Visualisation of Large Networks in 3-D Space: Issues in Implementation and Experimental Evaluation

Yan Xiao* Paul Milgram

Abstract

Three dimensional visualisation has become a widespread scheme for helping users to access and manage large information network. In this report, various techniques for displaying depth information are reviewed, with an emphasis on stereoscopic displays. Input devices used to interact with a 3-D space are also examined. Issues in 3-D network visualisation are elicited from three viewpoints: psychological, task-related and implementational. Consideration of these issues leads to the design of a preliminary experimental programme for evaluating various network visualisation techniques.

1 Introduction

In numerous application areas, large data structures pose difficulties to users. Problems occur in access and manipulation due to the complexity and the size of these data sets. For example, in software engineering, designing, managing, and maintaining software systems become more and more challenging[1] [12]. Along with this challenge comes the growing need to develop effective supporting tools or environments for users such as software designers.

Presenting the objects within a database and their relationships in diagrammatic form has a strong, intuitive appeal, as inferred by the saying "a diagram is worth ten thousand words." Graphical representation makes use of human perceptual capabilities, and tends to be intuitive and easy to manipulate, thereby increasing our ability to deal with abstract networks. With the availability of high-quality computer graphics, the appeal of visual forms has manifested by a flourishing of visualisation tools. These tools attempt to assist designers and others to access and manage information about large networks—sets of objects and their relationships [11, 5, 2].

However, computer graphics alone does not guarantee better and successful tools, as stated by Fairchild, Poltrock, and Furnas [5]: "...graphic representations do not automatically solve all problems associated with exploring, manipulating, and modifying very large knowledge bases. They simply transform very large knowledge bases into very large directed graphs." How to reduce visual complexity and make better use of the human perceptual system, as well as provide interactive control of what is being presented, are clearly primary concerns in building network visualisation tools.

A research project was started to address these concerns. Its aim is (a) to explore issues in implementing visualisation of complex networks in 3-D space, and (b) to evaluate various implementations experimentally. In particular, we are attempting to answer a number of questions related to network visualisation in three dimensional space: (1) Will network visualisation in 3-D help? (2) What are the benefits and costs of network visualisation in 3-D space? (3) What cues should be displayed to enable 3-D perception? (4) What control devices should be used to interact with networks in virtual 3-D space? Initial attempts to display networks in 3-D have been providing some guidance in defining issues

*IBM contact for this paper is A. G. Ryman, IBM Canada Laboratory, 81/894/895/TOR, 895 Don Mills Road, North York, Ontario, M3C 1W3. The first author is supported by a grant from Center for Advanced Studies, IBM Canada.

[1] This project was motivated by difficulties encountered in using GraphLog with very large networks (over 1500 nodes and arches) generated in software engineering [3].

and assessing implementation obstacles.

This paper starts with a general discussion of network visualisation in 3-D, then examines the psychology of human depth perception, that is, the various cues that humans use in depth perception. A number of techniques for providing these cues are reviewed, and issues in implementation and evaluation are raised, based on the initial experimental trials. Consideration of these issues leads to the design of a preliminary experimental programme for evaluating various network visualisation techniques.

2 Network Visualisation in 3-D

2.1 Essence of Network Visualisation

Visualisation has two components: to create a virtual space of abstract objects and relationships, and then to present the space visually. It exploits one or more human visual perceptual capabilities, and adopts ways in which we interact with objects in the real world to achieve its effectiveness.[2]

Part of the first component is to map a network into a virtual space (in 2-D or 3-D) by assigning each network node a coordinate. This process is known as *network placement*.[3] Usually the network structure specifies only linkage among nodes; therefore there is a large degree of freedom in the mapping or placement process. This freedom makes it possible to group nodes in many ways, and form certain familiar geometrical patterns. For example, nodes at the same level can be lined up, or combined to form the base of a cone. The virtual space need not be homogeneous—a fisheye view can be used. The present project does not deal with the creation of such virtual spaces, although there is a strong relationship between this component and the second to present the space visually.

The virtual space created should be displayed in such a way that users can perceive it easily. In other words, the question "which visual cues are to be exploited, and how?" has to be resolved. This is a central question that the project is attempting to answer.

One important factor in network visualisation is the mode of user interaction: the way in which users control the presentation of networks and manipulate them. This factor is also included in the scope of the study.

2.2 Pros and Cons of Network Visualisation in 3-D

Three-dimensional computer graphic display technology, which includes stereoscopic viewing, has become a practical reality, and many of the associated advantages are well accepted [4, 13]. Nevertheless, the utility of 3-D displays for specialised applications, in particular for visualising network software structures, must still be investigated within the specific relevant framework. The fundamental question being addressed in the current research is: Will 3-D graphical representation allow one to better visualise a network in terms of its system structure and properties?

2.2.1 Advantages of 3-D network visualisation

Network representation in 3-D space has a number of recognised advantages over representation in 2-D space.

First of all, it can eliminate, or at least alleviate, the arch crossing problem in 2-D space. One important aspect of visual complexity is the extent of arch intersections. For two-dimensional placement, this will clearly pose a serious problem. However, because any network of nodes and arches can be embedded within a three dimensional space with no intersecting arches. This suggests that using a 3-D representation of a network can greatly enhance the likelihood of its being satisfactorily visualised. One example of such a system is SemNet [5].

Secondly, a 3-D solution gives more freedom within the network layout to show relationships and structures that can not otherwise be represented by the network.[4]

[2] The effectiveness of visualisation, however, has hardly been proven experimentally. See [8, 1].

[3] In [5] the term "positioning knowledge elements" is used for network placement.

[4] In a 2-D space, proper network layout is constrained by having to avoid intersecting arches.

Thirdly, a 3-D representation affords operations such as changing view angles, "travelling" through [5], or rotating a part or the whole of the network. (This strategy has been used in, for example, the Cone Tree structure in the Information Visualizer [11]).

Finally, linear perspective in 3-D space provides a naturally scaled view along the depth dimension. That is, the scale along the depth dimension is not uniform; objects are smaller when more distant.

2.2.2 Disadvantages of 3-D network visualisation

There are, according to Wickens [22], three potential costs in deploying 3-D network visualisation techniques: (1) Human perception of depth (or z-dimension) information, both in artificial and natural viewing conditions, is poorer than that of its orthogonal image plane (or xy-plane) information. (2) In an integrated display of 3-D space, the added information to show depth may result in reduced precision in reading values along any one particular axis. (3) There are more design issues to be tackled. The network placement will certainly be more difficult, for example.

2.2.3 Unique properties of displaying network information

Displaying networks three dimensionally has some unique features. The dominant one is that a network does not have inherent dimensionality. Given a placement of a network in 3-D space, a simple projection of it onto the display monitor may still contain very little information on depth. Using lines to represent the linkage relationships,[5] the network displayed does not usually have volume objects, whereas most existing techniques of 3-D displays use volume and segregated objects. However, in the placement process, volume objects can in fact be created in some circumstances. For example, cone trees can be used to represent a hierarchical, tree-like structure. From a task point of view, network visualisation is different from other traditional 3-D application (such as in flight navigation), where speed, time, and path are important factors. Related to the dimensionality issues, in network visualisation, spatial location is not the central concern to a user because the particular spatial location is simply the product of the placement algorithm.

[5] Johnson and Shneiderman [7] used a Venn-diagram-like scheme to display hierarchical networks. This is the only method known to us that does not use lines and points to represent network information.

2.3 Summary

This section has reviewed the task related aspects of network visualisation. Of the two major components, the creation of a virtual space and presentation of the space visually, only the latter has been examined. From the task point of view, the questions to be examined are: (1) In what way do 3-D displays aid users in visualising and manipulating network structure? (2) Will navigation be harder or easier with 3-D displays?

Essentially, most applications of 3-D computer displays require the user of the system to perform one or more of the following perceptual skills: (1) recognition or visualisation of objects, (2) recognition or visualisation of relative spatial locations of objects, and (3) recognition or visualisation of physical and/or functional connections between objects. It is our belief that the latter two perceptual skills are the most relevant for visualising complex network structures.

3 Human Depth Perception

Perceiving objects in depth is somewhat mysterious, considering that images falling on the retina are essentially flat. The information or cues that enable humans to do this are numerous. This section reviews the various cues in depth perception, and possible ways of providing them. Wickens, Todd and Saidler's report [23] give a comprehensive review of the topic, but that review emphasises the application of real world images such as those found in aviation [10]. The following review is combined with a discussion of usage of these cues in a more synthetic (*i.e.*, virtual) world—network visualisation.

A systematic review of various ways of providing depth cues is conducted in this section, from both a psychological viewpoint and a technical (or implementational) viewpoint. The focus below is on integrated cues, that is, cues that are on one display. Other means exist for displaying three dimensional information, that provide depth information on more than one display at a time (either in parallel or in sequence). Typically, this method uses two projections of a network in 3-D space along different projection axes. Radiologists, for example, have shown that humans can form a 3-D representation mentally by looking at projected images. Such images allow greater precision in judgment in the projected plane, and thus also have a role in network visualisation.

For the purpose of the present project, the depth cues discussed are divided into two categories, according to whether or not a depth cue is present when the display is static or dynamic.

3.1 Static Depth Cues

Shadows. In the real world, light causes shadows, which in turn provide depth information. In a similar way, objects can be created graphically that have shadowed surfaces. The drawback of utilising this cue is that, in order to be effective, large surfaces have to be used, and thus the realisable number of nodes displayed on the screen is reduced. Cast shadows are also useful, as demonstrated in the Cone Tree in the Information Visualizer [11]. . An auxiliary plane has to be used for casting shadows, however.

Linear perspective.
Because parallel lines converge at distance. objects far away appear smaller than they really are; This perspective cue is very useful in displaying natural scenes, but not as useful in the network environment because of the lack of familiar geometrical forms such as symmetry.[6] Nonetheless, the linear perspective is analogous in many ways to fisheye-view effect, increasing displaying density as depth increases.

Colour-distance covariance. The filtering effect of the atmosphere reduces the luminance and colour saturation of distant real-world objects, and thus luminance and saturation provide implied depth information. These are not strong cues, but can be implemented jointly with other cues, if feasible.

Size consistency. Familiarity with the environment gives us clues about distance: objects of the same size appear in different sizes when not at the same distance. In the virtual space of network representation, acquiring familiarity with sizes of objects will be difficult, however, and hence there is little reason to assume that this cue will have a substantial impact.

Occlusion. In the real world, objects in the front always occlude those in the back. Although occlusion is considered to be a powerful cue in other environments [9], in displaying network structure, there are usually no volume objects, and thus occlusion is unlikely to be a strong cue for depth perception. In addition, occlusion gives information only about the relative depth of objects, not on their absolute depth.

Texture. This cue is associated only with surfaces, which networks do not have. (However, surfaces created can use this cue to show their position and orientation.)

Binocular depth cue. This cue is one of the most salient cues, as noted by Wickens, Todd & Seidler [23, p.100]: "Stereoscopic displays, ..., provide binocular depth cues which significantly enhance performance, particularly when visual enhancement cues are not presented". Under natural conditions, the left and right eyes locate in positions that are about 65 mm apart. The resulting difference in viewpoints for each eye gives information regarding the depth of an image, although how the human brain actually decodes this depth information is still under investigation.

3.2 Dynamic Depth Cues

Relative motion and motion parallax.
This is a very strong cue when a dynamic display is available [14]. The drawback of using this cue is that once the display is static,

[6] The cone tree display, in fact, generates symmetrical forms that allow the use of perspective cues. This particular method works only for hierarchical networks, however.

the effect[7] disappears. In addition, successive updates have to be smoothly produced and at a relatively high frequency, which is difficult when graphic drawings are complex, even for very powerful workstations.

Active parallax. This is the cue that comes from the integration of changes in perceived image and changes in the head position. Because active parallax requires close coordination between head movement and displayed images, and, therefore, rapid display updating, complex networks are usually beyond the possible application of this cue.

Preservation. When an object moves in depth, humans make the assumption that attributes of the object remains unchanged. For instance, a smaller size after movement signals an increase in depth.

3.3 Discussion of the Use of Depth Cues

From the body of existing studies and preceding review, it is clear that the following five cues are potentially the most useful cues for the network visualisation: (1) binocular disparity, (2) relative motion and motion parallax, (3) linear perspective, (4) proximity-luminance and proximity-saturation covariance, and (5) shadows.

A consistent finding in 3-D perception research is that the greater the number of cues provided, the better the depth perception [9, 22, 14, 10]. However, there is usually a cost associated with the display of each cue. Choosing a right combination requires both basic psychological studies, and empirical testing in a particular application domain.

Dynamic cues are usually very strong but require rapid updating. This fact gives static cues an advantage in the area of visualisation of large complex networks. Among the static cues, binocular disparity is recognised as the strongest. In contrast to dynamic cues, the binocular cue exists even when the displayed images are static, allowing viewers to gaze at the network. It is, therefore, relatively easy to maintain this cue no matter how complex the network is. The drawback is that the generation of the binocular depth cues always requires some dedicated hardware for separating images to the left and right eyes. In one way or another, lenses or goggles will have to be worn in the foreseeable future. Will the benefits of the binocular depth cue offset these drawbacks? This is a question that has to be answered empirically.

When using multiple cues simultaneously, there is a danger that different cues give conflicting information. That is, one cue might suggest that an object is in the foreground while another cue might suggest the opposite. Therefore, even though some cues are not significant, consistency has to be maintained to avoid conflicts.

In summary, we are proposing that the binocular depth cue will be the preferred primary cue to induce depth perception for complex 3-D network visualisation. Motion cues are also very salient, but, their implementation is limited by the complexity of graphics. Colour cues (luminance- and saturation-proximity covariance) are also useful, but are less salient. Shadows are a useful cue, but their use competes for screen space. The perspective cue depends directly on network placement and orientation, and is subject to certain conditions on the structure of networks (such as whether or not a network is hierarchical).

From the point of view of depth perception, the questions to be examined are: (1) which cues are most beneficial for depth perception? (2) will the advantage of providing various cues offset the associated cost?

4 Stereoscopic Display Implementation

As suggested by the discussion above, providing binocular disparity cue, or stereopsis, is probably the most promising cue in the 3-D network visualisation. In this section, the implementation of providing this depth cue on computer displays is described, and associated issues discussed.

The general approach is to provide each eye with a slightly (horizontally) shifted image of an object. There are, according to one taxonomy, two methods of providing stereopsis: time parallel and time multiplexed methods [6], that is, whether or not images are viewed by two eyes

[7] This is the so-called "kinetic depth effect", or "structure through motion" [18, 17].

simultaneously or alternately.

4.1 Time Parallel Method

The anaglyph method is probably the simplest method of showing two images at the same time. An obvious limitation is that only one or two colours can be used in one display. Random dot stereogramme is another simple way to achieve stereopsis, though no serious application has been known,

Using two optical paths to feed images from two computer screens is a powerful (and expensive) method of generating binocular disparity cue. This method is often implemented by a helmet-mounted display [15].

4.2 Time Multiplexed Method

Using one display monitor, but switching between each of the two eyes alternately, is by far the easiest way to achieve stereopsis. This method requires displaying two images (left and right) alternately to each eye. Users view through a liquid crystal shutter system that synchronises with the display. To eliminating flickering due to the alternation, a frequency of usually higher than 40 Hz for each eye (total of 80 Hz) should be used [16].

Two techniques can be used to implement this. By putting a shutter system that polarises light in front of the display monitor, simple polarised lenses will ensure that each image is viewed only by the appropriate eye. This (sometimes referred as passive) system is considerably more expensive than the following technique, but it results glasses that are lighter and simpler, and much less expensive.

The other technique puts the shutter system in the viewing glasses, which are synchronised to the display monitor either by direct wiring, or by infrared or ultrasound radiation signals.

Either of the techniques requires the generation of left and right images alternately on a single display monitor, which is described below.

4.3 Generation of Alternating Images

The popular way of generating alternating images is by the screen mapping method. With this method, left and right images are mapped onto the top and bottom halves of a single display buffer. Hardware alternately maps the top and bottom half display buffers to the whole display screen at a high frequency (usually 120 Hz in total). Therefore, there is no synchronisation problem associated with the non-mapping methods (described below). The biggest drawback is the difficulty for such displays to coexist with other regular (*i.e.,* non-screen-mapping) window applications. Loss of half of the vertical resolution in the mapping process is another severe shortcoming with present technology.

The non-mapping methods, on the other hand, require software to draw left and right images. In the SGI™ GL™environment, this can be done by either of the following two techniques.

4.3.1 Colour Map Cycling Methods

Mechanism. Instead of specifying directly what colour to use in drawing, an index can be used. This index is mapped to a colour (which in turn is specified by attributes in red, green, and blue). One of the advantages is that the assignment of an index to a colour, or colour map, can be changed during the course of displaying. The computer can alternate the assignment of colours at a fixed frequency, and therefore make an object appear in an invisible colour (that is, the same as the drawing background) by changing the index assignment. Thus, left images and right images can be displayed in turn.

Pros and Cons. Under multiple window conditions, using colour map swapping technique will interfere with other windows. Also the number of colours that can be used is quite limited (around 60 on a machine with a 24 bitplane graphic board). Despite these disadvantages, colour swapping has the advantage that left and right images are alternately shown at a fixed rate,[8] regardless of the complexity of the image to be drawn.

[8] In our experience so far, although the rate is fixed, for unknown reasons, occasionally the correspondence of left-right is reversed; and this is very difficult to detect and correct using software.

4.3.2 Buffer Swapping

Mechanism. A somewhat more straight-forward method of showing left or right images alternately is to use multiple graphic buffers and buffer swapping: one buffer holds each of the left and right images, and the buffers are swapped in high frequency (usually the same as the monitor frequency).

Pros and Cons. Since drawing with the buffer swapping method is the same as regular non-stereo drawing, there is no limitation on the number of colours that can be used. However, whenever a drawing takes longer than the basic cycle time (that is, the period for displaying one buffer, which is usually set to less than 1/60 second to avoid flickering), a swap will not take place until the next cycle time, and, therefore, the left image is shown to the right eye due to the missed cycle. In this case, a reversal of left-right correspondence occurs. With complicated drawings used for large networks, this loss of consistent left-right correspondence is unavoidable. Even simple drawings in a multiple window system will cause this correspondence reversal. A partial solution to this problem has been found by using the system clock to check missed cycles. With this solution, flickering is introduced as the missed cycles are detected *a posteriori*, and a correction is done by inserting extra missed cycles to bring the correspondence back to normal.

The flickering should not be a serious problem for network visualisation because it occurs only when a display has to be redrawn. However, the flickering will likely reduce the attractiveness of such stereoscopic displays. Some solutions have been proposed to reduce or eliminate the flickering, such as reduce drawing complexity when a rotation is being done.[9] This can be achieved either by reverting to monoscopic display mode or drawing a degenerated network. One special attribute of drawing a degenerated network and then incrementally filling in the details is that this will decrease the response time to the user's input, regardless of whether the display is monoscopic or stereoscopic.

[9] For one thing, moving objects provide more depth cues than static ones, and thus, losing binocular cue may be tolerable in some circumstances.

4.4 Summary

As stated above, there are a number of ways of providing the binocular disparity cue, each having some cost attached. The issues to be examined from the implementation point of view are: (1) the effects of display resolution problems when the screen mapping method is used, (2) the effects of flickering problems when non-screen mapping methods are used, (3) the coexistence of stereo-windows and non-stereo-windows in a multiwindow environment, (4) user acceptance of a particular implementations, and (5) the effect on the visualisation quality due to degenerated drawings dynamically displayed.

5 Input Devices

5.1 Various metaphors

Advances in computational power have made it possible to display and manipulate complicated 3-D scenes in real time. With the advances, there comes the need for devices to allow the users to interact effectively with a 3-D space. Especially after the availability of stereoscopic displays, the need for a better way to interact with objects in 3-D space is even greater.

The computer screen provides a virtual reality in a 3-D space. Because it is a virtual space, different metaphors have been proposed to mimic the ways in which we manipulate objects in the real world; each of them represents a class of spatial interactions that we have with our surroundings:

1. *World-in-hand metaphor*—manipulating the world with a hand. To present views from different angles, objects in the virtual world are rotated and translated as a whole.

2. *Object-in-hand metaphor*—manipulating an object with a hand. An object is manipulated as if it were held by hand, while the rest of the objects in the virtual world are still.

3. *Eyeball-in-hand metaphor*—locating a viewpoint. A viewpoint is placed in the 3-D space, and then the virtual space is presented according to that viewpoint.

4. *Flight metaphor*—locomotion with eyes. The *movement* of a viewpoint is controlled, giving a perception of travelling through the virtual world.

5. *Virtual-hand metaphor*—indicating a point in space. A point is placed in a 3-D space by judging its location visually while the virtual world remains still.

Clearly, each metaphor is a collection of controlling strategies, which are suitable or required for certain tasks. Instead of viewing these metaphors as rivals to each other, it is conceivable that combinations of these strategies are to be used to carry out a given task.

5.2 Taxonomy of Three Dimensional Input Devices

An object in 3-D has six degrees of freedom (d.f.). To completely control an object in 3-D requires an equal number of d.f. in the input signal. There are a number of ways that six dimensional (three translational and three rotational) controlling signals can be generated. They fall into two categories:

- Natural controller. The input device itself is configured in such a way that it can be viewed as an object in 3-D space, with each of its six dimensions (three translational and three rotational) mapping onto one dimension of the required input.

- Virtual controller. The input device does not have a spatial layout mapped naturally to a 3-D object, but nonetheless it can generate controlling signals of six d.f. An example is to use the traditional 2-D mouse with the help of button or key combinations.

5.2.1 Natural Controllers

Within this category, there are three major classes of 3-D input devices:

- *Force input* devices use forces applied as input signals, as in Spaceball™. Users get feedback of force, which is a function of input magnitude, *i.e.*, the bigger the input, the bigger the force needed.

- *Spatial input* devices use spatial positions and orientation as input signals, as in Flying Mouse™, Bird™, and IsoTrak™. Usually a "clutch" metaphor is used; that is, users define a starting point by pressing a button, and the input magnitude is proportional to the distance moved away from the starting point.

- *Gesture input* devices use hand gestures, as in DataGlove™, which is in fact a superset of the above class: hand position and orientation provide a spatial input, and finger positions (flexion and extension of fingers) indicate gestures.

5.2.2 Virtual Controllers

This category of controllers refers to controlling mechanisms that use 2-D input devices to generate 6-D controlling signals:

- Using dialogue windows. By using a dialogue window, a movement of a 2-D mouse or trackball can mean an input in different dimensions depending on where the movement occurs within the window, which is divided into several areas. More than one dialogue window can be used to indicate both translational and rotational inputs. Another method of using dialogue windows is to show miniature global views from different angles, as in SemNet.

- Using key combinations. A dialogue window is not always feasible or necessary. Keyboard or buttons on a mouse/track ball can increase the dimensionality. The same idea applies to extra dials on some trackballs (*e.g.*, FastTrack™). Typical use of this method is to add depth input by using buttons or dials.

5.3 Choices of Input Devices

As mentioned earlier, different tasks require the use of different strategies. For example, shooting a film is better done using the strategy of the flight-metaphor [21, 19]. The metaphors listed above encompass a wide range of tasks involved in the interaction with a 3-D space, and therefore 3-D input devices can be tested by serving

Metaphor	Natural Controller	Virtual Controller	
		Dialogue	Key
Flight-metaphor	Best	No	Yes
Eyeball-in-hand metaphor	Yes	Better	Yes
Object-in-hand metaphor	Better	No	Yes
World-in-hand metaphor	Yes	Better	Better
Virtual-hand metaphor	Yes	No	Yes

Table 1: Hypotheses

these metaphors. The *flight-metaphor, object-in-hand metaphor,* and *virtual-hand metaphor* require continuous input and constant eye-hand coordination, whereas the *eyeball-in-hand* and *world-in-hand* metaphors do not require input continuity. Table 1 shows the hypotheses as to which controllers can be, or are better, used to serve the corresponding metaphors.

5.4 Other Parameters of Input Devices

The choice of hardware is only one parameter in choosing input devices. Given a particular input device, there are still a number of other parameters or variables that govern the profile of that device. Of these the control mode is of particularly interest, that is, whether velocity control or position control is used [20].

6 Design of Experiment

The basic goals of the present research are to evaluate and to explore issues associated with providing 3-D displays and manipulation tools with various control and display implementation options.

6.1 General Methodology

Two methods can be used for the evaluation and exploration of a 3-D viewing system: (1) creating a "toy" world—a simplified version of the target world—so that the major issues are addressed in that world, and (2) testing a few exemplary tasks in the target world. The first method gives the flexibility of investigating each factor individually or in a desired combination, but at the same time it requires proper mapping of the toy world onto the real, target world. Although the second method does not have the mapping problem, it does not have much flexibility in experimental manipulation, either.

6.2 Choice of Experiment Task

To effectively fulfill the research objectives, a combination of both methods is proposed: toy world tasks are used to investigate individual factors, and exemplary tasks are used to test how well a particular implementation serves in the target world: helping software designers. On the choice of tasks, it is also important to note that our objective is not to investigate the user's abstract reasoning or problem-solving abilities, but to test the efficacy of using 3-D displays for this class of application. It is, therefore, important that our experiment be designed to concentrate on *perceptual* issues, rather than cognitive issues.

Task 1: Streamlining Networks

> A hierarchical network (*i.e.,* a tree) is initialised by randomising the location of all nodes, with the root node highlighted or marked. The subject is asked to put the randomised network back into a tree format according to the given root node so that no sub-tree is invading others.

Task 2—Reducing Cyclicity

> A network of a (possibly real) database describing a program is used as a stimulus, and the subject is asked to optimise the program in terms of its cyclicity—the number of arches that are participating in calling or referring loops.

Task 3–Overlaying Program

> A network of a (possibly real) database describing a program is used as a stimulus, and the subject is asked to rearrange the program so that it fits into a given size of computer memory by using overlay structure.

The first task is intended to test the dexterity of input devices. A 3-D visualisation system should allow users to manipulate a network with ease—moving around the network and relocating the elements in it without much difficulty. The latter two tasks are proposed because they have been used in the research on G^+ and GraphLog, and were shown to be effective in revealing how users deal with the underlying structures of a program [3]. Task 1 requires only that the subject explore a simple structure within a network, *i.e.,* the hierarchical relationship. Design tasks typically involve a richer network structure. Task 2 and 3 are chosen to investigate how an implementation of a 3-D visualisation system helps users perform design tasks.

6.3 Experimental Variables

Independent Variables. There are three independent variables in the experiment:

- Input device. The choice of input device (natural versus virtual controller) and user control mode (different metaphors, and velocity versus position control).
- Display. The mode of display (2-D versus 3-D display, cue combination, and fisheye view), and the display quality (screen resolution, update rate, labelling strategy, and degree of degeneration of drawing).
- Task. The task element (pointing, navigating, understanding, editing and re-placing), task environment (whether or not users can choose the input device or display mode), and task difficulty (the number of arches and nodes in a network, the properties of the network structure).

Dependent Variables. The dependent variables in the experiment tasks include:

- Subjective ratings
- Task performance measures (either task completion time or quality of the final network), and/or training time
- Subjects' choice of input devices
- Subjects' choice of viewing conditions.

7 Summary

Implementation and evaluation of a 3-D visualisation system involves technical, task-related, and psychological factors. In this article, all of these factors are examined and issues are raised for empirical testing and evaluation. In some situations hypotheses are presented. In conclusion, it is evident that graphical presentation in 3-D space has the potential to provide a better environment or interface for users to deal with large networks. However, much research needs to be done to examine the pros and cons of various techniques of 3-D visualisation, partly because of the uniqueness of the application of network visualisation. Major salient cues enabling depth perceptions include binocular disparity, linear perspective, and motion cues. It is our contention that binocular disparity holds the most promise for network visualisation. Three major factors in the design of an experiment to evaluate various implementations are: input devices and interactive mode, display mode and quality, and task requirement.

Acknowledgement

This project has received kind support from Shumin Zhai, David Drascic, Peter Wong, Ken Ruffo, Ferdie Poblete, and Sarah Zuberec in the Human Factors Laboratory, Department of Industrial Engineering, University of Toronto. I would like to thank Arthur Ryman for his encouragement, patience, and guidance, and an anonymous reviewer for his/her meticulous review on an earlier draft.

References

[1] F. P. Brooks. No silver bullet: Essence and accidents of software engineering. *IEEE Computer*, pages 10–19, 1987.

[2] F. P. Brooks, M. Ohu-Young, J. J. Batter, and P. J. Kilpatrick. Project GROPE—Haptic displays for scientific visualization. *Computer Graphics*, 24(4):177–185, 1990.

[3] M. Consens, A. Mendelzon, and Arthur G. Ryman. Visualizing and querying software structures. Technical report, IBM Canada Toronto Laboratories, Toronto, Canada, 1991.

[4] D. Drascic. Skill acquisition and task performance in teleoperation using monoscopic and stereoscopic video remote viewing. In *Proceedings of Human Factors Society 35th Annual Meeting*, pages 1367–1371, 1991.

[5] K. M. Fairchild, S. E. Poltrock, and G. W. Furnas. *Semnet: Three-dimensional graphic representations of large knowledge bases*, pages 201–233. Erlbaum, Hillsdale, NJ, 1988.

[6] L. R. Hodges and D. F. McAllister. Stereo- and alternating-pair techniques for display of computer-generated images. *IEEE Computer Graphics and Applications*, 5(9):38–45, 1987.

[7] B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of Visualization'91*, pages 284–291. IEEE, 1991.

[8] Jill H. Larkin and Herbert A. Simon. Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, 11:65–99, 1987.

[9] K. M. Mazur and J. M. Reising. The relative effectiveness of three visual depth cues in a dynamic air situation display. In *Proceedings of Human Factors Society 34th Annual Meeting*, pages 16–20, 1990.

[10] D. H. Merwin and C. D. Wickens. 2-D vs. 3-D display for multidimensional data visualization: The relationship between task integrality and display proximity. In *Proceedings of Human Factors Society 35th Annual Meeting*, pages 388–392, 1991.

[11] G. G. Robertson, J. D. Mackinglay, and Stuart K. Card. Cone trees: Animated 3D visualizations of hierarchical information. In *Proceedings of CHI'91*, pages 189–194, 1991.

[12] Arthur G. Ryman. Personal systems image application architecture: Lessons learned from the ImageEdit program. *IBM Systems Journal*, 29(3):408–457, 1990.

[13] Randy L. Sollenberger and Paul Milgram. Stereoscopic computer graphics for neurosurgery. In *Proceedings of Third International Conference on Human-Computer Interaction*, pages 294–301, Boston, MA, 1989. Elsevier Science Publishers.

[14] Randy L. Sollenberger and Paul Milgram. A comparative study of rotational and stereoscopic computer graphic depth cues. In *Proceedings of Human Factors Society 35th Annual Meeting*, pages 1452–1456, 1991.

[15] I. V. Sutherland. A head-mounted three dimensional display. *Proc. of Fall Joint Computer Conf.*, pages 757–764, 1968.

[16] P. Turner. Critical flicker fusion frequency and its modification by a conditioning stimulus of flickering light. *Journal of Physiology*, 171:6–8, 1964.

[17] S. Ullman. *The interpretation of visual motion*. MIT, Cambridge, MA, 1979.

[18] H. Wallach and D. H. O'Connell. The kinetic depth effect. *Journal of Experimental Psychology*, 45:205–217, 1953.

[19] C. Ware. Using hand position for virtual object placement. *The Visual Computer*, 6:245–253, 1990.

[20] C. Ware. Using velocity control to navigate 3D graphical environments: A comparison of three interfaces. In *Proceedings of Human Factors Society 35th Annual Meeting*, pages 300–304, 1991.

[21] C. Ware and D. R. Jessome. Using the Bat: A six-dimensional mouse for object placement. *IEEE Computer Graphics & Applications*, November:65–70, 1988.

[22] C. D. Wickens. Three-dimensional stereoscopic display implementation: Guidelines derived from human visual capabilities. *Stereoscopic displays and applications*, SPIE Vol. 1256:2–10, 1990.

[23] C. D. Wickens, S. Todd, and K. S. Seidler. *Three-dimensional displays: Perception, implementation, and applications* (CSERIAC Report 89-001). CSERIAC, Wright-Patterson Air Force Base, Ohio, 1989.

About the author

Yan Xiao is a PhD candidate in the Department of Industrial Engineering at University of Toronto. He is working in the field of cognitive engineering.

He can be reached at the Department of Industrial Engineering, University of Toronto, Toronto, Ontario, M5S 1A4. His e-mail address is `xiao@vered.rose.utoronto.ca` on Internet.

Paul Milgram is an associate professor in the Department of Industrial Engineering at University of Toronto. He has been working for a number of years on modelling of human performance and the design of information displays for complex systems in areas including space, underwater, nuclear, medical, and military applications.

He can be reached at the Department of Industrial Engineering, University of Toronto, Toronto, Ontario, M5S 1A4. His e-mail address is `milgram@gpu.utcs.utoronto.ca` on Internet.