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:
    - **Adjacency matrix**
      
      |   | 1 | 2 | 3 |
      |---|---|---|---|
      | 1 | 0 | 1 | 0 |
      | 2 | 1 | 0 | 1 |
      | 3 | 0 | 1 | 0 |
    - **Adjacency list**
      
      - 1: 2
      - 2: 1, 3
      - 3: 2

- **Graph Drawing**: automatic drawing of graphs in 2D and 3D
1.2 Definitions

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
1.2 Definitions

- Types of graphs
  - Trees
    - Properties
      - Special case of a graph
      - No cycles
      - Special root node
    - Free trees
    - Binary trees
    - Root trees
    - Ordered trees
  - Planar graphs
1.2 Definitions

- Types of graphs (cont.)
  - Directed/Undirected graphs
  - Extended graph models
    - Hierarchical graphs
    - Clustered graphs
    - Hypergraphs
    - …
1.3 Graph Drawing

- 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

- Research areas of GD
  - Graph layout and positioning of nodes
  - Scalability
1.3 Graph Drawing

- 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)
  - ...

Erasmus Teaching Exchange 12
DFM – Prof. Dr. Andreas Kerren
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, …
1.3 Graph Drawing

- 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]
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

[taken from S. Hong und P. Eades‘ course]
1.3.1 Aesthetics

- All layout algorithms fulfill more or less a set of aesthetics criteria

- Furthermore, the layout itself affects the perception of graphs

- **Problem**: These aesthetics criteria are sometimes contradictory and their computation mostly NP hard!

- Thus, the most GD algorithms are heuristics

[taken from S. Hong und P. Eades' course]
Aesthetics Criteria

- Edge crossings \(\downarrow\)
- Area \(\downarrow\)
- Symmetry \(\uparrow\)
- Edge length \(\downarrow\)
  - Maximal edge length, uniform edge length, total edge length
- Bends of edges \(\downarrow\)
  - Maximal bends, uniform bends, total bends
- Resolution \(\uparrow\)
1.3.1 Aesthetics

- **Example:** crossings and bends

```
  +-------+     +-------+     +-------+
  | monkey|     | eat    |     | obj    |
  +-------+     +-------+     +-------+
  | shell  |     | inst   |     | part_of|
  +-------+     +-------+     +-------+
  | walnut |     | matr   |     | spoon  |
  +-------+     +-------+     +-------+
```

Avoid edge *crossings*

Avoid edge *bends*

Avoid long edges

[less readable]

[taken from S. Hong und P. Eades’ course]
Example: Conflict between two criteria

Minimize edge crossings
Maximize symmetry

[taken from S. Hong und P. Eades’ course]
1.3.2 Force-directed GD

- 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
1.3.2 Force-directed GD

Spring Embedder

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

[taken from S. Hong und P. Eades‘ course]
1.3.2 Force-directed GD

- Problems of the classic Spring Embedder algorithm is the high runtime and its (possibly) breakdown with very large graphs

- 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
In another approach, called Layered GD, the layout method firstly looks for a suitable layering that assigns each node an integer number.

Most methods compute 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.
1.3.3 Layered GD

- Parallel layers
- Radial layers

[taken from S. Hong und P. Eades' course]
1.3.3 Layered GD

- Classic algorithm of Sugiyama, 1981
  

- Another heuristic defines a fixed order of the first and last layer and demands that each node is in the barycenter of its neighbors in the graph. This yields a linear system of equations
  

- Comparison of different heuristics, e.g.
  
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
**Definition:** A hierarchical graph \( H = (G, T) \) consists of directed graph \( G \) and a rooted tree \( T \), such that the leaves of \( T \) are exactly the nodes of \( G \).

\( T \) is called **hierarchy tree** and \( G \) the **underlying graph** of \( H \).
1.3.4 Clustered Graphs

- Approach of Noack and Lewerentz, 2005
  - [A. Noack and C. Lewerentz. „A Space of Layout Styles for Hierarchical Graph Models of Software Systems“. ACM SoftVis ’05, 2005]

- Three different layout criteria for views:
  (1) Cluster creation
  (2) Representation of the hierarchy
  (3) Distortion

- Modeling as 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, …
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 $h \in [0, 1]$.

Java packages at different levels. Three level views with $c = 0,5$ and $h = 1,0$. 
### 1.3.4 Clustered Graphs

- Distortion to magnify the areas of interest
  - Now, we add the parameter $d$ for the distortion with $d \in [0, 1]$
    (in figure below, let be $c = 0.5$ and $h = 0.9$)

<table>
<thead>
<tr>
<th>$d$</th>
<th>Graph 1</th>
<th>Graph 2</th>
<th>Graph 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td><img src="#" alt="Graph 1 $d=0.0$" /></td>
<td><img src="#" alt="Graph 2 $d=0.5$" /></td>
<td><img src="#" alt="Graph 3 $d=1.0$" /></td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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, …
Example: Linux kernel
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, …

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

- AS Internet Graph

http://www.caida.org/research/topology/as_core_network/AS_Network.xml
1.3.6 Applications

- Visualization of Biochemical Pathways

by F. Schreiber
1.3.6 Applications

- Map layouts like MetroMaps
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

1.4.1 Morphing

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
1.4.1 Morphing

- 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
1.4.1 Morphing

1. Network Vis.

1.4 Dynamic GD

http://graphael.cs.arizona.edu/graphael/
1.4.1 Morphing

Network Vis.

Dynamic GD

http://graphael.cs.arizona.edu/graphael/
If we know a sequence of graph in advance or if it is possible to precalculate it, then there is another method:

**Foresighted Graphlayout**

- [S. Diehl, C. Görg, and A. Kerren. „Preserving the Mental Map using Foresighted Layout“. In Proceedings of Joint Eurographics - IEEE TCVG Symposium on Visualization, VisSym `01, Springer Verlag, 2001.]

**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
1.4.2 Foresighted GD

- **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.)
1.5 InfoVis ↔ GD

- **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
1.5.2 Techniques for Special InfoVis Tasks

- Hierarchical Edge Bundles [Holten, InfoVis06]
  - Avoid Clutter in Networks
1.5.2 Techniques for Special InfoVis Tasks

- New solution with possibility to change the blending strength

[Demo]

http://flare.prefuse.org/apps/dependency_graph
1.5.2 Techniques for Special InfoVis Tasks

- **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

[Video]
1.5.2 Techniques for Special InfoVis Tasks

- **NodeTrix** [Henry et al., InfoVis07]
  
  - Combined techniques for a better structuring

(a) PARC Community

(b) Ed Chi’s influence

[Video]
1.5.3 Interactive Exploration and Clustering

- Visualizing online social networks

J. Heer and D. Boyd, InfoVis `05

[Video]

http://jheer.org/vizster/
1.5.3 Interactive Exploration and Clustering

- 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
1.5.4 Focus & Context

- 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
1.5.4 Focus & Context

**MoiréGraphs**


**Idea**

- Radial layout
- Several foci
- Animations
- Nodes can contain images
- …
1.5.5 Alternative Techniques

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
1.5.5 Alternative Techniques

- Each region corresponds to a level of jurisdiction in the legal system of the US.
- Nodes correspond to the different cases (1978-2005); Node size corresponds to the number of references on that case.
- Edges correspond to the single references.

[Video]