

# Constructing memorable asymmetric information diagrams in three dimensions

Brian Regan

University of Newcastle  
Callaghan  
NSW  
Australia  
mgbgr@newcastle.edu.au

**ABSTRACT** This paper presents a model for presenting relational information as a graph in three dimensions, where the emphasis is on the location of nodes by recognition of their locality in the drawing rather than the tracing of paths through the graph. The approach exploits the potential of 3D to present graphs with a high number of nodes specific to an individual user, such as a graph of web-page bookmarks. The key element is the creation of a memorable structure that contains a variety of sub-structures which act as reference points for the viewer as they manipulate their own viewpoint within the three dimensional virtual space of the graph layout. The layout algorithm used here challenges the need for an aesthetic of symmetry in graph drawing within 3D and exploits the user's ability to recall sub-structures within a drawing to replace edge-tracing as the means for locating nodes in the drawing.

**KEY WORDS** graph drawing, interaction models, World Wide Web; usability testing

## 1. INTRODUCTION

Graph drawing algorithms provide a mechanism for taking relational information and expressing this in a layout which can be displayed by a computer. With the increasing power of display technologies, graph drawing algorithms have moved from two to three dimensional presentations (Robertson, 1993). Three dimensions gives a new perspective on both the display space available for drawing a graph and the relationship of the users to the drawing of the graph on the display device.

The *Starship Algorithm* presented in this paper is a mechanism for displaying graphs with a large number of nodes (> 500) in three dimensions (3D) in

an environment where the users can move their viewpoint around the graph layout within the 3D drawing space.

In such an environment, the users need: (i) to be able to maintain their orientation such that they can identify areas of interest in the drawing, and (ii) to be able to recognise the location of a given node even though the resolution of the display or their own vision will be insufficient to read individual labels on nodes in a display of the full graph. The *Starship Algorithm* is designed to create a graph layout which is *memorable*, so that the users can both orientate themselves as they move within the 3D space and recognise distinct structural features as a guide to the location of individual nodes. One of the keys to

creating this memorable drawing is that the algorithm produces an *asymmetric* shape in the resulting drawing.

The algorithm takes an undirected graph, clusters the nodes to form a spanning tree over the graph which forms the basis for a structure based on a series of cone trees attached to a central spine. Of the many types of relational information that may be displayed - a user's collection of Web references and their inter-relationships are a candidate for representation by this algorithm. The result will be a distinctive and thus memorable shape, specific to a particular user.

## 2. MANAGING LARGE GRAPHS

In many applications, the size of graphs (in the number of nodes) to be presented to users is beyond the ability of a single display device to present all the nodes and their labels simultaneously on the screen such that individual items are discernable to the user. A number of strategies have been devised to present large complex graphs on displays of limited screen size.

### 2.1. Decomposition

Graph based software design techniques such as data flow diagrams (DFD) (Gane, 1979) have used a series of layers to present a large complex graph as a hierarchy of component diagrams. The component diagrams can be constrained so that they fit comfortably on a single display screen with enough space for each component to allow the user to read labelling information on both nodes and edges.

### 2.2. Windowing

Where the graph structure or the application is not amenable to decomposition, then the whole graph may be displayed as a single structure but with a scrolling window which highlights a section of the displayed graph drawing. The most common form of this can be seen in the tree structures displayed in file management software and software to manage bookmarks in web browsers, where the user has a scrollable window to scan a tree drawing of directories. Variations of this approach include the

fish-eye lens and perspective wall (Furnas, 1986) (Mackinlay, 1991) (Robertson, 1993).

### 2.3. Three Dimensional Motion

With the improving performance of 3D graphics software, large graphs can be drawn in a three dimensional virtual space with the user's viewpoint able to move within the virtual space. The complete graph drawing may still be too large for the whole graph to be seen with all the node and edge labelling readable, however, the user can control the position of their viewpoint to place it close to areas of the graph drawing of particular interest. Once the motion controls have been mastered, movement around the virtual space of the drawing is more intuitive, than the previous two techniques. The 3D drawings produced by the Starship Algorithm are device independent, such that they can be applied to planar display screens or sophisticated virtual reality environments. This approach is similar in effect to the scrollable windowing techniques mentioned previously, however, the 3D environment provides a depth of view which allows elements of the graph drawing to appear within view but at a distance which prohibits the display of their detail (such as labelling) whilst other elements are still close to the viewpoint. An example of this approach are cone trees (Robertson, 1991) (Hemmje, 1993) (Koike, 1993). The illusion of the 3D environment provides a sense that all components of the graph are viewable by simple manipulation of the user's viewpoint. It is into this last category to which the Starship algorithm belongs.

## 3. A MEMORABLE STRUCTURE

For each of the above drawing conventions, the focus has been on the user gaining information by tracing a path through the graph drawing (Ware, 1996). The emphasis on path tracing by users has encouraged aesthetics for evaluating graph layouts which focus on minimum number of bends in edges, minimum number of crossings and minimum edge length (Sugiyama, 1981) (Supowit, 1983).

Barreau (Barreau, 1993) studied the usage of file systems and found a preference by users for browsing the displayed tree structure of the file system

rather than waiting for the results of a search mechanism based on name. Even though the browsing time was often longer than the time taken by the search facility, subjects preferred to retain control of the searching process. In addition, the scanning of a graph drawing facilitates the identification of a node by 'recognition' of the pattern displayed by the graph drawing in the vicinity of the target node. On the other hand, specifying search criteria will require the 'recall' of the node's label and recognition memory has long been acknowledged as faster and larger in scope than human recall memory (Dix, 1993). Therefore a graph drawing and presentation environment which exploits 'recognition' of nodes or edges by their placement within shapes representing subgraphs within a drawing should facilitate convenient access to graph elements (nodes or edges).

The Starship algorithm creates a graph layout in three dimensions which forms a memorable shape for easier location of objects by recognition of their position. In a graph which must contain many nodes, the individual nodes will be difficult to identify individually, but the layout produced by the Starship algorithm creates a graph drawing with sub-structures which provide recognisable cues to locations.

#### 4. THE NEED FOR ASYMMETRY

Many 2D graph layout algorithms, particularly those concerned with tree layout (Supowit, 1983), have sought to satisfy aesthetic criteria which include a requirement for a symmetrical layout. Symmetry can enhance the aesthetic 'beauty' of a planar 2D layout, but Purchase (Purchase, 1995) found inconclusive results as to whether symmetry was useful in following the displayed graph.

The Starship algorithm seeks to create an asymmetrical layout because in the mobile viewing space of a 3D layout, the user needs as many distinctive features as possible to orientate themselves as they move their viewpoint around the layout of the graph. The primary objective of the Starship algorithm is to produce a layout which aids the user in remembering the locations of nodes in the layout. Symmetrical structures only increase the number of possible similar regions where a node may be located

and confuse the user's ability to locate the sub-structure to which they had previously visited.

#### 5. EXAMPLE APPLICATION

In order to place the discussion of the Starship algorithm within the context of an application domain, we will consider the requirements for maintaining references to information resources on the World Wide Web (Bookmark references to URLs). With increased usage of the Web, browser software packages have incorporated the facility to record the URL for sites of interest to a user in a 'hotlist' or 'bookmark list'. As the amount of time spent on the web and the number of interesting sites increases then the user will accumulate a lengthening list of interesting sites and management of this list becomes a bottleneck in terms of gaining access to previously visited sites.

The problem is analogous to the problem arising with early disk file systems (Eg IBM DOS/360 file system) where linear lists of files were considered satisfactory while disk drives were of limited capacity, however it was soon necessary to create the system of sub-directories in a tree structured hierarchy used by most contemporary file systems.

Commercial solutions to displaying web bookmarks have followed the same evolution as file systems to 2D layouts of a bookmark hierarchy. Graph layout methods for displaying hierarchies to exploit the potential of 3D layouts have been developed (Robertson, 1993) but these have not attempted to exploit the users familiarity with the information layout over time or move into 3D presentations.

Just as related files can be linked in a graph structure, so the details of interesting Web sites can be linked into a graph structure according to some semantic knowledge of their contents. The key attributes of the graph so formed are that it will be a network, contain a large number of nodes and be specific to an individual user. The Starship graph layout uses all these attributes to create a memorable shape for the user to locate their Web page references, with the possibility that the pre-existing hypertext links for a new page reference may be used to automatically provide the edges for the graph.

## 6. STARSHIP ALGORITHM

### 6.1. Definitions

A **cluster** is a connected sub-graph of nodes and clusters with maximum path length of two.

The **significance** ( $s$ ) of a node is a number representing the semantic importance of the node as a component of the information application by combining a weighting for the node and weightings for each edge attached to that node.

A **committed node** or cluster is one which has been assigned to a cluster.

A **core node** is the initial node in a given cluster.

A **major-edge** is an edge which is marked for inclusion in the spanning tree.

### 6.2. Algorithm Description

The Starship algorithm starts with an undirected graph over a set of nodes. In addition the algorithm may accept a weighting on a node or edge representing the semantic significance of some or all of the nodes or edges, which can be user defined and related to the semantics of the application domain. The algorithm goes through the following stages.

(1) **Order the nodes:** Each node is assigned a 'significance' derived by summing the weights of the edges attached to the node with the weight assigned to the node itself. The nodes are then sorted in order of decreasing significance.

(2) **Cluster the nodes:** Clusters of nodes are formed by scanning the node list, in a similar manner to the clustering used in (Tarjan, 1983). For each node a cluster is formed by grouping any node with an edge connected to the current node and which is not already committed to a cluster. Once every node is in a cluster, then the clustering continues by incorporating clusters into their neighbouring clusters. The choice of which cluster will incorporate the other is based on the cluster containing the node with the highest significance value. As each cluster is incorporated, the first edge identified as a link between the clusters is marked as a major edge. This process continues until all connected sub-graphs have been linked. Any uncommitted clusters apart from the first cluster created are linked by a created edge from their starting

node to the starting node of a cluster connected to the first cluster. This pattern of clustering mimics the growth of bacterial cultures; with each node being drawn into the cluster representing the local centre of activity in the graph.

(3) **Form Spanning Tree:** A spanning tree is now formed from the edges used to identify the clustering relationships, which minimises the path between nodes of greater significance.

(4) **Layout Nodes:** The longest path in the spanning tree is found and used as the spine of the layout along the x-axis. Thus nodes on this path are located along the x-axis. The distance between these nodes is determined by the maximum diameter of the cone trees formed for each of the sub-trees of the spanning tree at that given node.

### 6.3. Algorithm

i. Let  $G\{V,E\}$  be a graph over the set of nodes  $V$  and the set of edges  $E$ , such that:

(a) For every  $v_i \in V$  there is a weight  $w_{v_i}$  and for every  $e(i,j) \in E$  which connects  $v_i$  and  $v_j$ , there is a weight  $w_{eij}$

(b) For every node  $v_p$  with  $\text{degree}(v_p) = j$ , there is a significance  $s_p$ ,

where  $s_p = w_{v_p} + \sum_{i=1}^j w_{e_{pq}}$  and  $e(p,q)$  is an edge which

connects with  $v_p$  and  $v_q$ .

ii. Sort the node set,  $V$ , in order of descending node significance.

iii. For node  $v_i$  in  $V$  (where  $v_1$  is the node with the largest significance value and  $s_i > s_{i+1}$ ), if  $v_i$  is committed then repeat iii for  $v_{i+1}$ , otherwise form a new cluster  $C_j$  with  $v_i$  as the core node. Include in  $C_j$  all nodes  $v_k$ , where  $v_k$  is not currently committed and  $e(i,k) \in E$ . Mark all the nodes in  $C_j$  as committed. Mark edge  $e(i,k)$  as a major edge.

iv. For each cluster  $C_i$  (where  $i=1..c$  and  $c = |C|$ ), examine the nodes in order of their significance ( $s$ ).

For each node  $v_j$ , in  $C_i$ , if  $\exists$  node  $v_k$  in cluster  $C_p$  where  $e(j,k) \in E$ , and  $C_p$  is not committed (ie linked to another cluster), then mark  $C_p$  as committed, and mark  $e(j,k)$  as a major edge.

v. For any cluster  $C_i$ , apart from  $C_1$ , which is not committed; create a major edge from the core node of  $C_i$  to the core node of any other cluster.

vi. Form a spanning tree  $T(V,E')$  by including every major edge in  $E$ .

vii. Determine the longest path  $SP$  in  $T$ .

viii. Assign coordinates for all the nodes in  $SP$  along the  $x$ -axis.

ix. For each node  $v_{sp}$  in  $SP$ , create a cone tree for the sub-tree in  $T$  for every descendent of  $v_{sp}$  in  $T$ , which is not in  $SP$ , orthogonal with the  $x$ -axis.

x. The graph is then drawn with every cone tree initially replaced by a single cone to create the 3D drawing of the graph.

## 7. USABILITY TESTING

The key premise behind this layout is that by providing users with an asymmetric shape they will be able to remember the location of a node by recognition of the nearby features in the layout and that the features will be recognisable as the user's viewpoint moves around in the three dimensional viewing space. To validate these claims for the graph layout algorithm and more generally for the principle of drawing graphs in 3D using memorable asymmetric layouts, sample drawings will be trialed with some test subjects. Some user testing of graph drawings has already been conducted, [Purchase95] however, those tests have been able to present the test subjects with a variety of graph drawings and measure the ease with which subjects can trace paths on the drawing or locate objects in the drawing. The Starship drawings are designed to be optimal for the location of graph objects after the user has become familiar with the overall shape of the drawing. Thus a key aspect of the test

protocol will be the need for each test subject to become familiar with the location of information in the graph drawing which they may then be required to find again.

To ensure that all subjects understand the nature of the information contained in the graph drawing, yet also provide a large complex network graph, the test data is a graph structure based on movies, actors and directors. Each edge links either an actor or a director with a movie. The test procedure will consist of two phases:

(i) **Learning phase.** Subjects will be given a series of questions about movie actors and the system will highlight the location of the node containing the answer. The subject will then move their viewpoint to the node. In addition to training the subject on the location of information in the graph drawing, the time they take to navigate the user software to the highlighted node will be measured to provide a benchmark for that test subject's speed of manipulation.

(ii) **Testing phase.** The following procedure will be followed by subjects with either the Starship layout, a symmetric version of the Starship layout, or a linear alphabetical listing. The subjects will be given a series of questions, where each question contains either a request to find a node already visited in the learning phase or a new node which they had not previously visited but which is connected to a node visited in the learning phase. The software will measure:

(a) the time the user takes to locate the previously visited node - this will provide a measure of the memorability of the structure in the speed with which they return to the correct node.

(b) the distance along the axis of the regions where they hit in trying to find the given node - this will give some measure of locality to the subjects first positioning and so a measure of the memorability of the structure.

## 8. FURTHER DEVELOPMENTS

There are three main areas into which this

project is currently heading.

(i) The current area of activity is in implementing the software support required for the user testing. To be followed by the actual testing by users of the drawings produced by the algorithm.

(ii) The current version of the Starship algorithm creates a layout from input of a full graph theoretic structure. When slight variations occur in the input graph (for example, the addition of a single edge), the algorithm may generate a noticeably different layout. As an objective of the algorithm is to produce a layout wherein users will remember features of the resulting 3D drawing, it is important that the layout be stable under such minor change and thus an area of development is to find a relatively stable method for incremental change to the graph layout.

(iii) The other aspect of variability is in the user's viewpoint of the 3D structure being projected onto the plane of the screen or other viewing device. It is planned to include into the viewing software, an 'abridgement' procedure which will change some aspects of the drawing depending on the user's viewpoint. Thus when the viewpoint is distant from a sub-graph in the drawing, then the sub-graph may be replaced by a single shape.

## 9. CONCLUSION

The layout algorithm presented in this paper challenges the need for an aesthetic of symmetry in graph layout when we move into the realm of 3D and has proposed exploiting the user's ability to recall sub-structures within a drawing to replace edge-tracing as the means for locating information in a graph drawing. With the future movement toward 3D/VR presentations on the Web, it would seem appropriate to move the management of bookmark lists beyond the display formats of the first GUI's. Further work in this project is under-explored areas in graph drawing, such as, user testing of layouts, incremental graph drawing and abridgement of graph layouts.

## REFERENCES

- Barreau,Deborrah and Nardi,Bonnie. (1993) Finding and Reminding: File Organization from the Desktop. *SIGCHI Bulletin* 27(3):39-43.
- Dix, A. Finlay, J. Abowd,G. and Beale,R. (1993) *Human-Compter Interaction*. Hemel Hempstead: Prentice Hall.
- Furnas, G.W.(1986) Generalized fisheye views. *CHI'86 Proceedings*:16-23.
- Gane,C and Sarson,T. (1979) *Structured Systems Analysis*. Englewood Cliffs, N.J.: Prentice Hall.
- Hemmje,M.(1993) LyberWorld-A 3D-based User Interface for Computer-supported Information Retrieval in Documented Sets. *GMD-Spiegel*(1):56-63.
- Koike,H and Yoshihara,H. (1993) Fractal Approaches for Visualizing Huge Hierarchies. *IEEE* :55-60.
- Mackinlay, J. Roberston,G. and Card,S. (1991) Perspective Wall:Detail and context smoothly integrated. *Proceedings of SIGCHI'91* .
- Purchase,H. Cohen,R. and James,M. (1995) Validating Graph Drawing Aesthetics. Pp. 435-46 in *Graph Drawing*, (Ed) F. J. Brandenburg. Berlin: Springer.
- Roberston,G. Mackinlay,J. and Card,S. (1991) Cone Trees: Animated 3D visualizations of hierarchical information. *Proceedings of SIGCHI'91* 189-94.
- Robertson,G. Card,S. and Mackinlay,J. (1993) Information Visualization Using 3D Interactive Animation. *Comm of the ACM* 36(4):57-71.
- Sugiyama,K. Tagawa,S. and Toda,M. (1981) Methods for Visual Understanding of Hierarchical System Structures. *IEEE Transactions on Systems, Man, and Cybernetics* SMC-11(2):109-25.
- Supowit,K. and Reingold,E. (1983) The Complexity of Drawing Trees Nicely. *Acta Informatica* 18:377-92.
- Tarjan,R. (1983) An Improved Algorithm for Hierarchical Clustering using Strong Components. *Information Processing Letters* 17(July):37-41.
- Ware,C. and Franck,G. (1996) Evaluating Stereo and Motion Cues for Visualizing Information Nets in Three Dimensions. *ACM Transactions on Graphics* 15(April):121-140.