Implementation and Evaluation Scheme for Stereoscopic Display of Networks

Yan Xiao* Paul Milgram* A. G. Ryman†

Abstract

The use of three-dimensional computer graphics in visualisation has become widespread for helping users to access and manage large information networks. This report describes a project that used dynamic, stereoscopic displays for network visualisation in 3-D. Input devices used to interact with a 3-D space are also examined. The project shows the feasibility of stereoscopic technology for network visualisation, and highlights some important research issues, including the selection of multi-degrees-of-freedom input devices.

1 Introduction

With advances in information technology, the amount of data to which we have access is growing exponentially. Due to the complexity and size of many data sets, difficulties often arise in accessing, displaying, and manipulating large data structures. For example, in the design of software systems, the existence of thousands of objects with complex interrelationships often prevents the designer from having a proper understanding of the underlying structure among the objects, thereby creating serious obstacles to designing, implementing, and maintaining large software systems.

One recent effort aimed at overcoming these obstacles has led to a graphic query language, GraphLog (Consens, Mendelzon, & Ryman, 1992), that combines the powers of query and of visualisation. However, difficulties still exist when using GraphLog, many of which are due to the fact that graphs are conventionally laid out in either 2-D space or 3-D space, but are presented as 2-D projections. Using 3-D computer graphics in visualising complex databases has attracted great attention recently (*e.g.*, Hirose & Amari, 1991; Hirose, Myoi, Amari, Inamura, & Stark, 1990; Fairchild, Poltrock, & Furnas, 1988; Robertson, Mackinglay, & Card, 1991). These attempts represent an exciting trend in the use of 3-D presentation in user interfaces.

A research project was started to investigate various issues in displaying network information using stereoscopic 3-D. Often referred to as *true* 3-D, stereoscopic display technology utilises human stereopsis to enhance the ability to perceive depth. This report describes this project and discusses various issues associated with the implementation of stereoscopic display technology for network visualisation, including the choice of input devices for effective and dynamic interaction with 3-D displays. Though the project itself has a larger scope of research, the present report focuses on implementation issues, and proposes an evaluation scheme that can be used for determining how effective a particular implementation is. Section 2 discusses general issues of network visualisation; Section 3 reviews the psychological factors involved in depth perception; Section 4 explains the implementation of stereoscopic displays and summarises our experiences; and Section 5 gives an overview of the issue of choosing input devices, along with our experience with three input devices: a conventional 2-D mouse, SpaceBall,™ and Bird.™ Section 6 describes the the software package NTREE, which was used as a vehicle for exploration of the implementation space. In Section 7, the design of an experiment is proposed

---

*Department of Industrial Engineering, University of Toronto, 4 Taddle Creek Road, Toronto, Ontario M5S 1A4. Part of this report is published in Xiao and Milgram, 1992

†IBM Canada Laboratory, 3G/611/1150/Tor, 1150 Eglinton Ave. E.,North York, Ontario, M3C 1W3.

for evaluation of a network visualisation environment.

2 Network Visualisation

Presenting the relationships among objects within a database in diagrammatic form has a strong, intuitive appeal, as inferred by the saying "a diagram is worth ten thousand words" (cf. Larkin & Simon, 1987). Graphical representations make use of human perceptual capabilities, and tend to be intuitive and easy to manipulate, there by increasing our ability to deal with abstract networks. With the availability of high-quality computer graphics, the appeal of graphical representation has manifested itself in a flourishing of visualisation tools. In particular, a large number of tools have been developed in the context of software visualisation. These tools include visual programming (*e.g.,* Najork & Kaplan, 1991; Holland, 1991; Consens et al., 1992), code analysis (*e.g.,* Hirose et al., 1990; Koike, 1992; Calliss, Foley, & Ismail, 1991), and accessing and managing large networks (*e.g.,* Robertson et al., 1991; Fairchild et al., 1988; Brooks, Ohu-Young, Batter, & Kilpatrick, 1990).

However, the use of computer graphics alone does not guarantee better and more successful tools. As stated by Fairchild et al. (1988): "...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.

The aims of our research project are (a) to explore issues in enhancing the 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 actually help and improve task performance? (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 enable interaction with networks in virtual 3-D space?

2.1 Essence of Network Visualisation

In order to implement a visualisation tool, two stages are necessary: to create a virtual space of abstract objects and relationships, and then to display that space visually.[1] In this way we are able to exploit one or more human visual perceptual capabilities. To make the tool even more effective, it is important to make use of the ways in which we ordinarily interact with objects in the real world.

Part of the first stage involves mapping a network onto a virtual space (in 2-D or 3-D) by assigning each network node a coordinate. This process is known as *network placement.*[2] Usually the network structure specifies only linkages among nodes; therefore there can be many degrees of freedom in the mapping or placement process, as illustrated in Figures 1 and 2. This freedom makes it possible to group nodes in many ways, and to form certain familiar geometrical patterns where possible. For example, nodes at the same level of a hierarchy can be lined up, or combined to form the base of a cone, as shown in Figure 3. The virtual space need not be homogeneously mapped—a fish-eye view can be used, for example. Although the present project does not deal with the creation of such virtual spaces, there is a strong relationship between this stage and the second stage for presenting 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 one of the central questions that the project is attempting to answer.

One other important factor in network visualisation

---

[1] The effectiveness of visualisation, however, has not been sufficiently investigated experimentally (see Larkin & Simon, 1987; Brooks, 1987).

[2] In Fairchild et al. (1988), the term "positioning knowledge elements" is used for network placement.

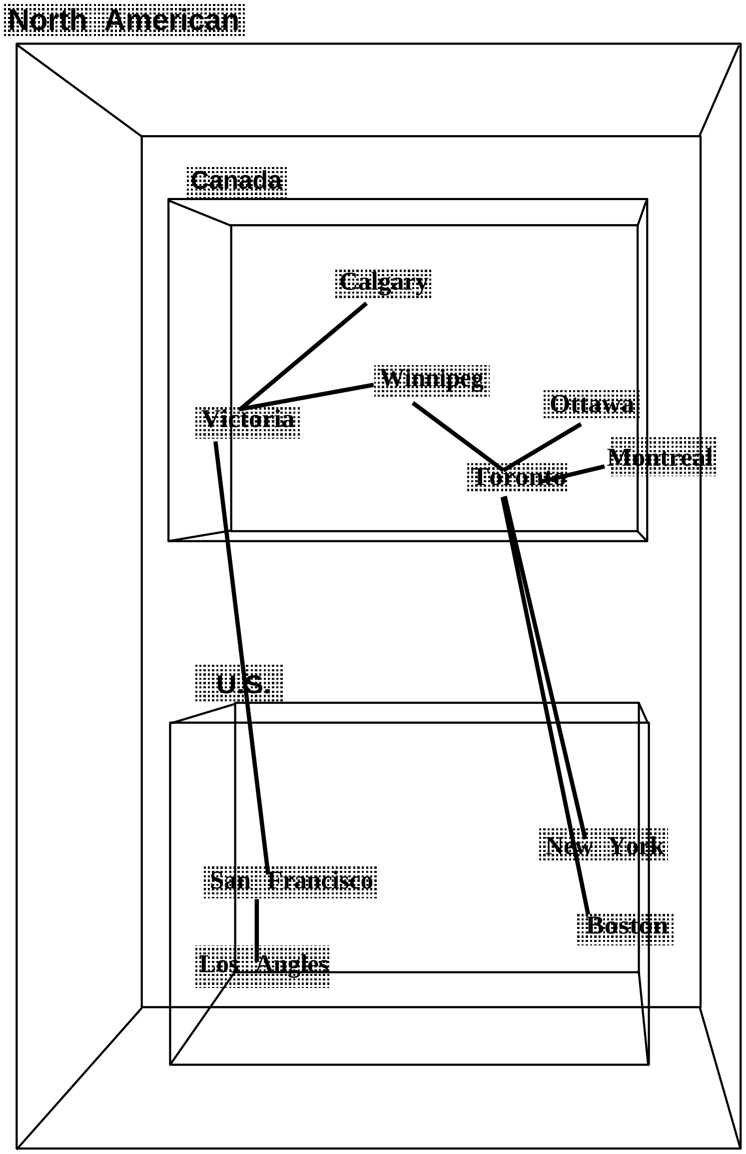


Figure 1: Displaying hygraphs using boxes. The database specifies linkage relationships between objects. To display the relationships graphically, a spatial location has to be defined for each object, and a geometrical identity has to be found to represent each relationship. There is no general way to map a database object into a spatial object. This hygraph (a graph with nodes containing subgraphs) shows one of many ways to map a database of cities and connections into a 3-D space.

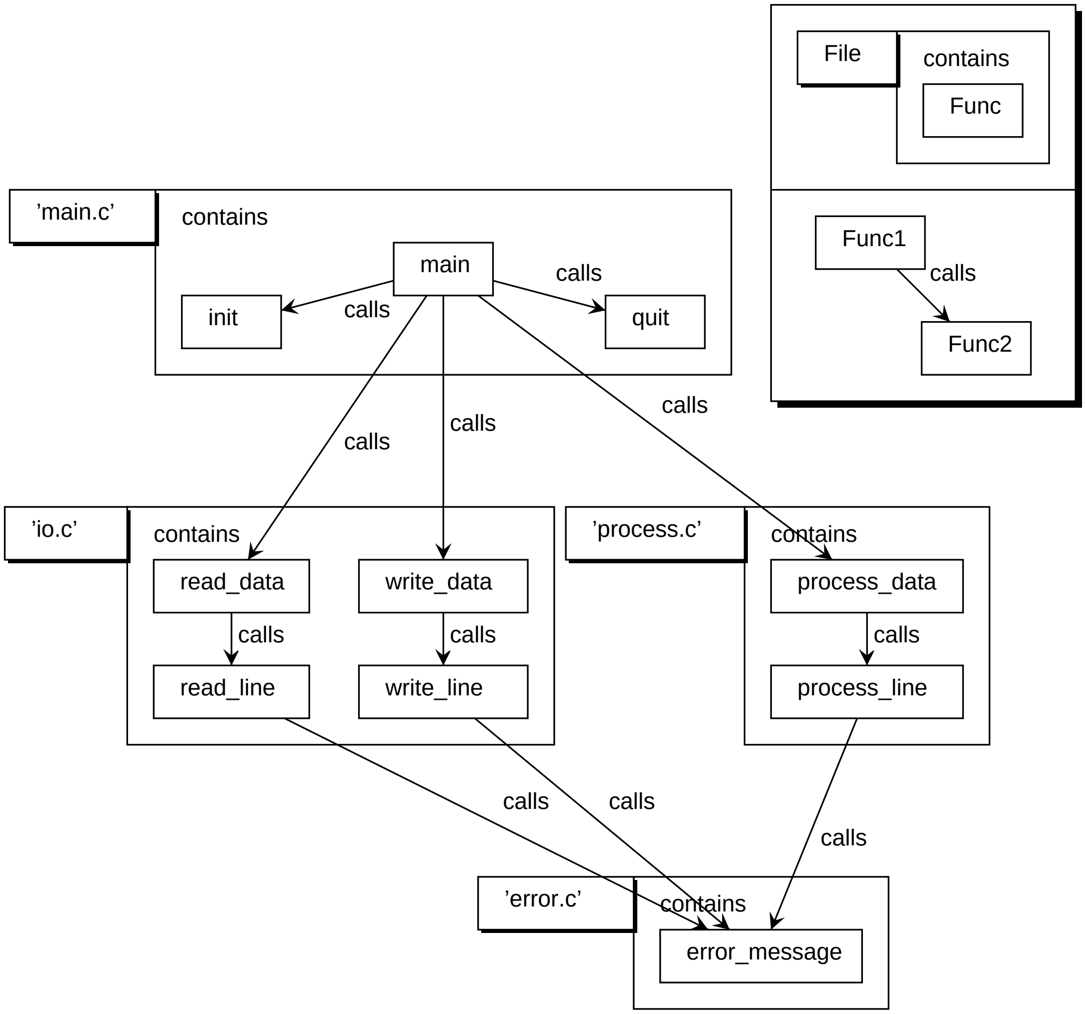


Figure 2: An example from a 4Thought display. Note that lines and boxes are used to represent different relationships. Boxes are used to show containing relationships (a file containing functions) and arrowed lines to show calling relationships.

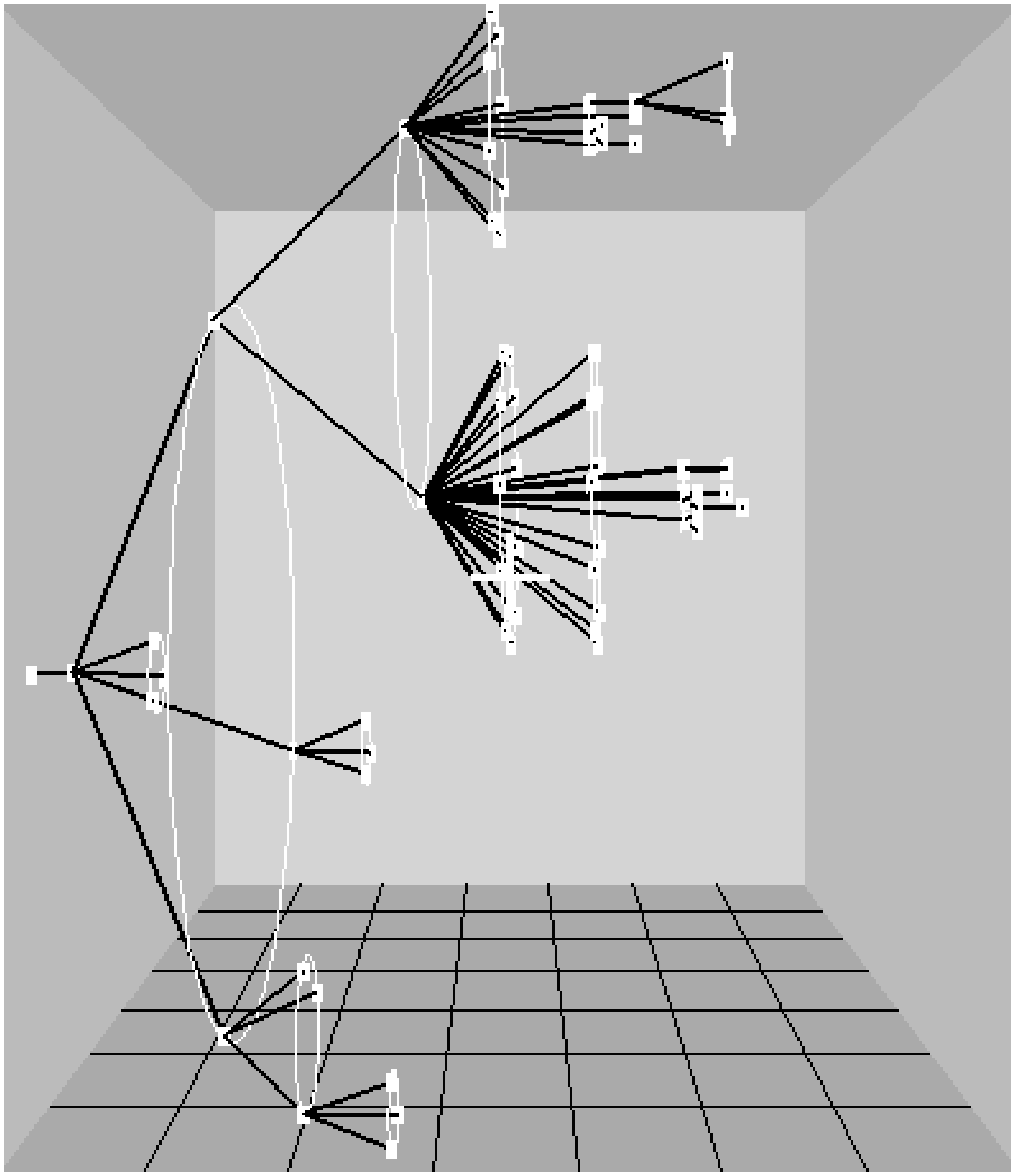


Figure 3: Displaying graphs using hierarchical cones (from an NTREE display). Imagery circles are added to enhance the perception of cones. Nodes that have the common parent node form the base of a cone, whereas the parent node is the top of a cone.

that is included within the scope of this study is the mode of user interaction: that is, the way in which users can interactively control the presentation of networks and manipulate objects within these networks.

2.2 Pros and Cons of Network Representation 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 (Drascic, 1991; Sollenberger & Milgram, 1989). Nevertheless, the utility of 3-D displays for specialised applications, in particular for visualising complex 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 representation

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 arc crossing problem in 2-D space. One important aspect of visual complexity is the extent of arc intersections. For two-dimensional placement, this will clearly pose a serious problem. However, it is important to realise that any network of nodes and arcs can be embedded within a three-dimensional space with no intersecting arcs. 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 (Fairchild et al., 1988).

Second, a 3-D network representation gives more freedom in laying out the network to show relationships and structures that could not otherwise be represented by the network.[3]

Third, a 3-D representation affords operations such as changing viewing angles, "travelling" through the network (Fairchild et al., 1988), 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 (Robertson et al., 1991). Recent research on the kinetic depth effect has demonstrated the important benefits to be gained from 3-D object rotation (Sollenberger & Milgram, in press; Sollenberger, 1993).

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; graphic objects are smaller when more distant from the observer.

2.2.2 Disadvantages of 3-D network representation

There are, according to Wickens (1990), 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 perception of 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 due to display clutter. (3) There are more design issues to be tackled, such as the network placement, which will certainly be more difficult.

In addition to Wickens' points, it must be borne in mind that certain 3-D visualisation techniques, such as object rotation and stereoscopic viewing, have associated with them certain costs, typically in terms of additional ancillary hardware and necessary minimal computing power. These are reviewed in Sections 3 and 4.

---

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

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 any inherent dimensionality. Given a placement of an arbitrary network in 3-D space, a simple projection of it onto the display monitor may still contain very little information on depth. Often lines are used to represent the linkage relationships,[4] and therefore the network displayed does not usually have any volume objects, whereas most existing 3-D display techniques employ volume and segregated objects. However, in the placement process, volume objects within a network can in fact be created in some circumstances. For example, cone trees can be used to represent a hierarchical, tree-like structure (Figure 3).

From a task point of view, network visualisation is different from other traditional 3-D applications (such as aviation displays), where speed, time, and path are important factors. Related to the dimensionality issues, specific spatial location is not always a central concern to a user in network visualisation, because the particular spatial location is simply the product of the placement algorithm, such that any particular element in a network could theoretically be placed anywhere within the 2-D or 3-D display space.

2.3 Summary

This section has reviewed certain task-related aspects of network representation. Of the two major stages, the creation of a virtual space and the presentation of the space visually, only the latter has been examined in the present project. From a practical task point of view, the questions to be examined are: (1) In what way do various 3-D display technologies aid users in visualising and manipulating network structures? (2) Will navigation be harder or easier with 3-D displays? (3) Will object manipulation be harder or easier? 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 the particular case of visualising complex network structures.

3 Human Depth Perception

Human's ability of perceiving objects in depth is somewhat mysterious, considering that all 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 Seidler's report (1989) give a comprehensive review of the topic, but that review emphasises applications with respect to real-world images such as those found in aviation (Merwin & Wickens, 1991). Another very useful recent review of this topic can be found in Sollenberger (1993).

The following review is combined with a discussion of usage of these cues in the more synthetic (*i.e.*, virtual) world—network visualisation. It is conducted from both a psychological and a technical (or implementational) viewpoint. The focus is on integrated cues; that is, multiple cues that are combined within 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, such methods use 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 orthogonally projected images. Such images allow greater precision in judgment in the projected plane, and thus also have a role in network visualisation; however, these methods may require an exceptionally high level of training and may also necessitate unacceptably high levels of

---

[4] Johnson and Shneiderman (1991) used a Venn-diagram-like scheme to display hierarchical networks. This method does not use lines and points to represent network information. See also Figure 2 which uses boxes to represent containing relationships.

mental efforts.

For the purpose of the present project, the depth cues discussed are divided into two categories, according to whether the display is static or dynamic.

3.1 Static Depth Cues

Shadows. In the real world, light projecting on objects causes shadows, which in turn provides 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, in terms of network visualisation, the realisable number of nodes displayed on the screen would have to be reduced. Shadows have been demonstrated to be useful in the Cone Tree in the Information Visualizer (Robertson et al., 1991). An auxiliary plane has to be used for casting shadows, however.

Linear perspective. Because parallel lines appear to converge at large distances, far away objects appear smaller. This perspective cue is very useful in displaying natural scenes, but is not as useful in the network environment because of the lack of familiar geometrical forms such as symmetry.[5] Nonetheless, the linear perspective is analogous in many ways to the fish-eye-view effect, since both increase display 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 difference in luminance and saturation provides implied depth information. These are not strong cues, but they 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, however, acquiring familiarity with sizes of objects will be difficult, 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 (Mazur & Reising, 1990), in displaying network structures 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 their absolute depth.

Texture. This cue is associated only with surfaces, and can be used to show their position and orientation. Such surfaces are not typically present in network representations, however.

Binocular depth. See Figure 4. This cue is one of the most salient, as noted by Wickens et al. (1989, p. 100): "Stereoscopic displays ... provide binocular depth cues which significantly enhance performance, particularly when other visual enhancement cues are not presented." Under natural conditions, the left and right eyes of an observer are located in positions that are about 65mm 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

Motion parallax and object rotation. These cues are also called "kinetic depth effect", or "structure through motion" (Wallach & O'Connell, 1953; Ullman, 1979), and are present when objects in different depths and the viewer move relatively to each other. For example, when in a moving car, objects near the car pass by faster than those further away. Motion parallax and object rotation are very strong cues when dynamic displays are

---

[5] 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.

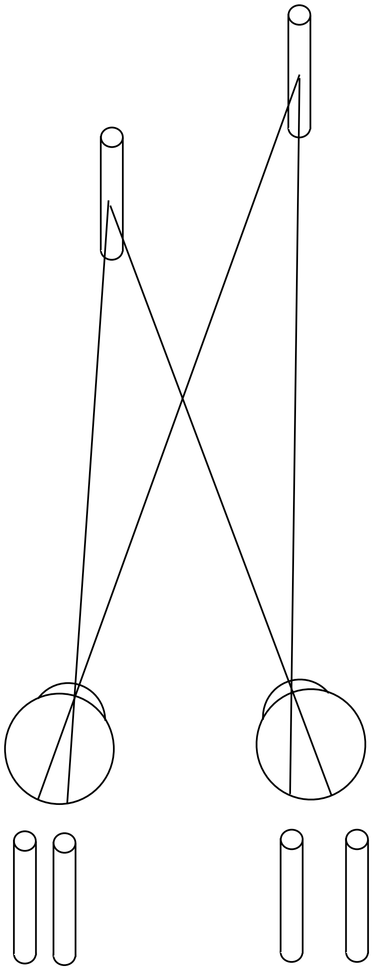


Figure 4: Binocular Disparity. Two rods are imaged differently on the retinas of the eyes. One eye sees the two rods closer than the other does. Binocular disparity is the difference in distance between the rods seen by left and right retinas.

available (Sollenberger, 1993; Sollenberger & Milgram, in press). The drawback of using them, however, is that once the display is static the effect disappears. In addition, successive screen updates have to be produced smoothly 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 images and changes in the head position. Because active parallax requires close coordination between head movement and displayed images, and, therefore, head movement monitoring and rapid display updating, complex networks are usually beyond the possible application of this cue.

Preservation. When an object moves in depth, the viewer makes the assumption that the object is rigid, and thus that attributes of the object remain 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 the preceding review, it is clear that the following two cues are potentially the most useful cues for displaying networks: (1) binocular disparity, and (2) motion parallax and object rotation. In addition, limited use can potentially be made of: (1) linear perspective, (2) proximity-luminance and proximity-saturation covariance, and (3) shadows.

A consistent finding in 3-D perception research is that the greater the number of cues provided, the better the depth perception (Mazur & Reising, 1990; Wickens, 1990; Sollenberger, 1993; Sollenberger & Milgram, in press; Merwin & Wickens, 1991). However, there is usually a cost associated with the display of each cue. Choosing a "good" combination requires both basic psychological research, and empirical testing in the 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 relevant to network visualisation, binocular disparity is recognised as the strongest. In contrast to dynamic cues, binocular disparity 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 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 for the foreseeable future. Will the benefits of the binocular depth cue offset these drawbacks? This is a question that has to be answered empirically.

One important aspect of weighing the advantages of various depth cues is to consider how they *complement* each other. It is well known, for example, that a serious drawback of rotation displays is *reflection ambiguity*, whereby the front and rear surfaces of an object appear to "flip" back and forth. One of the problems associated with binocular disparity, on the other hand, is the potential for perceived distortion in depth scaling, due to geometrical inconsistencies in the degree of disparity being employed. A significant advantage of using *both* these cues simultaneously, however, is that each tends to cancel out the associated problem of the other (Sollenberger, 1993).

When using multiple cues simultaneously, there is a danger that different cues could 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 stereoscopic depth cue—binocular disparity—will be the preferred primary cue to induce depth perception for complex 3-D network visualisation. Although motion cues are also very salient, their effective implementation is limited by the need for powerful graphics hardware. Colour cues (luminance- and saturation-proximity covariance) are also potentially 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) how important is depth perception? (2) which cues are most beneficial for it? (3) will the advantage of providing various cues offset the associated cost?

4 Stereoscopic Display Implementation

As stated above, providing binocular disparity, or stereopsis, is probably the most promising approach to enhancing 3-D network visualisation. In this section, various means of providing this depth cue on computer displays are described, and associated issues are discussed.

The general approach is to provide each eye with a slightly (horizontally) shifted (and rotated if non-orthographic projection is used) image of an object (see Figure 5). There are, according to one taxonomy, two methods of providing stereopsis: *time parallel* and *time multiplexed* methods (Hodges & McAllister, 1987); that is, whether images are viewed by two eyes simultaneously or alternately. Faris (1992) provided another taxonomy according to how the separation of left and right images is achieved (see Figure 6).

4.1 Time Parallel Method

The anaglyph method is probably the simplest method of showing two images at the same time. Left and right images are shown in two different colours (usually red and green), and coloured lenses are worn so that each eye only sees one image. An obvious limitation of that method is that only one colour can be used. Another disadvantage is that this method has been reported to be visually fatiguing. Random dot stereogramme is another simple way to achieve stereopsis (Julesz, 1971), although no practical application of it has not been reported.

Using two optical paths to feed images from two computer screens or video projectors is a powerful (but expensive) method of generating stereoscopic displays. This method is often implemented by a helmet-mounted display (Sutherland, 1968). For larger audiences, orthogonal polarising filters placed over both the image sources and the viewer's eyes comprise a common solution to the problem.

4.2 Time Multiplexed Method

Using one display monitor, but displaying two images (left and right) alternatively to each eye, is by far the easiest way to achieve stereopsis (see Figure 7). The viewer looks through a liquid crystal shutter system that is synchronised with the display. To eliminate perceived flickering due to the alternation, a frequency of more than 40 Hz for each eye (total of 80 Hz) should be used (Turner, 1964).[6]

Two techniques can be used to implement this. By putting a shutter system that alternatively polarises light by rotating the plane of polarisation 90° back and forth between cycles in front of the display monitor, a simple polarised lens over each eye will ensure that each image is viewed only by the appropriate eye. This system (sometimes referred as *passive*) is considerably more expensive than the following technique, but it has the advantage of glasses that are lighter, simpler, and much less expensive (see Figure 8). A large number of viewers can be accommodated using this passive system.

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 signals (see Figure 9).

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

4.3 Generation of Alternating Images

4.3.1 Screen mapping methods (120 Hz)

The most popular way of generating alternating images is by the screen mapping method (see

---

[6] *Critical flickering frequency*, or CFF, is the frequency below which flickering can be perceived. CFF depends on a few attributes of the display, one of which is the ambient luminance. In a darkened room, CFF can be as low as 33 Hz.

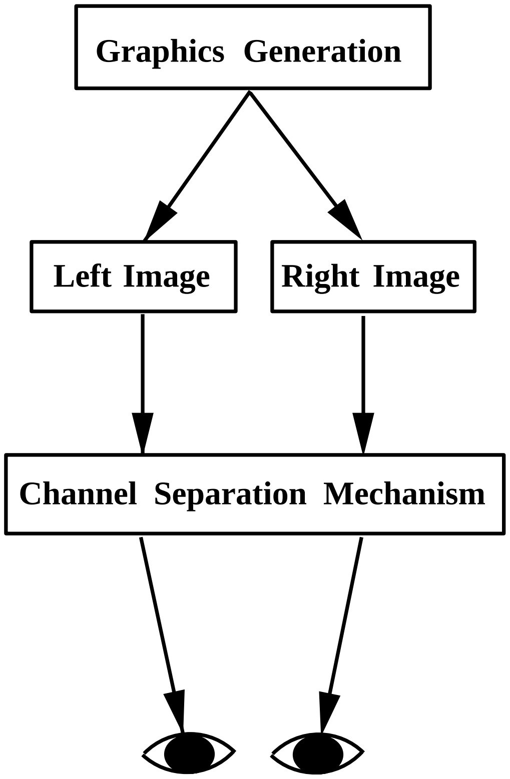


Figure 5: General scheme of stereoscopic display. Left and right images are generated and shown to each eyes through a separation mechanism. Depth information is constructed in the viewer's head by fusing the two images.

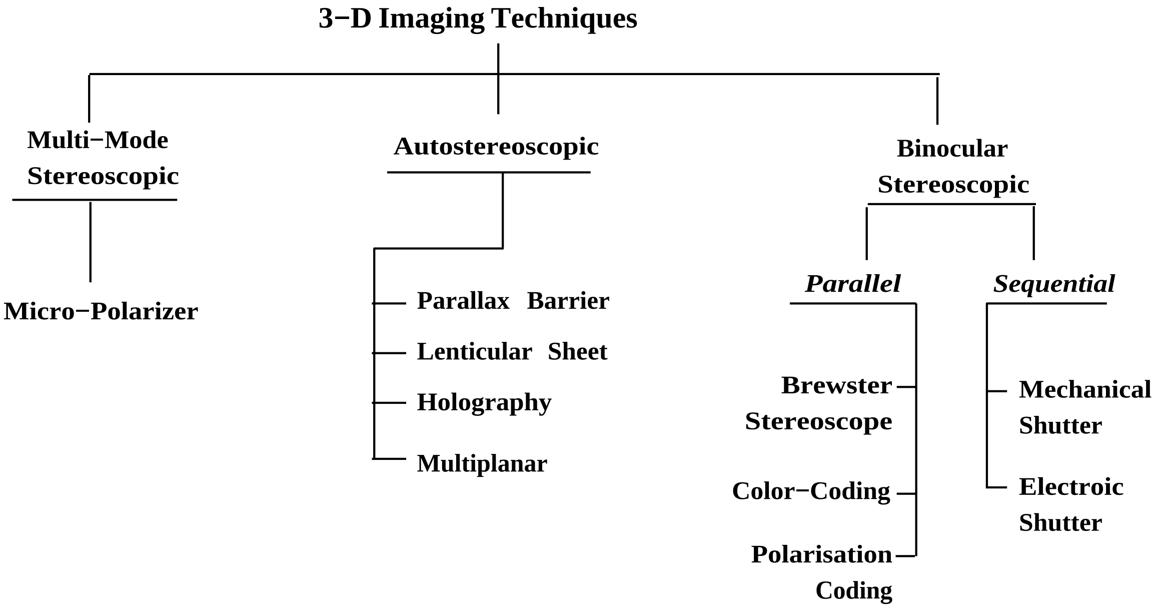


Figure 6: Stereo imaging techniques (from Faris, 1992)

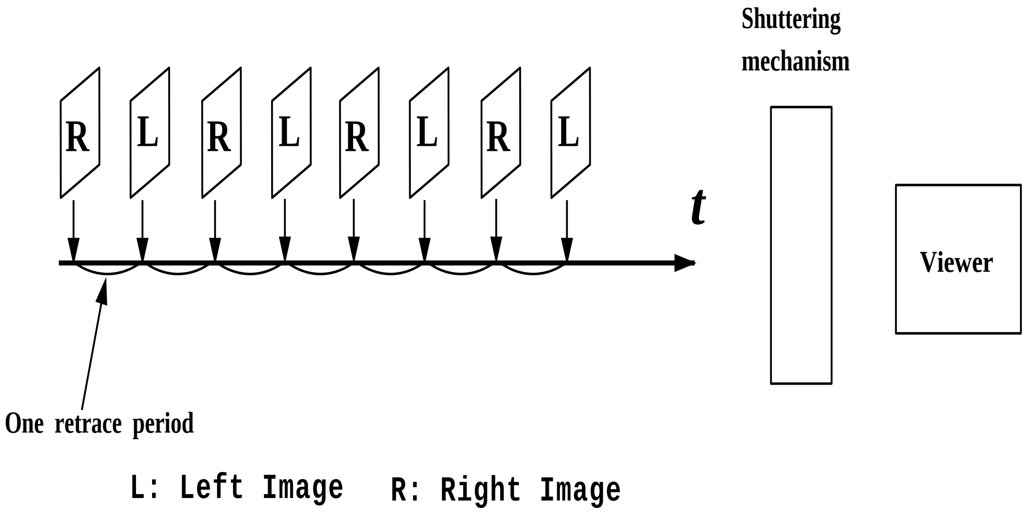


Figure 7: Time multiplexed method. Left and right images are shown in sequence. A shuttering mechanism occludes one eye at a time, thus each eye only seeing its corresponding image. The alternating frequency of the two images is usually the same as the monitor's refresh rate.

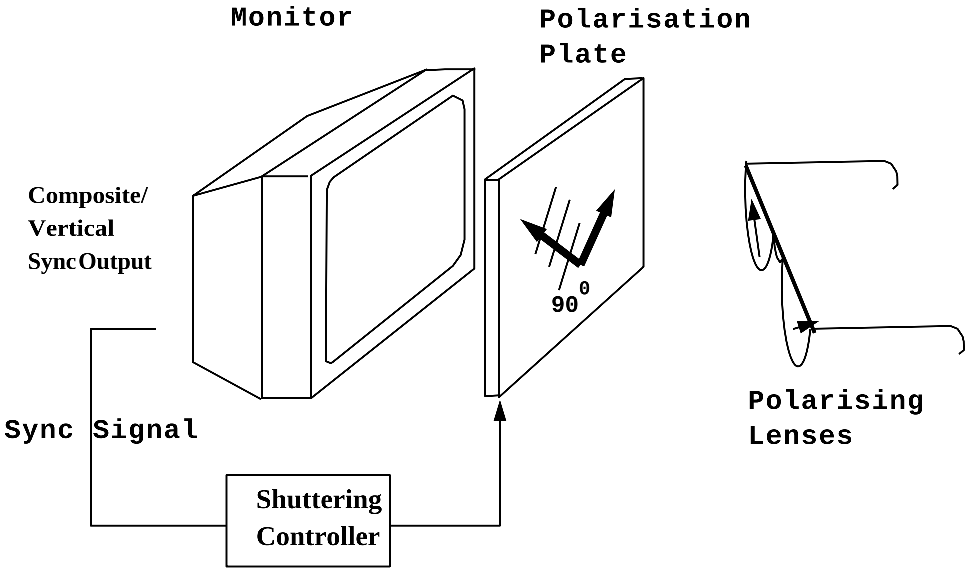
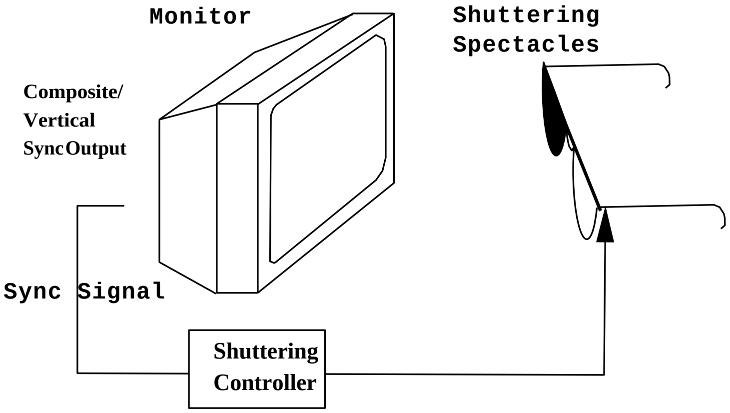
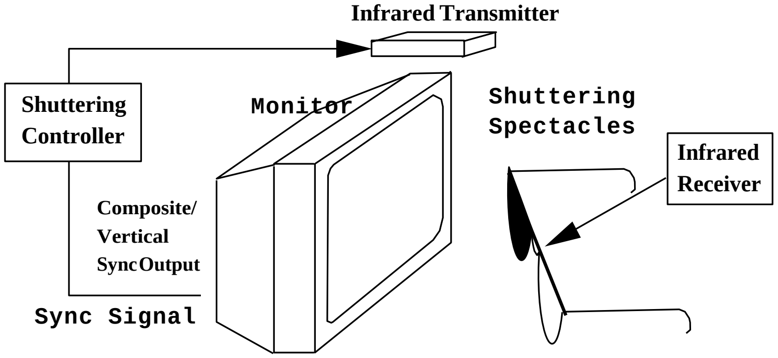


Figure 8: Passive display: using polarising plate. The screen image is polarised in two directions alternatively in high frequency. Through synchronisation with the display monitor, left and right images are polarised in one of the two directions. The viewer wears simple polarising lenses so that each eye sees one image.



(A) Active system with wires



(B) Wireless infrared active system

Figure 9: Active display: using active goggles. Liquid crystal shuttering goggles are controlled by the synchronisation signal from the display monitor. The two lenses are occluded alternatively. Some wireless systems make two lenses clear at the same time if the viewer looks away from the monitor, so that the viewer does not have to remove the goggles when not looking at the monitor.

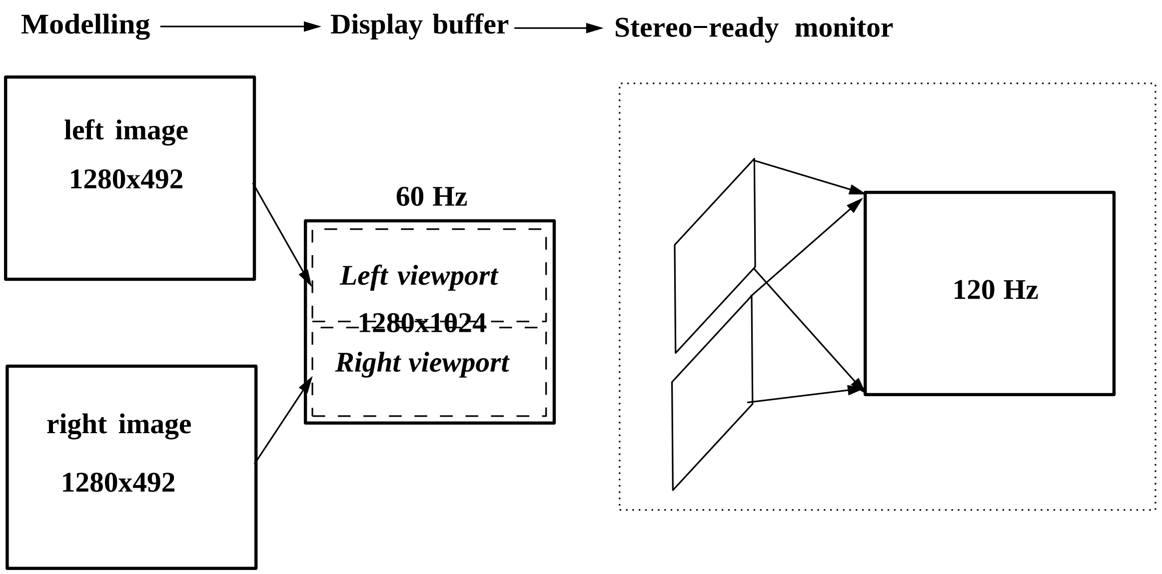


Figure 10: Screen mapping (120Hz) system. The monitor hardware maps alternatively the top and buttom portions of the display buffer onto the whole screen. The screen is refreshed at 120 Hz, with each of the left and right images refreshed 60 times per second.

Figure 10). With this method, left and right images are drawn in the top and buttom half portions of a single display buffer. By doubling the display frequency of the monitor (120 Hz), the top and bottom halves of each display buffer (which is refreshed at 60 Hz) are mapped alternatively to the display screen. Therefore, there is no synchronisation problem associated with the non-mapping methods (described below). The biggest drawback of this method is the difficulty for such displays to coexist with other regular (*i.e.*, non-screen-mapping) window applications (all known window managers are currently regular in this sense). Loss of half of the vertical resolution in the mapping process is another severe shortcoming with present technology. Needless to say, extra hardware is needed to accomplish the mapping (workstations with this hardware are called *stereo-ready*).

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

4.3.2 Colour map cycling methods (60 Hz)

Mechanism. Refer to Figure 11. 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, the left image and right image can be displayed in turn by making one of two images invisible.

Pros and Cons. One obvious disadvantage of this technique is the perceivable flickering associated with alternating images at 60 Hz. In addition, under multiple window conditions, using the colour map swapping technique will likely interfere with other windows. Also, the number of colours that can be used is quite limited (maximum 64 on a machine with a 24 bitplane graphic board). Despite these disadvantages, colour swapping has the advantage that left and right images are alternately

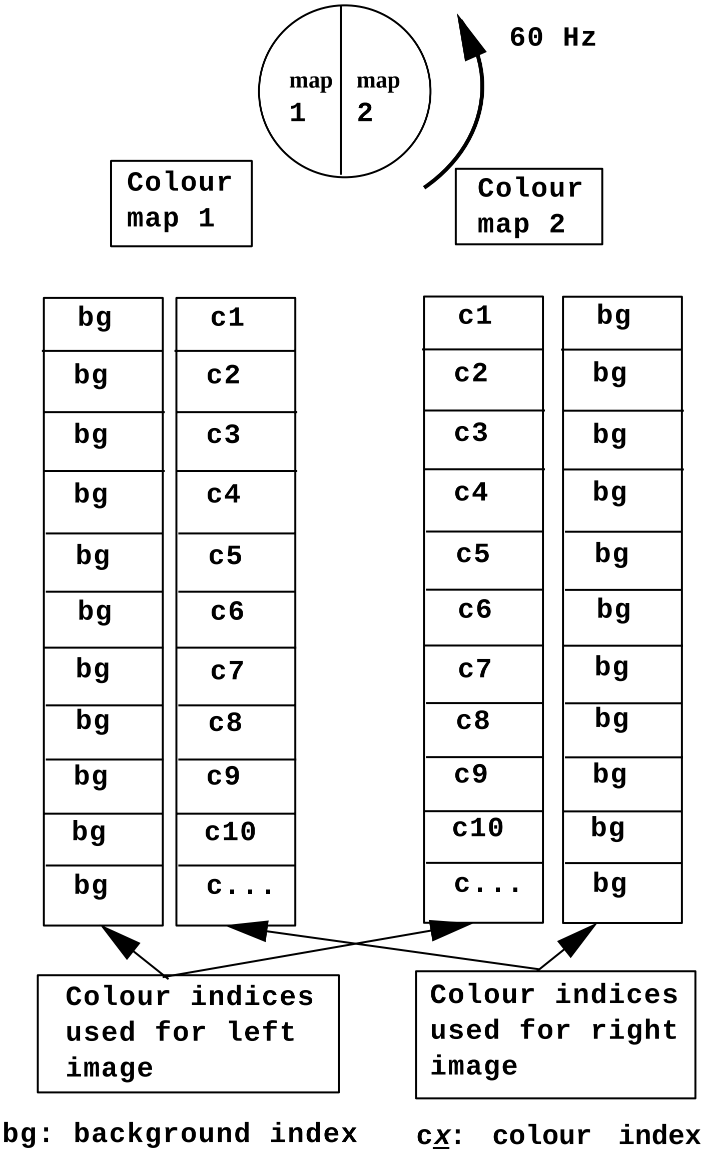


Figure 11: Colour map cycling method. Left and right image are shown on the screen simultaneously, but at any time the colour indices of one image are mapped to the same colour as the background colour. Thus that image is invisible. Two colour maps are used alternatively so that the left and right image are made visible in turn.

shown at a fixed rate,[7] regardless of the complexity of the image to be drawn and thus regardless of the actual image update rate.

4.3.3 Buffer swapping (60 Hz)

Mechanism. A somewhat more straightforward method of showing left and right images alternately is to use two graphic buffers and buffer swapping. One buffer holds each of the left and right images, and the two buffers are swapped in high frequency (usually the same as the monitor frequency). See Figure 12.

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 1/60 second), a swap will not take place until one or more cycles later, and, therefore, the left image might be displayed to the right eye due to the missed cycles. In this case, a reversal of left-right correspondence occurs. With the complicated drawings used for large networks, this loss of consistent left-right correspondence is unavoidable. Even simple drawings in a multiple window system can 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 introduced as the missed cycles are detected *a posteriori*, and a correction is done by inserting extra waiting cycles to bring the correspondence back to normal. This solution is shown in Figure 13. 14 shows the pseudo-code for the solution.

The flickering need 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 reducing drawing complexity when a rotation or translation is being done. This can be achieved either by reverting to monoscopic display mode or by drawing a degenerated network.[8] 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.

4.4 Summary

As stated above, there are a number of ways to provide 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 (120 Hz) is used, (2) the effects of flickering problems when non-screen mapping methods (60 Hz) are used, (3) the coexistence of stereo windows and non-stereo windows in a multiwindow environment, (4) user acceptance of 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 these advances comes the need for devices to allow the users to interact effectively with a 3-D space. Especially with 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 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 these metaphors represents a class of

---

[7] In our experience so far with both SGI Iris and IBM RS 6000 machines, although the rate is fixed, for unknown reasons, the correspondence of left-right is occasionally reversed. This is very difficult to detect and correct using software.

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

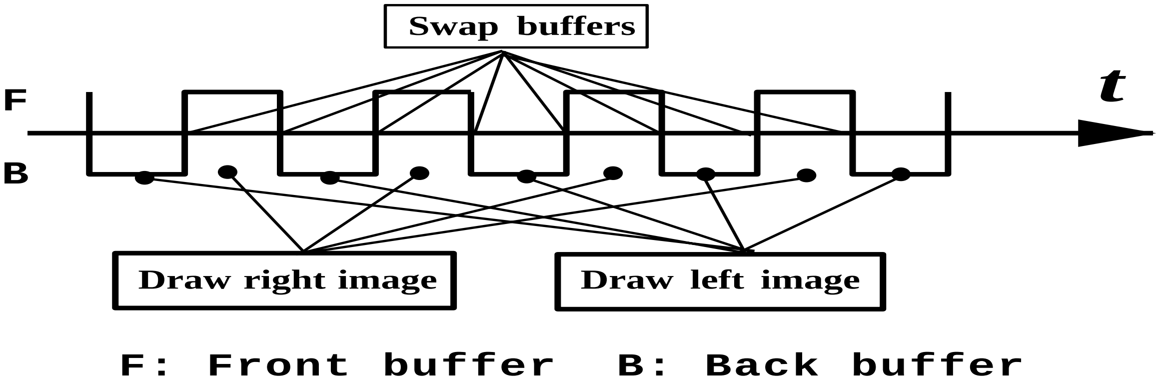


Figure 12: Buffer swapping method. Using two display buffers, each hold one of the left and right image, this method swaps buffers at high frequency. Usually double buffer mode is used so that while one buffer is displayed, the other buffer is updated.

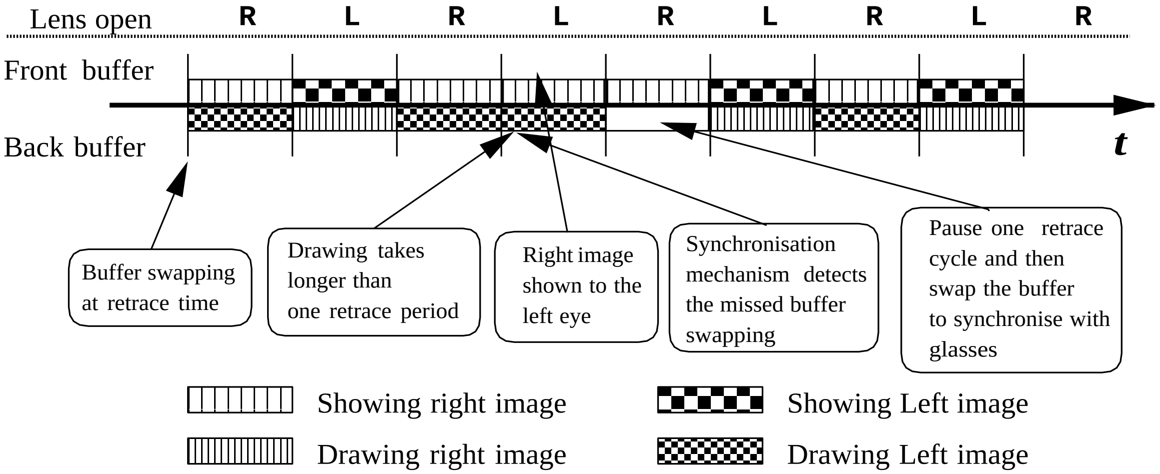


Figure 13: Synchronisation technique used in the swapping buffer method. Missed buffer swapping can cause the correspondence problem: the left eye sees the right image and the right eye sees the left image. By using accurate timing, missed buffer swappings can be detected and corrected.

Code for drawing left and right images:

```
draw_two_images():
begin:
     Draw_left_image()
     swap_buffers()
     num_elapse = calculate_elapse(clock_time)
     clock_time = get_time()
     Draw_right_image()
     swap_buffers()
     num_elapse += calculate_elapse(clock_time)
     clock_time = get_time()

     sync_buffer ( num_elapse )
end
```

where `calculate_elapse()` is to calculate the number of retraces that has elapsed since last saved `clock_time`; and `clock_time` is the machine CPU clock in microseconds.

Code for `sync_buffer()`:

```
sync_buffer(num_elapse):
begin:
     if (num_elapse is an odd number )
     do
          pause_a_retrace()
          swapbuffer()
          num_elapse = calculate_elapse(clock_time)
          clock_time = get_time()
          sync_buffer( num_elapse+1 )
     done
end
```

Figure 14: Pseudo-code for the synchronisation technique. Correspondence problem exists when the number of missed buffer swappings after having drawn both left and right images is odd. When that happens, `sync_buffer()` pauses for a retrace period to cause another missed buffer swapping, thus reverting the correspondence back to normal. Recursive call in `sunc_buffer()` is necessary since every `swapbuffer()` call has the risk of being missed by the graphics hardware and therefore needs to be checked.

spatial interactions that we have with our surroundings:

1. *World-in-hand metaphor*—Objects in the virtual world are rotated and translated as a whole, as if the entire world is being manipulated. Views from different angles are thus displayed.
2. *Object-in-hand metaphor*—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 defined 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 (Ware, 1991).
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, it is conceivable that combinations of these strategies could be used to carry out a given task.

5.2 Taxonomy of Three-Dimensional Input Devices

A rigid object in 3-D has six degrees of freedom (DOF). To completely control the position and orientation of an object in 3-D space requires an equal number of DOF 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:

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

5.2.1 Homologous 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 Heterologous Controllers

This category of controllers refers to controlling mechanisms that use 2 or less DOF 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 (Fairchild et al., 1988).

- 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.*, FastTrap^{TM}). A typical use of this method is to add depth input by using extra buttons or dials. An illustration of the key combination method is given in Figure 16, which is used in the NTREE package described below.

5.3 Choices of Input Devices

As mentioned earlier, different tasks require the use of different input strategies. For example, shooting a film is better done using the strategy of the flight-metaphor (Ware & Jessome, 1988; Ware, 1990). The metaphors listed above encompass a wide range of tasks involved in interactions with 3-D space, and some 3-D input devices can be implemented to realise more than one metaphor (*e.g.*, Zhai & Milgram, 1993). 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 lists a series of hypotheses as to which controllers can be used, 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 an input device. Given a particular input device, there are still a number of other parameters or variables that govern the profile of that device. Of these, control mode is of particularly interest: that is, whether velocity control or position control is used (Ware, 1991; Zhai & Milgram, 1993).

6 NTREE: A Stereoscopic Network Visualisation Package

The above mentioned factors were explored in NTREE, a software package that displays network information stereoscopically and allows manipulation of the network with 6 DOF Figure 15 shows the overall structure of the package.

Below are a few notes about the NTREE package:

- Hardware: NTREE has been tested in several models of SGI IRIS, including Crimson, 4D310/GTX, and 4D70, all equipped with Z-buffer and 24-bit graphics option, as well as IBM RS 6000 530 with GL, 24-bit graphics, and Z-buffer options. Stereo-ready hardware (monitor and graphics board) is necessary to use the 120 Hz screen-mapping method.
- Develop environment: SGI IRIX 4.0.5 C compiler, and IBM AIX 3.2 C compiler (with GL libraries).
- Two methods of multiplexing were implemented: (a) 60 Hz buffer swapping (see Figure 12), and (b) 120 Hz screen mapping (see Figure 10).
- Testing features include showing the effects of (a) background shadow, (b) labelling shadow, (c) floor grid, (d) incremental drawing of degenerated network during dynamic changes, (e) velocity control, and (f) stereo cursor display.
- A 3-button 2-D mouse was used to act as a 6 DOF input device, as shown in Figure 16.

7 Proposed Design of Experiment

The basic goals of the present research are to explore issues associated with providing 3-D displays and 6 DOF manipulation tools with various control and display implementation options.

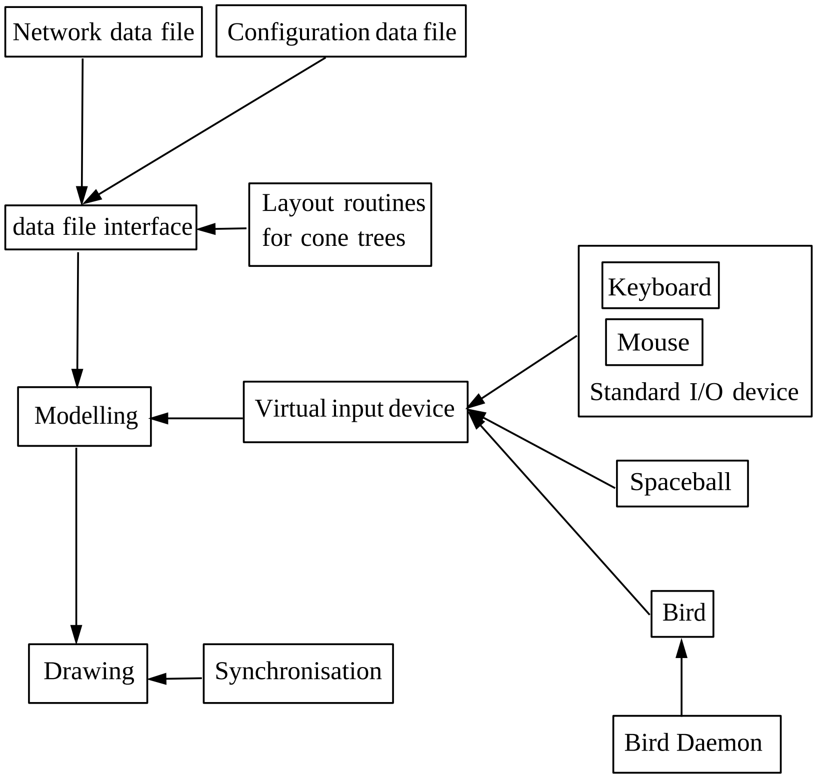


Figure 15: NTREE structure. NTREE is a test bed for various implementation of stereographically displaying networks with various input devices.

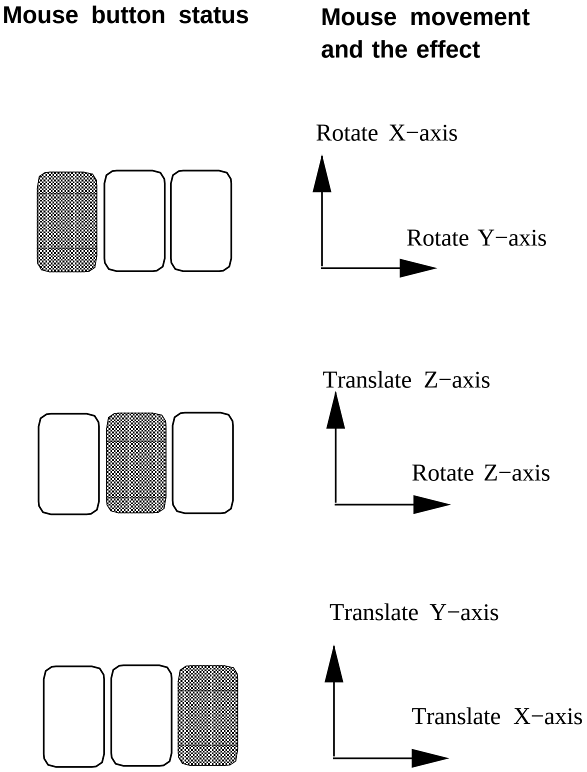


Figure 16: 2-D mouse used as a 6-D virtual controller. Shadowed buttons indicate the pressing down status.

Metaphor	Homologous Controller	Heterologous 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 of suitability of various input metaphors

7.1 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 representative tasks in the real target world. The first method gives the flexibility of investigating each factor individually or in a desired combination, but it requires proper mapping of the toy world onto the real world. Although the second method does not have the final mapping problem, it does not have the same flexibility for experimental manipulation.

7.2 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 representative real tasks are used to test how well a particular implementation serves in the target world: helping software designers. With respect to 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.

Proposed 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 subtree is intersecting any other subtrees.

Proposed 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, *i.e.*, minimise the number of arcs that are participating in calling or referring loops.

Proposed 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 re-arrange the program so that it fits into a given size of computer memory by using a code overlay structure.

The first task is intended to test both the display options and 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 (Consens et al., 1992). 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. Tasks 2 and 3 are chosen to investigate how a particular implementation of a 3-D visualisation system helps users perform design tasks.

7.3 Experimental Variables

Independent Variables. Three sets of independent variables are proposed for the experiment:

- Input device. The choice of input device (homologous versus heterologous controller) and user control mode (different metaphors, and velocity versus position control).
- Display. The mode of display (2-D versus 3-D, depth cue combination, and fish-eye 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 placing), task environment (whether or not users can choose the input device or display mode), and task difficulty (the number of arcs and nodes in a network, and the properties of the network structure).

Dependent Variables. The dependent variables proposed for 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.

8 Summary

Implementation and evaluation of a 3-D visualisation system involve technical, task-related, and psychological factors. In this report, all of these factors are considered and issues are raised for empirical testing and evaluation. For 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 complex 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 perception 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 a proposed experiment to evaluate various implementations are: input devices and interactive mode, display mode and quality, and task requirement.

Acknowledgements

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.

Yan Xiao was supported by an IBM Ph.D. Fellowship and worked at CAS for a term in 1992.

Reference

Brooks, F. (1987). No silver bullet: Essence and accidents of software engineering. *IEEE Computer*, 10–19.

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

Calliss, F. W., Foley, T. A., & Ismail, M. (1991). Inter-module code analysis using 3-dimensional interactive graphics. In *Proceedings of 25th Hawaii International Conference on System Sciences*, pp. 670–677.

Consens, M., Mendelzon, A., & Ryman, A. (1992). Visualizing and querying software structures. In *Proceedings of the 14th International Conference on Software Engineering.*

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

Fairchild, K., Poltrock, S., & Furnas, G. (1988). Semnet: Three-dimensional graphic representations of large knowledge bases. In Guindon, R. (Ed.), *Cognitive science and its applications for human-computer interaction*, pp. 201–233. Erlbaum, Hillsdale, NJ.

Faris, S. M. (1992). True stereo display and hard copy from one system. *Advanced imaging*, *7*(5), 18,20,22,80.

Hirose, M., & Amari, H. (1991). A study on visualization of control software design. In *MIT JSME Workshop on Computer Aided Cooperative Product Development.*

Hirose, M., Myoi, T., Amari, H., Inamura, K., & Stark, L. (1990). Development of visual 3d virtual environment for control software. In *NASA Conference on Human Machine Interfaces for Teleoperators and Virtual Environment.*

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

Holland, S. (1991). Two-dimensional visual programming and three-dimensional execution visualization in Prolog. In *IEE Colloquium on real world visualisation - virtual world - virtual reality (Digest No. 197).*

Johnson, B., & Shneiderman, B. (1991). Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of Visualization'91, October 22-25*, pp. 284–291. IEEE.

Julesz, B. (1971). *Foundations of cyclopean perception.* University of Chicago Press, Chicago.

Koike, H. (1992). Three-dimensional software visualization: A framework and its application. In Kunii, T. (Ed.), *Visual Computing: Integrating Computer Graphics with Computer Vision.* Springer-Verlag, Tokyo.

Larkin, J., & Simon, H. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, *11*, 65–99.

Mazur, K. M., & Reising, J. M. (1990). The relative effectiveness of three visual depth cues in a dynamic air situation display. In *Proceedings of Human Factors Society 34th Annual Meeting*, pp. 16–20.

Merwin, D. H., & Wickens, C. D. (1991). 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*, pp. 388–392.

Najork, M. A., & Kaplan, S. M. (1991). The CUBE language. In *Proceedings of IEEE Workshop on Visual Languages*, pp. 218–224.

Robertson, G. G., Mackinglay, J. D., & Card, S. K. (1991). Cone trees: Animated 3D visualizations of hierarchical information. *Proceedings of CHI'91*, 189–194.

Sollenberger, R. L., & Milgram, P. (1989). Stereoscopic computer graphics for neurosurgery. In *Proceedings of Third International Conference on Human-Computer Interaction*, pp. 294–301 Boston, MA. Elsevier Science Publishers.

Sollenberger, R. L. (1993). *Combining depth information: Theory and implications for* *design of 3D displays.* Doctoral dissertation, University of Toronto.

Sollenberger, R. L., & Milgram, P. (in press). Effects of stereoscopic and rotational displays in a 3D path-tracing task. *Human Factors.*

Sutherland, I. (1968). A head-mounted three dimensional display. *Proc. of Fall Joint Computer Conf.*, 757–764.

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

Ullman, S. (1979). *The interpretation of visual motion.* MIT, Cambridge, MA.

Wallach, H., & O'Connell, D. H. (1953). The kinetic depth effect. *Journal of Experimental Psychology*, *45*, 205–217.

Ware, C. (1990). Using hand position for virtual object placement. *The Visual Computer*, *6*, 245–253.

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

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

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

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

Xiao, Y., & Milgram, P. (1992). Visualisation of large network in 3-D space: Issues in implementation and experimental evaluation. In *Proceedings of 1992 CAS Conference*, Vol. 1.

Zhai, S., & Milgram, P. (1993). Human performance evaluation of manipulation schemes in virtual environment. In *Proceedings of IEEE Virtual Reality International Symposium* Seattle, Washington.

About the author

Yan Xiao is a PhD candidate in the Department of Industrial Engineering at the 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 the 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 particular for telerobotics.

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.

Arthur G. Ryman is a senior development analyst and the principal investigator of the Advanced Software Design Technology (ASDT) program at the IBM Canada Laboratory, Toronto, and a member of the IBM Academy of Technology.

He can be reached at IBM Canada Laboratory, 3G/611/1150/Tor, 1150 Eglinton Ave. E., North York, Ontario, M3C 1H3, and `ryman@vnet.ibm.com` on Internet.