Graph Drawing Techniques for Geographic Visualization

Peter Rodgers, University of Kent

P.J.Rodgers@kent.ac.uk

1. Introduction

Geovisualizers often need to represent data that consists of items related together. Such data sets can be abstracted to a mathematical structure, the graph. A graph contains nodes and edges where the nodes represent the items or concepts of interest, and the edges connect two nodes together according to some associational scheme. Examples of graph data include: network topologies; maps, where nodes represent towns and edges represent roads; and social diagrams where people are the nodes and edges represent some relationship between people, for instance, that they are friends. A good layout allows users to more easily investigate the data when performing tasks such as following paths, seeing clusters of closely related nodes and discovering general structures in data. Laying out graphs by hand is time consuming, however there are a number of proven graph drawing methods which can be used to automatically visualize the data.

Graph drawing concentrates on the automatic positioning of nodes and edges on screen or paper. Geographic visualization often includes edges that link regions, but often the spatial locations of regions that would be represented by nodes is already known. So why is graph drawing relevant to the visualization of geographic data?

- When regions are fixed, often the connections between them are not. Graph drawing includes techniques for distinguishing close edges, and routing them in the most comprehensible manner. For example visualizing train networks [3].
- In some cases fixing the exact position of locations is not desirable (such as underground railway maps), or even possible (such as globe map projections), but some essence of the position is required. For example, the relative positions between regions to the left, right, and below might need to be retained. In these cases graph drawing can help by nicely laying out the nodes, whilst still maintaining the desired relationships between them.
- Some data, even if it is connected to geographic position, is not best analysed by retaining physical locations. Data mining of this information might be best done by providing a visualization based on the relationship between items, rather than one based on their location [25]. For example, economic data might be best seen by spatially grouping countries that have strong links, see Figure 1. Mapping the internet also typically disregards location, both because of relevance, and because precise location information can be difficult to derive from web sites [4,7].
- The items of interest may not have an obvious physical location (such as data about corporations), hence an automatic graph layout is required to position the items based on how they are related.
- Familiar geovisualizations may be used effectively with alternate graph visualizations of the same data. Such linked multiple views [32] would allow an examination of the spatially oriented representation along side the relational representation for greater breadth of understanding. An example of such multiple views is given in Figure 2.

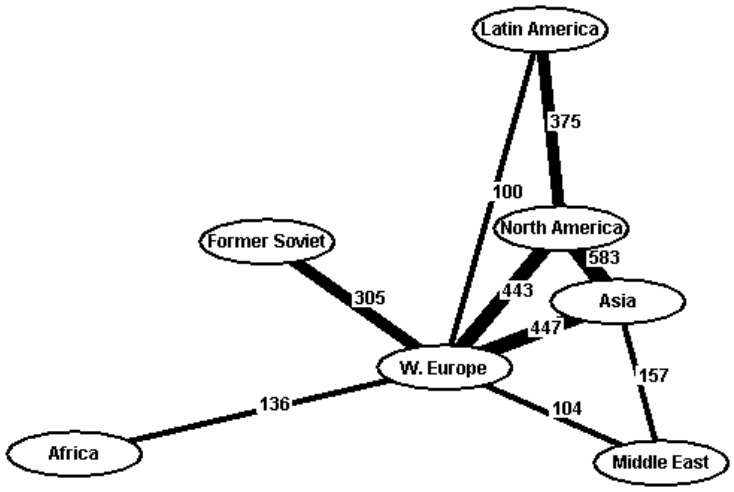

Figure 1. An example of a graph drawing. The edges in the graph represent total merchandise trade in 2001 between regions of the world. Only edges representing amounts over 100 Billion Dollars are shown. This has been drawn with a force directed graph drawing algorithm that includes an edge length heuristic [26].

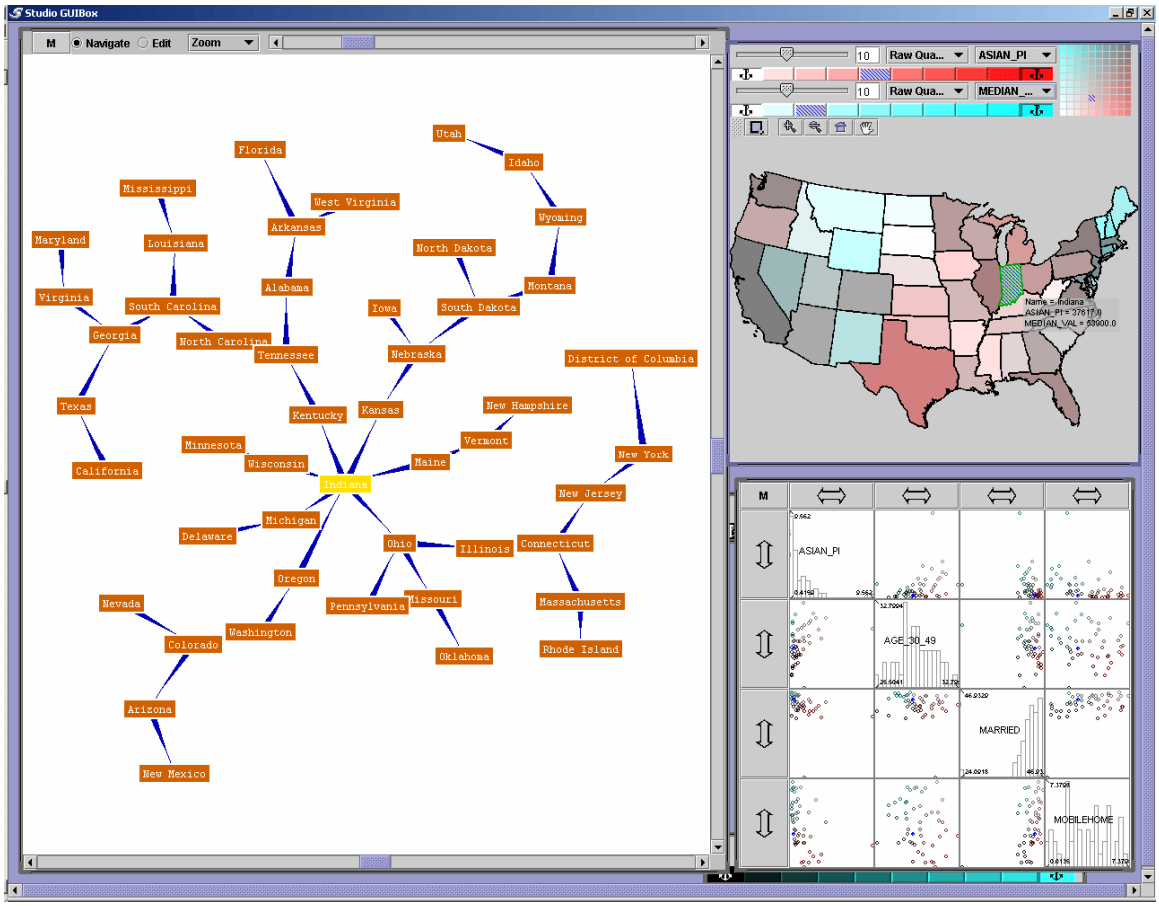

Figure 2. Linking a graph visualization of female cancer rates with other geovisualization views in Geovista studio [23].

Early graph drawing can be traced back to the barycenter drawing method of Tutte [42]. Early computer generated graph drawing included the work Sugiyama et al. [36] to address the need to visualize hierarchical structures. Later, Eades [8] adapted the Spring Embedder from microchip design methods and this became the first widely used technique for laying out general graphs. It is a testament to their effectiveness, and a comment on progress in the field that variations on these methods are still widely used. Further methods rapidly followed and overviews of common techniques can be found in [6, 19]. Methods to achieve a good drawing often depend on some knowledge of the graph semantics (that is, what the graph represents). If the

graph has some known structure then there are often quick specialist algorithms for producing an effective drawing. Such structures include trees and acyclic graphs, which can represent classification hierarchies and organizational charts. Another specialist feature of graphs is planarity. Planar graphs are those which can be drawn without any edge crossings, and have a particular application to geographic data, as graphs that have edges representing region borders are planar. Other algorithms work on more general graphs including force directed methods, which can produce layout reasonably quickly. Slower methods involve quantitatively measuring the aesthetics in a graph and then moving the nodes so as to improve the resulting score. It is common for these methods to be combined in a hybrid manner, so that a fast method can produce the initial drawing, followed by a slower technique for fine-tuning.

For those wishing to visualize graph based data without coding algorithms there are several effective commercial and academic tools for accessing most of the graph drawing techniques mentioned in this chapter. Tom Sawyers' Graph Layout Toolkit [37] is one of the most complete and widely used, but is not freeware, whereas AT&Ts' sophisticated GraphViz [13], containing spring embedder and hierarchical algorithms, is open source. Other systems that implement graph drawing techniques include TouchGraph [38] DaVinci [12], Graphlet [14], Jviews [17], Tulip [40] and VCG [33]. Note that this is not a complete list, and there are several general graph algorithm systems available that include graph drawing functionality.

The remainder of this chapter is organized as follows: Sections 2 to 5 give more detail on the graph drawing methods mentioned above; Sections 6 discusses dynamic graph drawing, which is the process of laying out a graph that changes over time. Section 7 address the issues involved in drawing graphs in three dimensions. Section 8 discusses the rendering of graph drawings. Section 9 concludes the paper and gives some idea of future research challenges, particularly those relating to geographic visualization.

2. Hierarchical Drawing Methods

Early graph drawing methods [36] dealt with specialist graph structures, where a notion of a single direction applies. This might be because the data is organized into a hierarchy, with the main concepts at the top, and other concepts below, or because there is actual flow, for example, abstract representation of rivers or data relating to population movement between regions. The main feature needed for this form of layout is that either the graph has no cycles, or that nodes participating in cycles are ordered. Figures 3 and 4 show illustrations of hierarchical layout.

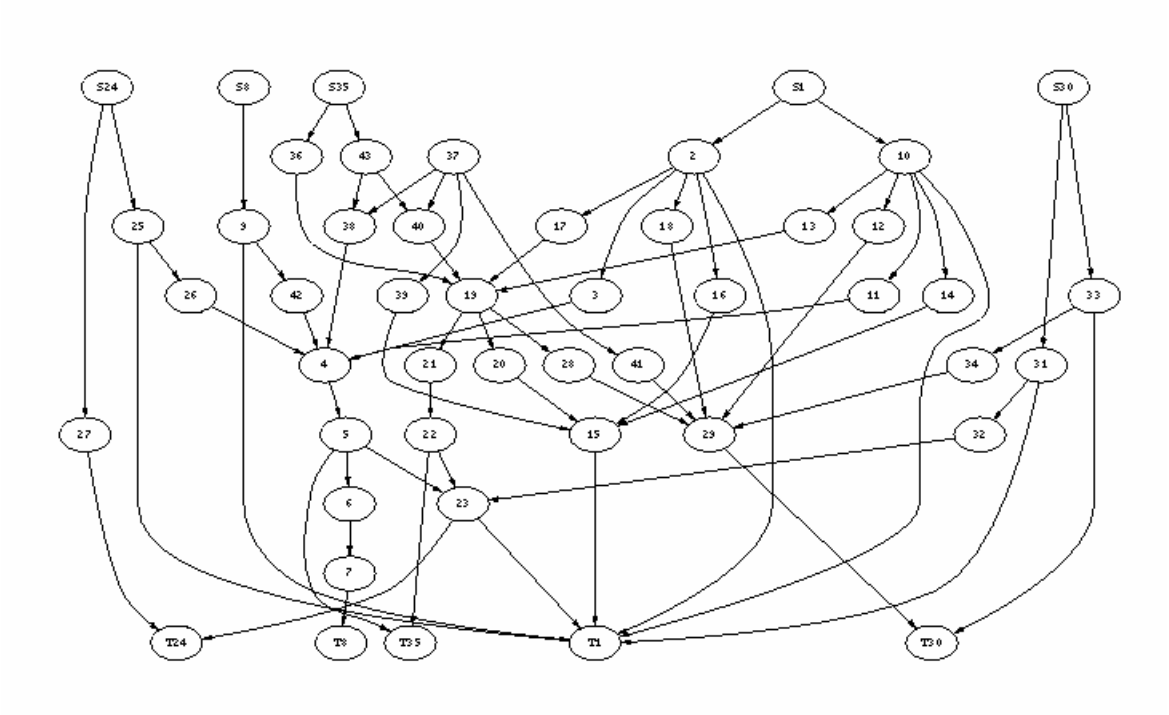

Figure 3. Top down hierarchical layout of world dynamics data [11] drawn with GraphViz.

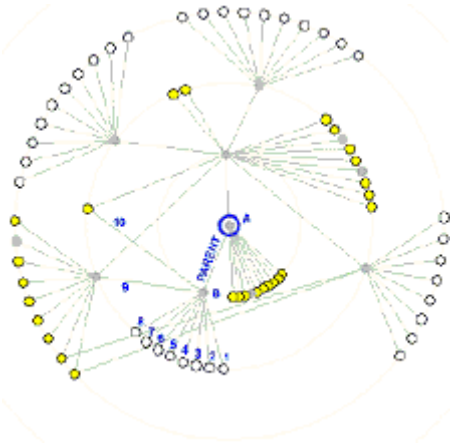

Figure 4. Radial hierarchical layout from [47]. This allows many more positions for nodes in lower layers, at the expense of a notion of flow that can be seen in the top down approach

The nodes in a hierarchical graph can be placed in layers, with the first nodes (called roots) in the top layer, and later nodes further down. The leaves at the bottom are the lowest nodes of all, as they have no following items. Edges that cross layers are often assigned dummy nodes at each layer crossing in order to aid the drawing process. Once nodes are layered their relative positions in the layer can be assigned. Various approaches are taken, but most are designed to reduce the number of edge crossings and edge bend points.

3. Planar Layout Techniques

A planar graph is one that can be drawn with no edge crossing. Many graph theoretic algorithms rely on a graph being planar, and there are quick methods to lay out such a graph. Graphs where edges represent adjacency between regions are planar (because no border can cross another border), as are graphs that represent Delaunay triangulations, a precursor to several analytical techniques in cartography [45,39]. A common method for drawing planar graphs is to give the nodes a canonical order, placing the first three nodes at vertices of a triangle and then adding lower ordered nodes in positions within the triangle, see Figure 5. This method does not always produce a satisfactory layout, but it is improved if followed by a force directed method, such as the spring embedder.

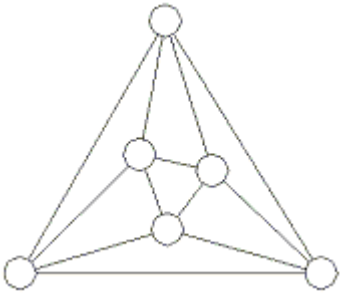

Figure 5. A planar graph laid out using Schnyder's method [35]. Planar graphs are those that can be drawn without any of their edges crossing.

It is also possible to planarize graphs, a process of converting non-planar graphs to planar graphs by removing crossing edges or placing new nodes at edge crossings. Planar drawing algorithms can then be applied to the planarized graph, after which the changes to the graph can be undone, and the modifications integrated into the drawing.

4. Force Directed Layout Techniques

Force directed methods are widely applied techniques. Here, repeated applications of a force model result in the graph attaining a minimal energy state, where the graph is considered to be well drawn. These methods have the significant advantage of being applicable to general graphs, with no requirement that the graph has a specific structure or that it meets some topological constraints. For example the spring embedder can be used with either planar and non planar graphs.

A fast approach for general graphs is the barycenter approach [42], see Figure 6. The nodes are drawn at the average distance of their neighbours (which is known as their barycenter). This method is repeated for a set

number of iterations, or until nodes stop moving, having reached a minimal energy state. A subset of the nodes must be fixed, to indicate the outer borders of the drawing space, or the nodes will simply collapse to a single point. This method is effective for sparse, symmetrical graphs, where fixed nodes can easily be defined.

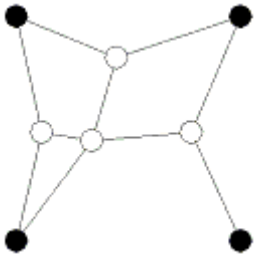

Figure 6. A graph after the barycenter drawing approach has been applied. The nodes with black centres are fixed.

Eades' spring embedder [8] is perhaps the best known graph drawing technique and is a popular method for visualizing geographic data such as network topologies and the internet [16,24,26]. A variant, which has an additional edge length heuristic is shown applied to a planar graph in Figure 7. Nodes are treated as charged particles that repel all other nodes, and edges are treated as springs that attract connected nodes. In an iteration of the method, the forces are calculated for all the nodes in a graph and the nodes are then moved in the direction that the forces indicate. Iterations continue until a least energy state is achieved, or a set number of iterations are performed. The result is a graph where nodes have decent minimal separation and connected nodes are reasonably close together. Recent work [41,43] has made the application of this method to large graphs possible, with graphs containing thousands of nodes being drawn in a few seconds.

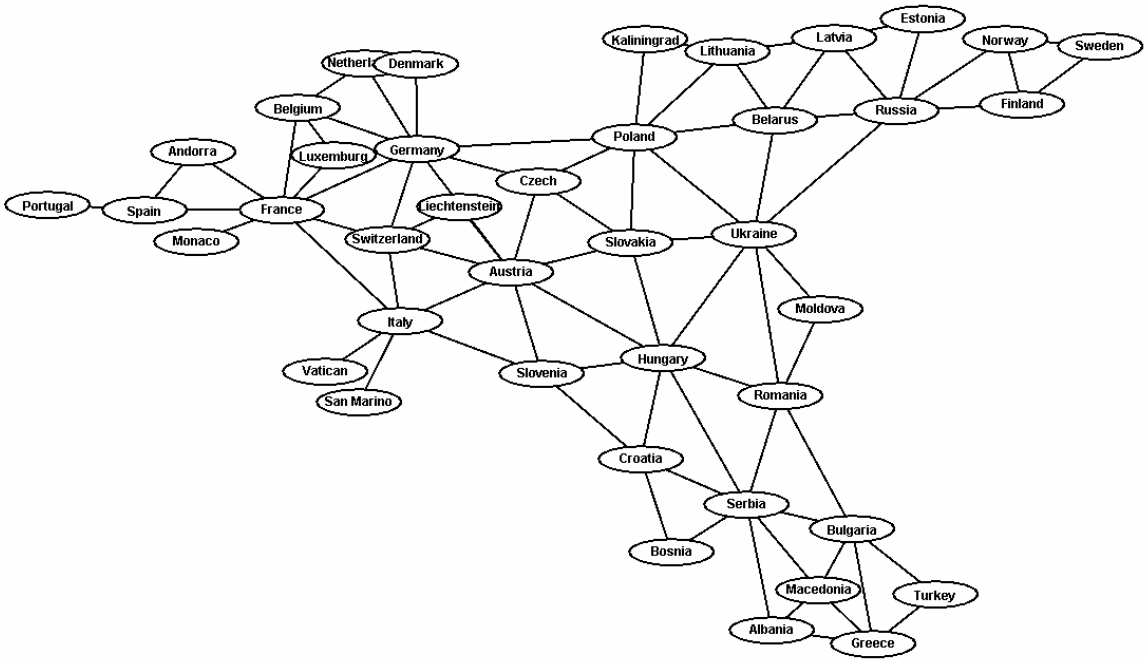

Figure 7. A planar graph representing bordering states in mainland Europe. A modified spring embedder has been applied. Such a graph could be attributed with relevant information such as traffic data or population flows.

5. Optimizing Aesthetic Criteria

The previous methods take implicit judgements about what is a good graph drawing. However, there are approaches that attempt to explicitly improve the aesthetics of a graph by allowing users to choose from a range of aesthetic criteria, weight them, and allow a multiple criteria optimizing technique to automatically lay out the graph based on these preferences. The major disadvantage of these techniques is the time taken to reach a reasonable drawing. Typically the criteria have to be recalculated on each iteration of the optimizer, and many iterations are required. Multicriteria optimizers that have been used include the (slow) simulated annealing method [5] and the (even slower) genetic algorithm approach [15].

The aesthetic criteria used must be quantifiable. Common criteria include: the number of edge crossings, statistics relating to even node distribution (such as the variation in distance of closest neighbours of each node); the area or aspect ratio of a rectangle containing the graph; the angular resolution of edges emanating from nodes (a small angle between edges means that they are difficult to distinguish); the number of edge bends; evenness of edge length; methods for measuring the symmetry in the graph; and encouraging directed edges in a particular direction. See [6,19] for fuller discussions of aesthetic criteria. A feature of such criteria is that empirical studies can be performed to discover user preference. This work is ongoing, but some results indicate that reducing edge crossings is a major factor in making graph layout comprehensible [30].

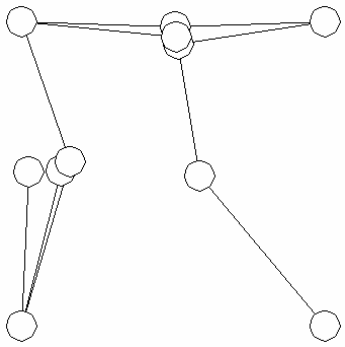

Figure 8. A graph with no edge crossings, even edge length and a 1:1 aspect ratio, but with poor node distribution and poor angular resolution

Criteria based methods may be effective in aiding the visualization of geographic data. In particular, even when the nodes are fixed to definite locations, criteria such as edge angular resolution [2] and the number of edge bends may be still be optimized, producing a better visualization.

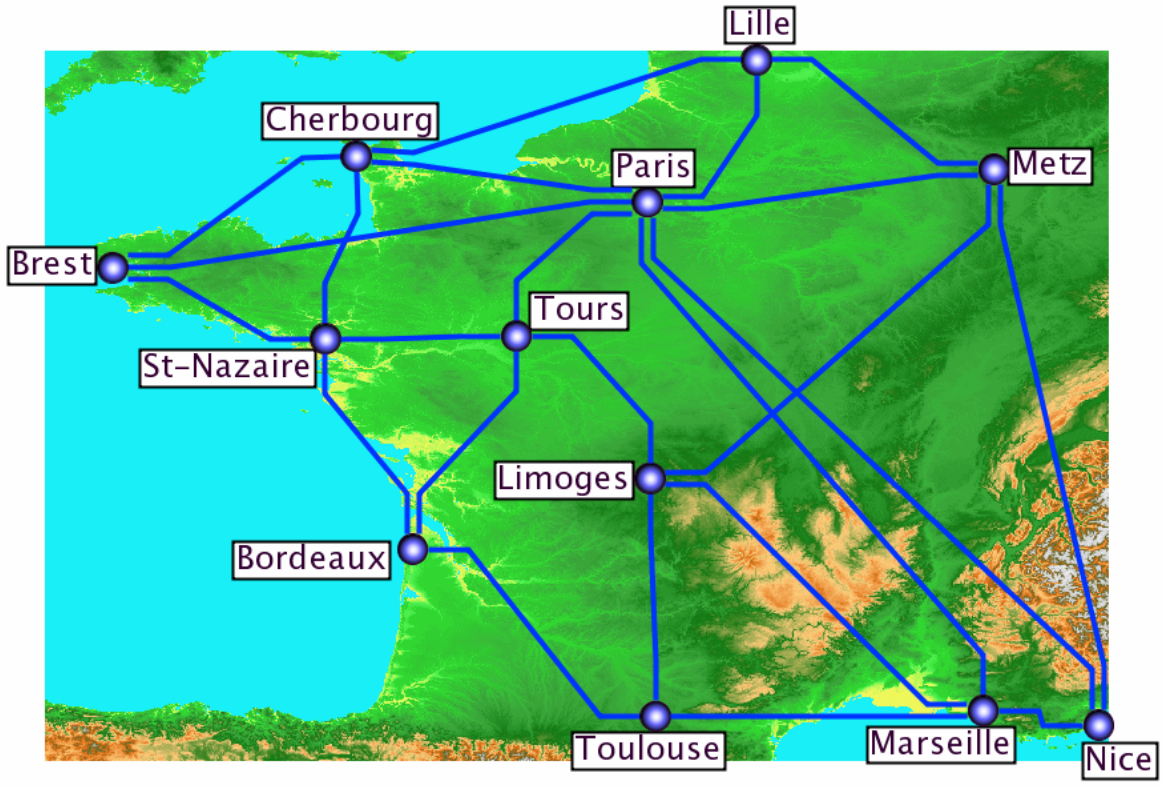

Figure 9. Graph created with Ilog Jviews [34]. It shows links between French towns, using a direct edge routing method. This also shows the application of labelling techniques to graph drawing [18,28].

Optimizing on aesthetic criteria is particularly effective where specialist requirements are present. For example, criteria might be developed for drawing public transport routes. Harry Becks famous tube map provides inspiration, as he straightened out train lines, evened out distances between station distance and altered the area of the map to fit within a poster. This is generally accepted as a much clearer way of

enabling the public to plan their routes. Generalizing these features might allow the development of a multicriteria optimizer for public transport schematics. Schematic generators have already been developed for other application areas, as shown in Figures 10 and 11.

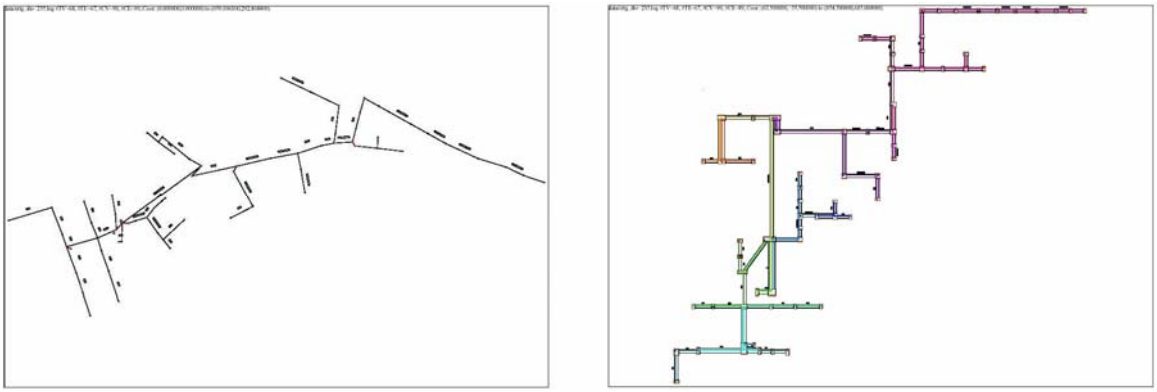

Figure 10. The diagram on the left shows a spatially exact representation of a cable plan. The diagram on the right shows the same cable plan, but using an automatically generated schematic representation [20]

6. Dynamic graph drawing

Graphs often change over time. This may be due to a user editing a diagram, or more commonly in information visualization, there is some alteration in the underlying data structure. For example, a real time map of the internet will be constantly be changing as WWW pages are added, modified and removed. The challenge here is to retain the users' mental map [9] of the graph.

A mental map is the current view the user has of the graph. For complex graphs, a user has to invest considerable time and effort into understanding the current layout, so that any changes should be integrated carefully and with minimal disruption to this mental map. Some iterative techniques, such as the spring embedder can do this naturally to some extent, but this relies on the particular technique being first used to draw the graph. There is a need for research that analyses and defines the user's current mental map, before deciding on the best method for making layout alterations, perhaps drawing on related work in cognitive aspects of cartography [21,29,31]. Figure 11 shows two possible rearrangements of a graph after the node 'X' has been added to the graph. Figure 11a (left) assumes that a hierarchical algorithm has been used to draw the original graph, and that node 'X' is a child of node 'a'. It attempts to find space on the same level as the other children of 'a', without disturbing any other nodes. Figure 11b (right) assumes that the spring embedder has been used to draw the original graph and so has produced a layout after moving the node 'a' to make space for the additional node. Although the drawing is nicer, the movement of the nodes may have disturbed the user's mental map.

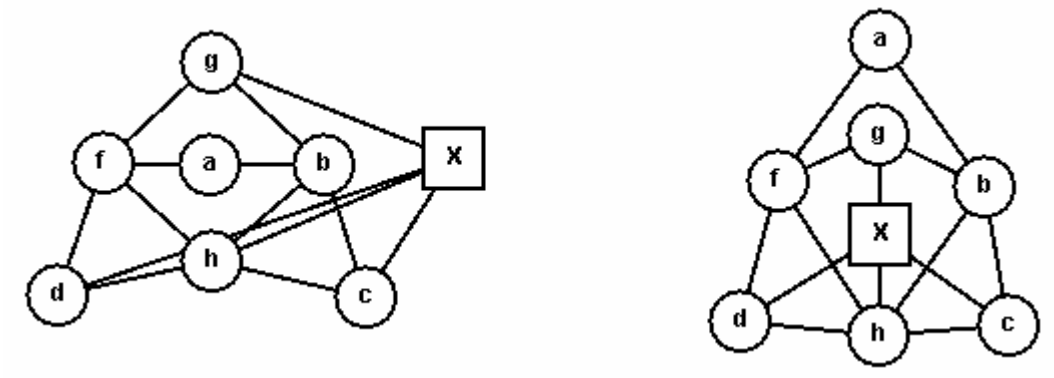

Figure 11. Examples of dynamic graph drawing. The node 'X' has been added.

7. Drawing Graphs in 3D

Whilst it is necessary in a document of this sort to display images in two dimensions, there are many methods that draw graphs in three dimensions. These are typically variants of the two dimensional techniques described in previous sections [27, 46]. For example, force directed approaches require very few modifications to deal with the extra dimension. However, not all graph drawing methods can be applied in

three dimensions, in particular, planar drawing methods by their nature are very dependent on being drawn on the plane. Others need considerable adjustment, including those that draw by measuring aesthetic criteria, as some criteria are not appropriate to three dimensions, such as measuring the number of edge crossings. In addition, new criteria may need to be developed, in particular in relation to viewpoints. Viewpoints [10] are a significant difference between three and two dimensional graph drawing. A viewpoint is a two dimensional projection of the three dimensional graph. This is of obvious importance, as with current display technology, a three dimensional layout must be visualized in two dimensions, and the choice of which viewpoint affords the best comprehensibility is a major issue.

When presented statically, graphs drawn with three dimensional techniques have very little advantage, and many disadvantages over drawing created with methods designed for two dimensions. However, three dimensional layouts become more useful when the user can interactively explore the graph [44]. Typically, as users rotates, pans and zooms around a graph, they gather a three dimensional model of the graph, allowing the structure to become clear. This sort of interactive system can allow complex graphs to be analysed effectively, whilst using a relatively small amount of screen real estate.

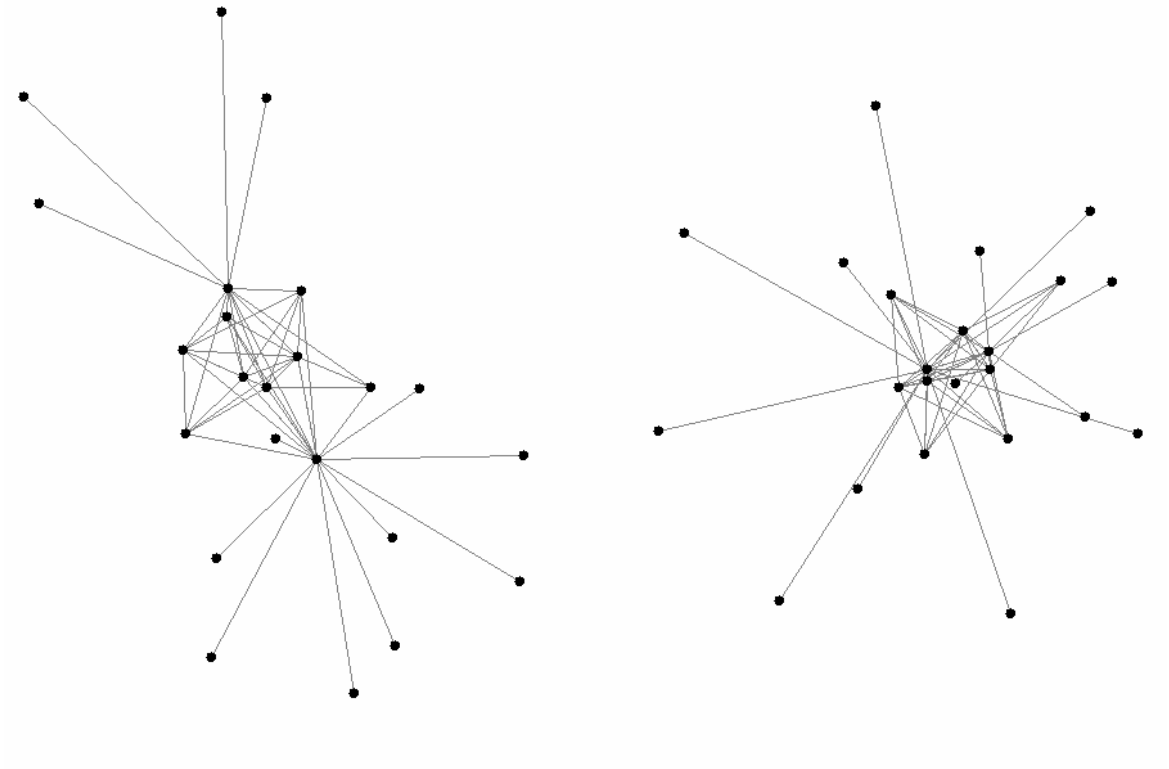

Figure 12. The two diagrams show different view points of one graph laid out by a 3D spring embedder. The graph represents the map of web site. Both diagrams have exactly the same layout, only the angles of the views are different.

8. Graph Rendering

Frequently in graph drawing research, graphs are treated as simple relational structures, circles connected by lines, with the only challenge relating to the layout of the nodes and routing of the edges. However, the comprehensibility of a graph not only relies on the layout, but also how the nodes and edges appear in the final realization. Standard visualization techniques such as colouring nodes and edges, and the use of transparency to deal with occlusion are widely used, and there is some work on distinguishing edge and node occlusion by the use of filleting [22], see Figure 13. However, much more research is needed in this area, particularly regarding application specific rendering, and geographic visualization would be a prime candidate for study.

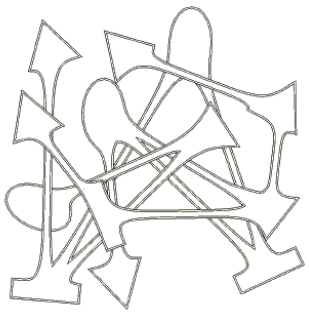

Figure 13. A graph rendered with fillets. From [22]. Fillets are curved joins between edges and nodes, and so help distinguish between edges that connect to nodes, and edges that don't connect to nodes, but where the nodes lie on top of the edges.

9. Conclusions

This chapter has briefly overviewed the current state of the art in graph drawing and discussed some applications in visualizing geographic data. A great deal of geographic data, both spatial and non spatial, has a relational element and is likely to benefit from graph drawing methods. There are interesting potential research areas in the application of graph drawing to geographic visualization, such as the development and use of new aesthetic criteria for geographic visualization, the empirical analysis of graph drawing in geovisualization, and the improvement of graph rendering for geographic data.

10. References

1. G. Battista, P. Eades, R. Tamassia and I. Tollis. Graph Drawing: Algorithms for the Visualisation of Graphs. Prentice Hall. 1999.
2. Ulrik Brandes, Galina Shubina, and Roberto Tamassia,. Improving Angular Resolution in Visualizations of Geographic Networks. Proc. Joint EUROGRAPHICS and IEEE TCVG Symposium on Visualization (VisSym'00). Springer. pp. 23-32. 2000.
3. Ulrik Brandes and Dorothea Wagner, Using Graph Layout to Visualize Train Interconnection Data. Journal of Graph Algorithms and Applications 4(3):135-155, 2000.
4. W. Cheswick, H. Burch, S. Brannigan. Mapping and Visualizing the Internet, Proceedings of the 2000 USENIX Annual Technical Conference, June 18-23, 2000, San Diego, California, USA. 2000.
5. R. Davidson, D. Harel. Drawing graphics nicely using simulated annealing. ACM Trans. Graphics, 15(4):301-331, 1996.
6. G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis. Algorithms for the Visualization of Graphs. Prentice Hall. 1999.
7. Martin Dodge and Rob Kitchin. Mapping Cyberspace. Routledge. 2000.
8. P. Eades. A Heuristic for Graph Drawing. Congressus Numerantium 42. pp. 149-60. 1984.
9. P. Eades, W. Lai, K. Misue, K. Sugiyama. Preserving the Mental Map of a Diagram. Proc. Compugraphics 91, pp. 24-33. 1991.
10. Peter Eades, Michael E. Houle, Richard Webber. Finding the Best Viewpoints for Three-Dimensional Graph Drawings. Graph Drawing (GD) 97, LNCS 1353. pp.87-98. 1998.
11. Jay W. Forrester. World Dynamics. Pegasus Communications. 1973.
12. M. Fröhlich, M. Werner. Demonstration of the Interactive Graph Visualization System DaVinci, Graph Drawing, (GD) 94, LNCS 894. pp. 266-269. 1995.
13. E. Gansner, S. North. An Open Graph Visualization System and its Applications to Software Engineering, Software—Practice and Experience 1–5. 1999.
14. M. Himslot. The Graphlet System, Proc. Graph Drawing (GD) 96, LNCS 1190, pp. 233-240. 1997.
15. M.H.W. Hobbs and P.J. Rodgers. Representing Space: A Hybrid Genetic Algorithm for Aesthetic Graph Layout. FEA'98 in Proceedings of JCIS'98, volume 2, pages 415-418, October 1998.
16. M.L. Huang, P. Eades, R. F. Cohen. WebOFDAV — navigating and visualizing the Web on-line with animated context swapping. WWW7: 7th International World Wide Web Conference. 1998.
17. JViews Home Page. http://www.ilog.com/products/jviews/graphlayout/
18. K. G. Kakoulis, I. G. Tollis. On the Edge Label Placement Problem. Graph Drawing (GD) 96, LNCS 1190, pp. 241-256. 1997.
19. M. Kaufmann and D. Wagner. Drawing Graphs: Methods and Models, LNCS 2025. 2001.
20. U. Lauther, A. Stübinger. Generating Schematic Cable Plans Using Springembedder Methods. GD2001. LNCS 2265. Springer-Verlag. pp. 465-466. 2001.
21. R. Lloyd. Self organizing cognitive maps. The Professional Geographer, 52(3), 517-531. 2000

22. T. J. Lukka and J. V. Kujala and Marketta Niemel. Fillets: Cues for Connections in Focus+Context Views of Graph-like Diagrams. In Proc. Information Visualization 2002. IVS, IEEE. pp. 557-562. July 2002.
23. A.M. MacEachren, F. Hardisty, M. Gahegan, M. Wheeler, X. Dai, D. Guo, M. Takatsuka. Supporting visual integration and analysis of geospatially-referenced statistics through web-deployable, cross-platform tools. Proceeding, dg.o.2001, National Conference for Digital Government Research, Los Angeles, pp. 17-24. 2001
24. D.S. McCrickard & C.M. Kehoe. Visualizing Search Results using SQWID. WWW6: 6th International World Wide Web Conference. 1997.
25. M. Monmonier. On the design and application of biplots in geographic visualization. Journal of the Pennsylvania Academy of Science, 65(1), pp. 40-47. 1991.
26. P.J. Mutton and P.J. Rodgers. Spring Embedder Preprocessing for WWW Visualization. In Proc. Information Visualization 2002. IVS, IEEE. pp. 744-749. July 2002.
27. T. Munzner and P. Burchard. Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space. VRML '95, special issue of Computer Graphics, ACM SIGGRAPH, pp. 33-38. 1995.
28. Gabriele Neyer. Map Labeling with Application to Graph Drawing. In Dorothea Wagner and Michael Kaufmann, editors, Drawing Graphs: Methods and Models, volume 2025 of Lecture Notes in Computer Science, pages 247-273. Springer-Verlag, 2001.
29. K. Perusich, M.D. McNeese. Using fuzzy cognitive maps to define the search space in problem solving. In G. Salvendy & M. Smith & R. Koubek (Eds.), Design of Computing Systems: Cognitive Considerations. pp. 805-809. Elsevier Science. 1997.
30. H.C. Purchase, J.-A. Allder, D. Carrington. User Preference of Graph Layout Aesthetics: A UML Study. Graph Drawing (GD) 2000, LNCS 1984, pp. 5-18. 2000.
31. M. Raubal, M.J. Egenhofer, D. Pfoser, N. Tryfona. Structuring space with image schemata: Wayfinding in airports as a case study, Spatial Information Theory: A Theoretical Basic for GIS (Vol. 1329, pp. 85-102). Berlin: Springer-Verlag Berlin. 1997.
32. J. C. Roberts. Multiple-View and Multiform Visualization. In Robert Erbacher, Alex Pang, Craig Wittenbrink, and Jonathan Roberts, editors, *Visual Data Exploration and Analysis VII, Proceedings of SPIE*, volume 3960, pages 176-185. IS&T and SPIE, January 2000.
33. G. Sander. Graph Layout through the VCG Tool, In Proc, Graph Drawing, DIMACS International Workshop (GD'94), LNCS 894, pp. 194-205, Springer-Verlag. 1995.
34. G. Sander and A. Vasiliu. The ILOG Jviews Graph Layout Module. GD2001. LNCS 2265. Springer-Verlag. pp. 438-439. http://www.ilog.com. 2001.
35. W. Schnyder. Embedding planar graphs on the grid. Proc 1st ACM-SIAM Symposium on Discrete Algorithms (SODA'90), pp. 138-148. 1990
36. K. Sugiyama, S. Tagawa, M. Toda. Methods for Visual Understanding of Hierarchical System Structures. IEEE Trans. on Systems, Man and Cybernetics, 11(2), pp. 109-125, February 1981.
37. Tom Sawyer Software, http://www.tomsawyer.com/
38. Touchgraph Home Page http://www.touchgraph.com/
39. V. Tsai. Delaunay Triangulations in TIN Creation: An Overview and a Liner-Time Algorithm. International Journal of Geographical Information Systems, 7(6), 501-524. 1993.
40. Tulip Home Page http://www.tulip-software.org/
41. D. Tunkelang. JIGGLE: Java Interactive Graph Layout Algorithm. GD ’98, LNCS 1547. pp. 413-422. 1998.
42. W.T. Tutte. How to draw a graph. Proc. London Math. Soc. (3) 13 pp. 743-767 1963
43. C. Walshaw. A Multilevel Algorithm for Force-Directed Graph Drawing. GD 2000, LNCS 1984. pp. 171-182. 2001.
44. C. Ware, G. Franck. Evaluating Stereo and Motion Cues for Visualizing Information Nets in Three Dimensions. ACM Transactions on Graphics.15(2) 121-139. 1996.
45. D.F. Watson, A.I. Mees. Natural Trees -- Neighborhood-Location in a Nutshell. International Journal of Geographical Information Systems, 10(5), pp. 563-572. 1996.
46. G.J. Wills. NicheWorks -- Interactive Visualization of Very Large Graphs. Journal of Computational and Graphical Statistics 8,2 pp. 190-212. 1999.
47. K.-P. Yee, D. Fisher, R. Dhamija, M. Hearst. Animated Exploration of Dynamic Graphs with Radial Layout. Proc. IEEE InfoVis. pp. 43-50. 2001.