1. An InfoVis View on Graph Drawing (Network Visualization)

Information Visualization (186.141)
TU Vienna, Austria
June 1 & 5, 2012

1.1 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
- VLSI
- ...

1.2 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:

<table>
<thead>
<tr>
<th>Node</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjacency list</td>
<td>1: 2</td>
<td>2: 1, 3</td>
<td>3: 2</td>
</tr>
</tbody>
</table>

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

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

Types of graphs

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

Types of graphs (cont.)

- Directed/Undirected graphs
- Extended graph models
  - Hierarchical graphs
  - Clustered graphs
  - Hypergraphs
  - ...

1.2 Definitions
1.3.1 Aesthetics

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

- Nodes
  - Shape, color, size, position, label, ...
- Edges
  - Color, size, thickness, direction, label, ...
  - Shape
    - straight, curved, planar, orthogonal, ...

- 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, ...

- Research areas of GD (cont.)
  - Navigation in large graphs
  - Dynamic graphs
  - Heterogeneous node and edge types
  - Massive node degrees
  - Visualization of isomorphic subgraphs
    - (Embedding of additional information)
    - (Focus & Context)
    - (Comparison of graphs)
    - ...

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

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

- Own research community
  - very large field!
  - I 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

- Graph layout and positioning of nodes
- Scalability

[1.3 Graph Drawing]

[1. Network Vis.]
1.3.1 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 ↑

1.3.2 Force-directed GD

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

Example: Conflict between two criteria

Example: crossings and bends

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

Layout example:
1.3.2 Force-directed GD

- There are many improvements of this approach, e.g.:
  - Inclusion of local, minimal energies
    - Algorithm of Kamada and Kawai, 1989
  - Iterative, force-directed node positioning
    - Fruchterman and Rheingold, 1991
  - Simulated Annealing
    - Davidson and Harel, 1996

1.3.3 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

1.3.4 Clustered Graphs

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

...
1.3.4 Clustered Graphs

- Approach of Noack and Lewerentz, 2005
- Three different layout criteria for views:
  1. Cluster creation
  2. Representation of the hierarchy
  3. Distortion
- Modeling an energy model with three parameters (c, h, d)
  - Computation of a layout by minimization of the total energy of the system
- Applications: inheritance graphs, call graphs, ...

1.3.5 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 3.3.5)
  - 3D (more space)
    - Classic 2D algorithms are extended to 3D
  - Problems
    - Navigation, massive overlaps, mental map, ...

---

**Clustering**

- Given is a layout. Furthermore, let c be the parameter for the cluster creation with \( c \in (0, 1] \)

1.3.4 Clustered Graphs

- Hierarchy
  - Given is a layout. Let c be the parameter as described before with \( c \in (0, 1] \) and h be the parameter for the hierarchy creation with \( h \in [0, 1] \)

- Distortion to magnify the areas of interest
  - Now, we add the parameter d for the distortion with \( d \in [0, 1] \)

  - Applications: inheritance graphs, call graphs, ...

- Java packages at different levels. Three level views with \( c = 0.5 \) and \( h = 1.0 \)

- Example: Linux kernel

  - [http://perso.wanadoo.fr/pascal.brisset/kernel3d/kernel3d.html]
1.3.5 Graph Drawing in 3D

- Example: 3D orthogonal GD

D. Wood et al.

1.3.6 Applications

- There are hundreds of applications (also in InfoVis) that use or extend classic GD techniques
- Tools
  - JUNG, Walrus, ...

1.4 Dynamic GD

- During the past years, 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
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

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


Here, mainly force-based methods are used

If we know a sequence of graph in advance or if it is possible to precalculate it, then there is another method:

**Foresighted Graph Layout**


**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

**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.)
Aim of Information Visualization

- InfoVis is a research area that 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
  - Interacting with the graph visualization
  - Exploring the possibly huge graph topology
  - Adding of additional information (attributes) into the drawing
  - ...

What is a network comparing to a graph?

- Network = graph + attributed information to nodes and edges (also called multivariate network)
- Just to give you some impressions, we will look to some specific aspects
  - Special graph drawing techniques that support InfoVis tasks
  - Interactive exploration and clustering
  - Focus & Context
  - Alternative techniques to show large graphs

Hierarchical Edge Bundles [Holten, InfoVis06]

- Avoid Clutter in Networks

Edge Clustering

- [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
1.5.3 Interactive Exploration and Clustering

1. Network Vis.
1.5 InfoVis ↔ GD

- Visualizing online social networks

[Image of a social network visualization]

J. Heer and D. Boyd, InfoVis ’05

1.5.4 Focus & Context

1. Network Vis.
1.5 InfoVis ↔ GD

- Huge graphs cannot be handled/perceived by only watching an image of the graph
  - The visualization only proves the complexity of the network
- **Focus & Context** is an important technique to explore huge graphs
  - Typically, one or more nodes are magnified in the center of the window
  - For example, we can use our fisheye technique studied in the InfoVis I course

1.5.5 Alternative Techniques

1. Network Vis.
1.5 InfoVis ↔ GD

- Network visualization through so-called **Semantic Substrates**

- **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

[Image of a semantic substrates visualization]

1.5.3 Interactive Exploration and Clustering

1. Network Vis.
1.5 InfoVis ↔ GD

- Overview+Detail with Constraint-based Cooperative Layout
  - [Tim Dwyer et al., “Exploration of Networks Using Overview+Detail with Constraint-based Cooperative Layout.” In Proceedings of Information Visualization 2008.]
  - **Idea**
    - Detailed view is NOT just a zoomed in view of the overview
    - Local optimizations but preserving mental map of the whole graph
    - Focus view uses a constrained-based graph layout

[Image of a cooperative layout visualization]

1.5.4 Focus & Context

1. Network Vis.
1.5 InfoVis ↔ GD

- **MoiréGraphs**

- **Idea**
  - Radial layout
  - Several foci
  - Animations
  - Nodes can contain images
  - …

[Image of a moiré graph visualization]

1.5.5 Alternative Techniques

1. Network Vis.
1.5 InfoVis ↔ GD

- 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

[Image of a legal system visualization]