vertices it connects
- Aesthetically pleasing
- Layout should display as much symmetry as possible
- Crossing free or minimal-crossing layout
- All edge lengths are approximately equal
- Even vertex distribution
- Distance between nodes in final layout should be as close as possible to "graph distance" (# of edges on shortest path between those nodes)

Graph Drawing Questions

- What is the metric of success for each of our goals?
- Can we guarantee to find a solution? The optimal or best solution?
- Can we use randomness? Does it help?
- How expensive/slow are the different algorithms to draw graphs?
- How does it scale with more nodes/edges?
  - Does it lose effectiveness in meeting our goals?
  - How is the running time affected?
- How do we label the nodes/edges with color/words/images?

Graph Drawing Challenges

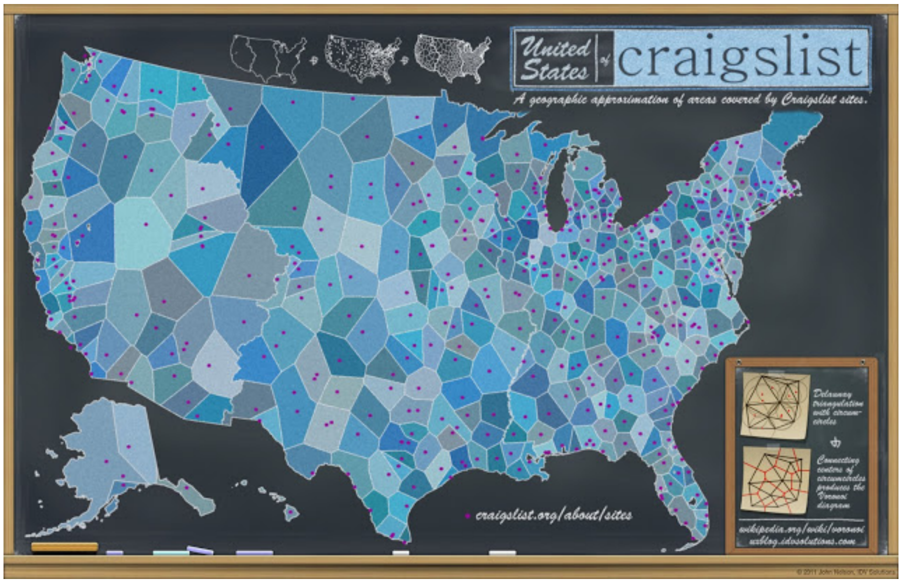
- What if the graph is non planar?
- What if the graph has many nodes & edges?
  - ~40-100 works well for simple force-based methods
  - # of springs = # of edges?
    Or # of springs >> # of edges?
  - Computation & convergence & getting stuck in a local minimum
- Does 3D (or 4D or …) or layout on the surface of a sphere or torus or … non Euclidean space help?

Today

- Questions about Homework 2?
- Some Graph Terminology
- Graph Drawing Goals, Questions, & Challenges
- Some Related Terms/Algorithms (mentioned indirectly in the reading)
- Reading for Today: "Force Directed Graph Drawing"
- Reading for Friday

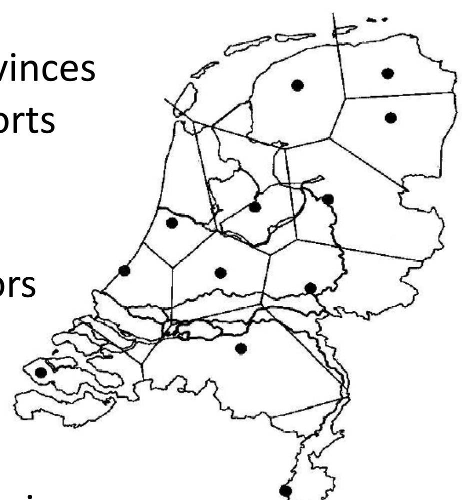
Voronoi Diagram

http://uxblog.idvsolutions.com/2011/07/chalkboard-maps-united-states-of.html



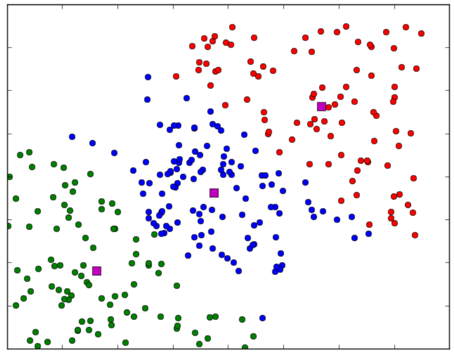
Voronoi Diagram/Cells/Regions

- How to re-district the Netherlands into provinces so that everyone reports to the closest capital
- Cell edges are the perpendicular bisectors of nearby points
- 2D or 3D
- Supports efficient *Nearest Neighbor* queries

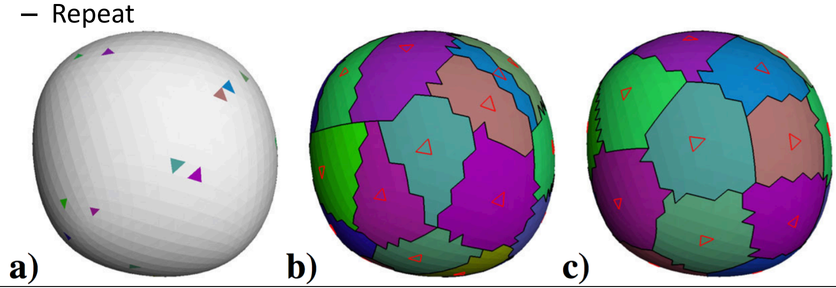


http://ccc.inaoep.mx/~rodrigo/robotica/Trigui.pdf

K-means clustering

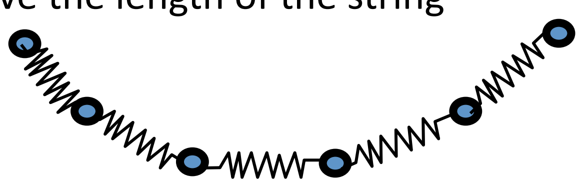


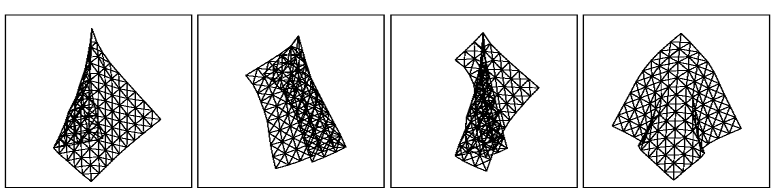
- Choose K
- Select K items at random, these are your leaders for the first round
- While (not done)
  - Every item joins the group of their "closest" leader
  - Each group elects a new leader
  - Repeat



String/Hair/Cloth Simulation

- Springs link the particles
- Springs try to keep their rest lengths and preserve the length of the string





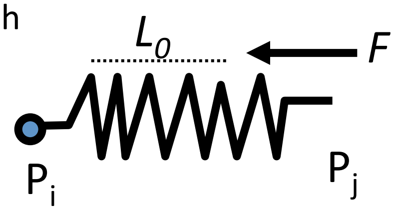
Interactive Animation of Structured Deformable Objects Desbrun, Schröder, & Barr 1999

Spring Forces

- Force in the direction of the spring and proportional to difference with rest length L_0

$$F(P_i, P_j) = K(L_0 - ||\vec{P_iP_j}||)\frac{\vec{P_iP_j}}{||\vec{P_iP_j}||}$$

- K is the stiffness of the spring
  - When K gets bigger, the spring really wants to keep its rest length



Using Springs for Graph Drawing

- Value for Spring Rest Length?
  - Rest length = 0, spring only attracts (doesn't repel)
  - Alternately, forces could act in
    - Both attract & repel (have non zero edge length?)
    - Only repel?
- What is the correct spring constant?
  - Too high/stiff -> system explodes (does not converge)
  - Too low -> takes too long to converge

exerting attractive and repulsive forces from one another." The attractive and repulsive forces are redefined to

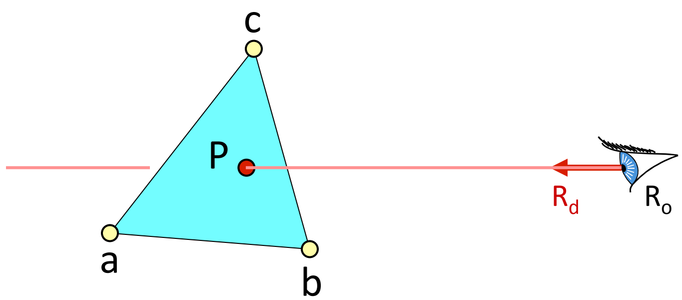
$$f_a(d) = d^2/k, \qquad f_r(d) = -k^2/d,$$

in terms of the distance d between two vertices and the optimal distance between vertices k defined as

$$k = C\sqrt{\frac{area}{number\ of\ vertices}}.$$

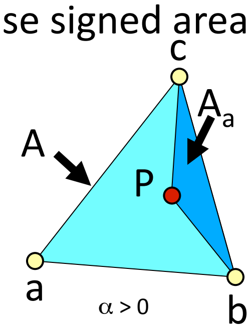
Barycentric Coordinates

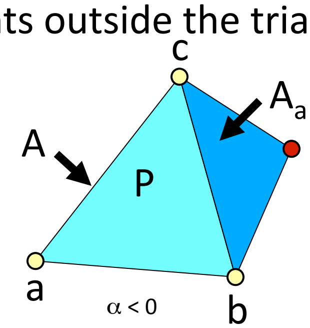
- $P(\alpha, \beta, \gamma) = \alpha a + \beta b + \gamma c$
  with $\alpha + \beta + \gamma = 1$
- If $0 < \alpha < 1$ & $0 < \beta < 1$ & $0 < \gamma < 1$
  then the point is inside the triangle!



How Do We Compute α, β, γ ?

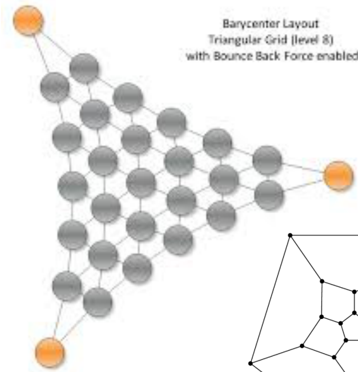
- Ratio of opposite sub-triangle area to total area
  - $\alpha = A_a/A$ $\beta = A_b/A$ $\gamma = A_c/A$
- Use signed areas for points outside the triangle



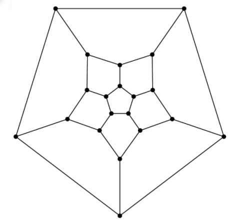


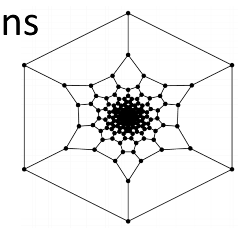
But how do I know if the point is outside the triangle?
That's what I was trying to determine!

Barycentric Graph Layout



- Planar graphs (only?)
- Start with some fixed positions
- Each vertex is placed at the average of its neighbors
- Can be solved as a system of equations
- Results can have poor resolution & poor distribution of vertices





Non Euclidean

- Hyperbolic layout has better equal distance layout for leaves of a "complete tree"
- Related to
  - Fisheye view
  - Focus + context
  - assumption that center is more important

Daina Taimina
http://www.math.cornell.edu/~dtaimina/Artexhibits.htm

Simulated Annealing

- “Temperature” changes (constants within the algorithm change) as # of iterations of algorithm increase.

Genetic Algorithms

Today

- Questions about Homework 2?
- Some Graph Terminology
- Graph Drawing Goals, Questions, & Challenges
- Some Related Terms/Algorithms (mentioned indirectly in the reading)
- Reading for Today: "Force Directed Graph Drawing"
- Reading for Friday

Reading for Today

- "Force Directed Graph Drawing" chapter by Steven Kobourov from the book *Handbook of Graph Drawing and Visualization 2013*

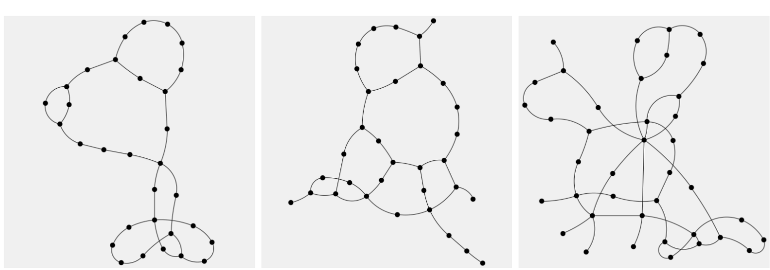


Figure 12.11 Examples of force-directed Lombardi drawings: note that every edge is a circular arc and every vertex has perfect angular resolution [CCG+11].

Your Comments

- Inspired by nature
  - RNA folding, molecular bonds
- Would be interesting to watch evolution of solution, incremental construction
- What about directed?
- Genetic algorithms too slow for graph drawing?
- Now understand why graph drawing algorithms crash when given 10,000 nodes
- Handle big graphs like divide & conquer for sorting
- Edge curvature can be detrimental
- Would like to see same graph, drawn by each different algorithm, and timing numbers for comparison
- Interesting to see inter-relations between publications, and which publications were most referenced

Force Directed Graph Drawing

- No domain-specific knowledge required
- Basic algorithm works well for ~100(s) nodes
  - Not good at larger, local minima, unreadably dense
- Use non linear forces (e.g., log), even distribution of vertices, temperature adjustments (simulated annealing)
- Keep vertices away from non-incident edges
  - How is this done? Extra vertices (w/ repuslsion springs) in the middle of edges?

Other Graph Drawing Variants

- Optimize a planar layout (no crossings)
- Improve a layout with intentional, meaningful crossings
- Non-Euclidean spaces (allow uniform distribution vertices, not biased towards nodes in center of drawing)
  - How do this work with interactive visualization? Would help for understanding of these non-Euclidean spaces.
- Multi-level layout technique
  - a.k.a. Hierarchical / Level of Detail

Dynamic Graphs

- Present relationships as they evolve over time
- Stack of graphs, slices of time, relationships in space & time
- Applications
  - Extracting information from a CVS version control system, what parts of the code are unstable over periods of time?
  - Inheritance graphs, program call stacks, control-flow graphs
- Goals:
  - Readability
  - Maintain mental model, consistency of placement of nodes over time
- Types
  - Aggregated views – multiple graphs displayed at once, in series?
  - Merged views – stacked on top of each other
  - Animations – one graph shown after the other
  - Morphing – fade in, fade out
- Animated vs Interactive?
  - User gets to move nodes around, expand, zoom in, focus, highlight nodes with certain properties

Writing quality/improvements

- Paper could have more illustrations ******
  - Problem of exponential area of barycentric method
  - Methods that prevent separation into clusters, or put high degree vertices at center
  - Is pseudocode effective/sufficient as an explanation?
  - Difficult to understand these algorithms without being able to form a mental model of how they work **
- Good overview, sufficient detail that could be implemented by person with a computer graphics background
  - but difficult to follow for students w/o ACG :(
- Want/need more details!
- Hard to keep track of papers (better to use technique/title than author name)
  - Chapter was somewhat disorganized
  - Order of papers was not always chronological
- Want more discussion, side-by-side comparison/contrast/pros-and-cons of the different algorithms
- Who is the audience for this chapter?
  - Chapter in book for new-to-field students –or– survey paper for experts in the field? Is there a difference?
- How do you choose which algorithm to use for specific data? No clear instructions
  - Show same (specially constructed to show differences) example graph stuffed into each algorithm
- Reads like a history book, would be nice to have more discussion of where these algorithms are used, which algorithms are more appropriate for what situations, etc.

Algorithms with arbitrary constants

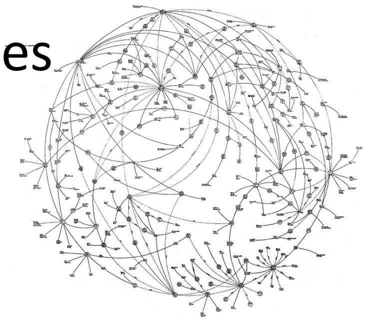
- When reading an algorithm…

---

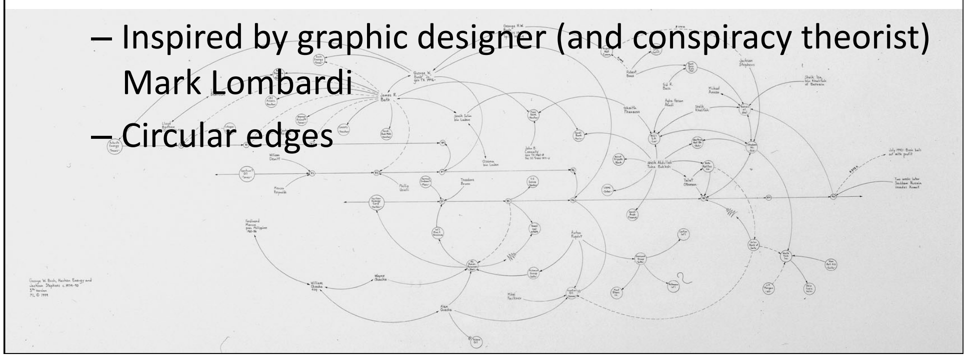
algorithm SPRING(G:graph);
place vertices of G in random locations;
repeat M times
    calculate the force on each vertex;
    move the vertex $c_4 * (\mathit{force\ on\ vertex})$
draw graph on CRT or plotter.

The values $c_1 = 2$, $c_2 = 1$, $c_3 = 1$, $c_4 = 0.1$, are appropriate for most graphs. Almost all graphs achieve a minimal energy state after the simulation step is run 100 times, that is, $M = 100$.

Using Non-Straight Edges

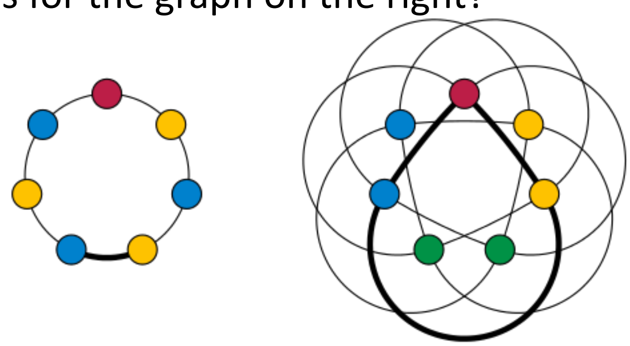


- Increase/maximize angular resolution at vertices
  - Can use Bezier Curves
- Lombardi or “near-Lombardi”
  - Inspired by graphic designer (and conspiracy theorist) Mark Lombardi
  - Circular edges



Effectiveness of Curved Edges

- What are the advantages of straight vs curved edges for the graph on the right?

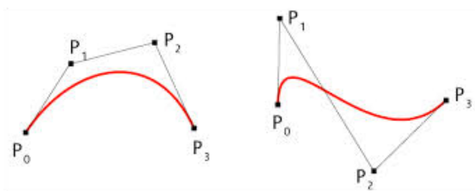


A seven-vertex cycle and its complement, showing in each case an optimal coloring and a maximum clique (shown with heavy edges). Since neither graph uses a number of colors equal to its clique size, neither is a “Perfect Graph”.

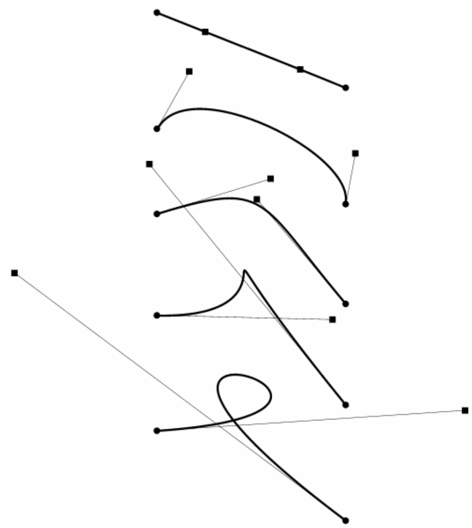
https://en.wikipedia.org/wiki/Perfect_graph#/media/File:7-hole_and_antihole.svg

Cubic Bézier Curve

- 4 control points
- Curve passes through first & last control point
- Curve is tangent at $\mathbf{P_0}$ to $(\mathbf{P_1}-\mathbf{P_0})$ and at $\mathbf{P_3}$ to $(\mathbf{P_3}-\mathbf{P_2})$



http://www.e-cartouche.ch/content_reg/cartouche/graphics/en/html/Curves_learningObject2.html



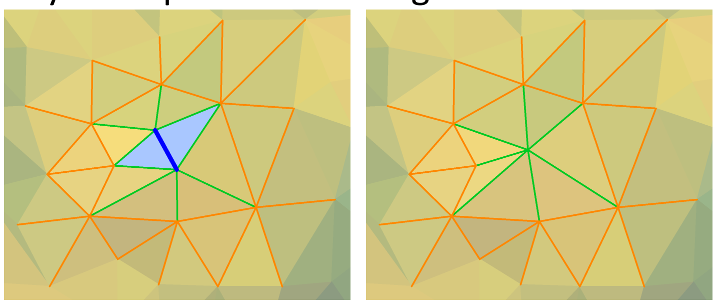
http://www.webreference.com/dlab/9902/bezier.html

Today

- Questions about Homework 2?
- Some Graph Terminology
- Graph Drawing Goals, Questions, & Challenges
- Some Related Terms/Algorithms (mentioned indirectly in the reading)
- Reading for Today: “Force Directed Graph Drawing”
- Reading for Friday

Edge Contraction / Edge Collapse

- Goal: Reduce number of vertices/edges while minimize shape/color/attribute loss
- Possible algorithm for 3D meshes:
  Always collapse shortest edge

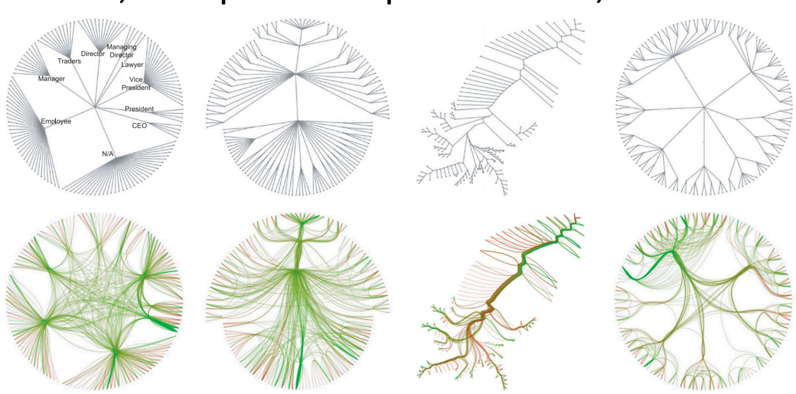


- But what to do for data without obvious spatial coordinate system?

Reading for Friday

Post a comment/question on LMS by Tuesday @10am

- "Social Network Clustering and Visualization using Hierarchical Edge Bundles", Jia, Garland, & Hart, Computer Graphics Forum, 2011.



Manual (ideal) | Strength clustering [ACJM03] | BC communities [GN02] | Our method