

Force Directed with Annotation (prefuse beta)

Description

Spring embedding algorithms also called FDP (Force Directed Placement) can be used to sort randomly placed nodes into a desirable layout that satisfies the aesthetics for visual presentation (symmetry, non-overlapping etc.) See Eades (1984) and Jones' webpage.

FDP (Battista et al., 1984) views nodes as physical bodies and edges as springs connected to the nodes providing forces between them. Nodes move according to the forces on them until a local energy minimum is achieved. In addition to the imaginary springs, other forces can be added to the system in order to produce different effects (see table). Many visual examples of these force models can be found in Battista et al. (1984).

Force Model	Formula	Example of Usage
Spring Force	$F = k(1-a)$ k- stiffness of spring a- natural length of spring	Assigning different k and a to different edges to separate nodes by different distances.
Gravity Force	$F = g/r^2$ g- associated with mass of node, usually equals 1.	Apply gravity force between node pairs to prevent node overlapping.
Electrical and Magnetic Force	$F = eE$ $F = qB$ E- electric field strength B- magnetic field strength	Changes nodes distribution along a direction. FDP was extended to three dimensions by Kumar & Fowler (1996).

A simple algorithm to find the equilibrium configuration is to trace the move of each node according to Newton's 2nd law. This takes time $O(n^3)$, which makes it unsuitable for large data sets. Rob Forbes (1987) proposed two methods that were able to accelerate convergence of a FDP problem 3-4 times. One stabilizes the derivative of the repulsion force and the other uses information on node movement and instability characteristics to make a predictive extrapolation. In a recent paper, Huang et al. (1998) describe a system that uses a "logical frame" to present a subgraph of the entire graph. Algorithms are provided to smoothly migrate from one logical frame to another.

Links

- [Source Code](#)

Pros & Cons

The FDP method is easy to understand and implement, but is very slow for large graphs.

Applications

Force directed graph drawing algorithms are increasingly popular in information visualization. They have been used in BEAD (Chalmers, 1992), Narcissus (Hendley, 1995), SHriMP (Simple Hierarchical Multi-Perspective views) (Storey, 1995) and SPIRE (Hetzler et al., 1998) see (Young, 1996). They can/have also been used to generate graphical representations of [Pathfinder networks?](#).

Some Demo's & Examples:

[XHDG - X toolkit Hierarchical Directed Graphs](#)
[WWW3D: A 3D Multi-User Web Browser](#)
[DA-TU STORYBOARD](#)

Implementation Details

In class we will use a java code that was originally implemented by Sun Microsystems, Inc. and was modified by Larry Mongin. In particular, the appearance of the nodes was changed, the forces between these nodes were modified, and several variables like system energy, step bound, iteration step and several functions were added to observe (and interact with nodes during) sorting.

Usage Hints

The Spring package can be found on ella in '/home/www/ella/htdocs/classes/L697/code/infovis/gui/spring'.

The implementation in the [Information Visualization XML Toolkit](#) can be found on ella in '/home/www/ella/htdocs/classes/L697/code/infovis/xmlbridge'.

To use only the spring portion of the toolkit you will need an entire copy of the Spring Package and the Information Visualization XML Toolkit, as listed in the above directories on ella. You need to maintain the package (directory) structure of these copied files for it to work with your own code. You will need to use XMLSpring as a visual component just like any other Swing component (initialize it and add it to a container in an applet or application). Please review the following applet Example.java.

The demo is also in the Information Visualization XML Toolkit. The demo can be found on ella in '/home/www/ella/htdocs/classes/L697/code/infovis/demo/'. Copy the demo files to a machine with the proper JDK (1.3 with the XML pack or 1.4) and then double click on infovis.jar to open the demo. Click on 'Select File' to select a XML file to visualize and select one of the XML files from the demo. Note: there are 3 files (hierarchy.xml, tabular.xml, list.xml) and their names indicate their structure type so you should know which visualizations can use which XML file.

Variations: Change appearance of the nodes, the forces between these nodes, energy, step bound, iteration steps.

Acknowledgements

The java applet was implemented by Yuezhen Zhou. Nihar Sanghvi integrated the code into the software repository.

References

1. Battista, G. , Eades, P., Tamassia, R., and Tollis, I.G. (1994) Algorithms for drawing graphs: An annotated bibliography. *Computational Geometry: Theory and Applications*, 4(5), 235-282.
2. Dobkin, D. P., Hausner, A., Gansner, E. R., & North, S. C.(1999) Clustering force-directed graph layouts. In *Proceedings of the 15th Annual Symposium on Computational Geometry*, 425-426.
3. Eades, P. A heuristic for graph drawing Stanton. Ralph G. (Ed.). *Congressus numerantium*. Winnipeg: Utilitas Mathematica.
4. Young, P. (1996) Software Visualization. <http://vrg.dur.ac.uk/misc/PeterYoung/pages/work/documents/lit-survey/soft-vis/>
5. Forbes, R.(1987) Heuristic acceleration of force-directed placement. In *Proceedings of 24th ACM/IEEE Conference on Design Automation Conference*, 735-740.
6. Huang, M. L., Eades, P., & Wang, J. (1998) On-line animated visualization of huge graphs using a modified spring algorithm. *Journal of Visual Languages and Computing*, 9(6), 623-645.
7. Jones, C. Graph Drawing webpage<http://replay.waybackmachine.org/20060901121831/http://orcs.bus.okstate.edu/jones98/drawing.htm>
8. Kumar, A. & Fowler, R.H. (1996) A Spring Modeling Algorithm to Position Nodes of an undirected Graph in Three Dimensions. Technical Report Department of Computer Science, University of Texas - Pan American. http://www.cs.panam.edu/~rfowler/papers/1994_kumar_fowler_A_Spring_UTPACSTR.pdf

See Also



The license could not be verified: License Certificate has expired! [Generate a Free license now.](#)