STEREOSCOPIC HEAD-TRACKED DISPLAYS: ANALYSIS AND DEVELOPMENT OF DISPLAY ALGORITHMS

A Thesis
Presented to
The Academic Faculty
By
Zachary Justin Wartell

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Computer Science

Georgia Institute of Technology
August 2001

STEREOSCOPIC HEAD-TRACKED DISPLAYS: ANALYSIS AND DEVELOPMENT OF DISPLAY ALGORITHMS

Approved:

_____
Larry F. Hodges, Chairman

_____
Elizabeth Davis

_____
William Ribarsky

_____
Thad Starner

_____
Greg Turk

Date Approved_____

DEDICATION

I give thanks and love to:

- To my family
  - Mom, Dad and Arlena.
  - Grandma Wartell and Grandma Salo
  - In memory of Grandpa Wartell and Grandpa Salo

  Your love, guidance and support made this possible.
- To my teachers:
  - Dr. Larry F. Hodges, my thesis advisor
  - Dr. William Ribarsky, Dr. Elizabeth Davis, Dr. Greg Turk and Dr. Thad Starner, my thesis committee
  - Each author cited in this text
  - Mrs. Kerns, Mr. Morgan, Mr. Hampton, Mr. Faulkner, Ms. Counts, Mrs. Mecom, Mrs. Bidlack, Mrs. Swan, Mr. Wertzburger, Mrs. Druva, Mrs. Armor, Mrs. Swindel, Mrs. Tostrude and the many others at Druids Hills High School, Briarcliff High School, Ridgeview Junior High, Gaithersburg High, The Epstein School and Briar Vista.
- To the staff at the College of Computing and GVU Center
- To my friends:
  - William, Lloyd, Afif, Sarah and Anna – who saw me through the many phases of my student journey
  - All fellow students at the GVU Center and in the Virtual Environments Group at Georgia Tech especially Dave Krum, Frank Jiang, Don Allison, Jeff Wilson and Doug Bowman
  - John Griffin – for your wisdom
- To the sponsors who supported various aspects of this research and the research it builds upon:
  - The Nation Science Foundation
  - Office of Naval Research
  - U.S. Army Research Laboratory

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF NOMENCLATURE

Matrices are represented in bold italics such as: $\boldsymbol{M}, \boldsymbol{\alpha}$

Points and vectors are represented in bold such as: **A, B, C**

Scalars are represented in italics, usually lower-case such as: *a*, *b*, *c*

SUMMARY

Virtual reality has much potential and many challenges. We investigate geometric image distortion issues arising from human factors concerns in systems using stereoscopic head-tracked displays. These displays are stationary and attached to a desk, tabletop or wall. The user perceives a true 3D image and can examine the virtual scene from different viewpoints by physically moving around the display. Stereoscopic displays raise concerns beyond those found in simpler monoscopic display systems. To maximize viewing comfort and user interaction, viewing parameters must be automatically and dynamically adjusted. This thesis contributes the following:

- a framework for understanding, classifying, and comparing software techniques that help the viewer fuse the two stereoscopic 2D images into a single 3D perception
- analytic descriptions of the distortions induced by the following fusion control techniques:
  - false eye separation
  - image shifting
  - image scaling
- a comparison of the geometric properties of the above three techniques with each other and with the other fusion control techniques
- a fusion control technique with fewer dynamic, geometric distortion components than prior methods
- a technique that balances multiple stereo viewing issues when traveling through an extensive, global terrain virtual environment while maintaining an exo-centric, or orbital view
- geometric guidelines for matching an application's geometric requirements to a set of appropriate fusion control techniques

CHAPTER I

INTRODUCTION

1 Overview

Virtual Environments (VEs) aim to perceptually place the user in an artificial world through computer-generated sight and sound. Through this medium, VEs offer a human-computer interaction paradigm which can improve the user's understanding of and interaction with 3D information spaces. Proposed and developing VE applications include 3D design work; data visualization for science, engineering, and medicine; training; education; and certain types of psychotherapy. The primary component of generating a virtual world is interactive 3D imagery. Stereoscopic VE systems generate imagery by presenting a unique perspective image to each eye. As a result, the user perceives a single, true 3D image that appears to exist in front of and behind the physical display surface. While stereoscopic display is an important and common feature in VR systems, further research is needed. Stereoscopic viewing raises concerns beyond those raised by monoscopic VR systems. Special techniques are necessary to maintain good stereoscopic viewing conditions as the user travels through and manipulates the virtual environment. This is especially true for extended virtual environments where the scene contains rendered geometric detail at scales covering several orders of magnitude. In such environments, users need to zoom in and out in order to move between detailed and global views.

This thesis researches techniques for maintaining good stereoscopic viewing conditions during user interaction with extended virtual environments. The research will cover stereoscopic HTD (Head-Tracked Display) systems such as the virtual workbench [Krug95][Serr95][Rose97]. HTDs are distinguished from HMDs (Head-Mounted Display) [Suth68] in that a HTD does not mount the display surface on the user's head. Instead, the displays are stationary, attached to a desk, tabletop or wall. Other examples include the CAVE [Cruz93] and 'fish-tank' VR [Ware93]. HTDs offer higher resolutions and less encumbrance than HMDs. HTDs may also more easily bridge the gap between desktop interfaces and virtual reality interfaces and can easily integrate both work environments. While stereoscopic viewing issues in HTDs are not

problematic for small scenes placed close to the screen, these issues become troublesome for general and extended environments. As the user travels through these environments, the displayed portion of the scene changes continually and the viewing parameters should be automatically and dynamically adjusted in order to maintain productive stereoscopic viewing conditions.

Maintaining good stereoscopic viewing conditions in a stereoscopic HTD requires balancing several goals. First, limitations exist on a person's ability to fuse stereoscopic image pairs into a single 3D image. When a user cannot fuse the images, double vision or "diplopia" results. Diplopia needs to be avoided. Second for fusible stereoscopic images, the added value of stereoscopic depth varies with scene content and geometry. Clearly, it is desirable to maximize the depth information added by stereoscopic display. Third, the stereoscopic HTD exists in a physical environment. This physical environment contains visual cues that can conflict with the stereoscopic imagery. Frame-cancellation occurs when a virtual object geometrically extends outside the view frustum. Since the object is not real, it cannot occlude the display's physical frame as a similarly positioned real object would. This can lead to a collapse of the stereoscopic depth illusion. Another issue is that when a user directly manipulates an object it should be brought close to the user and in front of the screen if possible. Finally, the 3D geometry perceived by the user should be equivalent to the 3D geometry modeled by the system. If this is not possible, the application designer should at least precisely understand the discrepancies.

To maintain good stereoscopic viewing conditions, the software must dynamically adjust the user's view of the environment as she travels through and manipulates the virtual world. There are numerous degrees of freedom which control the view. We are concerned with software-controlled degrees of freedom and we partition these into: view placement, view scale, and view optics. View placement refers to the location and orientation of the projection window. The projection window is the virtual representation of the HTD's physical display surface in the virtual world. View placement does not refer to eye point locations because in a HTD the user's head position is a physical parameter controlled by the user and is not under software control. View scale is a single degree of freedom that represents the viewer's size in the world. View optics include all other parameters modeled by the pin-hole camera model in interactive

computer graphics. This includes modeled eye separation, the position of the near and far clipping planes, field of view, and other distortions such as depth compression or expansion.

This thesis contributes to the following areas concerning the maintenance of good stereoscopic viewing:

1) View Optics: A variety of methods are used for controlling image fusion and/or maximizing stereoscopic depth. However, key geometric aspects are not understood for all techniques. In particular the techniques of image scaling, image shifting, and false eye separation manipulate the displayed 3D image indirectly through manipulations of the viewing model and 2D images. What these manipulations imply about the resulting 3D image has not been fully analyzed with respect to stereo HTDs. This thesis will present the 3D transformations that are equivalent to these indirect displayed 3D image manipulations. This thesis will quantify and analyze these transformations. We also investigate alternative methods of fusion control and geometrically compare our new methods to previous methods.

2) View Placement and Scale: The challenge for these view parameters is not understanding the distortion, but rather finding ways to dynamically adjust the parameters. Standard types of travel techniques must be carefully implemented and augmented with automatic view positioning in order to maintain productive stereoscopic viewing conditions.

   Our second goal is to investigate and develop automatic view placement methods to augment user controlled travel in order to maintain good stereoscopic views. Specifically, we develop a travel method for a whole-planet terrain visualization system using a map-like interaction metaphor.

3) Geometric Guidelines – Intuitively, view positioning is sufficient for avoiding frame cancellation and positioning objects. For controlling image fusion, however, the more complicated view optic techniques are often used. The question then becomes which image

fusion technique is most appropriate for a given stereoscopic application. Through comparisons of the geometric properties of various fusion techniques, this thesis provides guidance for matching an application's geometric characteristics to a fusion technique.

2 Thesis Organization

Chapter 2 reviews general background information concerning stereoscopic displays. The chapter reviews human stereoscopic vision, the algorithms and hardware that implement stereoscopic display, the physical and psychophysical limitations are of these displays, and how prior researchers have dealt with these problems.

Chapter 3 develops a framework for investigating stereoscopic HTD applications. The first section describes and motivates a classification scheme for view parameters in the computer graphic viewing model. The second section describes key geometric differences between two stereoscopic view manipulations. The third section describes and motivates two parallel classifications for stereoscopic applications. The final section builds on these classifications and makes several key observations concerning prior techniques for image fusion control.

Chapter 4 develops a travel technique suitable for a specific stereoscopic application. The system displays a whole-planet terrain database viewed from an orbital vantage point. The technique uses automatic adjustment of the view placement and scale parameters in order to maintain good stereo viewing conditions. This work is also presented in [WarZ99b].

Chapter 5 analyzes the problems with the false eye separation technique when applied to stereoscopic HTDs for controlling image fusion or maximizing stereoscopic depth. We show that false eye separation distorts the displayed 3D image by a non-affine collineation. This preserves lines but not parallelism. We derive and analyze an analytic description of this distortion. The analysis shows that even with *perfect* head tracking, the displayed 3D image dynamically shears with head motion parallel to the display and dynamically expands and compresses with head motion perpendicular to the display [WarZ99a]. This is vexing because one plus of adding head-tracking to a stereo display is to remove qualitatively similar

dynamic distortions found in non-head-tracked systems. We also investigate the distortions for two other image fusion techniques, image scaling and image shifting. In a limited case they have similar properties to false eye separation. In general, however, image scaling and image shifting have geometric side effects not present in false eye separation. Some of these side effects are anecdotally linked to viewing discomfort.

Chapter 6 investigates new image fusion techniques based on false eye separation. We design these techniques to counteract the dynamic nature of false eye separation's induced distortion. The initial method removes the lateral shearing effect. This method, called α-false eye separation, is presented in [WarZ99d]. Finally Chapter 6 presents the development of techniques called β-distortion and γ-distortion. These methods remove all dynamic dependence of the induced distortion on the head position. When applying these techniques, our anecdotal observation is that there are tradeoffs to be made between the rigidity of the displayed image and the ease with which one can look around nearby objects to see farther occluded objects. Chapter 6 discusses this tradeoff in detail. Finally, Chapter 7 concludes the thesis and discusses future work.

CHAPTER II

BACKGROUND AND PREVIOUS WORK

1 Overview

This chapter contains general background information concerning stereoscopic displays. The next section, Section 2, reviews the basics of human stereoscopic vision. The third section reviews how virtual environments implement stereoscopic displays and what their physical and psychophysical limitations are. The fourth section discusses how prior researchers have dealt with these limitations.

2 Physical Environments and Depth Perception

2.1 Concepts in Binocular Vision

In order to understand human factors issues in stereoscopic virtual environments, one must begin with a basic understanding of human depth perception in physical environments. Psychophysics texts cite roughly a dozen visual cues which the human visual system uses to determine the depth of viewed objects [Gold96]. Binocular vision or stereopsis is a key depth cue. Stereopsis is the perception of depth based on the discrepancy between the left and right eye views of the environment. The following description of binocular vision comes from [Hersh99] Chapters 2 and 3 and Davis et al. [Dav95].

In order to "look" at a particular point on an object in space, a human's eyes must rotate to look toward the fixation point and also adjust the lens's shape in order to bring the fixation point into focus. The rotation guarantees that the retinal image of the fixation point lies on the central portion of each retina. This central area contains the highest visual resolution and is called the fovea. This rotation process is called vergence. Verging onto the current fixation point may require convergence or divergence depending on the spatial relationship between the new fixation point and the previous fixation point. Convergence is rotating the eyes inward relative to their previous orientations while divergence is rotating the eyes outward relative to their previous orientations. In addition to the vergence movement, the eyes must adjust the

shape of the lens of each eye in order to bring the new fixation point into focus. This process is called accommodation.

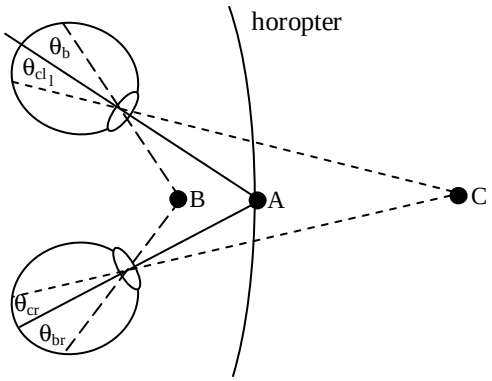


Figure 1: Retinal Disparity

Figure 1 illustrates a pair of eyes fixated on point A. Two other points, B and C, are also visible to the viewer. A pair of lines from point A through the eyes' lenses shows how point A is projected onto the fovea. Call these lines projectors. Two other points, points B and C, are also projected onto the retina. For point B we can measure the angle, θ_{bl}, between its left eye projector and the left eye projector of the fixated point, A. Similarly we can define the angle θ_{br} between B's right eye projector and A's right eye projector. Angular displacements inward from A's projector are positive while those outward from A's projector are negative. The sum $\theta_b = \theta_{bl} + \theta_{br}$ is called the retinal disparity of point B. While points B and C have non-zero disparities, the point A has zero disparity. There is an entire set of 3D points with zero disparity. These points define a special surface called the horopter. There are in fact multiple empirical [Hersh99,Chapter 3][Dav95] and theoretical [Gul76, Chapter 3] ways to define the horopter. However, it is beyond the scope of this review to discuss these details. For points not on the horopter, the sign and magnitude of their disparity is a function of the point's location. In general a larger magnitude indicates the point is located farther away from the horopter. Negative disparities, exemplified by point B, indicate the point lies between the eyes and the horopter, while positive disparities, exemplified by point C, indicate the

point lies beyond the horopter. A negative disparity is commonly called a crossed disparity since the eye must cross (converge) to re-fixate on these points, while a positive disparity is called an uncrossed disparity since the eye must uncross (diverge) in order to re-fixate on these points. Retinal disparity only indicates depth relative to the horopter. One might guess that knowing the rotation angles of the eyes and the separation between the eyes, the human visual system could determine the horopter's absolute depth and then in turn determine absolute depths for other points off the horopter using disparity information. Empircally, however, it is unclear what information is used to impart a sense of absolute depth to the horopter relative depth available from binocular disparity [Dav95].

When fixating on a point on a physical object, one can observe two phenomena. First, the images of objects at depths farther in front or behind the fixation point are increasingly blurred. Second, there is only a limited range of depths about the horopter for which an object's two retinal images can be fused into a single perceived image. Objects too far in front or behind the horopter will be seen as double images. The occurrence of double images is called diplopia. Typically we are unaware of these double images, but we can easily see them by fixating on a finger at arm's length and slowly moving a finger of the other hand to and fro relative to the fixated finger. Fusion limits can be visualized both as a retinal image measure and as a horopter relative measure in space. Fusion limits are described by Panum's fusion area which is illustrated in Figure 2 (pg 9). Figure 2 shows the eyes fixated on a point **F**. Two other points **A** and **B** and their retinal images are also shown. Figure 2A shows an overhead view while Figure 2B shows the retinas as seen from behind the head. In such diagrams one must distinguish between corresponding retinal locations and corresponding image locations. Two points on the retina correspond when the points are the same vertical and horizontal distance from each fovea. Two points in the retinal images correspond when the image points are both generated by light from the same point 3D space. For point $\mathbf{A_l}$ in Figure 2B, the corresponding retinal point in the right eye is $\mathbf{A_r}$. If we assume 3D point **A** exists but not point **B**, then **A** yields corresponding image points $\mathbf{A_l}$ and $\mathbf{A_r}$ which fall on corresponding retinal locations at these points. If we assume 3D point **B** exists but not point **A**, then **B** yields corresponding image points $\mathbf{B_l}$ and $\mathbf{B_r}$ which do *not* fall on corresponding retinal locations. In Figure 2B the crosshatched ellipse about $\mathbf{A_r}$ illustrates

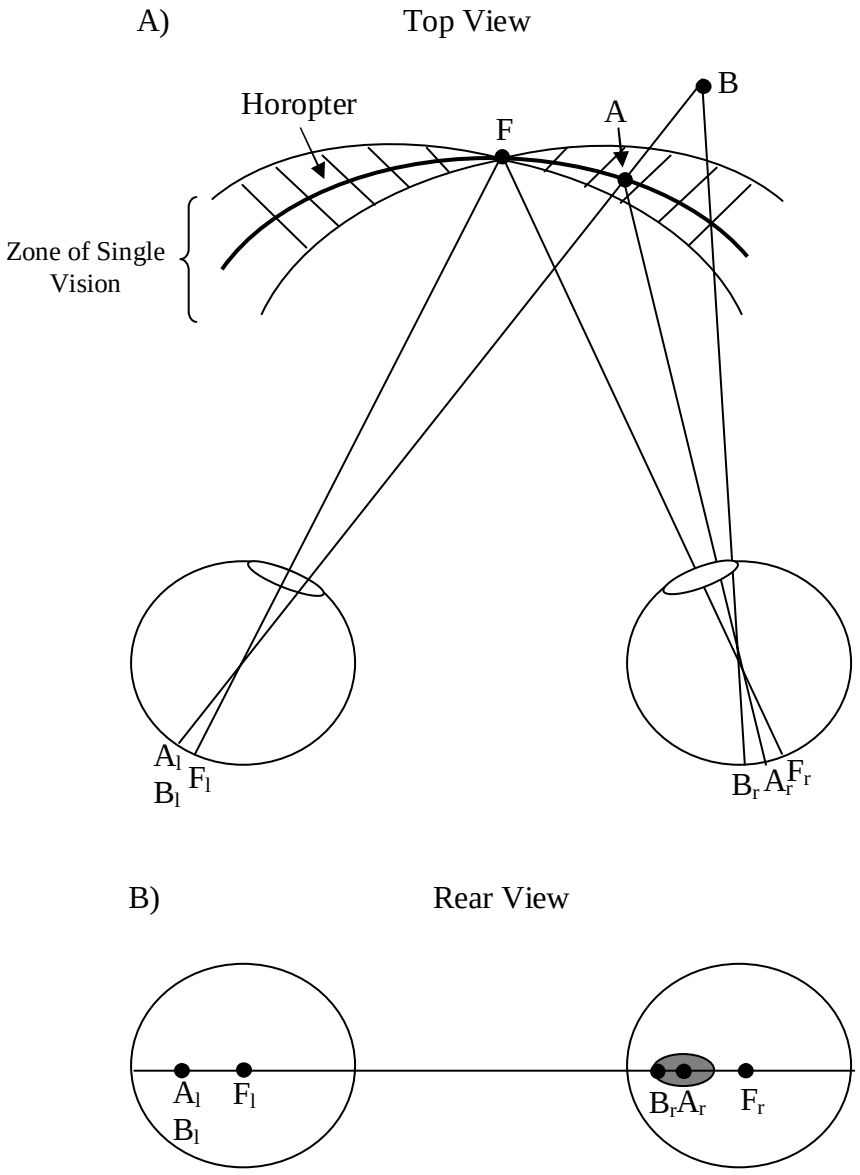


Figure 2: Panum's Fusion Area - (A) is an overhead view while (B) is a view of the eyes' retinas from behind the head. See text for details.

Panum's fusion area for retinal location $\mathbf{A_r}$. This ellipse is a region about $\mathbf{A_r}$ where an image point can fall and still be fused with a corresponding image point falling on $\mathbf{A_l}$. If we assume the 3D scene contains 3D point **A** but not point **B**, point **A** will yield images $\mathbf{A_r}$ and $\mathbf{A_l}$ which are fusible. If we assume the 3D scene contains point **B** and not point **A**, **B** will yield retinal images $\mathbf{B_l}$ and $\mathbf{B_r}$. Again since $\mathbf{B_r}$ is within Panum's fusion area for $\mathbf{B_l}$, these two points are fusible. If we move **B** outward along the line **AB**, $\mathbf{B_r}$ will move outside Panum's area and the point **B** will appear diplopic. This illustrates that Panum's fusion area on the retina defines a region of space about the horopter in which 3D points can appear and remain fused. This is often called the zone of single vision [Mart96]. Panum's fusion area (on the retina) is elliptic showing a larger tolerance for fusing horizontal disparities than vertical disparities. A typical range allows 10' to 20' for horizontal disparities and 2.5' to 3.5' for vertical disparities. Panum's area grows larger toward the periphery of the retina. It also varies with the spatial and temporal properties of the stimulus as well as the procedures used to measure it [Dav95,pg156]. Vertical disparities can arise due to perspective deformations that occur in stereo geometry as a function of the distance and spatial location of objects [Dav95, pg160].

The prior discussion ignored the issue of how the human visual system determines what points of each retinal image correspond to the same 3D stimulus. This complex topic of computer and human vision is far beyond the scope of this text. However, two relevant concepts are local and global stereopsis [Dav95]. Local stereopsis is disparity processing at one location in the visual field without reference to disparities at other locations in the visual field. For example, if presented with a single line segment in space there is little ambiguity as to what retinal image points correspond since each retinal image contains only one line. This stimulus's disparity processing can be done locally. Global stereopsis involves disparity processing at multiple locations in the visual field. In more complex stimuli which image point corresponds to which other image point on the other retina can become ambiguous. This ambiguity can be resolved by taking a more global approach. Psychophysical measurements using simple, single line stimuli (a local stimulus) can yield different results than more complex stimuli (a global stimulus).

Stereopsis, i.e. the perception of depth, does not necessarily require image fusion [Dav95]. When the viewer can determine both the direction (nearer or further) and the magnitude of the stimulus depth, the

percept is called quantitative or patent stereopsis. Quantitative stereopsis does not require the images be fused but only occurs for small disparities. Qualitative, or latent, stereopsis occurs for larger disparities. In qualitative stereopsis the images are always diplopic and only the direction of depth is perceived. For still larger disparities the double image's depth may appear at the fixation plane or may appear to have no well-defined depth at all.

2.2 Spatial Distortion

Even in natural environments, humans often do not have correct perceptions of spatial organization. There is a discrepancy between the physical geometry and the perceived geometry. The psychology research on this phenomena is large and ongoing. It will be important to later discussions of virtual stereoscopic environments that the reader have at least a sense of the distortions in physical environments. Below, two representative experiments are cited to illustrate natural perceptual distortions.

Empirical studies of natural visual distortion use several techniques [Wagn85]. In magnitude estimation, subjects report a perceived measure such as size or distance of some target object. Subjects are given a reference object which defines the reported units. In category estimation, subjects place the target object's measure in one of a given set of categories. In perceptual matching, the subject matches a reference stimuli to a target stimuli based on some measure. For example, they might view multiple distant target stimuli and have to pick the one of the same size as a nearby reference stimulus. In mapping tasks, the subject constructs a small model, perhaps via hand drawing, of the environmental stimuli.

Wagner [Wagn85] asks subjects to estimate the distance between a pair of stakes stuck in the ground in an outdoor field in a 40m by 40m region. Subjects view the scene from three fixed locations at 0, 20, and 40 meters from the edge of the square region. The general result is that the distance between pairs of stakes is increasingly underestimated as the stake pair orientation becomes closer to being along the line of sight. On average the same physical distance is seen as slightly less than half as large when presented in the in-depth orientation as compared to a frontal orientation. When asked to plot the positions of stakes, subjects also show compression in perceived depth. Researchers have proposed several theoretic accounts for visual space including Euclidean, spherical, and hyberbolic geometries. Wagner finds two models that best fit his

data. These are an affine-model that scales along one depth axis and a vector contraction model that accounts for head rotations by scaling along the line of sight to a given target. In the affine case, the best-fit scale factors for Wagner's data vary from 0.38 to 0.53 depending on task. In the vector-contraction case, the factors vary from 0.40 to 0.49 depending on task.

Johnston [John91] showed subjects squashed or elongated cylinders on a stereoscopic display. Displays distances were 53.5, 107 and 214 cm. The images were devoid of any other depth cues (shading, texture gradient, etc). Subjects performed a categorization task. Subjects had to determine whether the presented cylinder was squashed or elongated compared to a true circular cylinder. Responses show that at close viewing distances, cylinder depth is overestimated. At the intermediate distance, perception is close to veridical. At the far distance, cylinders appeared flattened.

3 Generating Stereoscopic Imagery

Stereoscopic VR aims to stimulate the human experience of stereopsis. To generate 3D imagery, a typical VR system has a location and orientation tracking device, an image generator, and one or more displays. The tracking device determines the positions of the user's head and/or eyes and possibly of the displays. The image generator computes the image that each eye would see on a display surface if the eye and the display existed inside the virtual world at their tracked positions. This image is then fed to the physical display. Typical VR systems are configured either as a head-mounted display (HMD) or as a head-tracked display (HTD). In a HMD, the display is attached to a helmet or headset worn by the user; hence as the user looks around the environment both the eye points and the displays are in continuous motion. In a HTD, the display is stationary, attached to a desk, tabletop, or wall. In this case only the eye points move as the user physically looks around. HTD examples are the CAVE [Cruz93], the virtual workbench [Krug94] and desktop or "fish tank" VR [Deer92][Ware93]. The CAVE surrounds the user with four or more large projected screens. The virtual workbench uses a single, large projected screen laid horizontally or at an angle. Desktop VR augments a desktop display with stereoscopic and head-tracking capabilities.

In order to create stereoscopic imagery, the system generates a unique perspective view of the scene for each eye. Software methods for stereoscopic display are well known [Hodg92, Sou95, Robi95]. Stereoscopic display for virtual reality has been shown to improve user depth perception and task performance in a variety of tasks [Rose93, Ware93] and under a variety of conditions [Dav95]. This is not surprising since real world experience shows that stereopsis is an important depth cue, especially for objects within the user's personal space (1.5 meters) [Cutt97].

The following subsections review stereoscopic display hardware and software and define key terms and conventions used in this work.

3.1 3D Display Hardware

3D display dates back to the early 1830's with the introduction of the Wheatstone and Brewster stereoscopes for viewing 3D hand drawings and later stereoscopic photographs [McAl93]. Stereoscopic displays are a subclass of more general 3D displays. While we are only concerned with stereoscopic displays, a brief overview of 3D displays illustrates the characteristics and limitations of our specific interest, the stereoscopic displays. In this dissertation the operational definition of a 3D display is a display with two properties:

(1) "retinal disparity"- (also called stereo parallax) The display presents a left eye perspective view of a virtual scene to the observer's left eye and different right eye perspective view to the observer's right eye. The retinal disparities in the left and right retinal images induce a perceived 3D image of the scene. Implicitly the user experiences ocular vergence as she fixates on objects at different depths.

(2) "multi-viewpoint"- (also called motion parallax) The image pair presented to the user's eyes is dependent on the user's head position so that by moving or walking around the display, the user perceives the virtual objects from different vantage points. "Multi-viewpoint" means continuous and correct changes to the perceived images as the head moves in *any* direction.

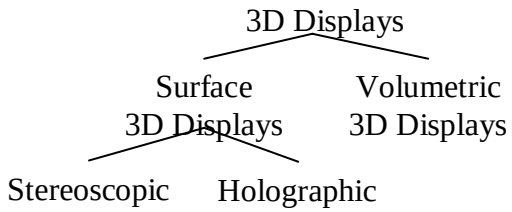


Figure 3: A Taxonomy of 3D Display Technology.

Researchers define several taxonomies for 3D displays [Hodg85][McAl93][Hall97]. In this thesis, we use the taxonomy in Figure 3. There are two classes of 3D displays, volumetric 3D displays and surface 3D displays. In a volumetric display, the display's light emitting elements sweep over or occupy a physical volume [McAl93]. In a surface display the light emitting elements occupy a single, physical surface [Hall97]. An immediate question for surface displays is how do we satisfy the retinal disparity and multi-viewpoint properties, which are properties of 3D dimensional objects, when we use a 2D dimensional display? The *ideal* surface display must recreate all light that would reach the observer's eye if the virtual scene were replaced by a real, physical scene. This light could come from any possible direction. Figure 4 A and C illustrates the two eyes of one observer fixated on a corner of a physical cube. Figure 4 B and D illustrates the two eyes fixated on the corner of a virtual cube as displayed by an ideal surface display. In the real world case (A and C), the light wavefronts corresponding to a single corner on the box are illustrated in gray. A triple of rays traces the formation of the corner's retinal image on each eye. As the person moves left to right (A to C), the eyes receive and focus a different part of the wavefront onto the retina. Wavefronts are, of course, emitted from every point on the cube but we've only illustrated the wavefronts for a single point. The complete set of wavefronts would yield the retinal images of the whole scene. Figure 4B and D illustrate the wavefronts that the ideal surface display must emanate in order to recreate the image of just a single point, the box corner. The ideal surface display is the vertical line. A triple of rays traces the formation of the box corner's retinal image. This time, of course, the box corner is

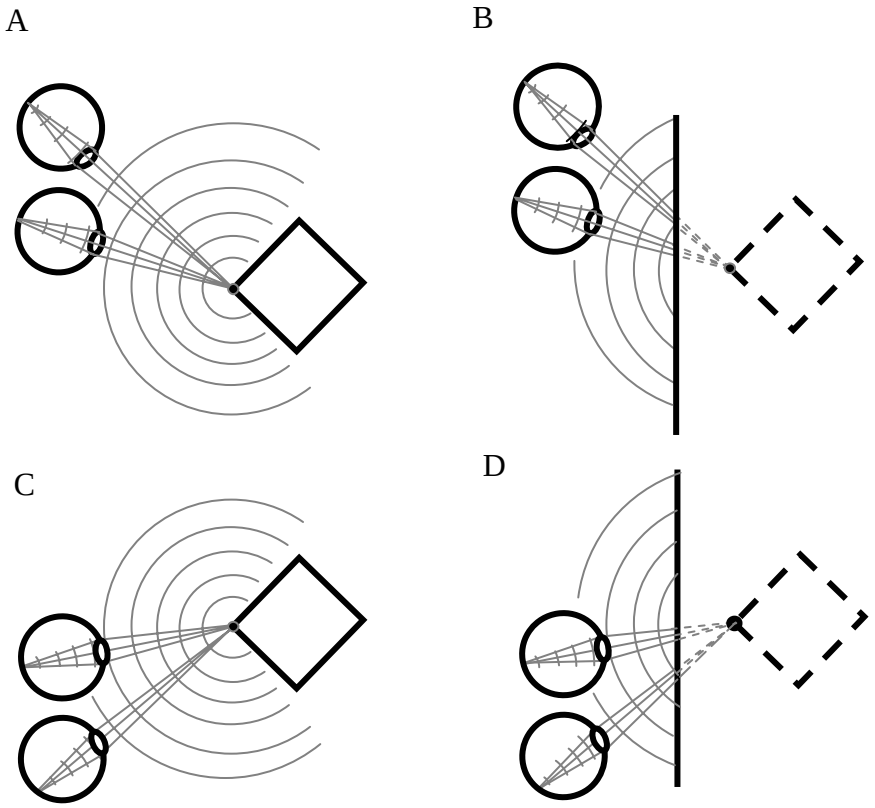


Figure 4: Viewing in Physical World and in an Ideal Surface Display -- (A) and (C) show a user fixated on a corner of a block in the physic world. (B) and (D) illustrate the wavefronts that the ideal surface display would generate.

virtual which is indicated by dashed lines. To recreate the entire scene, the display must recreate the wavefronts created by every point of every object in the scene in order to yield the virtual box shown as a dashed box. It should be clear that an ideal surface display satisfies the retinal disparity and multi-view properties. Finally, implicitly the ideal surface display induces ocular accommodation and creates depth of field effects. This is shown in Figure 5. In Figure 5 the synthetic 3D image consists of two boxes. The eye fixates on the corner of the larger box behind the display. This requires the eye to accommodate to the depth of virtual image of this fixation point. At the same time the image of other objects, such as the smaller box corner, will be out of focus.

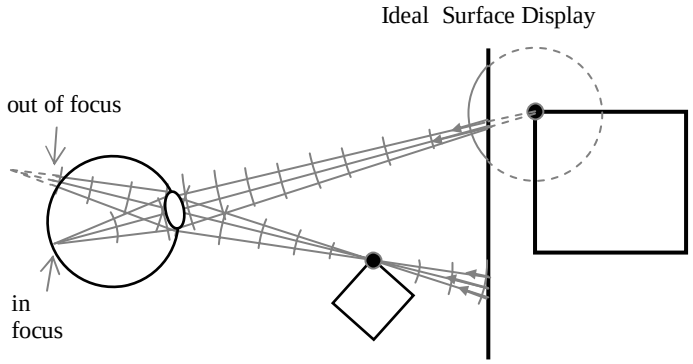


Figure 5: Retinal Image Focus in an Ideal Surface Display

The only surface display capable of generating conditions equivalent to the idealized surface display is the full holographic display. Through diffraction and interference effects of a fringe pattern, the hologram recreates the wavefronts of all the light flowing from the virtual scene through the display surface [Hall97][McAl93]. Full holograms therefore satisfy the retinal disparity and multi-viewpoint property and they induce ocular accommodation and depth of field effects [McAl93]. Full holograms achieve all these goals when viewed by the naked eye. Hardcopy holograms are routinely generated and important advances have been made in holographic video, but interactive holographic displays are not yet a commercial reality. They remain an active area of research but appear several decades away [Hall97].

The second type of 3D surface display is the stereoscopic display. All commercially available 3D surface displays fall in this category. Stereoscopic displays are built using common technology such as CRTs and LCDs. These surface displays cannot create the wavefronts of the synthetic 3D image as illustrated in Figure 4 and Figure 5. To satisfy the multi-viewpoint property a stereoscopic display must therefore: (1) determine the user's head position; (2) render a left and right eye image specifically computed for that head position; and (3) channel each of the two images to the appropriate eye. So while at a given moment a full holographic display outputs an image for every possible eye position, a stereoscopic display outputs an image for only the two current eye positions. (There are exceptions to

this. Some hardcopy stereoscopic images generate multiple views simultaneously, but in interactive VR only the two images are generally computed and presented).

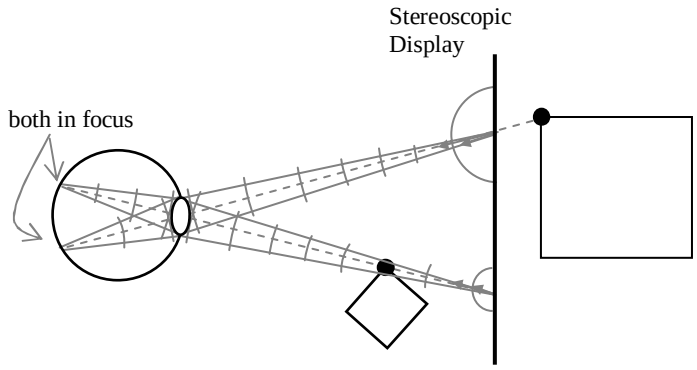


Figure 6: Retinal Image Focus in Stereoscopic Display

Computationally an interactive stereoscopic display system only simulates the principle ray from a virtual object to the eye by determining what pixel on the display is intersected by that ray and then lighting that pixel with the appropriate color. Physically, of course, an entire wavefront is produced by each pixel on the stereoscopic display. Figure 6 illustrates how a stereoscopic display renders corners of two boxes. Ray and wavefront diagrams are shown for a corner on each box. The simulated principle ray is shown as a dashed gray line. The pixel intersected by the principle ray is lit with the box corner's computed color. The wavefronts generated by these pixels are, of course, centered at the physical display pixels, not the virtual object's 3D locations. Whereas in a full holographic display the wavefronts appear to emanate from the 3D location of the boxes' corners (Figure 5), in a stereoscopic display they appear emanate from the display surface itself (Figure 6). In a stereoscopic display, the eye tends to accommodate to the physical display plane, not to the depth of the synthetic 3D points [Yeh93]. Virtual 3D objects at depths different from the fixation point will generally remain in focus. Therefore, unlike full holographic displays, stereoscopic displays neither induce proper ocular accommodation nor create depth of field effects. These limitations apply to the entire variety of stereoscopic display hardware including both those displays that

require users to wear glasses and the "auto-stereoscopic" displays which do not require users to wear glasses. See [McAl93] for a general technology overview.

3.2 Stereoscopic Display Computational Models

This section reviews the geometry of the common model for stereoscopic virtual reality image generation, focusing on stereo HTDs. A viewing coordinate system hierarchy convention is defined as are a number of other important terms.

The basic methods for interactively rendering these types of perspective projections using the standard graphics pipeline is discussed in [Fol92]. These basic methods are well-known, but complexities arise when adding stereoscopic display and head-tracking found in VR displays. First there is a conceptual issue. Textbooks talk about the "camera model." This terminology can lead to misconceptions when modeling VR displays. In particular, some programmers apply the "camera model" by modeling the eye pupils and the retinas. This is fundamentally incorrect! For VR displays the software must model the eye pupils and the display surfaces. The previous section should have made this clear. (More precisely, the software should model the "entrance pupil" of the eye [Robi92]. An entrance pupil of an optical system is the optical image of the limiting aperture as seen from the object side of the first lens.)

This thesis focuses on "planar-coincident" stereoscopic displays. "Planar-coincident" will mean (1) the display surface is planar and (2) that the displays perceived by each eye are coincident. Flat screen stereo HTDs fit this model exactly and to a degree so do CRT based stereo HTDs [Deer92]. HMDs often do not fit this model because the displays for the left and right eye can be rotated at different angles and the optics of HMDs can distort the image of the planar display surface into a curved surface [Robi92].

This discussion shows that the monocular projection model is easily extended to the stereoscopic case by using two different asymmetric, off-axis view frustums. Methods for implementing this model are discussed in [Hodg92] and [Sou95].

Just as stereoscopic display complicates the basic viewing model, so does head-tracking. When rendering in a VR system with head-tracking, the position and orientation of the projection window and the eye point positions must be modeled so as to conform to the physical display environment. This requires

using a tracking subsystem such as [Mey92]. Typical VR systems track the head location and orientation and assume the eye pupils are at fixed offsets from the head coordinate system. To properly integrate the tracking system information into the viewing model, one uses a hierarchy of coordinate systems that relate the tracker's coordinate system, the eye points, and the projection surfaces [Sou95, Robi95]. This thesis uses the coordinate system hierarchy illustrated in Figure 7. This hierarchy convention covers most VR display configurations based on 6DOF trackers. For a more general discussion see [Sou95, Robi95].

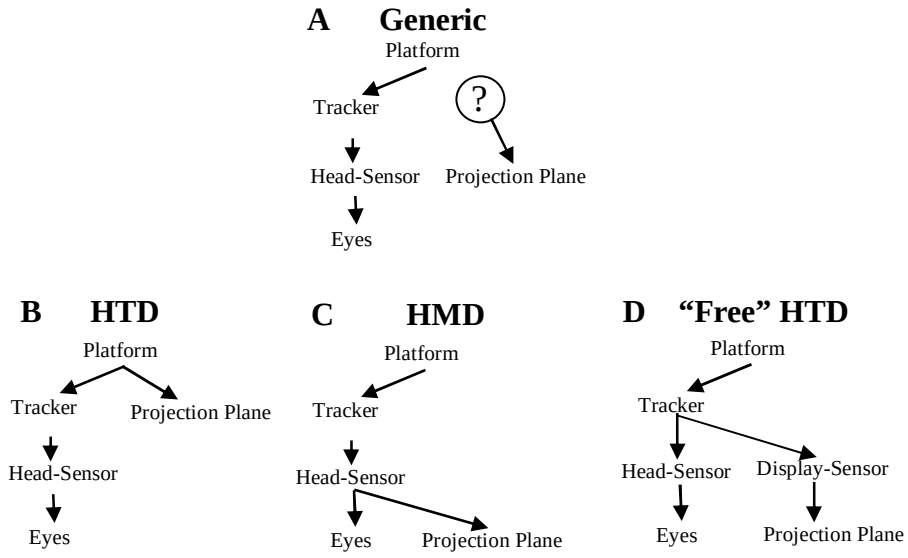


Figure 7: Viewing Geometry Coordinate System Hierarchies.

The top coordinate system is the Platform Coordinate System. Manipulating this coordinate system moves the viewpoint through the virtual space. Directly attached to this coordinate system is the Tracker Coordinate System. The Tracker Coordinate System simply represents the physical coordinate system defined by the tracking system. Attached to the Tracker Coordinate System is the Head-Sensor Coordinate System and attached to that is the Eyes Coordinate System. The two eye points are on the x-axis of the Eyes Coordinate System and are symmetric about its origin. The Projection Plane Coordinate System always contains the projection window in its XY plane with the window centered about the origin.

The Projection Plane Coordinate System may be arranged in one of three hierarchies. In a HTD, where the physical display is attached to a desk or wall, the Projection Plane Coordinate System is attached to the Platform Coordinate System (Figure 7B). In a simple, planar coincident HMD, a single Projection Plane Coordinate System is attached to the Eyes Coordinate System (Figure 7C). A third possibility is that the Projection Plane Coordinate System is attached to another tracker receiver (Figure 7D). In this scenario, the user is able to reposition and reorient the physical display in addition to moving her head. The display might be a light-weight flat panel display held in the user's hands or a large, heavy display mounted on some sort of pivot mechanism [McKe92].

Regardless of the display configuration, the position and orientation of each coordinate system relative to its parent is measured physically from the physical display setup. These measurements are taken either statically when the display system is physically configured or dynamically at run-time by the tracking system. The projection window dimensions are also computed from physical measurements.

The position matrix of the Platform Coordinate System defines the mapping between the physical space of the real world and the virtual space of the virtual world. In order to move the user through the virtual world, the application manipulates the Platform Coordinate System. In addition to specifying location and orientation, the Platform Coordinate System can also be uniformly scaled. From the user's vantage point, this causes the virtual world to grow and shrink. In a multi-user environment, other users would see the user's graphical representation changing in size [Leig96].

One might question the necessity of adding scale as a 7^{th} degree of freedom. Why it is not sufficient to simply move the viewpoint away from the scene to get a global view or conversely to simply move the viewpoint toward an object to get an arbitrarily detailed view? The problem is that this "zooming by translation" works very poorly in a VR interface employing a head-tracked display, a stereoscopic display, or direct manipulation with six degree-of-freedom devices. In these cases, zooming must be performed by scaling. With head-tracking, zooming out by moving the viewer away from an object will rapidly increase the sensitivity of the projected image to head-position (Figure 8A, next page). Such sensitivity can be quite distracting. By using an independent scale dimension, the system can scale down the object and preserve the object-viewer distance in order to avoid this problem. In the stereoscopic case, zooming by moving the

viewer towards or away from an object can bring the object either far above the display surface or far beyond the display surface making the image uncomfortable to fuse (Figure 8B). Scaling the object while keeping the object near the projection plane solves the problem [Ware95a].

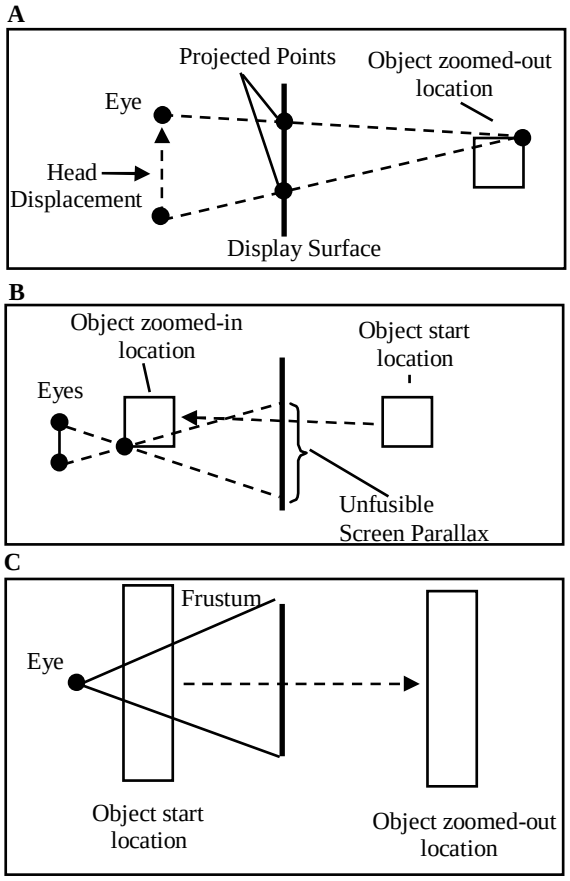


Figure 8: Problems with Translation Zooming

Finally direct manipulation using a six DOF device will be difficult for large objects. Manipulating objects is easiest when the user can see the complete object and the object is within arms reach of the viewer [Mine97]. If, in order to see the complete object, the viewer must move away from the object, both of these requirements cannot be satisfied for large objects (Figure 8C). Again the solution is to scale so that the object is small enough to be brought close to the user and still be viewed in its entirety. In all cases,

properly implementing this scaling requires manipulating the scale factor of the Platform Coordinate System.

Table 1: Four Eye Separations

	Physical	**Virtual** (Physical * Platform Scale)
True	True Physical Eye Separation (subject's interocular)	True Virtual Eye Separation
Modeled	Modeled Physical Eye Separation (software value)	Modeled Virtual Eye Separation

Table 2: Eye Separation Example

	Physical	**Virtual** (Physical * 10^6)
True	True Physical Eye Separation 6.0 cm	True Virtual Eye Separation 60km
Modeled	Modeled Physical Eye Separation 3.0 cm	Modeled Virtual Eye Separation 30 km

To determine the eye points' locations on the x-axis of the Eyes Coordinate System, we must specify an eye separation. This value is clearly a distance measurement, but when we speak of distances we must specify in which coordinate system we are taking the measurement. We define the "physical eye separation" to be the eye separation measured in the real world. This is the value used to determine the eye separation relative to the Eyes Coordinate System. Next, we define the "virtual eye separation" to be this physical eye separation multiplied by the Platform Coordinate System's scale. If, for example, the user's physical eye separation equals 6 cm and the user views a virtual Earth at a 10^{-6} scale (so that the planet appears as a small globe) then the virtual eye separation is 60 km. A further complication arises because researchers often deliberately miss-model the physical eye separation. They may deliberately exaggerate the physical eye separation or they may deliberately underestimate the physical eye separation. In this work, the term "false eye separation modeling" refers to the discrepancy between the user's true physical eye separation and the modeled physical eye separation. The term "false eye separation

modeling" does not refer to the discrepancy between the virtual eye separation and the physical eye separation. Table 1 illustrates these concepts while

Table 2 illustrates them applied to a specific example. To simplify notation, the terms "true eye separation" and "modeled eye separation" will henceforth refer the physical eye separations. Only if there is a possibility of ambiguity, we will include the prefix "physical." When occasionally discussing the virtual counterparts we'll use the terms "true virtual eye separation" and "modeled virtual eye separation." Again these are the true physical eye separation and modeled physical eye separation multiplied by the Platform scale factor.

Next we define a stereo VR display configuration to be orthostereoscopic when:

(1) the physical modeled eye separation equals the user's true physical eye separation

(2) the physical dimensions of the projection window are correctly measured and modeled

(3) the physical position and orientation of the Projection Plane Coordinate Systems are correctly measured and modeled

(4) the physical position and orientation of the user's head are correctly measured and modeled.

This definition allows for an arbitrary view (Platform) scale factor and therefore it does not require that the virtual eye separations equal the physical eye separations. This differs from the definition of orthostereoscopic presented in some literature which require a scale factor of one. Allowing the scale factor to vary freely within the definition of orthostereoscopy makes sense in stereoscopic HTDs because all other properties are easily met with interactive, head tracked image generation.

4 Depth Perception and Interaction in Stereoscopic Displays

There are 5 goals for maintaining productive stereoscopic viewing conditions:

- generating fusible stereoscopic imagery
- generating accurate stereoscopic imagery
- maximizing the added value of stereoscopic depth images
- minimizing frame cancellation

- bringing manipulated stereoscopic imagery within arms' reach to improve direct manipulation

Having reviewed the necessary background material we can now describe these problems in detail.

4.1 Image Fusion in Stereoscopic Displays

Ideally, in a stereoscopic HTD, the user's visual experience of the virtual objects would exactly match her visual experience of similarly lit and arranged physical objects. Unfortunately, this is often not the case. One problem for many stereo HTDs is cross-talk or "ghosting" [Yeh90]. Cross-talk is the presence of a faded or 'ghost' image of the right eye image in the user's left eye view and the presence of a ghost image of the left eye image in the user's right eye view. Cross-talk can occur due to imperfection of the filtering mechanism in the glasses or screen imaging mechanism. Yeh and Silverstein [Yeh90] have shown that the CRT issues which increase cross-talk will correspondingly *decrease* the observer's ability to fuse stereo images. While improved filter and shutter technologies have reduced cross-talk, the problem remains in modern systems.

Experiments in bulkier displays with completely separate left/right optic channels have still shown that user's may be unable to fuse images even in these more ideal systems which have no cross-talk [Surd97]. Chapter II, Section 3.1 (pg 13) illustrated that stereoscopic HTDs do not properly reproduce wavefronts so neither depth of field nor proper ocular accommodation is induced. These two issues appear to yield fusion problems. The figures in the following pages (Figure 9, Figure 10, Figure 11) help summarize and bring together the prior discussions on stereoscopic displays and the psychophysics of vision. In all three figures the eyes are the circles on the left. They view an either a physical environment or a stereoscopic virtual environment both of which consist of either physical or virtual boxes. The fixation point is indicated by dashed projector lines extending from the eyes. At the fixation point are several horopter related curves. The central curve is the horopter itself. The gray region about the horopter is the region where objects will appear singular (non-diplopic). This is called the zone of single vision [Mart96]. The horizontal arrowed-line at the top of each figure shows the distance to the fixation point. This line is labeled with three letters. O is the distance to the object. C is the distance where are eyes are converged.

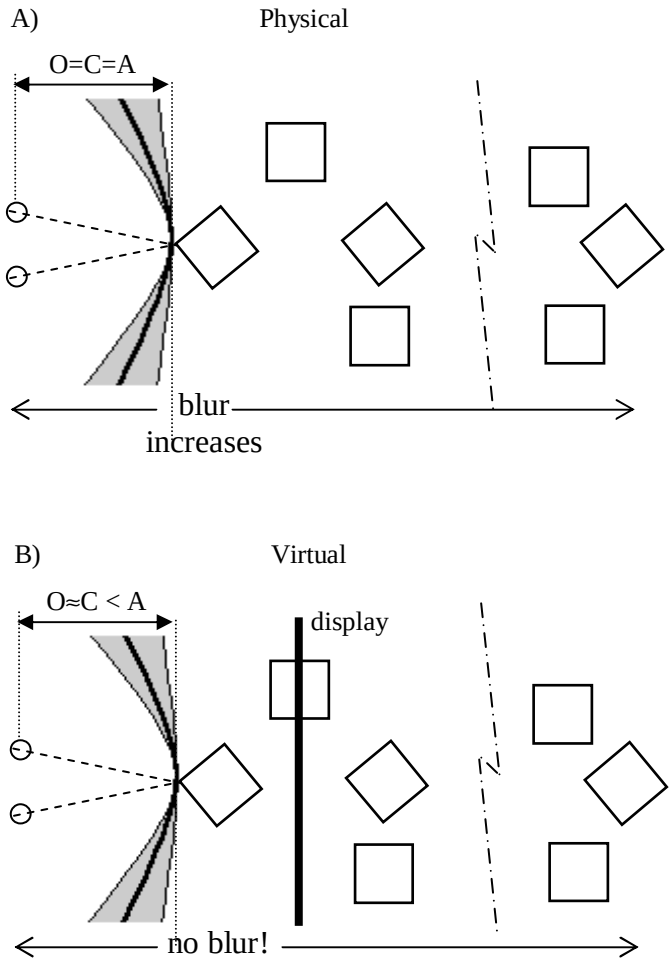


Figure 9: Differences between Fixating on Real and Virtual Nearby Objects.

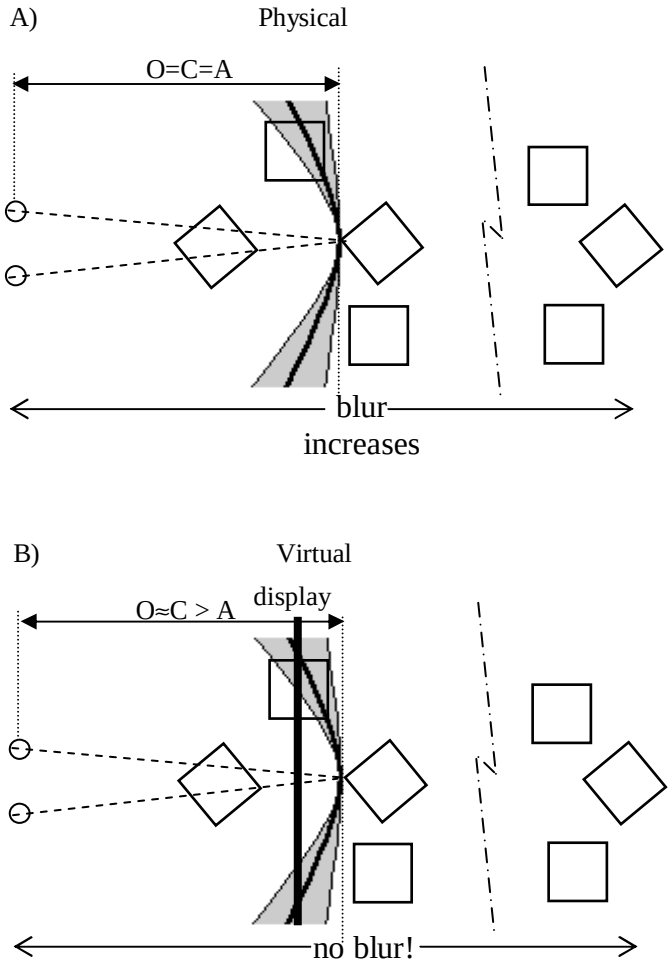


Figure 10: Differences between Fixating on Slightly Farther Virtual and Real Objects.

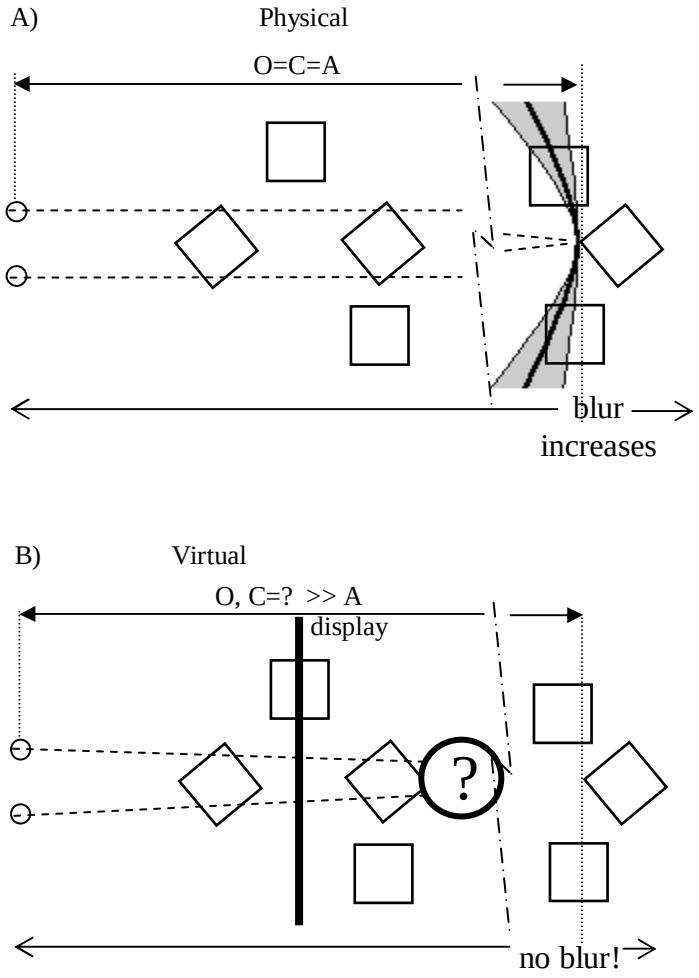


Figure 11: Differences between Fixating on Real and Virtual Distant Objects.

A is the distance to the focal plane to which the eyes are nominally accommodated. The zig-zag vertical line on the right indicates that the blocks on the far right are at a distance much greater than indicated by the scale of the diagram. In each figure, part A illustrates the physical environment case and part B illustrates the virtual environment case.

Figure 9 shows the eyes fixated on a nearby object. In the physical environment, object, vergence and accommodation depths are equal (labels O,C and A). Objects farther from the horopter are increasingly blurred (depth of field). In the virtual environment, object and vergence distance are roughly equal but these depths are smaller than the nominal accommodation depth which is at the screen. Objects farther from the horopter are not blurred. Since Panum's fusion area increases with lower frequencies which occur with blurring [Dav95] and since the virtual scene has no depth blurring, the zone of single vision in the virtual environment will be thinner than in the physical environment. Moreover, the double images for objects outside the zone of single vision will be more distinct in the virtual case than in the physical case due the lack of blur. This can make the double images more distracting in the virtual environment. Figure 10 illustrates similar information for a slightly farther fixation point. In the virtual case (B), object and vergence distance are now greater than the nominal accommodation distance.

Figure 11 illustrates a different problem. In the physical case (A), people have no problem verging their eyes to very distant objects. The projectors (dashed lines) show that the optical axes of the eyes are basically parallel. In the virtual case (B), however, people have problems. They may be unable to diverge their eyes away from the accommodation plane depth at the display. The discrepancy between the accommodation depth required to keep the image in focus and the vergence depth required to fixate on the distant target is too great. Similar vergence-accommodation conflicts can arise for virtual objects too close to the viewer. For brevity we do not illustrate this case.

The vergence-accommodation conflict arises because stereoscopic displays violate the natural correlation between these two ocular motor processes [Lipt93]. Mon-Williams et al. [Mon95] show that exposure to non-head-tracked stereoscopic displays leads to a temporary alteration of the natural correlation between accommodation and vergence. At extreme distances the user may be unable to diverge his eyes enough in order to fuse the images [Surd97]. At lesser distances where fusion is possible, users can still

experience eyestrain, headaches, and fatigue. While the lack of depth of field blurring probably reduces the size of the zone of single vision and makes doubles images more distracting, the vergence-accommodation conflict is considered the primary cause of visual fatigue in these systems [Lipt93][Sou95].

4.1.1 Stereoscopic Fusion Metrics

There are a number of metrics used to quantitatively characterize fusion limits on a stereoscopic display. These metrics commonly distinguish between virtual points that are in front of the projection window from those which are behind the window. We will refer to the space in front of the window as *near space* and the space behind the window as *far space*. (Lipton [Lipt82] uses the terms screen and theater space in reference to stereoscopic cinema).

The first common metric is *screen binocular parallax* or just *screen parallax* [Hodg93a]. This is the signed distance measured on the display surface between a single 3D virtual point's left and right eye 2D image points, L and R (Figure 12A, page 30). In Figure 12A the screen parallax for the illustrated virtual point is *p*. The screen parallax is positive if the virtual point is behind the projection window and negative if the virtual point is in front of the projection window. If the virtual point is on the window, the parallax is zero. Negative screen parallax values are referred to as crossed-parallax. This is because if the eyes are initially fixated on the projection plane they must cross to re-fixate on a point in front of the screen. For analogous reasons positive parallax values are referred to as uncrossed-parallax. Figure 12B illustrates how horizontal parallax changes with a virtual 3D point's location. Let *es* be the modeled eye separation. A 3D point in the projection window has zero parallax (point B, Figure 12B). Moving the 3D point from the projection window (point B) to a distance halfway towards the eyes (point A, Figure 12C) takes the parallax from zero to *–es*,. As the point is moved even closer towards the eyes the parallax grows towards negative infinity. If we move a 3D point from the projection window (point B) out towards infinity (point C, Figure 12D), the parallax goes from zero towards *+es*.

The second metric is the *horizontal visual angle* or *HVA* [Yeh90]. HVA of a 3D point is the visual angle subtended by the vector LR (Figure 12A). The HVA for points in near space is defined to be negative while the HVA for points in far space is defined to be positive. HVA is generally more useful

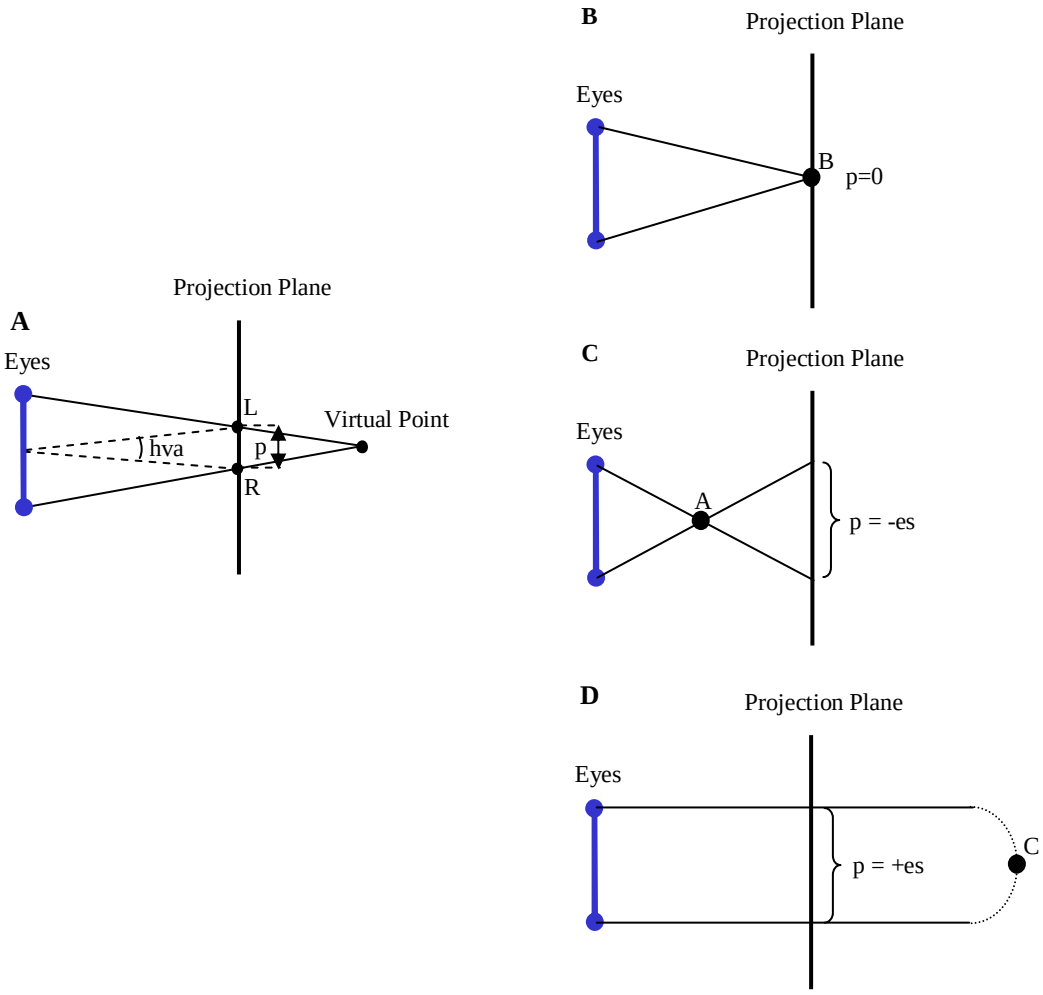


Figure 12: Stereoscopic Metrics -- (A) Horizontal parallax (p) and horizontal visual angle (hva). (B-D) Variation of horizontal parallax with virtual point location.

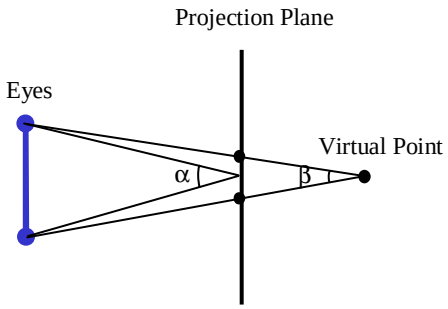


Figure 13: Vergence Difference

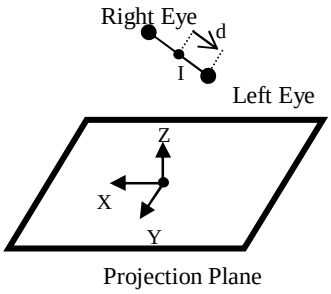


Figure 14: Labeled Stereo Viewing Geometry

than screen parallax since experimental results can be cited in HVA independent of the distance between the user's eyes and the screen.

The final metric is *vergence* difference [Sou92][Hodg93a][Valy66]. (Note, in [Valy66, pg 42] the term "differential parallax" is equivalent to vergence difference). Vergence is the measure of the change in vergence angle which an observer's eyes must rotate through when they re-converged from the display surface onto a virtual 3D point (Figure 13). For the illustrated point the vergence angle is α-β. Vergence for points in near space are negative and are called crossed-vergence and vergence for points in far space are positive and are called uncrossed-vergence. Note the difference in vergence angles between two points is equal to the retinal disparity between the two points in space [Hodg93a]. Therefore, the vergence

difference is a measure of the retinal disparity of virtual points assuming the user fixates on the display. Recall, retinal disparity of a 3D point changes with the fixation point.

The literature contains a number of recommended cross and uncrossed limits for these metrics. A given metric and recommended limit value pair implies a limited range of depth that should be displayed in front of and behind the screen. Generally, the range of allowable parallax grows with observer distance from the screen [Lipt93]. During application development it is useful to translate angular measures into the nearest and farthest comfortably fusible depths. Southard [Sou92] provides a useful approximation for mapping vergence angle to near and far fusible depths. For this and future equations, we define the following conventional variables shown in Figure 14. **I** is the center of the eye axis. **d** is the vector from the center of the eye axis to the left eye. These coordinates are measured relative to the Projection Plane Coordinate System. This system has an origin at the center of the projection window, X and Y axes parallel to the projection window sides, and a Z axis extending out of the window toward the user. Southard's equation is:

$$far\_fusible = I_z - \frac{2|d|I_z}{2|d| - I_z\theta_{\max}} \qquad (1)$$

where $\theta_{\max}$ is the positive uncrossed vergence angle limit

Valyus [Valy66] gives a vergence difference range of +/- 1.6 degrees. Yeh and Silverstein [Yeh90] experimentally find a fusible HVA range of -4.93 to 1.57 degrees for viewing durations that allow ocular vergence (2 s) and a HVA range of -27 min arc to 24 min arc for viewing durations that don't allow ocular vergence (200ms). They recommend keeping applications to the smaller of these ranges. William's and Parrish's experiments suggest a viewing volume of –25% through +60% of the head-to-screen distance. These data use the criteria of comfortable, fused vision in front of the screen and less than 10% perceived depth error behind the screen. Subjects view a virtual rod while adjusting the depth of a physical marker to match the virtual rod's depth. Seigel and Nagata [Seig00] empirically investigate using the minimum possible stereo parallax that still yields a sense of stereoscopic depth. Subjects view stereoscopic video of a

scene appearing behind the screen. Even when the maximum screen parallax is reduced to 1 mm there is a significantly greater than chance probability (~2/3) that viewers could correctly discriminate a "microstereoscopic" picture from a flat one. This parallax is less than two percent of the typical maximum screen parallax of 65 mm (derived from the average human eye separation). When using 1-3 mm camera/eye separation out of the nominal 60-65 mm separation, they found the parallax big enough to stimulate binocular stereopsis but small enough so that cross-talk is perceived as a blur instead of ghosting.

This dissertation does not answer the question of which recommended fusion limit metric and range is best. Rather, the goal is to examine the geometric consequences of applying different software fusion control techniques which will be listed shortly. When making certain technique comparisons, we must choose some fusion metric and recommended limits to apply to the geometric fusion techniques. Most recommendations, which we will review shortly, are fairly consistent at least for the first meter or so. Typically we will use the limit yielding the most liberal behind-the-screen depth range. If the depth compressing nature of a fusion technique yields undesirable geometric artifacts under a more liberal limit, then these artifacts will only grow stronger if we apply a more stringent fusion limit that would require larger depth compression.

Interestingly, for stereoscopic cinema Lipton advocates allowing a slight divergent positive parallax, that is a positive parallax greater than the nominal eye separation, under certain conditions [Lipt82, pg 103]. Lipton allows for a positive parallax which causes each eye to diverge ½ degree outward beyond its parallel orientation. Geometrically such divergent parallax cannot be converted to a proper depth and it clearly indicates a maximum fusible depth "greater than infinity." This heuristic would be the most liberal possible. However, the use of divergent parallax in stereoscopic cinema arises from an artifact of stereo filming. The common way to film stereo cinema is with a cross-lens-axis (CLA) system [Lipt82, pg 103]. This is quite similar to rotating the viewing frustums in a planar-coincident VR display instead of using proper asymmetric viewing frustums. The reason for using a CLA system is that while producing asymmetric viewing frustums is trivial in VR it is difficult to do optically in real cameras. A side effect of using the CLA technique is that positive screen parallax on the movie screen will increase without bounds. This is unlike the situation with proper asymmetric viewing frustums in VR where positive parallax reaches

a limit equal to the modeled eye separation. As a result, in stereo cinema only a finite range of depth in model space (i.e. the physical filming space) beyond the convergence plane will map to a given positive parallax limit. (In contrast, in VR the infinite range of depth in model space beyond the convergence plane will map to positive parallax values up to the modeled eye separation). Therefore, for stereo cinematography disallowing divergence would greatly limit the range of depth beyond the cameras' convergence plane that physical objects could occupy in the physical filming space (see [Lipt82], figure 3.8). Hence a motivation for allowing divergent parallax in stereo cinema is to counteract an artifact of the CLA system; but this motivating artifact does not exist in VR at all. Furthermore, with respect to stereoscopic VR, Lipton [Lipt93, page 15] later suggests that practitioners use the vergence difference limit of Valyus as a rough rule of thumb. For these reasons, we will not consider the practice of allowing divergent positive parallax when choosing among fusion limits appropriate for stereo HTDs.

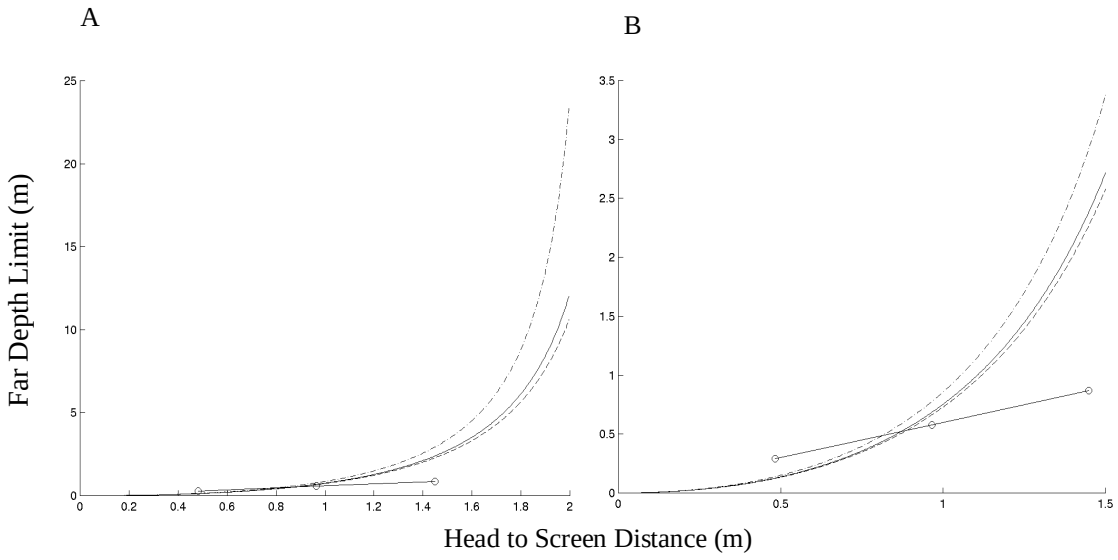


Figure 15: Farthest Fusible Depth Versus Eye-to-Screen Distance -- Solid line – Valyus's+1.6 vergence difference; Dash line – Yeh's +1.57 HVA; Dash-dot Valyus's max parallax approximation; Circles – William and Parrish limits.

Figure 15A shows Valyus's +1.6° limit is more liberal than Yeh's 1.57° HVA limit. The figure plots the resulting behind-screen-depth against head to screen distance. The solid curve is depth from Valyus's limit computed according to Southard [Sou92]. The dashed curve is depth from Yeh's 1.57° HVA computed by mapping HVA to screen parallax and then screen parallax to distance. All computations use a 6.5 cm eye separation. The dashed dotted curve is Valyus's own approximation where *max_parallax* = 0.03 * *head-screen-distance*. Again, this parallax is converted to maximum depth using simple trigonometry. In Figure 15B a smaller range is shown along with empirical data from William and Parrish [Will90] shown as circles. William and Parrish don't use a fusability criteria for their limit in far space. Rather these values represent the distance at which the perceived depth error is under 10%. They suggest limiting far space to 60% of the head-screen distance. In their experiment participants viewed a physical target next to a virtual target. Having a physical target to converge and accommodate towards next to the virtual target might account for the apparent ease with which participants could fuse the virtual target as compared to the sometimes smaller fusible depth indicated by the other recommendations. Nonetheless, these metrics are reasonably consistent for the first meter or so.

Like the far fusible point, the nearest comfortable fusible point also varies with head position. So similar plots can be made for space in front of the screen. While there is no asymptotic effect for near space, the farther from the screen a user stands, the farther in front of the screen geometry can be and remain comfortably fusible. Overall, these results indicate that while in a large form factor display with a user standing far enough away, fusion of far space is not problem on average, in the smaller form factor display systems fusion of far space is a problem. Near-space fusion can be a problem for any form factor if virtual objects appear too far in front of the screen.

These analytic models can only be treated as heuristics, because they are only average subject values measured from simple, specific environments. In real VR applications, contrast, screen parallax and geometry distribution interact in complex ways when determining the fusibility of different regions of an image. As a result, the fusion limits outlined above may be either too liberal or too conservative for a given

view of a particular scene. The literature does not appear to contain any attempts to dynamically analyze via software all the fusion related properties in order to determine a more precise indication of what parts of the VR scene will be least comfortable to fuse at any given moment. This thesis will often need to analytically examine the effects of different fusion control algorithms. When doing so, parameters of the control algorithms will be set based on how much a typical scene must be compressed to make it fusible to a typical viewer. As mentioned, we will use the most liberal published fusion limits of which we are aware when determining this geometric compression factor.

4.1.2 Addressing Fusion Issues

Because stereoscopic displays do not properly reproduce wavefronts, they do not create proper depth of field effects. As discussed this can reduce the zone of single vision around the horopter and make double images of more distant objects more distracting. Some experimental and anecdotal results indicate that simulating depth of field reduces these problems [Naga96][Mart96]. However, to allow the viewer to fixate on a variety of depths, the simulated depth of field must be dynamically generated based on the fixation point as determined by ocular vergence. This requires additional hardware for proper eye gaze tracking and requires multi-pass rendering techniques that significantly reduce frame rate [Woo97, pg402]. Finally, these techniques do not directly address the vergence-accommodation conflict.

The vergence-accommodation conflict might be reduced with additional hardware. HMD researchers are trying to dynamically adjust the optical depth of the display surface. This requires tracking the eyes' convergence angle, computing the 3-space fixation point and optically adjusting the distance of the display to the depth of the fixation point. Researchers are pursuing this in HMDs [Omura96][Sugi98] but it is unclear how such techniques could practically translate to HTDs.

More commonly, researchers have pursued the following software techniques to reduce fusion problems by reducing screen parallax:

(1) 'underestimated false eye separation' – This method sets the modeled eye separation to underestimated value either statically [Hodg92] or dynamically [Ware95a][Ware95b]. Doing so reduces screen parallax of all virtual geometry. Note, in Ware et al. the term "virtual eye

separation" is equivalent to our term "modeled eye separation." So when Ware et al. discuss varying the "virtual eye separation" in our terminology they are discussing a discrepancy between the modeled and true eye separation. In our terminology the term "virtual eye separation" means something completely different that Ware's use of the same term. Recall when we use the term "virtual eye separation" we are referring to the physical eye separation (modeled or true) multiplied by the platform scale factor. We consider it a minor contribution of this thesis to fully distinguish between the four eyes separations "virtual true" , "virtual modeled" , "physical true" and "physical modeled."

(2) 'image scaling' – In VR this method scales down the projected images about the center of the screen [Sou92]. Southard refers to this as frame magnification. Such scaling reduces screen parallax of all virtual geometry. In stereo cinema "frame magnification" is the ratio of the film frame size to screen size denoted by the variable *M* ([Lipt82], pg 91). We, however, note that the effect of Southard's "frame magnification" in VR is subtly different from the stereo cinema variable *M*. In particular in VR, Southard's frame magnification overestimates the size of the modeled projection window with respect to the physical screen size. Like the *M* factor this clearly alters the size of image on the screen. However, unlike the *M* factor, Southard's VR frame magnification also has the effect of enlarging the view frustum's field of view and bringing more of the scene into view than would be visible otherwise. In order to keep this difference in mind, we refer to Southard's frame magnification method as image scaling. To summarize, image scaling in VR and the traditional frame magnification factor *M* have the same effect on screen parallax; but image scaling additionally brings in more imagery into the screen image.

(3) 'image shifting' – Image shifting translates the left and right images toward one another. If done carefully, this can reduce the maximum absolute screen parallax over the entire image. Its use to alter the stereo image dates back to early stereo photography of the 1800's [Valy66,201-202]. Various computer graphics specialists discuss it too [Lipt93][Hodg93b][Akka99] (Akka's frustrum asymmetry factor is basically the same as image shifting). Note, again as with image scaling, in VR image shifting also brings into view a bit more of the scene which is something

image shifting doesn't do when viewing stereo photographed prints. Seigal's and Nagata's [Seig00] center-of-interest correction in non-head-tracked, stereoscopic video is essentially the same geometric manipulation too.

(4) 'fusibility clipping' – This method sets the near and far clipping planes so as to clip out unfusible geometry [Sou95].

(5) 'perpendicular scaling' – This method scales the world perpendicular to the projection plane to bring objects closer to the projection plane prior to 2D perspective projection. It is an example of linear depth mapping technique mentioned in [Will90].

(6) 'asymmetric/asymptotic technique' -- [Will90] contrasts linear depth mapping techniques with techniques which map the depth coordinate asymptotically. (Presumably asymptotic techniques include false eye separation, however, no explicit techniques are cited). They give an improved 'asymmetric/asymptotic' matrix that also maps the depth coordinate asymptotically.

(7) 'view placement' – This method only adjusts viewer location and orientation. By itself, its use for fusion control is limited to simple scenes.

(8) 'view scaling' – This method adjusts the uniform Platform scale factor to make the scene fusible. [Ware95a][Ware95b][Ware98] mixes view scaling with underestimated eye separation in non-head-tracked displays. The scale is applied to bring geometry close to screen reducing far fusion problems. However, the scale is also effectively controlled by the user as a zoom factor. It is possible that the user might zoom in so as to see a small detail and as a side effect set the scale so that distant objects are too far to fuse. Therefore, an automatic eye separation adjustment is made to compensate for this possibility. For a HTD, [Ware97] uses view scaling alone but again the scale is under user control as a zoom factor. This raises the potential fusion problem mentioned above. For their particular application this doesn't appear to be a problem.

4.2 Image Accuracy in Stereoscopic Displays

In addition to desiring the stereo image to be comfortably fusible we desire the perceived 3D image to be an accurate portrayal of the modeled 3D image. The *modeled scene* can be defined precisely and

exactly as the scene ideally presented to the viewer taking into account the view placement and the view scale factor and eye positions. Unfortunately, the perceived scene can only defined empirically. No theoretic model will exactly fit a viewer's perception. At best one takes empirical measurements and then examines what geometric model statistically best fits the data (Chapter II, Section 2.2). These difficulties arise even when viewing real, physical scenes and as discussed in that section there are discrepancies between physical space and perceived space. Call these distortions "natural distortions". Virtual reality adds further complications and distortions since technical limitations and errors will create a discrepancy between the user's perception of the virtual scene and the user's perception of the mimicked physical scene. Call these distortions simulation distortions.

Simulation distortions have many sources. HMD optics or display curvature should be taken into account [Deer92][Robi92]. Failure to do so yields warped 2D images that imply some warping of the presented stereoscopic 3D image. In HTDs, tracker latency can lead to observable swaying in the 3D image as the rendered imagery lags behind the viewer's true head position. Systematic tracker error introduces further error. Additionally, many of the described fusion methods distort the scene in someway.

Simulation distortions can be examined both empirically and theoretically. For empirical examinations virtual reality researchers typically operationally define the perceived scene through a registration experiment. The user is asked to position or register a physical pointer at the same perceived location as a displayed virtual, stereoscopic object. Ellis and Menges [Ellis97] display a virtual tetrahedron through a stationary see-through HMD. The tetrahedron is rendered at a sequence of distances between 33 and 108 cm. Subjects position a physical pointer at the perceived depth of the virtual tetrahedron. When the accommodation distance is optically adjusted to the tetrahedron distance, the distance responses were completely linear, unbiased and unskewed with errors of ±0.3 cm. In a second experiment, the tetrahedron is rendered at 58 cm but accommodative depth was at 50 cm or infinity. Again, the distance responses were almost perfectly accurate and accommodation depth did not significantly affect the response. Yoshida et al. [Yos99] examine a similar task using a large projected display. The subject's head was held stationary while eye-screen distance varied between 70 cm to 100 cm and eye-object distance varied between 30 cm and 50 cm. Registration errors are generally under 1 cm and the authors present a

numerical technique that theoretically can reduce the RMS (root mean square) errors to within 0.013 cm for the tested subjects.

Simulation distortions can also be examined theoretically by asking the question: "if the projected 2D stereo images are distorted due to artifact X, what 3D distortion of the 3D scene would yield a geometrically equivalent end result?" The focus is on algebraically deriving geometrically equivalent 3D distortions. When the modeled scene is modified by this geometric distortion, we refer to the new scene as the *displayed scene*. (Our use of the terms *modeled scene* and *displayed scene* are equivalent to the terms *object* and *image* in various stereoscopic photography and cinema literature.) These geometric distortions contrast with perceptual distortions which can only be assessed by picking some operational definition for the perceived scene and performing an experiment with human subjects. Geometric analysis of stereoscopic images indicates the expected nature of the perceptual distortion due to the simulation artifact. These distortions are above and beyond the natural perceptual distortions that would occur even if the display exactly reproduced the wavefronts found in simulated physical environments. Some geometric distortions may be more perceivable than others. For instance, stereoscopically induced curvature of a "plane" may be perceivable only for curvatures that are "strong enough" [Jule71].

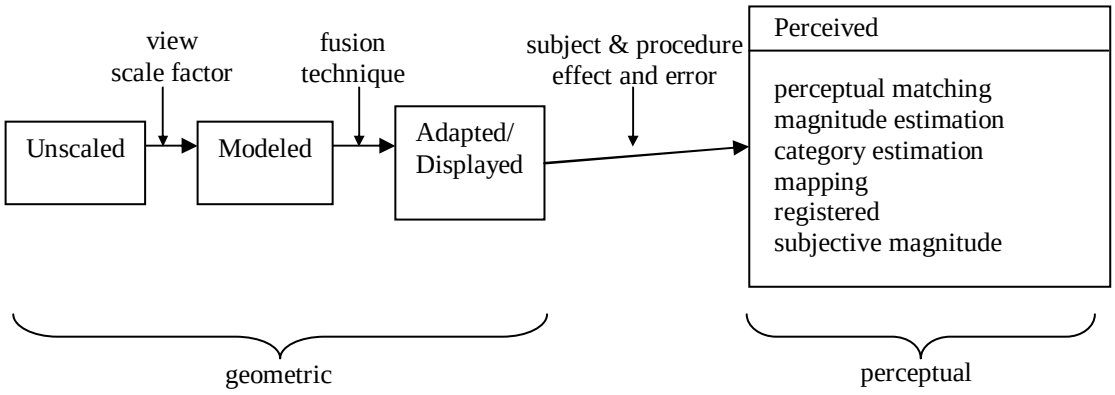


Figure 16: Geometric Versions of the Virtual Scene.

This dissertation focuses on the geometric distortions deliberately generated by fusion control methods. We refer to the modeled scene as modified by the fusion technique as the *adapted scene*. As

discussed above this is a geometric, theoretical construct. Typically we will have to assume that the only source of distortion is the fusion technique when performing algebraic and geometric derivations. Therefore we will often treat the adapted and displayed scene as equivalent. Figure 16 illustrates the various defined scenes and indicates their character.

To a large degree research dealing with the stereo image accuracy has proceeded separately from that dealing with fusion problems. The fusion management techniques deliberately distort the 3D image in various ways. On the other hand, methods investigating image accuracy limit their virtual environments, display configurations, and viewing durations to those which the experimental subjects can comfortably fuse. If we have to deliberately distort space for fusion control, we should at least have an accurate analytic understanding of the distortion so that we can account for it if possible. This thesis will address this issue.

Various researchers have geometrically analyzed distortions in stereoscopic displays. However, none fully analyze the effect of image shifting, image scaling or false eye separation in a manner that completely accounts for the effects on stereoscopic HTDs.

Robinet and Rolland [Robi92] present a computational model for HMD optics describing how these optics distort straight lines into curves. Watson and Hodges [Wats95] demonstrate a real-time method for compensating for this distortion.

Deering [Deer92] discusses several aspects of accurately modeling stereoscopic HTDs. First he points out the variation in the true eye separation due to ocular vergence and he suggests a few solutions. He then qualitatively discusses the distortions due to tracker lag. Finally, he presents a quantitative description of the distortions due to the curvature and refraction of the front glass in CRTs. He also gives a run-time method to compensate for these latter two problems.

Hodges et al. [Hodg92, Hodg93a] discuss qualitative aspects of the incorrect modeling of the user's head position in non-head-tracked stereoscopic displays. As the user's head is displaced from the modeled location, the perceived stereo image appears to contract or grow and shift side to side. Similar observations are made in the stereoscopic display literature [Spot53][Valy66][Lipt82]. Additionally, Hodges et al. quantitatively analyze the change in eye separation due to convergence.

Hodges and McAllister [Hodg90] present an analytic description of the distortion of the 3D image for planar-coincident, stereoscopic displays if view frustum rotations are used to model the binocular viewing geometry. They discuss the induced vertical parallax and non-line preserving distortion and they conclude that the rotation method is inappropriate for single screen stereo displays.

Across a number of stereoscopic media, a number of researchers investigate the distortion due "discrepancy between eye and camera separation." The literature yields two different distortions results. One group finds a uniform scale distortion while another finds a non-affine collineation. As pointed out by Lipton [Lipt82], some authors have stated that prior authors' results were incorrect [MacA54]. To our knowledge there has been no published attempt to precisely investigate where the differences lie in the varying geometric analyses. We will elucidate the distinction between these in a separate chapter, Chapter III. We consider this elucidation to be a contribution of this thesis. For now accept without proof that the non-affine collineation group fits what we have called false eye separation while the uniform scale group fits what is often called hyper/hypo stereoscopy.

Ware et al. [Ware95a] present a brief discussion of the change in the perceived depth of a point for false eye separation modeling in non-head-tracked stereo displays. (As noted earlier, Ware uses the term "virtual eye separation" to refer to what we call the physical modeled eye separation).

Woods et al. [Wood93] derive an analytic description of distortions in stereoscopic tele-operator systems. They assume the viewer is looking at a single display surface while the image generating cameras may be parallel or angled-inward. In the parallel case using false eye separation, the distortion preserves lines but not parallelism. In the angled-in case the distortion maps lines to curves. Woods' treatment assumes the eye axis is parallel to the display plane and is not titled and that the center of the eyes lies on a line perpendicular to the display and through its center. These assumptions are, of course, not true in a stereoscopic HTD system and therefore Woods' results do not cover this case.

Another stereo distortion is due to stereo base mismatch or hyper/hypo-stereoscopy. This was originally studied in the context of various binocular devices for viewing the real-world. For example, a telestereoscope is a non-magnifying, binocular viewing apparatus. Using mirrors a telestereoscope gives the left and right eye two widely, horizontally displaced views of the real world. The stereo base is the

distance between the first apertures of the left and right eye components. Stereo base mismatch is the discrepancy between the device's stereo base and the user's eye separation. Hyperstereoscopy has the stereo base larger than the viewer's eye separation while hypostereoscopy has the stereo base smaller than the viewer's eye separation. The distortion due to hyper/hypo stereoscopy is to uniformly scale down the displayed space. So buildings might appear as small models. The distortion due to hypo-stereoscopy is to uniformly scale up the displayed space. Various authors provide various proofs and examples illustrating that the distortion is a uniform scale. Bercovitz [Berc98] provides geometric and intuitive proof in the context of non-head-tracked stereoscopic video systems (see Figure 2, [Berc98]). Valyus [Valy66,pg 47-50] provides an intuitive geometric proof motivated by telestereoscopes and stereo photography (see Figure 26, Valy66). Dewhurst presents example diagrams (figure 6-2, figure 8-2 [Dew54]) and Spottiswoode [Spot53, pg65] discusses it too. Rule provides a diagrammatic and analytic proof of this uniform scaling effect in the context of stereo photography (figure 4 [Rule41a], figure 2 [Rule41b]). (As a word of warning, Rule [Rule41b] uses the term "stereoscopic magnification" to refer to the uniform scale of hyper/hypo stereoscopy. Spottiswoode generally uses the term "stereoscopic magnification" [Spot53] similarly, but he also uses the term stereoscopic depth magnification and stereoscopic width magnification. The latter are differential quantities depending on position in display space. In the case of hyper/hypo stereoscopy, however, these terms are constants equal to the uniform scale factor of hyper/hypo stereoscopy). Norling [Norl39] simply states that hyperstereoscopy scales the displayed scene. In reference to telestereoscopes and stereo photography, Helmholtz analytic derivation originally published in 1866 [Helm62,pg343] is undoubtedly the earliest treatment. Finally, in reference mainly to stereoscopic cinema and photography, MacAdam appears to disagree and presents a geometric construction that is a non-affine collineation.

A number of these authors examine other distortions as well. MacAdam [MacA54] illustrates the distortion due to frame magnification (image scaling) in stereo cinema. Focusing on tele-presence video systems, Woods et al. [Wood93] also have an analytic distortion description that accounts for image scaling and alterations of convergence distance which is roughly equivalent to image shifting. These

descriptions, however, assume the eye axis is parallel and centered relative to the screen which is not true in stereo HTDs.

Valyus [Valy66,pg. 389] gives an analytic description of the distortion arising from a discrepancy between the true and model eye separation and the true and modeled central eye position. However, the derivation assumes the eye axis is parallel to the projection plane and is not tilted (i.e. roll and yaw angles equals zero) which is not true in a stereo HTD. Nor does Valyus focus on the case where head position is correct but eye separation is incorrect which is of interest for understanding false eye separation on stereo HTDs.

For stereo cinema, Dewhurst [Dew54, Chapter 2] diagrammatically examines the consequences of rotating the projectors at a sharper angle than the camera's were initially. From the diagram it is clear the induced distortion is at least a non-affine collineation. That is it is clear that parallelism is not preserved but the diagrams appear to assume without proof that straight lines are preserved. Such rotations also cause the displayed image to move forward and backward relative to the stereo window. These rotations' effects on the screen image is similar but not equivalent to the effect of image shifting.

Rule [Rule41a] derives an expression for display space distortion within a single 2D plane that contains the eye axis and is perpendicular to the projection window. His equation explicitly accounts for distortion induced by frame magnification, the discrepancy between actual observer-screen depth and observer-screen depth required for orthostereoscopy and hyper/hypo stereoscopy. Implicitly his equation also accounts for image shifting. Rule explicitly notes that the distortion is a plane-to-plane projective transform (i.e. a 2D non-affine collineation) and he illustrates how this plane-to-plane projective transform can be constructed by taking the model and display planes into 3-space and performing a 3D perspective projection. Rule, however, assumes that the camera (equivalent to the model eye locations in VR) and the observer are both centered relative to the screen and that both the camera and eye axes are parallel to the screen. These assumptions will not capture the geometric situation in a modern stereoscopic HTD system.

For stereoscopic cinema Spottiswoode [Spot53] examines the distortion effects taking into account the camera positions, the effects of the optical printing process, the effects of the projector arrangement and the effect of viewing position. Given the generality of this approach, the equations do account for frame

magnification, image shifting, false eye separation and hyper/hypo stereoscopy. However, the primary focus is on the effect on the Z or depth coordinate ([Spot53], equation 15) and how depth ranges in camera (i.e. modeled) space map to depth ranges in image (i.e. display) space. He does not focus on the complete 3D space transformation description.

As far back as 1866, Helmholtz analytically examined the effects of image shifting in Chapter 30, "Perception of Depth," in his *Treatise on Physiological Optics* [Helm62,pg 333]. He investigates the distortion in some detail focusing on the various analytic properties of the distortion. The properties described are those of a non-affine collineation. The equations can also account for a discrepancy between modeled and true head position. If combined his equations appear to have enough degrees of freedom to account for the effect of image shifting as the true and modeled head move together. This could be used to describe the effect of image shifting on a stereo HTD if we limit the eye axis to be parallel to the projection window. However, Helmholtz does not investigate this combination for the simple reason that the notion of a stereoscopic head-tracked display simply did not exist in 1866!

Finally motivated by robotic vision, the most general description of stereoscopic distortions is by Baratoff [Bara97]. Given a completely arbitrary relationship between the model and true eye points and projection planes, Baratoff shows the induced 3D transformation is a quadratic Cremona transformation-"a rational transformation which is one-to-one almost everywhere, but which does not in general preserve collinearity." Given its generality, all other possible descriptions of specific stereoscopic distortions must be a specific subcase of Baratoff's equations. Baratoff's motivation is to describe what happens in general in computer vision when the stereo cameras are poorly calibrated and vision algorithms attempt to reconstruct a 3D model of the world from the two camera images. Baratoff uses methods of algebraic geometry to perform the derivation using high level operators in order to avoid the exceedingly complex individual coordinate equation derivations. Due to the very generality of his analysis, Baratoff does not cover specific instances of stereo distortion such as false eye separation, image scaling and image shifting that arise from image fusion considerations in stereoscopic displays. Nor does he investigate the specific dynamic repercussions for stereo HTDs where the eye points are moving independent of the stationary projection plane. This simply does not occur in stereo cameras.

4.3 Maximizing Stereoscopic Depth

Stereopsis is a psychological depth cue. By simulating stereopsis a 3D display can improve the user's depth perception of the virtual scene. Clearly it is desirable to maximize the effectiveness of the stereoscopic effect. There are several considerations. First stereopsis is strongest as a depth cue within a human's "personal space", the first 1.5 meters in front of her body [Cutt97]. Based on geometric considerations and the limits of visual acuity, Valyus calculates that the minimum discernible difference in depth between two adjacent planes increases with distance from the eyes [Valy66]. He also calculates that an average person cannot stereoscopically distinguish depths beyond 1350 meters. That is, as far as stereopsis is concerned, depths from 1350 meters out to infinity are all the same. (Modern texts consider stereo vision to be ineffective past 20 meters). This limiting distance is called the radius of stereoscopic vision. In order to improve an observer's ability to discriminate distant depths in the real world a binocular viewing apparatus such as a telestereoscope can be used. As discussed earlier this has the effect of looking at a small model of the scene which brings the stereo image closer to the viewer.

Here are several examples of how these facts can be applied to stereo HTD applications. First when viewing distant objects we can either move or scale the view to bring them closer; also if the objects of interest are already near, we can enhance their stereoscopic depth by overestimating eye separation [Ware95a][Ware95b][Ware98]. In principle any of the geometric manipulation techniques for image fusion could also exaggerate stereoscopic depth by setting their parameters to the opposite extremes.

4.4 View Volume Frame Cancellation

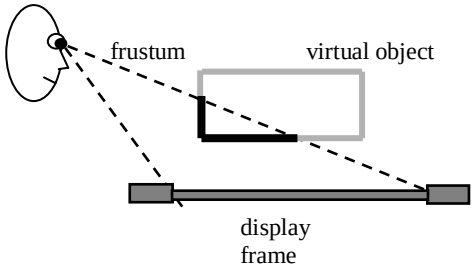


Figure 17: Frame Cancellation.

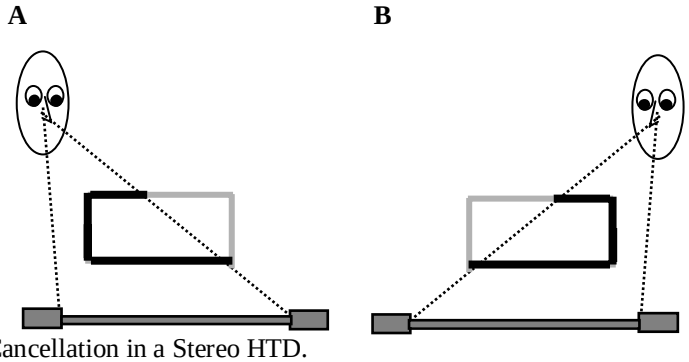


Figure 18: Frame Cancellation in a Stereo HTD.

The volume of the perceived 3D image in a surface 3D display is potentially infinite. The perceived 3D scene can stretch out to infinity beyond the display surface. There are, however, regions in space which the perceived 3D image cannot occupy. These regions are the exterior of the viewing frustums of the screens in the system. This leads to a depth cue conflict between occlusion and stereopsis. Valyus [Valy66] recognized this in stereoscopic cinema and he called the problem 'frame cancellation.' Frame cancellation refers to the fact that a virtual object jutting outside of the view frustum cannot occlude the physical frame of the display as a similarly position physical object would. In Figure 17 the gray portion of the virtual box is outside the view frustum and cannot occlude the far end of the frame. From a psychophysical point of view, the frame appears to occlude an object that is in front of the frame. This is a clear violation of real-world experience. In this case, the visual cue of occlusion is inconsistent with the stereopsis cue. The result is a reduction or loss of the illusion of depth in the perceived 3D image. In a stereoscopic head tracked display, frame cancellation becomes even more obvious as the user moves back and forth (Figure 18).

4.5 Object Positioning

Humans have an innate sense of the position of their limbs. This sense is called proprioception. When manipulating virtual objects, an application can take advantage of this sense by bringing the manipulated geometry within arms reach. A number of techniques have been developed using this idea [Mine97]. Additionally, in a stereo HTD this entails bring the manipulated geometry in front of the

display surface so that the user can physically contact the imagery. Note that this objective matches that of bringing objects close to the user in order to improve stereo depth perception but it can conflict with the need to avoid frame cancellation.

CHAPTER III

FRAMEWORK FOR UNDERSTANDING STEREOSCOPIC APPLICATIONS

1 Introduction

This chapter develops our framework for investigating stereoscopic applications and also makes several new observations. The chapter describes and motivates a classification scheme for parameters in the computer graphics view model and a classification of stereoscopic applications. The chapter illustrates the fundamental difference between hyper/hypo stereoscopy and false eye separation and the final section builds on these results to make several key observations concerning view scaling and perpendicular scaling when used for image fusion control.

2 View Parameter Classes

Maintaining productive view conditions in stereoscopic HTDs requires automated control of the view parameters. There are numerous degrees of freedom which control the view. We are concerned with software-controlled degrees of freedom and we partition these into: view placement, view scale and view optics. View placement refers to the location and orientation of the projection window. The projection window is the virtual representation of the HTDs physical display surface in the virtual world. View placement does not refer to eye point locations because in a HTD the user's head position is a physical parameter controlled by the user and is not under software control. View scale is a single degree of freedom that represents the viewer's size in the world. Note, while it is the position of the projection plane relative to the scene that is important, we must actually manipulate the Platform Coordinate System to alter this relationship.

The view placement parameters are generally sufficient for managing object placement for manipulation and frame cancellation. For the remaining two stereoscopic viewing issues, fusion control

and maximizing stereoscopic depth, researchers often use additional parameters. This next class of parameters are called view optic parameters which includes all other standard computer graphic viewing parameters such field-of-view, modeled eye separation, plus any deliberate distortions. Most image fusion control techniques manipulate view optic parameters. These include underestimated eye separation, image shifting, image scaling, perpendicular scaling and fusibility clipping. To maximize stereoscopic depth overestimated eye separation is often used.

3 False Eye Separation versus Hyper/Hypo Stereoscopy

As mentioned a number of researchers investigate the distortion due to a "discrepancy between eye and camera separation." The literature yields two different distortion results. One group finds a uniform scale distortion while another finds a non-affine collineation. Understanding the difference between these two cases in earlier stereo media is important for understanding how the parameters available to the stereo HTD viewing model relate to stereo image manipulations used in these other stereo media. As discussed in Chapter II, Section 4.2 (Image Accuracy in Stereoscopic Displays) various authors analyze the effect of hyper/hypo stereoscopy in various contexts including telestereoscopes, stereoscopic photography and stereoscopic tele-presence video. They all conclude that hyper/hypo stereoscopy induces a uniform scale transformation on displayed space. Interestingly, MacAdam [McAd54] states that "the effect of camera separation" greater or less than the eye separation does *not* uniformly scale the displayed scene. He explicitly states that the uniform scale assertion is incorrect. When he illustrates the distortion of the "effect of camera separations less than interocular," his diagram (Figure 11, [McAd54]) clearly illustrates a distortion that is at least a non-affine collineation. (By "at least" we mean that the diagram shows a loss of parallelism but that he assumes, i.e. gives no clear proof, that the distortion really preserves straight lines). MacAdam does not explicitly use the term "hyper stereoscopy" but his use of the phrase "effect of camera separations greater/less than interocular" and his assertion that the "uniform scale" hypothesis is false, appears to indicate that he is addressing the hyperstereoscopy method. Lipton [Lipt82], points out this discrepancy between MacAdam and others but offers no explanation.

For telepresence video, Woods [Wood93] investigates the distortion due to the difference between the "eye separation and camera separation" and derives a non-affine collineation. Ware et al.'s [Ware95a] results with respect to non-tracked stereo CRTs are consistent with this too (see Figure 2 [Ware98]).

To our knowledge an explicit comparison pointing out the essential geometric difference between the uniform scale and non-affine collineation literature has not been made. With the exception of Ware's work, prior works have only discussed one or the other distortion.

Ware et al. [Ware95a][Ware95b][Ware97][Ware98] do, however, make it clear that there is a difference between false eye separation (sometimes called "virtual eye separation" in their terminology) and a uniform scale (in particular a scale about the center of the eyes, a "cyclopean scale"). This is apparent because they perform both manipulations in the same VR application. However, Ware et al. do not make it clear whether to equate hyper stereoscopy (i.e. the effect of a telestereoscope) with false eye separation or a uniform scale. These papers are seminal work in the field of dynamic stereo adjustment for VR and this ambiguity does not detract from these works. Ware et al. show both geometric manipulations are useful.

In the introduction to [Ware95a] and [Ware98], Ware et al. state:

> When observing a mountain range at a distance of 30 km stereo vision contributes nothing to our understanding of the spatial shape. However, if we were to create a stereo pair of images with the viewpoint separated by 5 km we will obtain a good enhanced "hyper stereo" image. This technique is, of course, used extensively in stereo photogrammetry. [Ware98]

In the same section five paragraphs later, they begin discussing false eye separation (Ware's "virtual eye separation") and they point out this leads to a non-linear mapping of depth (Figure 2, [Ware98]). While they do not say so, their geometric construction must induce a non-affine collineation. Therefore, their introductory section seems to imply that they are equating hyper stereoscopy with a non-affine collineation transformation and *not* with a uniform scale. Later, however, when discussing the effect of their uniform scale (Ware's "cyclopean scale") they remark:

> It increases the eye separation relative to distant images. A distant large object, such as a mountain will have no useful disparity under normal viewing conditions. However, after a cyclopean scale, the eye separation relative to the object will be increased [Ware98].

This statement would appear to indicate they are equating hyperstereoscopy with a uniform scale.

Then in [Ware95b] they state:

> In stereo photogrammetry and in certain kinds of range finders it is common to create stereo images which have an effective eye separation much larger than any actual eye separation [5]. The reason for this is obvious; human eyes are only placed approximately 6.3 cm apart, which means that stereo information is only a useful depth cue up to 30 meters or so. However, if we can effectively change the eye separation then far more distant objects can be resolved by stereopsis. In viewing a mountain 10 km distant a virtual eye separation of 1 km might be appropriate. [Ware95b]

Later in [Ware95b] they list three methods "to change the effective eye separation by a number of means." Their method (1) is clearly what we've called false eye separation. Method (2) is a uniform scale and method (3) is a description of a telestereoscope. These works tend to refer false eye separation and uniform scaling both as changes to "the effective eye separation" or "virtual eye separation." For this purpose, we believe our terminology that explicitly distinguishes four eye separation values--physical true, physical modeled, virtual true and virtual modeled (see Table 1, pg 22)—is useful for precisely disambiguating the difference. Further, these works do not fully analyze the non-affine collineation transform of false eye separation as compared to the simpler uniform scale. They do, however, examine the non-linear mapping of the z coordinate.

There is one key difference between the geometric constructions used by the uniform scale group, Bercovitz, Drewhurst, Rule, Valyus, and Spottiswoode (see Chapter II, Section 4.2 for specifics), and the constructions used by the non-affine collineation group, Woods, Ware, and MacAdam. Figure 19 illustrates the difference in the geometric constructions using a color scheme that we will use throughout this text. Figure 19A shows a single box in model space. The modeled eye points are shown in light blue. The central thick vertical line is the projection plane. The corners of the box are projected onto the projection plane through black projector rays.

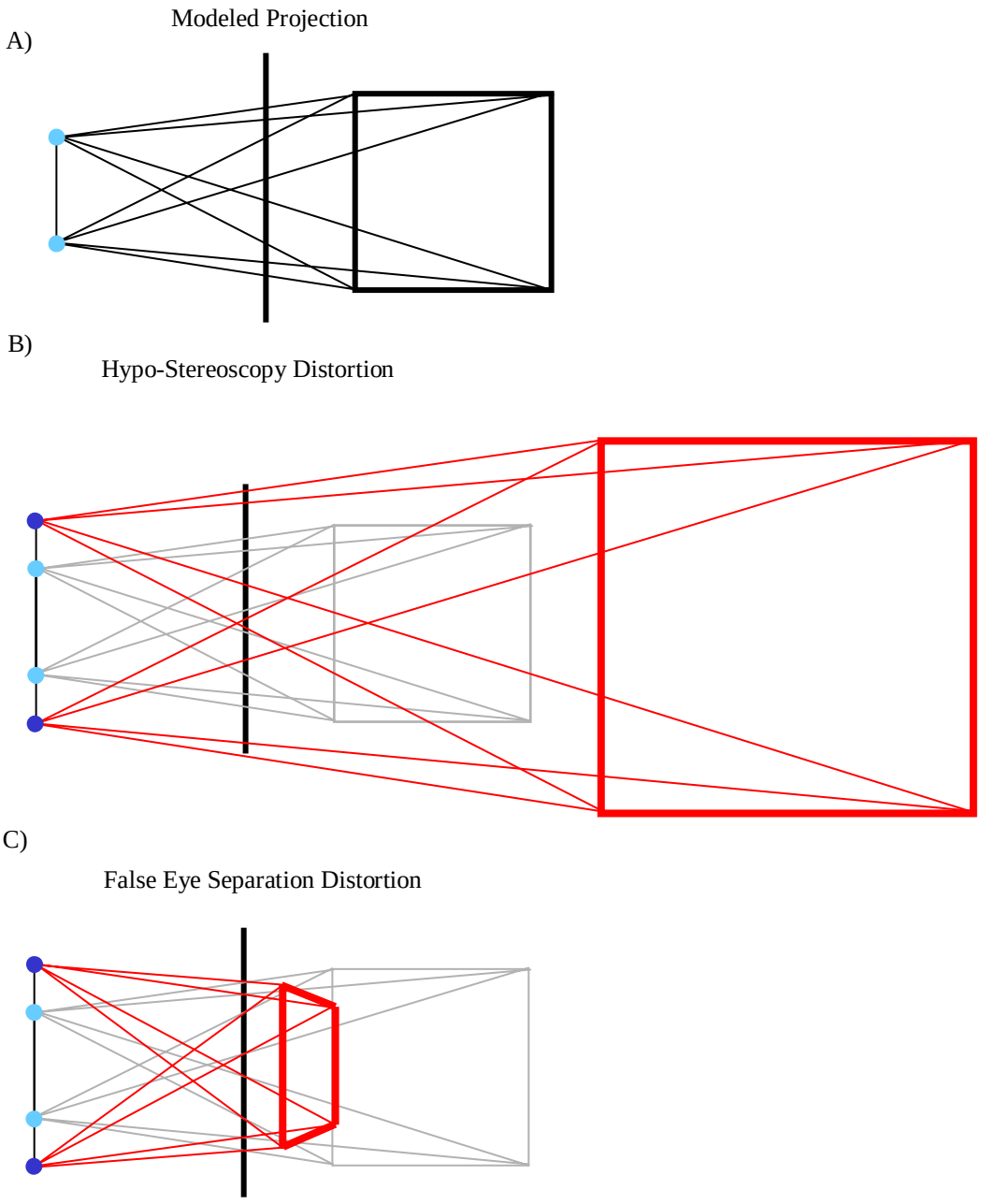


Figure 19: Comparison of Hypo Stereoscopy and False Eye Separation

The uniform scale group assumes that as the human eye separation differs from the camera/modeled eye separation (1) the left eye projector rays move rigidly with the left eye and (2) the right eye projector rays move rigidly with the right eye. Figure 19B shows the result of hypo-stereoscopy. The true eyes are in dark blue. Here both the true eye points and the projector rays are moved rigidly outward. This is equivalent to moving the projected image points rigidly too. Again, the reconstructing projectors are shown in red. The result is that the box in displayed space is a larger, uniformly scaled version of the box in modeled space.

More formally define a *projection* to be a triple: a center of projection (COP), an image on a finite projection surface, a set of projector rays connecting the COP to each image point on the projection surface. A *stereo projection* is simply a pair of projections. Further, let there be a modeled stereo projection and a true stereo projection. For stereo photography and video in the modeled stereo projection, the COPs are the centers of the camera lens and the images are the camera CCDs or camera film plates. For VR in the modeled stereo projection the COPs are the modeled locations of the entrance pupil of the user's eyes and the images are the modeled location of the display screens. Define a modeled and true stereo projection to be *projector-similar* when the COP and projectors of the modeled stereo view can be mapped to the COP and projectors of the true stereo view by a positive uniform scale. Define a modeled and true stereo projection to be *image-similar* when the COPs and images of the modeled stereo view can be mapped to the COP and images of the true stereo view by a positive uniform scale. It should be clear that if the modeled stereo projection and the true stereo projection are projector-similar then the displayed stereo image is a uniform scaled version of the modeled stereo image. This is because the display stereo image points are defined by the intersection of corresponding true projectors and these intersection points uniformly scale along with the stereo projection geometry. (This is the basic argument by Valyus and Helmholz for telestereoscopes). Next, image-similar is sufficient but not necessary for projector-similarity. Figure 19B shows projector-similarity but not image-similarity. To have image-similarity we'd need to add a different true projection plane (instead of the single one shared by the true and modeled projections) with the true projection plane at some scaled up distance from the illustrated shared projection plane.

The uniform scale group's projector-similarity image assumption makes sense for tele-stereoscopes. Valysus's Figure 24 satisfies projector-similarity [Valy66] and Helmoltz's Figure 57 [Helm62] would satisfy projector-similarity. For more complex optical systems, however, the precise optics of the system determines whether it is appropriate to transform the projected images/rays with the COPs. This rigid relation between model and true projectors and COP is satisfied for non-tracked tele-presence video if each eye views a separate display through a separate lens and either (1) the mechanics of the viewing system adjusts lens and the display positions to match the relative relation of the camera lens and CCD or (2) the mechanics of the camera adjusts the camera lens and CCD to match the relative relation of the viewing lens and display. Bercovitz [Berc98] clearly makes this assumption in his Figure 2 which satisfies image-similarity. Rule's figure 2 [Rule41a] for stereo photography also satisfies image-similarity.

In contrast, the non-affine collineation group assumes that the left and right eye projected image planes are not adjusted when the eye points are moved. So when comparing the modeled stereo projection to the true stereo projection we find the two are not projector-similar. Figure 19C illustrates underestimated false eye separation. Again, the modeled COPs are in light blue while true COPs are shown in dark blue. Here the modeled eye separation underestimates the true eye separation by one-half. False eye separation does not move the left and right projected image points. This is clear in Figure 19C since the reconstructing red projector rays pass through the same points on the screen as the original projectors shown in faded gray. The result is that the box corners are distorted into a trapezoid. This red trapezoid shape is the modeled box in display space.

The non-affine collineation group's lack of projector-similarity is easy to understand for the cited VR cases but a bit harder to understand for the tele-presence video and stereo cinema. For a stationary stereoscopic VR display (be it a tracked HTD or an untracked display), manipulating the modeled COPs relative to a static screen makes an intuitive sense. Again this is illustrated in Figure 19C. This is geometrically what Hodges et al. [Hodg92] mean when manipulating "eye separation". Ware et al. perform similar manipulations. With respect to tele-presence video Woods et al. [Wood93] assume both eyes look at a single screen (Figure 3 [Wood93]) and that the image from each CCD is map directly to the full width of the screen with the CCD centers at the screen's center. In this case, we move the two cameras (CCDs

and lenses) farther apart, the CCD image centers still map to the screen center. In order to preserve projector-similarity of the model and true stereo projections, Woods would have to either (1) re-adjust the CCD position relative to the lens as the cameras are moved further apart or (2) adjust the separation of the left and right display image via an image shift. However, Woods et al. do *not* do this. This leads to the lack of projector-similarity which results in a non-affine collineation. Note, that choice (1) would be mechanically difficult while choice (2) would lead to regions on the sides of the CRT with only left image and no right eye image and visa versa. More difficult to understand, however, is MacAdam's seeming assertion that hyperstereoscopy is not a uniform scale. Figure 11 in [MacA54] is equivalent to our Figure 19C. The reason why he considers this the proper geometric construction for illustrating the "effects of camera separations greater [or less] than the interocular" in stereoscopic cinema unfortunately remains unclear given the results of Rule, Valyus, etc. Stereo photography and cinema involve multiple geometric considerations include camera geometry, film development and printing, film projector geometry and view to screen geometry. Unfortunately, it is unclear what are all the assumptions concerning these geometries made by the various authors. Clearly, however, there must be some unstated difference in assumptions.

While both hypo-stereoscopy and underestimated eye separation involve an "underestimation," hypo-stereoscopy enlarges the displayed space while underestimated eye separation shrinks displayed space. If we were to illustrate the opposite case, we'd find that while both hyper-stereoscopy and overestimated eye separation involve an "overestimation", hyper-stereoscopy shrinks displayed space while overestimated eye separation expands displayed space. This may seem counter-intuitive, but it simply is the way the geometric constructions behave!

To conclude, hyper/hypo stereoscopy, defined as the effect of a telestereoscope, is simulated in VR by uniformly scaling the Platform Coordinate System matrix roughly around the center of the eye points. The exact center location would vary a bit depending on the optics of the simulated telestereoscope, but Ware et al.'s cyclopean scale, a scale about the exact center of the modeled eye points, was shown to be very effective in VR. The exact scale center is probably of little importance. Since, the view scale factor comes for free in the VR coordinate system model, we will henceforth simply refer to "view scale" manipulations instead of using the terms hyper/hypo stereoscopy. Section 5.2 of this chapter will illustrate cases where

false eye separation can be preferable to view scaling for fusion control despite having a more complex 3D transformation description.

4 Application classes

Interactive applications targeted for stereoscopic displays can be partitioned into several classes based on image fusion control issues. Fusion problems with stereo images grow as the modeled objects move farther away from the projection plane. The simplest solution is to simply move the virtual view to bring the modeled objects close to the projection plane. This only works, however, if the modeled scene extends over a small range of depth. This is the case for a stereoscopically simple environment with a few objects that are close together. We must be careful, however, in defining the term "stereoscopically simple." A scene consisting of a small 10 cm ball viewed at a 1:1 scale is clearly stereoscopically simple. But if we zoom to bring the scale to 1:1000, the ball will appear 10 m wide and now covers a wide depth range.

This raises the question of whether or not a scene and the application encourage or even allow zooming. This depends on the level of geometric detail in the model. For example take a simple floor planning application showing one floor of a virtual house populated with furniture. Assume the user views this on a virtual workbench. We can embed the floor in the projection plane and scale the model so the scene appears like a dollhouse. From this perspective the user would experiment with arranging furniture and wall layout. In this application, the user is not concerned with modeling the surface geometry of walls or furniture down to the centimeter, so there is no need to model the objects at this detail. Therefore, zooming down to a wall will reveal no new useful geometric information. In this application zooming is unnecessary. To avoid fusion problems in this application, we can statically place the virtual scene close to the projection plane.

We now more carefully define a *stereoscopically simple application* as an application whose scene covers a small depth range and which contains geometric detail covering a small range of scales. For a *stereoscopically simple environment,* only a minimal amount of extra work is needed to account for stereoscopic display issues [Lipt93]. The scale factor can be set to a fixed value and the virtual object

placed partially in front and partially behind the projection plane. For a single object, user initiated translation and rotations can generally be allowed without further consideration. For scenes such as the simple floor planning application, the application need only limit the orientation of the scene to make the floor flush with the screen.

In general, however, 3D applications are not stereoscopically simple. For example, in a more general-purpose architectural design application, one might model the pipes and wires running through the walls. Viewing these details immediately implies zooming in on the environment. This raises image fusion issues since the virtual scene will now cover a larger range of depth. Such stereoscopically non-trivial applications can be further partition into two classes: *locally shallow applications* and *deep applications.*

A locally shallow application contains enough geometric detail to require zooming, but when viewing any particular detailed region the perceived geometry covers only a small depth range. For any given detailed view, these scenes must be dynamically brought to and aligned with the projection plane. Since the geometry is locally shallow this maneuver is sufficient. For these types of applications, the key issue is handling the transitions between different local views of the locally shallow geometry. This can be done by manipulating the view parameters: location, orientation and scale. An example of this is a whole-planet terrain visualization that maintains an overhead or map-like view of the terrain. Chapter IV describes just such a travel technique that uses a variety of automated parameter adjustments in order to maintain good stereoscopic viewing conditions.

Unfortunately, many applications are not locally shallow either. We call these applications *stereoscopically deep applications.* For example, the whole-planet terrain application becomes a deep application as soon as the user uses an ego-centric or "flying" travel technique. Here the user looks over the horizon and the perceived scene can stretch out for miles. As another example, even if an application maintains a top-down view of the terrain, it might make the planet semi-transparent. Again, the perceived scene can stretch arbitrarily far beyond the projection plane as the user peers into the center of the earth. For these deep environments, the image fusion techniques that manipulate view optic parameters are needed when fusion problems arise.

5 Comparing Fusion Control Techniques

Chapter II, Section 4.1.2 listed eight software techniques for controlling image fusibility: underestimated eye separation, image scaling, image shifting, fusibility clipping, perpendicular scaling, asymmetric-asymptotic transform, view placement and view scaling. This section makes a number of key observations and generalizations and develops several useful classifications of the view optic techniques. Note, that while this section focuses on image fusion, in principle any of the geometric manipulation methods can be used for exaggerating stereoscopic depth by setting the parameters to the opposite extreme. Therefore certain aspects of this section also apply to view optic distortions used for exaggerating depth. These aspects will be noted in the text.

5.1 General Considerations

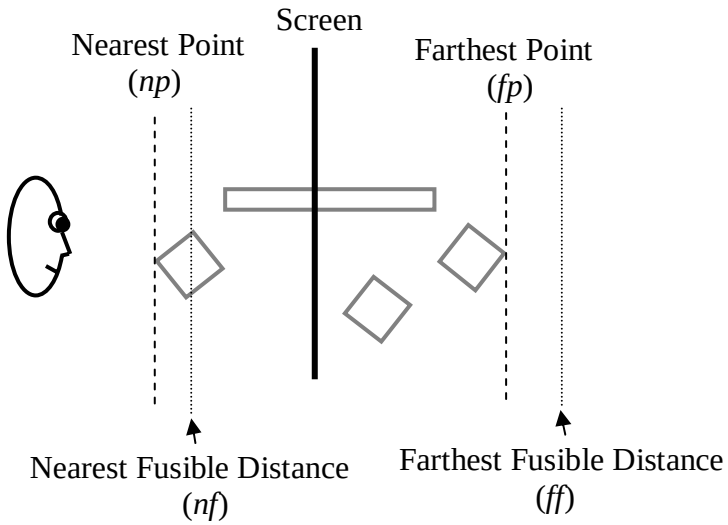


Figure 20: Four Important Distances in Fusion Control--This diagram corresponds to Case 3 of the fusion control algorithm in which far geometry is all comfortably fusible but near geometry is not all comfortably fusible.

Except for fusibility clipping all the fusion techniques geometrically manipulate the scene. Fusibility clipping clips out geometric data outside the fusible range. Since throwing out of data is often not

tolerable, the geometric manipulation techniques are needed. A generic implementation of a geometric manipulation technique is as follows:

```
Algorithm 1:

fp = the farthest geometric point on any virtual geometry
ff = the farthest fusible point
np = the nearest geometric point on any virtual geometry
nf = the nearest fusible point
if |fp| > |ff|
   if np > nf
         /* CASE 1 */
         map np to nf
         map fp to ff
   else
         /* CASE 2 */
         map fp to ff
else
   if np > nf
         /* CASE 3 */
         map np to nf
```

As shown in Figure 20, *np* and *fp* are the depths (z-coordinate in Projection Plane coordinates) of the nearest and farthest geometric point on any virtual geometry while *nf* and *ff* are nearest and farthest comfortably fusible depths. This illustrates the four important distances. The figure corresponds to Case 3 of the algorithm. Note three things about the algorithm. First, the above algorithm lists Case 1 separately because many geometric techniques lack the degrees of freedom needed to simultaneously map *np* to *nf* and *fp* to *ff*. This means either multiple techniques are needed or the most conservative value of a technique's parameter must be chosen. Second, Algorithm 1 only compresses the scene if it is originally non-fusible. Alternatively, unconditionally mapping [*np,fp*] to [*nf,ff*] would occasionally exaggerate scene depth. While this can be useful [Ware98], we do not consider exaggerated stereo here. Finally, some geometric techniques allow $fp=\infty$ which maps all of far space into the fusible range

Algorithm 1 has two characteristics that are open to different implementations. First, *np* and *fp* can be either dynamically calculated based on the current scene [Ware98] or they can be preset to assumed maximum values. We call this characteristic "scene-depth sensitivity." An algorithm is either scene-depth sensitive or scene-depth insensitive. Second, *nf* and *ff* can also be either dynamically calculated based on

the current head position or preset to fixed values. (Recall, the fusible depth range varies with head position). To distinguish between options of this second consideration, we say a fusion technique can be implemented for either a "resting-head" or an "active-head." A resting-head implementation determines a fixed fusible range, [*nf, ff*], based on the distance between the screen and a fixed head position. When the scene is unchanging, [*np,fp*] are constant and a resting-head implementation then holds the fusion technique's software controllable fusion parameters at fixed values. As long as the user does not move closer than the resting distance the scene remains comfortably fusible. This is analogous to Lipton's [Lipt82,196] recommendation for stereoscopic cinema where he computes comfortable depth ranges based on the position of the closest seats in the theater. In contrast, an active-head implementation continually recalculates the fusible range, [*nf, ff*], based on the current head position. This implies the algorithm continually adjusts the technique's fusion parameters as the user moves her head. The advantage of an active-head implementation is that theoretically the user can move through a larger distance range and maintain comfortable image fusion. The disadvantage is that the active-head implementation will dynamically change the technique's 3D distortion as the user moves her head. This may lead to a rubbery appearance to the virtual world as the user moves forward and back. The literature does not appear to make any direct comparisons of resting-head versus active-head implementations. Intuitively, the choice might vary with the application requirements and user preferences and the extent of forward/backward head movement associated with the particular application and the physical display environment.

Most of the geometric manipulation techniques distort the perceived scene. We define a distortion to be any geometric manipulation that people do not experience in the real world. Manipulating viewer location and orientation are not distortions. All remaining geometric manipulation techniques distort the perceived space in some way.

When comparing these distortions, we identify three major considerations: (1) whether the distortion itself is static or dynamic with respect to head motion (as will be discussed this is a separate issue from whether the implementation is based on an active or resting head position); (2) what the general properties of the distortion are; (3) whether the distortion preserves in-screen geometry; and (4) how many software-

controllable degrees of freedom the distortion has. Note, these considerations are relevant whether the view optic technique is used for either image fusion or stereo depth exaggeration.

First, we distinguish static distortions from dynamic distortions. A static distortion is one that does not change with head position. A dynamic distortion is one that changes with head position. Dynamic distortion qualities are only noticeable when the user moves his head. As we will discuss, dynamic artifacts occur in several techniques. The analytic distortion equation is dependent on head position and the displayed virtual object's shape will continually change in an unrealistic manner. There is an important relationship between static versus dynamic distortions and resting-head versus active-head implementations. A dynamic distortion lends a rubbery appearance to a resting-head implementation, which would otherwise appear rigid. A dynamic distortion also creates problems in active-head implementations. The rubbery effect induced by an active-head implementation only occurs due to forward/backward head motion. Therefore, a dynamic distortion that changes with lateral head-motion will add a qualitatively new dynamic distortion component to an active-head implementation. Possibly a technique's distortion may only change under perpendicular head-motion. In that case, the dynamic distortion probably won't add to the rubbery quality much beyond that already present. However, in an active-head implementation, dynamic distortions will complicate the programming. These difficulties indicate that an ideal technique's distortion should be static under all head motion.

The second distortion consideration is what are the general, static properties of the distortion. These considerations include:

(1) aspect ratios – Does the distortion preserve aspect ratios or does it non-uniformly scale the displayed space? If a non-uniform scale occurs what axes in projection plane coordinates are preserved, if any?

(2) angles – Does the distortion preserve angles? Does it shear perceive space? Are angles in certain planes such as those parallel to the projection plane preserved?

(3) parallelism – Does the distortion map parallel lines to parallel lines?

(4) collinearity – Does the distortion map straight lines to straight lines?

There is a third distortion characteristic deserving special attention. Researchers have found it useful to use the screen surface as a natural work surface to limit the degrees of freedom over which the user must operate [Cutl97]. For example, imagine a user viewing a city block with the streets flush with the screen and the buildings sticking out of the screen. The user might lay route points on the street between the buildings. Here the screen reduces the degrees of freedom through which the user must physically move his controller or hand. If buildings were to jut too far above the screen to be comfortably fusible, the application might translate the view and push the scene back into screen. This pushes down the street level which is the plane with which the user wants to work. The screen is no longer a natural working surface for the routing task. This illustrates that for tasks that utilize the screen as a physical working plane, it is advantageous to keep in-screen geometry in the screen. Additionally, we should consider that a scale along either the X or Y screen axes alters the 3D object's screen footprint. This could make interaction more difficult. In the building example, it would crowd the buildings closer together making laying the route points more difficult.

The fourth distortion issue is the degrees of freedom available. Many geometric techniques lack enough software controllable degrees of freedom and cannot simultaneously map *np* to *nf* and *fp* to *ff*. For this reason Algorithm I separates the case where both near and far geometry are not comfortably fusible (Case I). In such a case, either multiple techniques must be used or the most conservative value of a single technique's controllable parameter must be chosen.

Assessing these four distortion issues is easy for perpendicular scaling, viewer scaling and view placement because the transforms are directly given. Assessing false eye separation, image scaling, image shifting and the asymmetric/asymptotic transform is more complex. The first three methods work indirectly on the 3D scene through the 2D projected images or viewing parameters. As a contrast to linear techniques like perpendicular scaling, Williams and Parrish [Will90] give the asymmetric/asymptotic transform matrix. The presented matrix also embeds the standard world-to-screen transform components. Unfortunately, a matrix describing only how displayed 3D space is affected independent of the world-to-screen component, is not given. No verbal or pictorial description of how the matrix affects 3-space is given beyond describing the asymptotic depth behavior. Clearly, however, the technique must necessarily

be a non-affine collineation given the asymptotic depth behavior and its matrix representation. Finally, the given matrix does not have the degrees of freedom needed to support head tracking.

5.2 Utility of Non-Affine Distortions

Given that false eye separation induces a geometrically more complex distortion than perpendicular scaling and view scaling even in non-head-tracked displays, why would researchers commonly use this technique for image fusion? Answering this question is important. If false eye separation is blatantly inferior to view scaling and perpendicular scaling, further investigating false eye separation in order to account for head-tracking is unwarranted. Additionally, view scaling and perpendicular scaling preserve a large number of 3D geometric properties. If this alone is the rubric by which we should rank image fusion techniques, further investigating of image scaling and image shifting is also of little value since their distortions could hardly be any simpler than view or perpendicular scaling. This section will illustrate, however, that in a number of conditions false eye separation is superior to view scaling and perpendicular scaling.

A demonstration helps illustrate a variety of these cases where false eye separation is better than the scaling methods. We used a calibrated, desktop VR setup with a 24 inch monitor and a Polhemus Fastrak tracker. Screen resolution was 1024x768 in stereo with liquid crystal shutter glasses. A comfortable sitting distance was 75 cm. Eye separation is 6.5 cm. Figure 21A (pg 68) shows the view of the scene. At the bottom is a matrix of 5 cm cubes. The closest row just rests on the view plane. In the far distance, 1000 meters away, are a set of large, 100 meter cubes. Viewed on a desktop stereoscopic HTD, the distant geometry is very difficult to fuse. The measured screen parallax of these cubes is 6 cm. The 1000 meter depth is certainly well beyond the 45 cm depth yielded William and Parishes 60% limit. 45 cm may seem amazingly small, but it is consistent with other heuristics for average users. For instance by Southard's equations and Valyus's convergence limit we get 35 cm. Akka [Akka99] suggests a maximum screen parallax of 3.5% of screen width. On our 48 cm wide screen viewed at 75cm by 6.5 cm separated eyes, this limit yields a maximum depth of 26 cm.

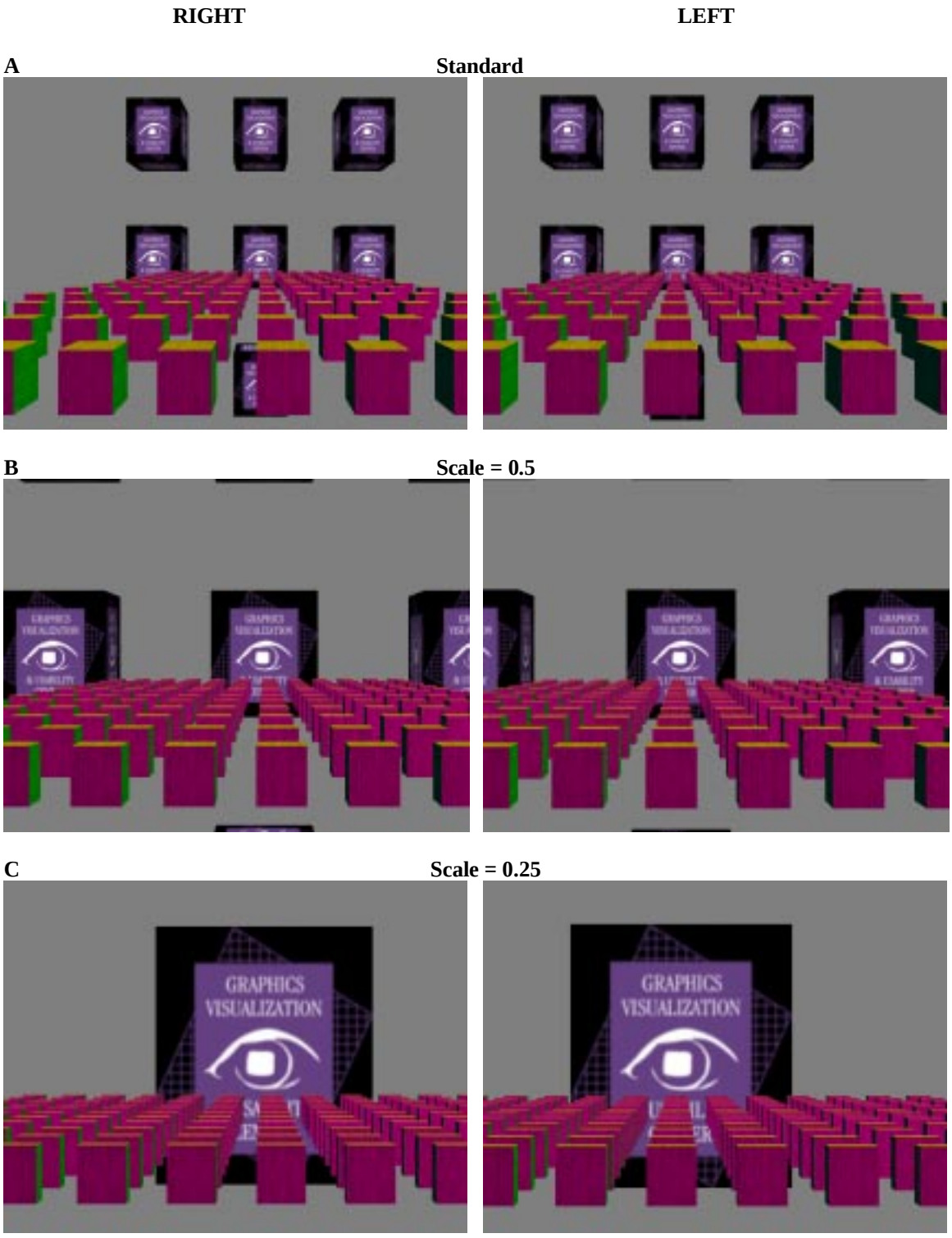


Figure 21: Perpendicular Scale Effect (figure continues on next page)

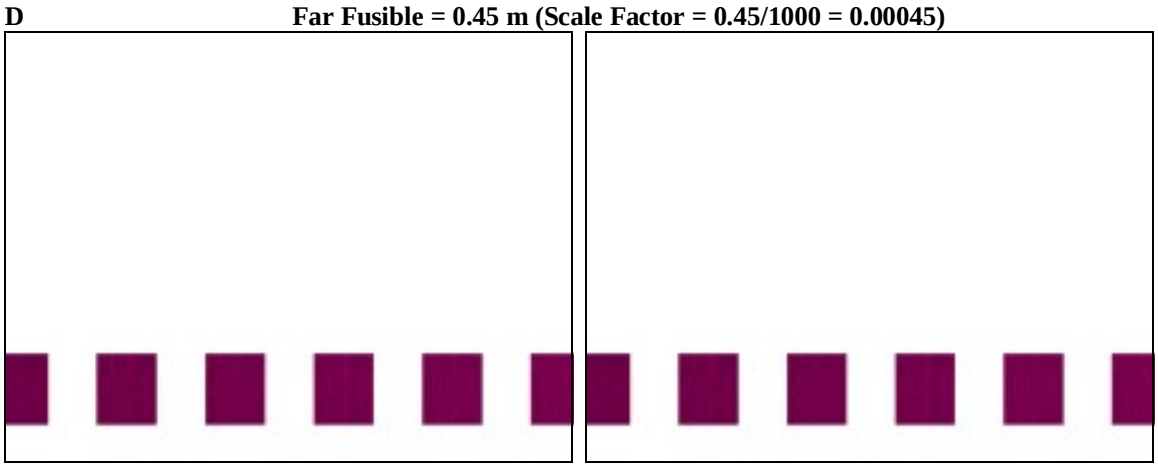


Figure 21: Perpendicular Scale Effect

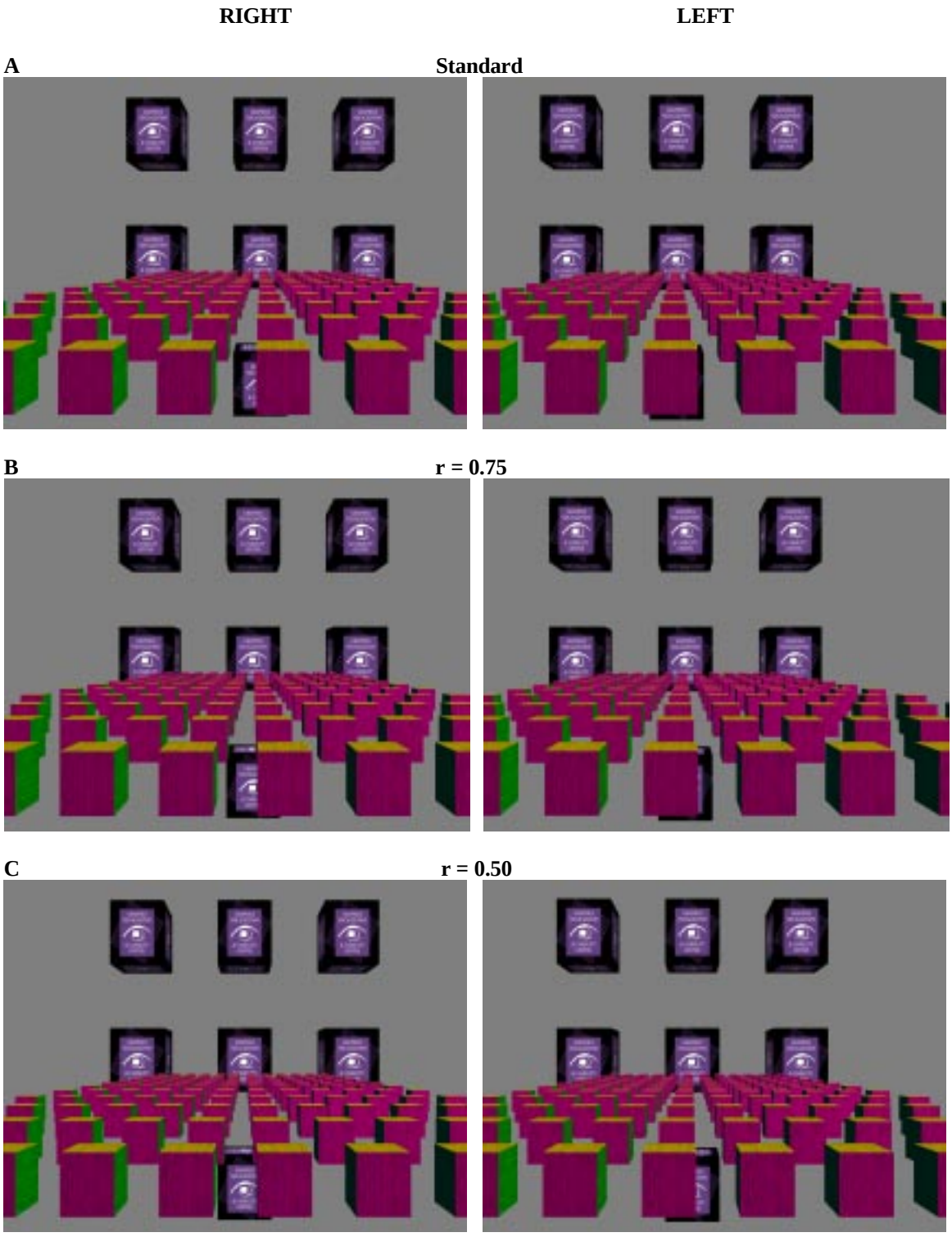


Figure 22: False Eye Separation Effect (figure continues on next page)

D **Far Fusible = 0.45**

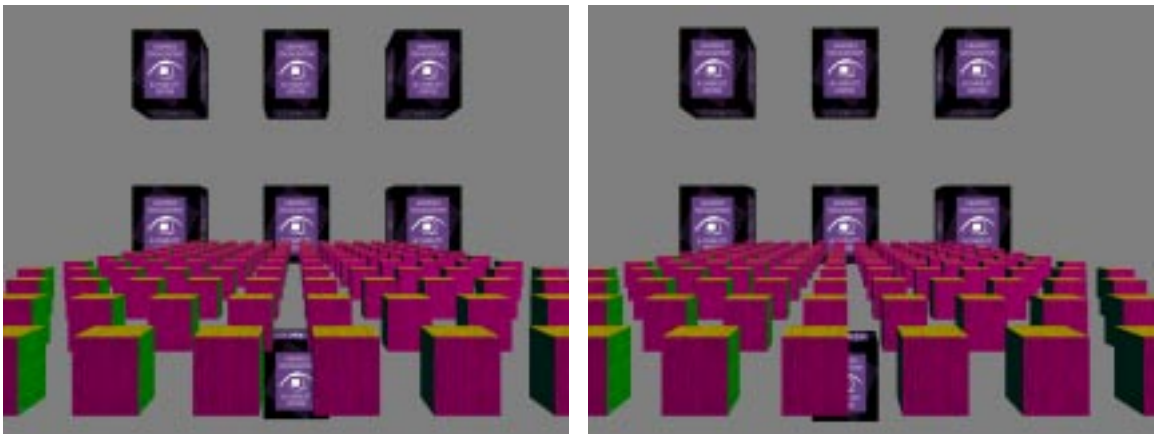


Figure 22: False Eye Separation Effect

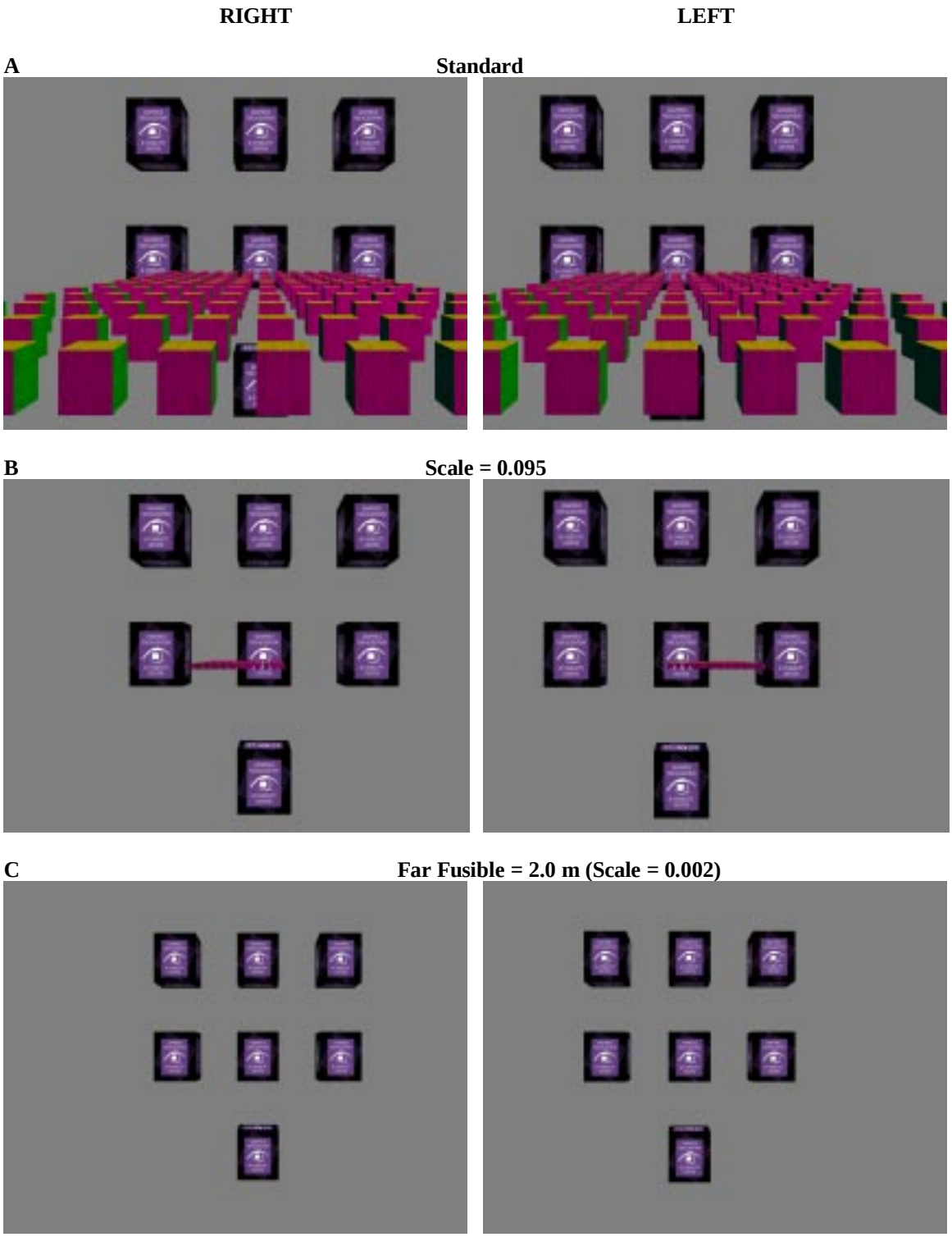


Figure 23: View Scaling Effect (figure continues on next page)

D **Far Fusible = 0.45 m (Scale = 0.00045)**

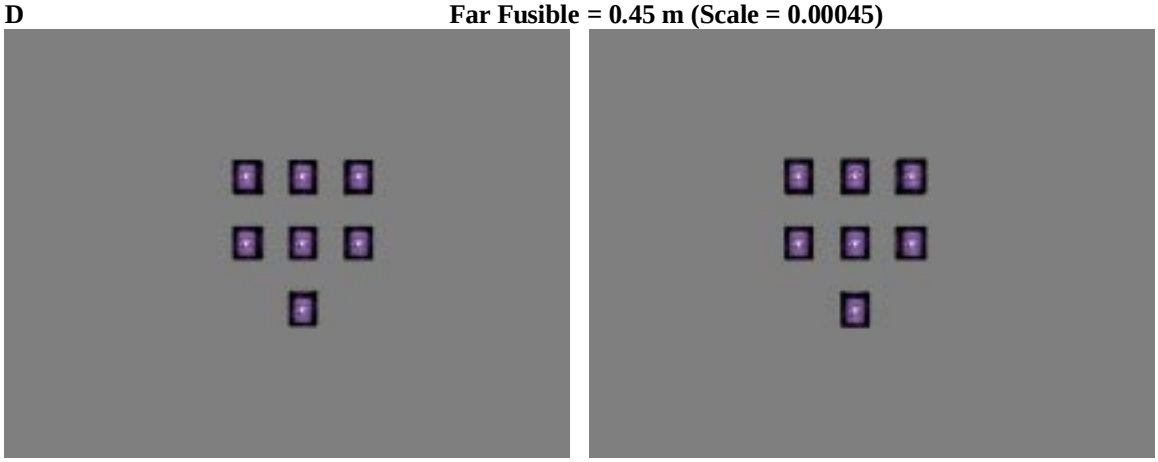


Figure 23: View Scaling Effect

Figure 21 illustrates perpendicular scaling. The side by side images are stereo pairs. Note, the right eye image appears on the left and the left eye image on the right. This allows viewing of the 3D image by crossing one's eyes while looking at a given image pair. (Learning to see such 3D images can take practice). Figure 21A shows the initial view. Figures B through D illustrates perpendicular scaling for increasing scale factors as we approach the scale factor needed to map the 1000 m distant blocks to 45 cm. In B through D, the visual angle subtended by the distant object changes drastically when compared to the original image (Figure 21A). Additionally, the change in aspect ratios of the nearby cubes are extremely noticeable, especially when viewed on the actual stereo HTD. By the time we reach the 45 cm / 1000 m scale factor (Figure 21D) the image grossly differs from the original. To make distant geometry fusible, perpendicular scaling substantially changes the overall monoscopic image properties. For an application that simulates a walk-through or fly-through of a natural environment these changes are not acceptable. For more abstract data visualization applications, they are also troublesome. In contrast, the underestimated eye separation's change to the monoscopic image properties is rather negligible because it only moves the eye points closer together. For the small eye separation adjustment needed to make this scene fusible, the change in the right eye image is barely monoscopically noticeable. Figure 22 illustrates the results of false eye separation for decreasing modeled/true eye separation ratios until reaching the mapping of 1000 m to 45 cm (Figure 22D). r is simply the ratio of modeled to true eye separation. Note, reducing eye separation causes the distant blocks' images to move to the right in the left eye images (right column) and to the left in the right eye images (left column).

Figure 23 illustrates the effect of view scaling. In Figure 23B the view is scaled about the view plane's center to a point just before a change in horizontal parallax of the distant cubes is first measurable on the screen. The scale factor is 0.095. (Note, the near blocks are roughly shrunk by $1/10^{th}$). As we approach the 45 cm / 1000 m scale factor in Figure 23 C and D the near geometry completely disappears! If we instead scale the view about the center of the eye axis, then the near blocks shrink and move far above the projection plane and very close to the eye. This also can create near fusion problems.

Additionally, in many applications, dynamically adjusting view scale is not appropriate. This makes a scene-depth sensitive implementation impossible. An example is an application simulating driving or

walking through a natural environment. A scene-sensitive view scaling would cause the world to dynamically grow and shrink as the user moves through it and the near and far points change. This is clearly at odds with everyday experience, which the application aims to convey. Using false eye separation to control fusion wouldn't create the uniform scaling effect.

Scene-insensitive view scaling would use a constant scale, but fusion concerns could force us to shrink the world down to a fixed size that is unnatural for an application. Imagine an application, a game perhaps, where we walk over a planet surface. Depending on the screen size we'd want to pick some view scale factor. For aesthetics on a desktop display we might want human characters to appear 6 inches tall when they cross the plane of the screen. This implies a view scale factor of 6/72 to make 6 ft characters an appropriate size. However, we might have distant clouds (like the large distant blocks in Figure 23) and the scale factor needed to make these clouds fusible would conflict with the aesthetically chosen scale factor. In fact, as with the cubes in Figure 23, the fusion based scale factor could make the virtual humans too small to see. Using false eye separation, however, would not interfere with the aesthetics-based scale factor in this application. Abstractly, let *S* be the diameter of a sphere bounding the scene. Let *F* be the size of the smallest significant geometric feature. To make the scene comfortably fusible, we must scale by *ff*/*S*. (Recall, *ff* is the farthest comfortably fusible point). View scaling is problematic, if at its scaled size of *ff/S* * *F* the smallest feature is too small to manipulate or see.

Finally if view scale is under user control as a zoom factor, then scale cannot also be set by the fusion control algorithm. If the user zooms in to see details, distant objects may exceed comfortable fusion limits. This could be managed with false eye separation.

In summary, perpendicular scaling, while having a simpler 3D transformation, appears to have little actual value. We were motivated to investigate perpendicular scaling due to its simple 3D transformation. No literature appears to recommend this technique per-se; William and Parrish's [Will90] figure 6 hints at the perpendicular scale method but their text description actually describes a non-linear mapping which must be at least a non-affine collineation. Next, view scaling is useful for fusion control under certain circumstances. Ware et al. [Ware97] is a prime example where view scale is effectively controlled by the user as a zoom factor. In this particular application, dynamic scaling was quite acceptable as it was a

scientific visualization application and having a dynamically scaling world didn't violate any real world experience that the application was trying to portray. Note, however, that the view scale was varied as a user controlled zoom so it is theoretically possible that at certain scales the far point could exceed the comfortable fusion limit. It appears that in this particular application this did not occur. Above we illustrated certain geometric conditions in which view scaling can create problems that false eye separation does not create. These examples should provide some geometric guidance in choosing between these two methods.

Overall this chapter illustrated the fundamental difference between hyper/hypo stereoscopy and false eye separation and illustrated when false eye separation may be geometrically preferable to view scaling. This chapter described several classes of applications which require view parameter adjustments of increasing distortion complexity in order to control image fusion. These range from non-distorting viewer placement, to view scaling and finally to a non-affine technique, false eye separation. The next chapter begins the more detailed investigation by describing how to manage orbital viewing of a locally shallow application by automating adjustments to view placement and view scale.

CHAPTER IV

EXO-CENTRIC TRAVEL IN A LOCALLY SHALLOW ENVIRONMENT

1 Overview

This chapter discusses a travel technique for a global terrain database on the virtual workbench. This work is also presented in [WarZ99b]. The technique maintains a map-like point of view of the terrain. Since the underlying terrain system [Lind96] can model terrain at resolutions varying from 8km down to 1mm, this environment is not stereoscopically simple. However, when viewed from a map-like point of view, the global terrain database is a locally shallow environment and stereoscopic display issues can be handled by adjustments to the view placement and view scale parameters.

2 Previous work

Much literature exists concerning navigating large information spaces such as [Leib94] [Bed94][Bart95][Furn95]. However, this body of work concerns 2D GUI interfaces. 3D work for large spaces, exemplified by the April issue of Presence: Teleoperators and Virtual Environments [Barf98], has focused on psychological studies, not specific techniques.

Many general travel techniques for virtual environments have also been developed, studied and compared. Travel techniques can be partitioned into several classes. In one class of techniques the user drives or flies herself or a vehicle through the environment. We refer to this class as ego-centric travel [Bow97]. [Ware90] calls this class the flying metaphor. Bowman et al. [Bow97] discuss a number of these ego-centric methods. In a second class, the user appears to grab and manipulate the environment itself. We refer to this class as exo-centric or orbital travel. [Ware90] also calls this class the scene-in-hand metaphor. (Note this dissertation uses a slightly different terminology than in [WarZ99b] which presented the work of this section. In [WarZ99b], we used the terms first-person and third-person navigation instead of ego-centric and exo-centric travel. Increasingly, however, researchers [Bow99]

define “navigation” as a broader process including a human cognitive component called way-finding and a computer interface component, the travel technique. In this work we use this newer terminology and only discuss travel techniques. Also increasingly the terms “third person viewpoint” means an interface that shows the user’s avatar by placing the virtual camera slightly behind the avatar, while “first person viewpoint” implies the virtual camera provides the view directly from the avatar’s eye point. This is contrary to [WarZ99b] where we used the terms first and third person travel as synonyms for ego-centric and exo-centric travel. In this thesis we no longer use the terms first and third person as such synonyms.)

Parallel to these classifications of travel techniques are those that focus on the input devices themselves [Zhai95]. The input device can be isotonic, elastic, or isometric. Isotonic devices offer no resistance; examples are a standard mouse or a “flying mouse” moved freely through air. Elastic devices are those such as common joysticks. Isometric devices do not move at all and register the applied force of the user’s hand. Another issue is how device input is mapped to object movement. Zero-order techniques map device input directly to object position while first-order techniques map device input to the velocity of the controlled object. Since we are aimed for a map-like interface to our terrain application, we will strive for zero-order, isotonic control.

A number of specific techniques are relevant to our objectives. Chen et al. [Chen88] describe a mouse interface for rotating 3D objects on a regular monitor. Their ‘virtual sphere’ technique works as if the 3D object is encased in an invisible sphere which is grabbed and rolled with the mouse pointer. Mackinlay et al. [Mack90] describe a technique for both zooming and moving through a more general environment by grabbing and moving around individual objects. Hix et al. [Hix95] describe pre-screen projection. It is used for map navigation, but the technique is controlled by head movements which conflict with the use of head-tracking and stereoscopic display on a virtual workbench. In Netra [Gobl95], the user manipulates the orientation of a virtual MRI of a patient’s head with a tracked doll’s head. While similarly manipulating a planet would work at small scales this method would not work well at large scales. More importantly for our goals, Netra uses neither head-tracking nor stereoscopic display. Next the WIM (Worlds In Miniature) [Stoa95] technique displays a small copy of the world which the user holds. The user can both manipulate objects and travel using the WIM model. We are unaware, however, of WIM

extensions to the workbench that incorporate zooming. Furthermore, even with such an extension, a basic travel method addressing the challenges of Chapter II would still be needed. Johnson [John95] explores several methods of panning a 2D map with a mouse. We use some of his results in Section 3.3 of this chapter.

Stereo HTDs have been studied for several years [McKe92][Deer92][Ware93] and more recently, the virtual workbench has gained much attention [Froh95][Krug95][Obey96][Rose97][Serr95][Guan98] [Durb98]. However, this body of work discusses applications with relatively small scale models such as human organs, cars, molecules and local terrain. More importantly, these papers do not address in detail the challenges listed in Chapter II, Section 4.

There is a small set of work which addresses stereoscopic terrain display and navigation. Some of the earliest work [Vero90] avoids diplopia in the following manner. The authors first fix the distance between the viewer and display surface so that objects at infinite distances beyond the projection plane remain fusible. Next they fix the near clipping plane to avoid rendering objects which are too close to the viewer to be fusible. However, they do not use head-tracking nor 6 DOF control devices. Also they do not address scale issues nor whole-planet terrain.

Ware and co-workers refine the display of non-head-tracked, stereoscopic terrain [Ware95a][Ware95b][Ware98]. Their methods are based on the premise that correct stereo is not always the best choice. They first scale the world about the central eye point to bring the nearest visible terrain point to the screen. Then they adjust the modeled eye separation to enhance stereoscopic depth based on the nearest and farthest sampled pixel in the Z buffer. Later work [Ware97] does address head-tracked systems making a slight modification to the scale step and leaving out the eye separation adjustment altogether.

More recently Durbin et al. reported on a command and control application called Dragon on the virtual workbench [Durb98]. Dragon presents terrain and military icons on the virtual workbench. The author designed and implemented the head-tracked stereo software and helped develop the navigation system for Dragon. Navigation in Dragon addresses some, but not all, of the issues from Chapter II. Dragon does treat scale as a separate degree of freedom and couples user scale and user location in the map-centric (exo-centric person) travel method. However, Dragon's travel technique does not explicitly address the

stereo issues such as image fusion or frame cancellation and the Dragon system does not support global, whole-planet terrain.

3 Exo-centric Travel

Our display environment consists of a Fakespace Immersive Workbench with Polhemus trackers. The user wears stereo Crystal Eyes glasses with a tracker receiver mounted on the side. The user holds a custom-made "laser pointer"-like device which is tracked and has 5 buttons. The physical device appears to emit a virtual laser beam. To travel, the user activates one of 3 modes by separate buttons on the laser pointer. These modes are zoom, pan, and rotate. The basic metaphor is that the user is manipulating the terrain with the laser pointer. As the user employs these modes there are several automatic adjustments to view position parameters in order to keep the terrain stereoscopically viewable.

3.1 Start position

Any 2D or 3D map travel technique must obviously have an initial point of view. For 2D maps and small scale 3D maps, the start point is trivially the view point that displays the entire map. When navigating whole-planet terrain in a head-tracked stereoscopic display, choosing a good starting point is more complex. From experimentation, we characterize an ideal start position as follows:

- The position should display the most complete view of the planet possible.
- The user should be able to view the entire planet without diplopia or frame cancellation.
- The user should be able to physically reach as much of the displayed planet as possible directly with his hands.
- Within the previous three constraints, the planet should appear as large as possible.

We choose a start position that empirically balances these goals. We parameterize this start position based on a canonical user resting head position. Specifically we use:

- the display size and position
- a standard height (SH) measuring an average user's eye height
- a default viewer distance (D) from the front of the workbench (Figure 24, next page).

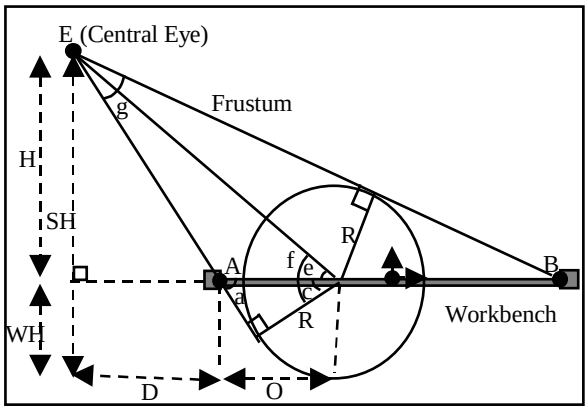


Figure 24: Start Position

We compute the radius, *R*, and offset, *O*, of the largest sphere which is contained in the default view volume. Figure 24 shows a side view of the situation. We fix the sphere center to be in the projection plane in order to keep half of the planet above the physical display. While we could place more of the planet above the display, this would require using a smaller scale planet.

In Figure 24, *WH* is the height of the display surface. Points **A** and **B** represent the edges of the display area. Lower case letters represent the illustrated angles. Computation is then done in a 2D coordinate system whose origin is at the display center. *O* and *R* are computed as follows:

$$O = H/\tan e - D$$
$$R = O \sin a \qquad (2)$$

Angles *e* and *a* can be calculated from the other labeled points in the figure. The *O* and *R* values are then used to set the initial user scale and position.

Note that while this start position balances the four goals, the user can still move his head to violate these goals. Only dynamic placement of the planet can prevent this. However, to avoid confusing the user we do not use dynamic adjustment the start placement and scale.

3.2 Zoom

The first travel mode is for zooming. The zooming technique involves a user-controlled scale plus an automatic rotation. The scale and translation work as follows. When the user presses the zoom button, the current pointer position is recorded. As the button is held and the pointer is moved towards or away from the projection plane, the magnitude of the displacement from the initial position is computed. The magnitude determines the zoom speed. The direction of zoom, either in or out, is determined by whether the pointer is displaced closer to or farther from the projection plane. To perform the zoom, we first scale the Platform Coordinate System up or down based on the magnitude and direction of the pointer movement. This has the effect of changing the physical-world to virtual-world scale factor, making the perceived world shrink or grow. Next, if the pointer intersects the terrain, the Platform Coordinate System origin is simultaneously scaled about this intersection point. This causes the user to zoom about the selected point. This technique gives the user control of zoom speed and direction plus control of the zoom-in point. Note that since we zoom by a fixed scale factor at every step, the user moves in and out at a logarithmic rate which appears to be more effective than simple high velocity motion [Mack90].

Since giving the user control of the zoom-in location gives her two more degrees of freedom to manage, we initially considered simpler approaches. For example, we tried always zooming about a fixed point such as the display center. However, we informally observed that this leads to repeated switching between zooming and panning while moving to a target. A user aligns her target with the display center by panning and then starts to zoom. Soon she is zooming slightly off target, which requires a correcting pan. This occurs repeatedly while approaching a single target and soon becomes a nuisance. Giving the user control of the zoom point avoids this problem.

In addition to this scale and translation activity, the user is automatically repositioned so that the planet appears to smoothly rotate about the selected terrain point. In detail, the planet rotates so that the planet normal vector at the selected terrain point becomes perpendicular to the projection plane (Figure 25B, pg 80). Without this automatic rotation, a zoom quickly brings too much of the planet out of the projection plane leading to image fusion problems and severe frame cancellation (Figure 25A). The automatic

rotation also keeps the viewing adjustment step, discussed in Section 3.5 , from pushing the planet deeper into the display plane and driving the target location farther away.

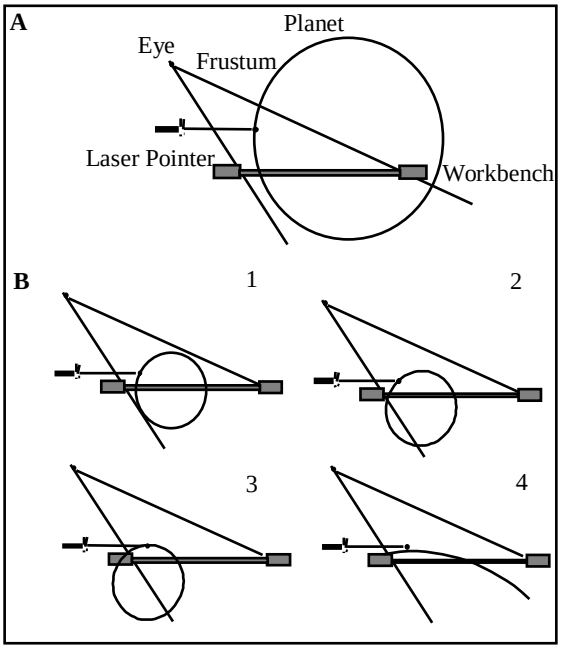


Figure 25: Zoom Problem (a) and Solution (b)

There are a few additional automatic activities and rules applied when zooming. First we only activate the adjustment step (see Section 3.5) when the world scale is past the threshold at which the planet diameter is 2.5 times the display width. This prevents the adjustment step from dynamically repositioning the carefully chosen start position. The next automatic activity guarantees a return to the initial position when zooming out. When the planet diameter is less than 2.5 times the display width and the user is zooming out, we smoothly move the viewpoint so as to slide the planet back to its starting position while maintaining its current orientation. The repositioning occurs as a function of the difference between the view's current scale and the starting scale. This guarantees the initial scale and initial position are reached simultaneously.

3.3 Panning

The next travel mode is panning. The user grabs a point on the terrain with the laser ray and then drags the terrain around by that point at the end of the ray. This technique is motivated by the results of Johnson for panning 2D scenes in a touch-controlled display [John95]. Our panning method is analogous to Johnson's 'Background panning' method, also known as the 'click and drag' method, which he found to provide the best accuracy and to be the method most users expected.

In more detail, the panning method works as follows. When the user presses the pointer's pan button, the point on the terrain intersected by the virtual laser is recorded. As the user moves the laser pointer, we track the intersection of the laser with a planet-centered sphere which intersects the recorded point. We then move the viewpoint to effectively rotate the planet about its center and bring the terrain point to the sphere intersection point. This panning method is effective at all zoom levels from those at which the planet appears as a small globe to those where the terrain appears planar. Also note that our ray-terrain intersection code [WarZ98a][WarZ99c] allows for intersection with terrain outside the view frustum. This adds a bit more flexibility since the user can grab unseen terrain and therefore make larger panning gestures than would be possible if only on-screen terrain could be selected.

Our panning technique differs from the virtual sphere technique [Chen88]. In the virtual sphere technique, a user rotates a 3D object with a mouse. This technique treats the object as if it were encased in an invisible sphere which the user can click-and-drag with the mouse pointer. The differences with our panning technique are that we use 6-DOF devices instead of a mouse and we use polygon accurate point selection [WarZ98a][WarZ99c] instead of computing intersections with a virtual sphere. We found this to be *absolutely necessary* when viewing higher resolution data. In these cases, spheres or ellipsoids were far too rough an approximation to the visible terrain to allow the user to usefully select and drag the terrain.

Mackinlay et al. [Mack90] does polygon accurate picking for panning movements along with an automatic orientation step that brings the look-at vector perpendicular to the selected polygon. However, they do not address head-tracking or stereo and zooms occurs by a translation.

Finally, our panning method differs from [Durb98]. In [Durb98] the grabbed terrain point is always the point directly underneath the 6-DOF device. In fact, [Durb98] does not compute intersections with the

terrain at all but effectively computes the intersection with a single, abstract plane which approximates the terrain. Clearly, this technique only works for small, local terrain regions and not global, whole planet terrain. Also note that since a user can easily point our pointing device straight down, our method subsumes the method used in [Durb98].

3.4 Rotation

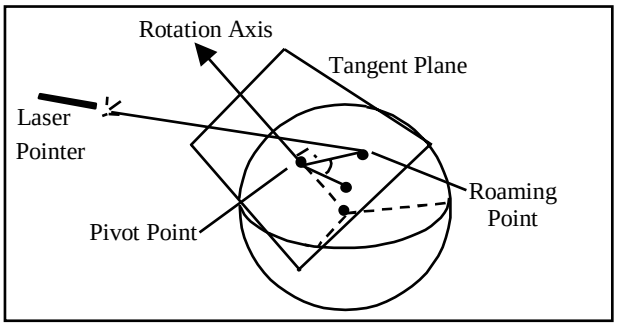


Figure 26: Rotation Geometry

The final mode is rotation. When the user presses the rotation button the point on the terrain intersected by the virtual laser is recorded. This point is called the pivot point. Next the plane which intersects the pivot point and is tangent to the planet's spheroid is also calculated. While the button is held, the intersection point of the virtual laser and this plane is computed. This point is called the roaming point. As the user moves the roaming point, a thick line between the pivot point and roaming point is displayed. This thick line is rendered as a green transparent cylinder with a diameter set to be 5% of the display width. We call this cylinder the "rotation cue". While the user moves the roaming point about the pivot point, the planet rotates about the axis defined by the tangent plane's normal and the pivot point. Figure 26 illustrates these details.

It is also necessary to implement a small dead zone around the pivot point. Unless the user draws the roaming point beyond a threshold distance from the pivot point, the planet is not rotated. Without this dead zone, the user tends to create unintended large rotations due to small hand motions, especially when initiating the rotation maneuver. The dead zone's threshold distance is 10% of the display width.

Finally during rotation the pivot point is sometimes offset from the initial terrain intersection point along the spheroid normal. This becomes necessary when, at the current scale, the terrain undulations appear more than a few centimeters high. Here the terrain can obscure the rotation cue especially if the pivot point occurs at a lower elevation. To implement this feature, we use some information recorded by the adjustment step. Specifically, we use the highest visible terrain peak measured perpendicular to the projection plane. We simply offset the pivot point along the spheroid normal by a distance equal to the difference between the high-peak height and the pivot point height. Ideally we should use the distances measured perpendicular to the plane tangent to the spheroid; however, this adds computational expense. Instead, we only perform this offset step if the spheroid normal is within 5 degrees of the projection plane normal. This works well since, due to the zoom-in auto-rotation feature, whenever we reach scales at which terrain undulations can obscure the rotation cylinder, the visible portion of the planet is fairly flush with the projection plane. Finally, we also implemented a two handed version of this rotation using two virtual laser pointers. One pointer defines the pivot point while the other defines the roaming point. Both methods are suitable for all scale levels.

3.5 Adjustment step

Having covered the user controlled travel activities, we now discuss the automatic adjustment step for maintaining good stereoscopic imagery. Our goal is to take maximum advantage of stereoscopic depth cues while minimizing diplopia, frame cancellation and image distortions.

While others working with non-head-tracked stereo vary modeled eye separation [Ware95a][Ware95b] [Ware98], we do not do this. The primary reason is that false eye separation in head-tracked stereo induces a distortion with a head-position dependent shearing component (Chapter V). This causes the viewed scene to shear back and forth with head movement despite otherwise perfect display calibration. Additionally, the induced distortion will cause the hand-held six DOF devices and their virtual representations to be miss-aligned. Poor correspondence between hand held six DOF devices and stereo imagery is problematic [Schm83][Deer92][McKe92]. Finally in many command-and-control applications, users demand accurately portrayed height data which can be warped by false eye separation (Chapter V, Section 2). For

these reasons we use a nominally correct eye separation. Therefore to maximize stereoscopic depth cues, we aim to keep the terrain within 1.5 meters of the user, a distance where stereo is strongest as a depth cue [Cutt97]. For the virtual workbench, this means keeping the terrain as close as possible, while considering image fusibility and frame cancellation. Also keeping the terrain slightly above the display plane puts most objects within arms reach and lets the user contact objects which are stereoscopically above the display.

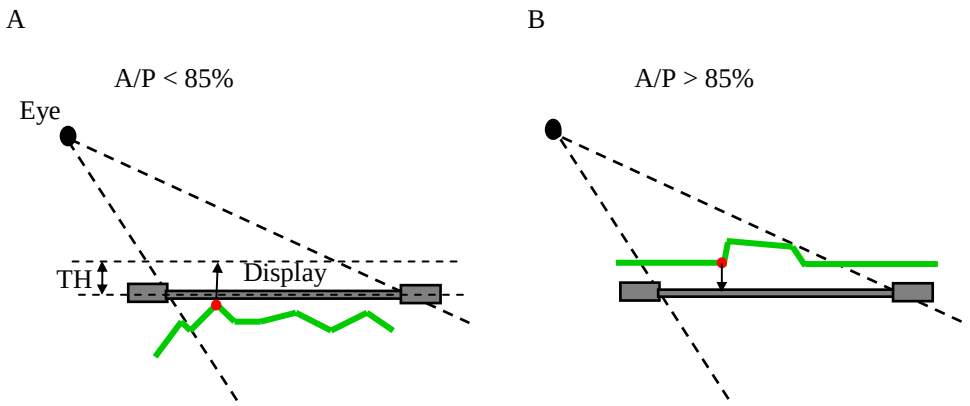


Figure 27: Illustration of Adjustment Step

The adjustment step works as follows. The terrain rendering thread renders display lists which are created by various other threads such as the terrain renderer and the object renderer. For the adjustment step, the render thread copies a sample of the right-eye depth buffer generated from the terrain display list and then the navigation thread examines this copy. The navigation thread scans the depth buffer copy and finds both the farthest point above the projection plane and the point nearest to the eye. During the same loop, we also record the number of pixels, *A*, above the projection plane and the number of pixels, *P*, not equal to depth buffer clear-screen value. Two rules are then applied. First if *A*/*P* is less than a threshold (85%), we move the view along the projection plane normal in order to bring the near point to a predetermined target height, *TH*, above the display plane (Figure 27A). Second if *A*/*P* is greater than the threshold, then we move the view along the opposite direction so that the far point is flush with the display plane (Figure 27B).

While the first rule draws the terrain peaks out of display, the second rule counters the following problem. At certain scales and terrain formations, the first rule can bring an unnecessary amount of the terrain above the display plane. For example, a particular data set might contain a few peaks and then mostly flat land. At certain scales the first rule would cause all of the flat terrain to be floating above the display. Effectively there is a large plane which extends far beyond the window limits hovering above the display. Even at target heights as small as 5 centimeters, the uniformity and extent of this plane creates a strong frame cancellation effect. In contrast for the same target height, if the terrain is more undulating, then the frame cancellation effect is less disturbing. We surmise this occurs since with undulating terrain only some of the terrain at the display edges is clipped by the view frustum, while with the flat planar terrain *all* terrain at the display edge is clipped. Given this situation, the more natural position for this problematic terrain is with the planar area flush with the display plane. The second rule catches such cases--where too much terrain is above the display plane--and pushes the terrain back down. Note, while Ware et al. [Ware95a][Ware95b][Ware98] effectively move the near point instantaneously to the screen, we move the view at an exponentially decaying rate towards the target position, as determined by one of the two rules. Using these smooth transitions keeps our rule set from causing abrupt displacements especially when first zooming in from the start position. Also the smooth transition serves as a dampening factor to prevent oscillations between the two rules.

For the Target Height, *TH*, we use a constant value that empirically works well. While *TH* could be adjusted as a function of the nearest fusible image plane [Sou95] this would cause the terrain to be pushed down into the display plane when the user leans down for a closer look. We informally observed that such behavior is more unnatural than diplopic conditions. While people experience real-world diplopia when peering too closely to an object, they do not experience inanimate objects autonomously moving away when closely examined. Therefore *TH* is set to 10% of the standard user height above the workbench, a position within Yeh's [Yeh90] fusibility constraints for the standard eye height (Figure 24, pg 78).

4 Results and Observations

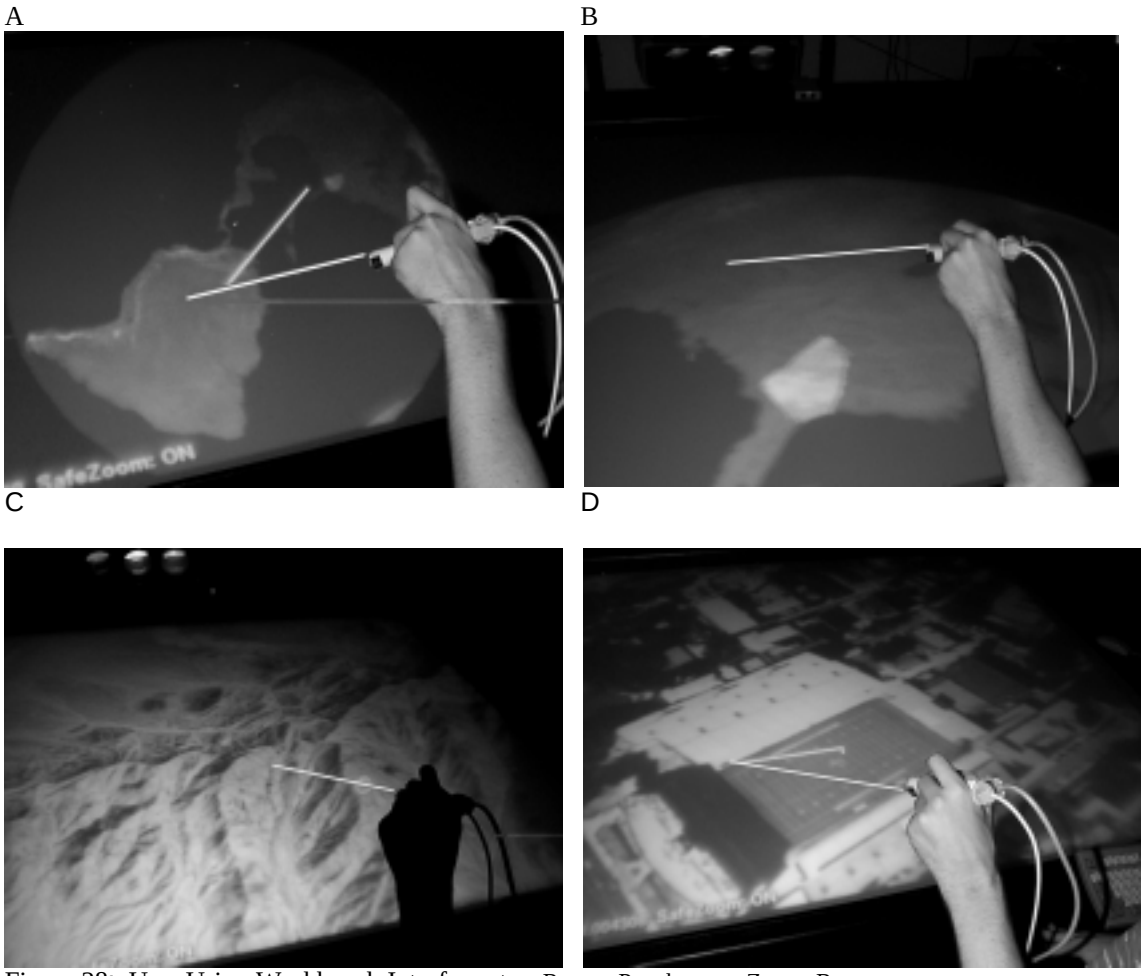


Figure 28: User Using Workbench Interface--top: Rotate, Pan; bottom: Zoom, Rotate

This navigation interface has been demonstrated to dozens of technical and non-technical users. Figure 28 illustrates the interaction on a virtual workbench. Figure A is rotation. Figure B is panning. Figure C is zooming and Figure D is again rotation. The described design has been illustrated to perhaps 100 users in demonstrations over the past three years. Anecdotally, several interesting issues have arisen. First, how the user holds the virtual laser pointer greatly affects their ability to use the interface. Users who grasp the

pointer like a pen as shown in Figure 28B anecdotally become reasonably adept after 15 minutes. (Admittedly the user's in our demonstration population are biased. Those attendees who are probably uncomfortable with 3D computer games, etc. and might perform poorly with our interface undoubtedly shied away from active participation). Some users however insist on grasping the pointer like a sword handle. This works very, very poorly. The pen like grip orients the pointer downward which is the way in which the interface was conceived. A sword like grip orients the pointer upward. This forces the user to twist her wrist in an awkward position in order to try and point downward. Some of the most memorable users, who had the greatest difficulty, insisted on adopting the sword grip. Part of the problem appeared to be the form factor of the pointer. Originally it was made of 14/16 inch PVC pipe with tactile buttons mounted on the outside. This thickness appeared too large for most users to comfortably grasp like a pen. (The author who built the pointer, has relatively large hands. With fingers spread apart the author's pinky tip to thumb tip distance is 23.5 cm.) Therefore we rebuilt the pointer with 10/16 PVC pipes and cut the pipes to embed the buttons into the pointer. Anecdotally this smaller form factor appeared to greatly help users hold the pointer in a pen like position. Unfortunately, some users still adopted a sword like grip to the detriment of their experience with the interface. Perhaps this occurs because with the lighter PVC pipe, the weight of tracker receiver on the back of the pipe plus the weight of the dangling wires makes the pointer unbalanced and back heavy. One user commented that it's like trying to hold a pencil with a lead weight at the eraser end. A more professionally designed pointer could avoid this problem. Unfortunately the commercial ones we are aware of have only a single button. We suspect that a design that looks like a pen will probably provide a better affordance [Norm90] toward taking a pen like grip. In our case, however, users always observed the author or demonstrator using the pen grip first and as mentioned most adopted the grip without further comment.

In terms of the stereo effectiveness, users and passive viewers routinely remarked that the globe, terrain and buildings appeared to protrude directly out of the display screen. This would simply not happen without the adjustment step which brings the terrain to a point slightly above the screen. Without the scaling and adjustment step, we'd be viewing the terrain at a great distance. This would yield positive screen parallax that is known to yield 3D images appearing behind the screen plane [Lipt82].

CHAPTER V

GEOMETRIC DISTORTION ANALYSIS

1 Overview

A VR system that interactively generates the presented imagery and displays it on a stereoscopic HTD has great control of all viewing parameters at the time of viewing. In contrast, in stereoscopic video and photography most of the viewing parameters cannot be altered once the photo is taken. Image shifting and image scaling are primary mechanisms to alter the stereo image at view time in non-interactive stereo media. Given the greater flexibility of VR, it makes sense to geometrically analyze image shifting and image scaling to see how the implied 3D distortion compares to and interacts with simpler viewing parameter adjustments such as view position and view scale as well as false eye separation methods. Again, in VR these simpler parameters can always be manipulated at view time due to the interactive image generation. This chapter presents analytic expressions, illustrations and comparisons of the geometric distortions induced by false eye separation, image shifting and image scaling.

2 False Eye Separation

The false eye separation technique was originally developed in non-head-tracked stereoscopic displays. Underestimated eye separation controlled the fusibility of deep scenes while overestimated eye separation enhanced the depth of shallow scenes. When applied to modern head-tracked systems, however, we observed that false eye separation has dynamic distortion effects. Figure 29 illustrates this phenomenon when underestimating the eye separation by one half its true value. In Figure 29A, a user views a horizontal stereo HTD such as a virtual workbench. The physical display is outlined in black. The gray cube is the modeled virtual geometry. Figure 29A-E are a frontal view of this display system. Here the display surface appears as a horizontal black line. While the system internally models the gray cube, the

user actually perceives a trapezoidal shaped object shown in red. Figure 29B and C illustrate the how the perceived object compresses and expands due to up and down head movement while D and E illustrate the left/right leaning due to side to side head movement. The user would observe similar forward/back leaning as the user moves his head into and out of the plane of the diagrams D and E.

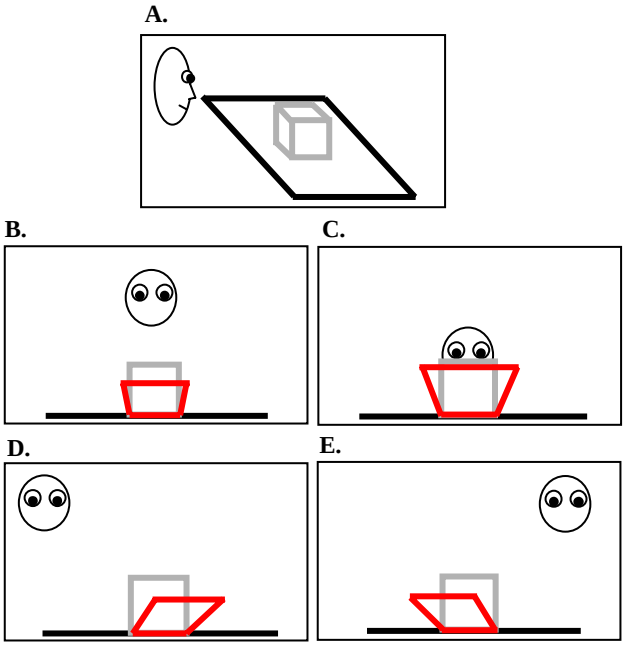


Figure 29: Observed Effect of Underestimated Eye Separation -- (A) A user viewing a horizontal stereo HTD such as the virtual workbench. The gray cube is a virtual cube. (B-E) Four front views of the user viewing the cube on the stereo HTD. The horizontal black line is the display. Underestimated eye separation causes the user to perceive a trapezoidal version of the cube (red) instead of the modeled cube (gray). (B) and (C) illustrate the compression and expansion of the perceived object due to up/down head motion. (D) and (E) illustrate the left/right shifting of the perceived object due to side to side head motion.

VR systems orient displays in a variety of ways. To discuss arbitrary displays, we define the term "perpendicular head motion" to be head motion perpendicular to the screen and we define the term "lateral head motion" to be head motion parallel to screen. Using this terminology, the prior distortion observation

shows that perpendicular head motion yields an expansion/compression effect while lateral head motion yields a leaning effect. This behavior is particularly irksome because one of the advantages of adding head-tracking to stereoscopic displays is to *remove* similar distortions that occur in the absence of head-tracking [Hodg92].

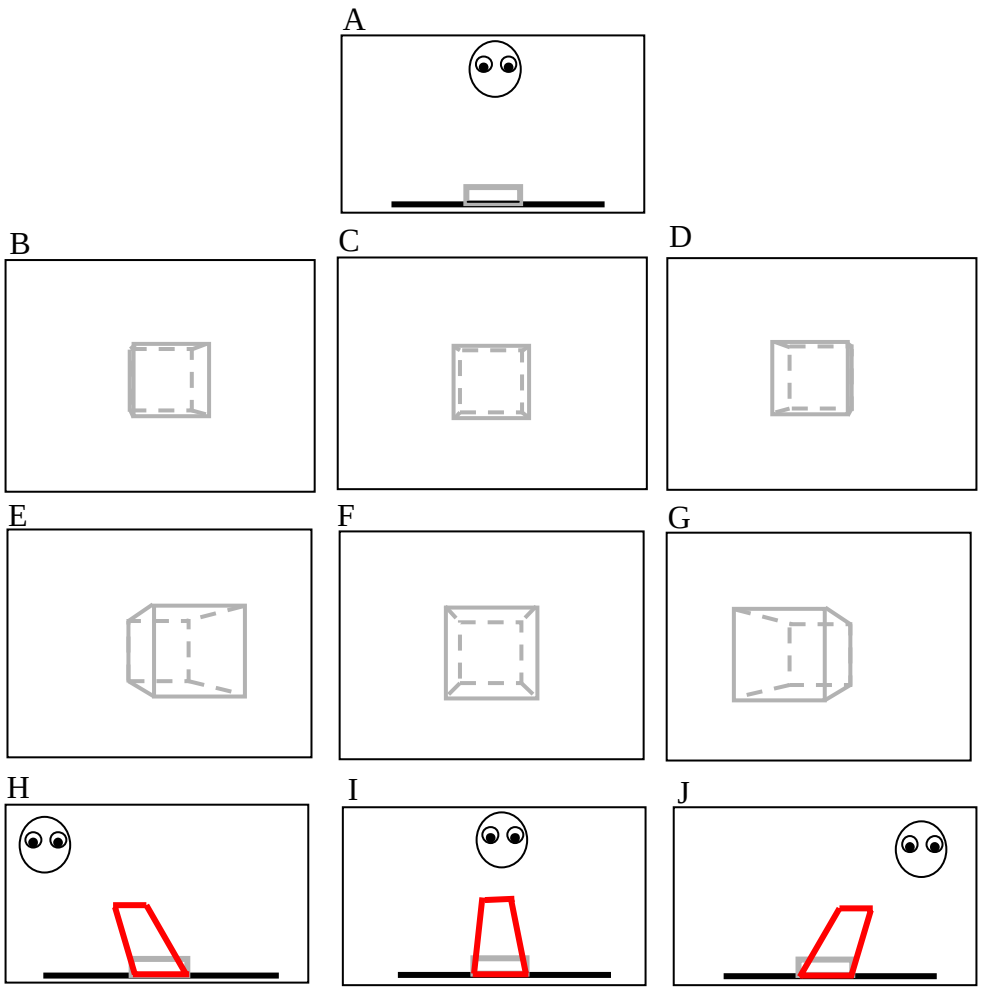


Figure 30: Effect of Overestimated Eye Separation -- (A) User viewing a flattened cube. (B)-(D) the image the user observes when moving left to right. (E)-(G) is what a user would see when moving left to right if the flattened cube is replaced by a true cube, i.e. one that is not flattened and hence extends higher above the display screen. (H)-(J) show our hypothesis as to why overestimated eye separation yields enhanced depth obscures the details of the cube sides as compared to in the true separation case (E)-(G).

While underestimated eye separation makes deep scenes fusible, overestimating the eye separation makes flat scenes appear with greater stereoscopic depth. Ware et al. [Ware98] applies this to non-tracked VR. When applied to a head-tracked stereo system, however, we again observed dynamic distortions. Figure 30 helps illustrate our observations. The user views a flat box (Figure 30A). Figures B through G illustrate the box as seen by the user under two conditions. In figures B through D the eye separation is exaggerated. This makes the top of box appear to rise farther out of the display. In figures E through G, the eye separation is correct but we scale the box height until the box top is at similar depth as in the overestimated eye separation case. As the user moves her head left to right (B through D and E through G), there is a noticeable difference in the ability to see the sides of the box when comparing the overestimated eye separation case (B through D) to the scaled height case (E through G). The box behaves as if it is leaning towards the user (figures H through J) thus obscuring the user's view of the sides. Again, figures H through J compare the perceived stretched box (red) with modeled box (gray). Perpendicular head motion with overestimated eye separation also yields expansion and compression. Further discussion is found in later analytic sections.

2.1 Construction of the Distortion

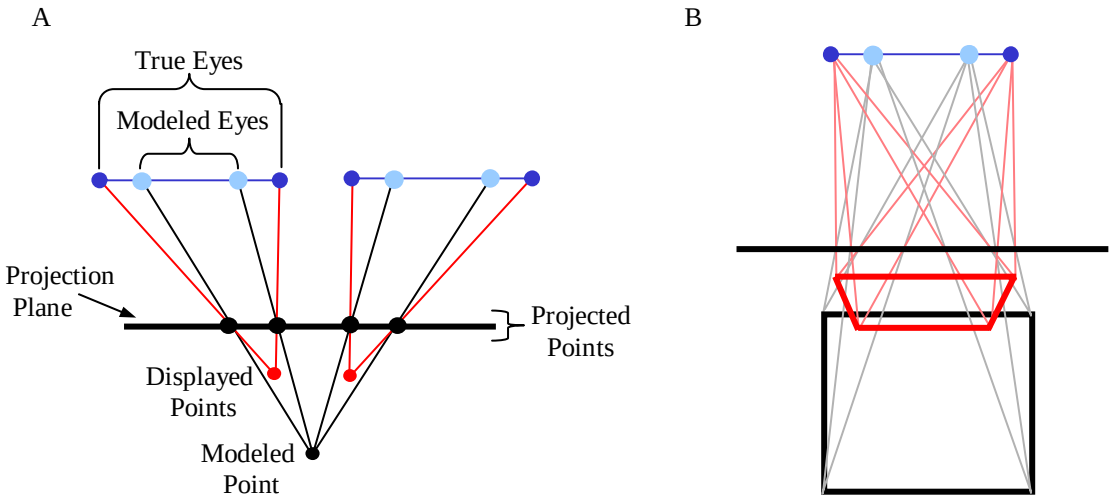


Figure 31: False Eye Separation Geometric Construction -- (A) Illustrates the two displayed points (red) associated with two different head positions. (B) Illustrates the construction's effect on a box.

At first observation, one might guess that the distortions observed were actually due to latency in tracking. However, by moving our head very slowly we mitigated the effects of latency and we still observed warping of the 3D image. To understand this warping we began with hand drawn constructions like Figure 31. In Figure 31A, two sets of eye points are illustrated in blue. Within each set the true eye points are on the outside in dark blue and the modeled eye points are on the inside in light blue. The projection plane is the horizontal black line. Below a single modeled point is shown in black. For each head position, the modeled point is projected onto the projection plane through the modeled eye points. These projectors are drawn in black. If we had used true eye separation, to get the same projected screen points we would have to first transform the modeled point (black) to the displayed point (red) and then project this new point onto the screen. The location of the displayed point is found by drawing lines (red) between the true eye points and the projected points. This construction defines the effect of false eye

separation on the displayed space. If we let Δ be the 3D transformation mapping the modeled point to the displayed point then we can say that transforming 3-space by Δ and then projecting this displayed space onto the image plane using the true eye points yields the same projected images as projecting the original modeled space onto the image plane using the underestimated eye separation. Note how the red displayed point moves as the user moves her head between the two illustrated positions. This indicates that the model-to-display space transformation changes with head position. Figure 31B illustrates the construction's effect on a more general shape. (Shortly we will justify connecting the displayed points with straight lines by proving that the construction preserves straight lines).

This construction assumes that all the important physical measurements relevant to the viewing coordinate systems (see Chapter II, Section 3.2), except for the modeled eye separation, are correct. This implies the system is *perfectly* tracking the head-position and orientation in order to yield the correct location and orientation of the eye axis as used by the rendering subsystem. The construction also assumes any distortion due to curvature of the screen or any optics is negligible or accounted for by other means [Deer92]. Additionally, it neglects the fact that the true eye separation changes during vergence.

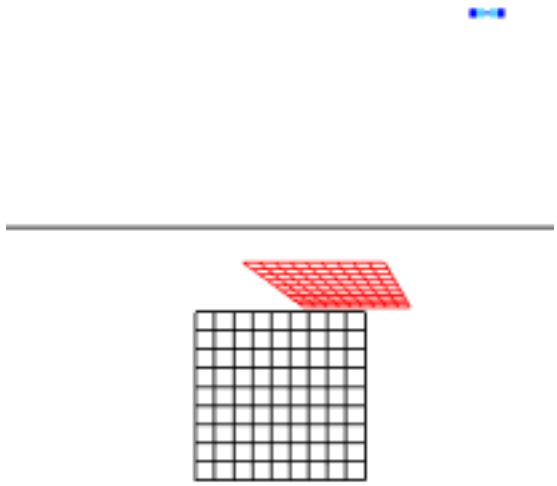


Figure 32: Software Generated Diagram of Underestimated Eye Separation -- The geometric construction is used to map grid points only (not the lines between them) but it *appears* that linearity is preserved.

A key remaining question is whether the construction actually preserves lines or whether it maps lines to curves. To pictorially explore this issue, we need to perform the construction for a large number of modeled co-linear points. Here manually drawing the construction becomes too tedious, so we use a simple program written in C and OpenGL that graphically illustrates the construction's effect on a set of points arranged in a grid. The program generates the construction by computing the intersections of the lines in the construction for each grid point and by then connecting the transformed grid points with lines. Some of the results of this computed construction are shown in Figure 32. The modeled grid is black and the displayed grid is red. Note, the software only transforms the grid points using the construction. It is possible that the line segments between the modeled grid points (black) should actually be mapped to curve segments and not to line segments as illustrated in the displayed grid (red). However, the fact that the display space grid points did *appear* to remain co-linear convinced us that the construction was indeed a collineation. In order to *prove* that the construction is a collineation, one must develop an axiomatic proof or an analytic description showing the construction preserves straight lines. Because we ultimately desired an analytic description of the distortion and because we were pictorially convinced that the construction was a collinearity, we pursued the analytic description directly.

2.2 Analytic Derivation

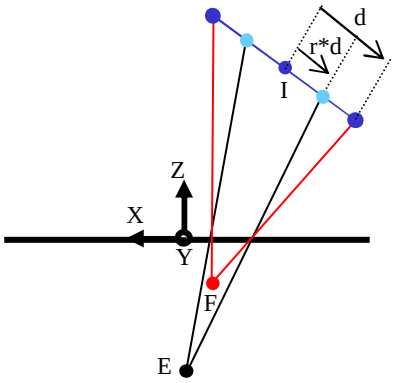


Figure 33: Parameterization of the Distortion Due to False Eye Separation -- The projection plane lies in the X-Y plane. The user's true left and right eye are displaced by vectors **d** and **-d** from the central eye, **I**. The modeled eye points are displaced by $r \bullet$ **d** from the central eye. r is the ratio of the modeled eye separation to the true eye separation. **E** is a modeled point while **F** is the displayed location of this point.

To derive an analytic description of this distortion we parameterize important points as illustrated in Figure 33. First we place the Projection Plane Coordinate System at the center of the projection plane with the plane containing the X-Y axes. Next we add a central eye point, **I**. The true left and right eyes are displaced from **I** by the vectors **d** and –**d**. 2|**d**| is the true eye separation. The scalar r is the ratio of the modeled eye separation to the true separation. Hence the left and right modeled eyes are displaced by r***d** and $-r$***d** respectively, and 2r|**d**| is the modeled eye separation. **E** is the modeled point and **F** is the displayed point.

Appendix A derives the analytic description of the construction. In projection plane coordinates the matrix is:

$$\Delta = \begin{bmatrix} 1 & 0 & \frac{(1-r)(I_x I_z + d_x\ d_z\ r)}{d_z{}^2 r^2 - I_z{}^2} & 0 \\ 0 & 1 & \frac{(1-r)(I_y I_z + d_y\ d_z\ r)}{d_z{}^2 r^2 - I_z{}^2} & 0 \\ 0 & 0 & \frac{r(d_z{}^2 - I_z{}^2)}{d_z{}^2 r^2 - I_z{}^2} & 0 \\ 0 & 0 & \frac{I_z(1-r)}{d_z{}^2 r^2 - I_z{}^2} & 1 \end{bmatrix} \tag{3}$$

In the context of a rendering pipeline the distortion acts as follows. Let a matrix, $\boldsymbol{M}_{\boldsymbol{A}}^{\boldsymbol{B}}$, denote the coordinate transform from coordinate system A to coordinate system B. Then matrix stack during rendering is:

$$\boldsymbol{M}_{\boldsymbol{Model}}^{\boldsymbol{Screen}} = \boldsymbol{M}_{\boldsymbol{World}}^{\boldsymbol{Screen}} \cdot \boldsymbol{M}_{\boldsymbol{Model}}^{\boldsymbol{World}} \tag{4}$$

Let $\boldsymbol{[M]_A}$ be the representation of a transform $\boldsymbol{M}$ in coordinate system A. Then using false eye separation effectively induces the complete transformation:

$$\boldsymbol{M}^{*\boldsymbol{Screen}}_{\boldsymbol{Model}} = \boldsymbol{M}_{\boldsymbol{World}}^{\boldsymbol{Screen}} \cdot \left[\Delta\right]_{\boldsymbol{World}} \cdot \boldsymbol{M}_{\boldsymbol{Model}}^{\boldsymbol{World}} \tag{5}$$

Therefore, using false eye separation will produce the *same displayed 3D image* as using the true eye separation and adding $\boldsymbol{[\Delta]_{World}}$ on the matrix stack. Note, that as equations (4) and (5) describe virtual space, $\boldsymbol{[\Delta]_{World}}$ will include a scale component inherited from the Platform Coordinate System scale. However, when analyzing $\boldsymbol{\Delta}$, it is more convenient to ignore this scale issue and consider the Projection Plane Coordinate System as it exists in the physical world. We can then discuss the effects of $\boldsymbol{\Delta}$ in

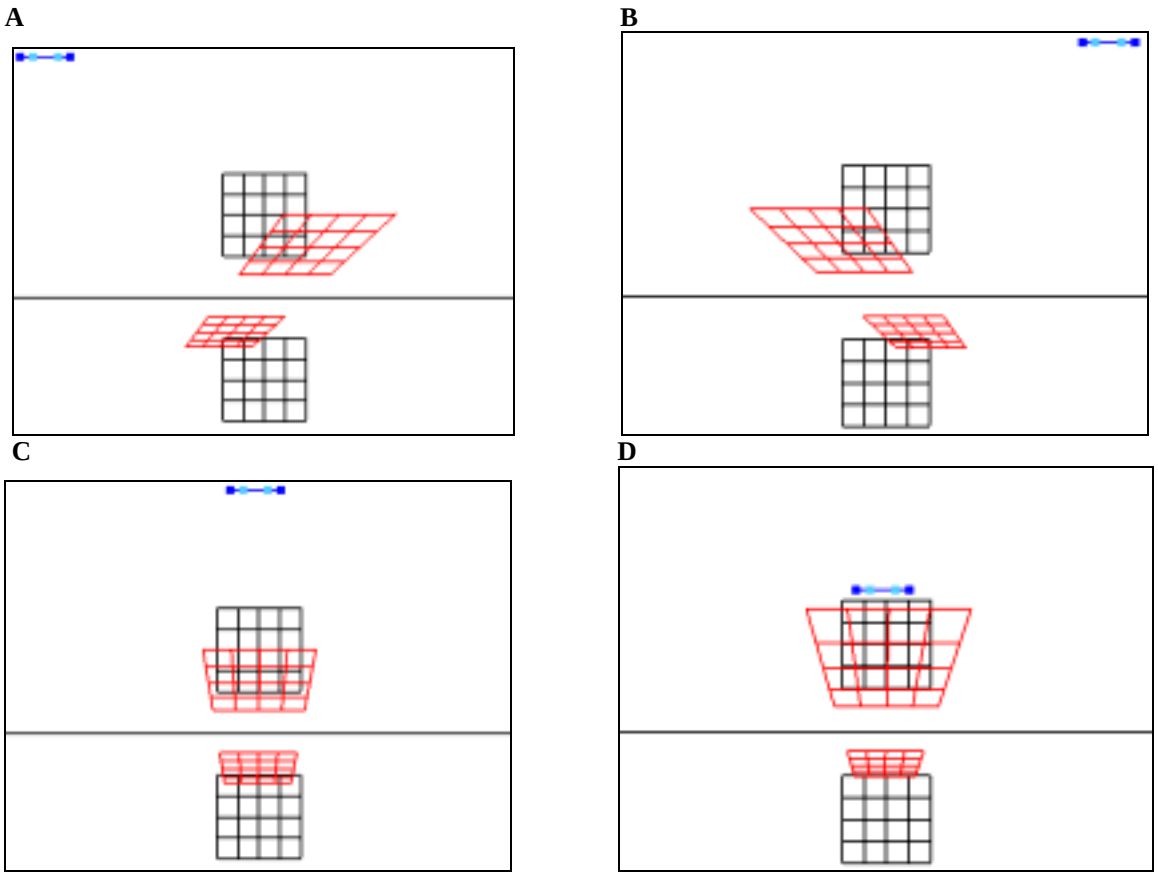


Figure 34: Effect of Underestimated Eye Separation -- (A) and (B) show the sideways shifting of lateral head motion, and (C) and (D) show the compression/expansion of perpendicular head motion. The color convention follows that of Figure 31.

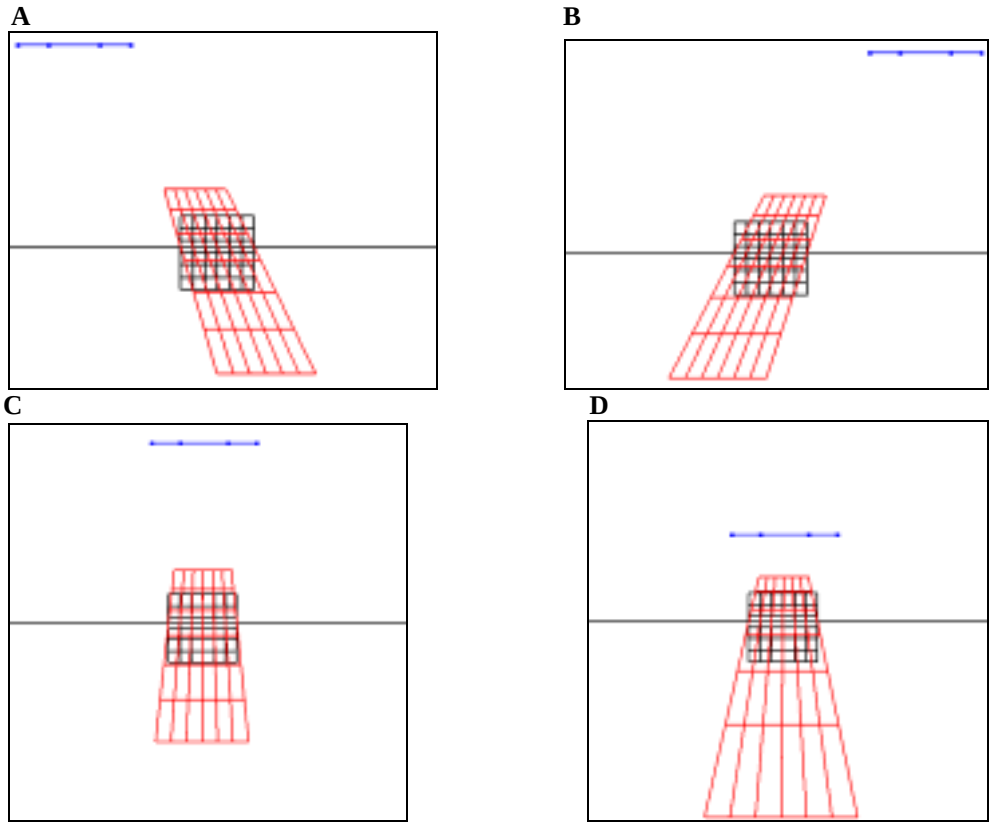


Figure 35: Effect of Overestimated Eye Separation-- (A) and (B) show the sideways shifting of lateral head motion, and (C) and (D) show the compression/expansion of perpendicular head motion. The color convention follows that of Figure 31.

physical units such as meters and consider how Δ behaves independent of the Platform Coordinate System scale.

2.3 Qualitative Analysis of Δ

Δ is a collineation so it maps straight line to straight lines. However the element in the fourth row and third column is non-zero indicating the distortion is a non-affine collineation; so Δ does not preserve angles, distances nor parallelism. Next in the third column contains a shearing component in rows 1 and 2 and a scaling component in row 3. All these components depend on head position and eye separation ratio so the value of Δ varies with these parameters. This correlates with the dynamic distortions of the displayed scene as the user moves his head.

We can now make more formal qualitative observations concerning the interaction of head movement, modeled eye separation and modeled object position on the distortion components. Figure 34 illustrates the underestimated eye cases using two modeled grids, one in front of the screen and one behind the screen. For underestimated eye separation, objects in front of the screen will shear *opposite* to the direction of lateral head movement while objects behind the screen with shear *with* the direction of the lateral head movement (Figure 34 A and B). Objects in front of the screen will expand as the head approaches the screen (compare C to D) while objects behind the screen will contract as the head approaches the screen.

Figure 35 illustrates the overestimated eye cases for a single grid, half of which is in front of the screen and half of which is behind the screen. For overestimated eye separation, objects in front of the screen will shear *in* the direction of lateral head movement while objects behind the screen will shear *opposite* the direction the lateral head movement (Figure 35 A and B). Objects in front of the screen will compress as the head approaches the screen (compare C to D) while objects behind the screen will expand as the head approaches the screen. The results are summarized in Table 3 (next page).

Table 3: Effects of Head Motion on Display Objects for Under and Over Estimated Eye Separation

Underestimated Eye Separation

	Head Left	**Head Right**	**Head Closer**	**Head Farther**
Object In Front	Shear Right	Shear Left	Expand	Contract
Object Behind	Shear Left	Shear Right	Contract	Expand

Overestimated Eye Separation

	Head Left	**Head Right**	**Head Closer**	**Head Farther**
Object In Front	Shear Left	Shear Right	Contract	Expand
Object Behind	Shear Right	Shear Left	Expand	Contract

2.4 Quantitative Analysis of $\boldsymbol{\Delta}$

Having gained an intuitive understanding of $\boldsymbol{\Delta}$, we now return to a more rigorous analysis. Section 2.4.1 examines degenerate cases in the matrix. Section 2.4.2 discusses the presence of a maximum depth plane in the underestimated eye separation case. Section 2.4.3 concludes with various plots of specific components of the transformation.

2.4.1 Degenerate Cases

$\boldsymbol{\Delta}$ contains three degenerate cases which must first be addressed. All these cases correspond to similar degeneracies in the original construction. In this section we will refer to the construction as $\boldsymbol{\Delta_c}$ ("Δ Constructed"). Once we show that these cases occur in rare circumstances, we will ignore them in further analysis.

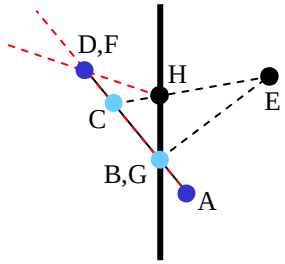


Figure 36: Embedded Modeled Eye Degeneracy

First $\boldsymbol{\Delta}$ is only well-defined when the modeled eye points are not contained in the projection plane. If they are, the denominators in the 3rd column become zero (see Equation (3), page 96). However, recall $\boldsymbol{\Delta}$ is a homogenized form of $\boldsymbol{\Delta'}$ (Appendix A, Section 2, pg 185) which assumes this eye configuration did not occur. In $\boldsymbol{\Delta'}$ this configuration leads to the lower-right term being zero and the matrix becomes singular in this case. This is in accordance with the ray construction which also becomes singular, or non-invertable. Specifically, in such a configuration, the construction maps all points to the point **D**. In Figure 36, the true

eye points are **A**,**D** (dark blue); the modeled eye points are **B**,**C** (light blue); black dashed lines show the projection of input point **E** to points **H**,**G** on the projection plane; red dashed lines show the reconstructed point **F**. Since **B** and **G** are coincident, the reconstruction line **AG** is embedded in the eye axis. Hence, **F** is constrained to the eye axis and therefore Δ_c maps 3D space to this line, in particular point **D**. Clearly, this degenerate case occurs rarely so it is permissible to ignore it in Δ_c and to ignore the corresponding degeneracy in Δ.

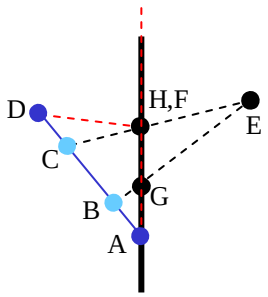


Figure 37: Embedded True Eye Degeneracy

The second degeneracy occurs when the true eye points are contained in the projection plane. In this case, Δ is singular. This follows from the fact that the rows of Δ are no longer independent if the third element in row 3, $r(d_z^2-I_z^2)$, equals zero (see Equation (3), page 96). Assuming r is non-zero, the complete term is zero precisely when a true eye point is embedded in the projection plane. Again, this result is in accordance with the ray construction, Δ_c, which becomes singular in this case. Specifically, Δ_c maps all points to the projection plane if a true eye point lies in the projection plane. Figure 37 illustrates this. The points are labeled as described in the previous paragraph. Since the reconstruction line **AG** is constrained to the projection plane, it follows that **F**, the intersection of **AG** and **DH** is also constrained to the projection plane. Hence, Δ_c maps 3 space to the plane. Again, this degenerate case occurs rarely so it is permissible to ignore it in Δ_c and to ignore the corresponding degeneracy in Δ.

The final degeneracy is the most interesting. It primarily occurs for values of $r>1$ (overestimated eye separation). Rather unexpectedly, both Δ and the original construction Δ_c flip some objects in front of the viewer to behind the viewer (Figure 38).

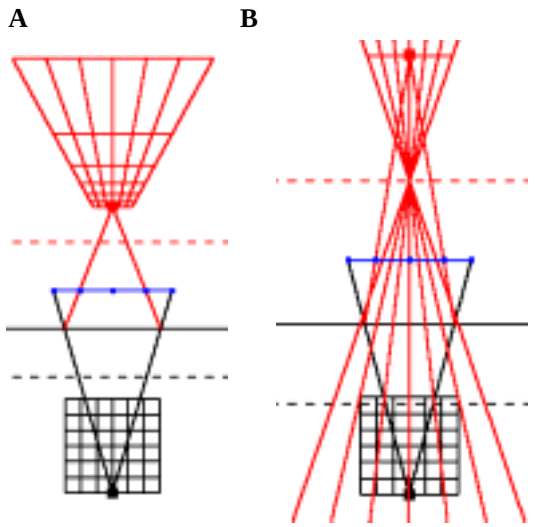


Figure 38: Excessive Positive Parallax Degeneracy

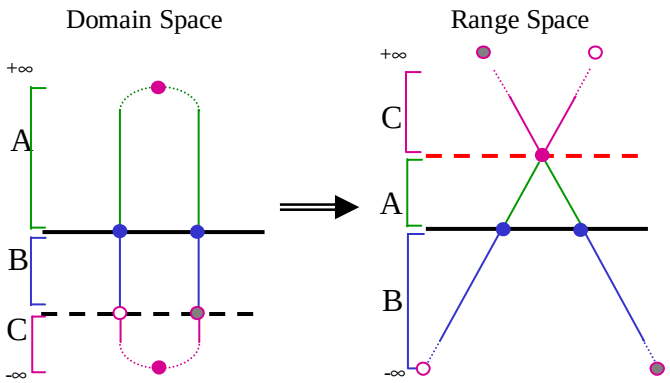


Figure 39: Effects of Typical Projective Transform. See text for details.

Such behavior is inherent in a perspective transform for objects that cross the vanishing plane of the transform [Wyli70]. Recall that the vanishing plane is the plane of points (affine points) which are mapped

to points at infinity (ideal points). For review, Figure 39 illustrates how a perspective transform maps different regions of space. Three regions in space are color coded green, blue, and purple. The fixed plane is a solid black horizontal line. The vanishing plane is a dashed black horizontal line. The vanishing plane of the inverse transform is a dashed red horizontal line. Two parallel lines in the domain space, color coded by the region containing them are mapped to intersecting lines in the range space. Ideal points are illustrated by placing a circle at both ends of the line that contains the ideal point. Some ordinary points become ideal points and ideal points become ordinary points as mapped between the domain and range space. Also note (1) how a region A, a region of infinite depth, is compacted to a region of finite depth; (2) how a region B, a region of finite depth, is expanded to a region of infinite depth; and (3) how region C, a region of infinite depth, is mapped to another region of infinite depth but is flipped around to the opposite end of space.

Returning to Figure 38, the vanishing planes and fixed plane is colored coded as in Figure 39 (fixed plane in black, vanishing plane in dash-black, inverse vanishing plane in dash red). Note the fixed plane of the transformation equals the projection plane in our stereoscopic display. Figure 38A shows the effect on an object, the black grid, beyond the vanishing plane while Figure 38B shows the effect on an object intersected by the vanishing plane. Again the ray construction is illustrated for a single point on the grid. Note when $r < 1$ (not illustrated), the vanishing line is generally behind the eyes where no stereoscopic imagery ever appears. Therefore, the effect of flipping an object in front of the eyes to a position behind the eyes is generally only arises for $r > 1$ (overestimated eye separation).

At first this degeneracy makes the basic construction, Δ_c, appear somewhat flawed from a psychophysical perspective since it does not predict what a user will perceive in this degenerate case. The problem lies in the fact that for true a eye separation, *es*, screen parallax varies from –infinity, to zero and to +*es* as the modeled point moves from the eyes' center, to the projection plane and towards infinity beyond the projection plane (Figure 12 B-D, page 30). This degenerate case, however, generates a *positive* screen parallax that is *greater* than +*e*s. As soon as we cross the veridical +*e*s limit, we have reached a situation that has no analog in real world experience. Such excessive positive parallax can yield diplopia and user discomfort. This is also referred to as divergent parallax.

Interestingly, this exaggerated eye separation ($r > 1$) has been used quite successfully in a non-head-tracked real-world application [Ware95a][Ware95b][Ware98]. Most likely, since this application only exaggerates the eye separation for scenes with little depth, most of the virtual objects lie on the closer side of the vanishing plane where they do not experience excessive positive screen parallax and a flipping under Δ_c. In such cases the result is an effective exaggerated depth shown in Figure 40.

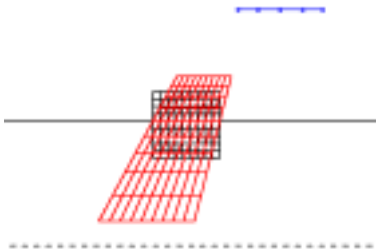


Figure 40: Effect of Overestimated Eye Separation — This is a case which yields exaggerated depth without the flipping degeneracy.

2.4.2 Maximum Depth Plane

We can use Δ^{-1} to compute the maximum possible depth in displayed space when the modeled eye separation is smaller than the true eye separation ($r < 1$). The existence of a maximum depth in the displayed space has been noted before [Wood93] [Ware95a]. Figure 41 illustrates this idea. For a point beyond the projection plane the screen parallax reaches its maximum value, equal to the modeled eye separation, for a point infinitely far away, **E**. This places a limit on the depth of the reconstructed displayed point, **F**.

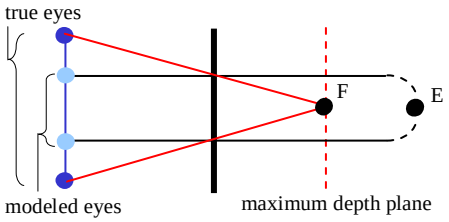


Figure 41: Maximum Depth Plane due to Underestimating Eye Separation

For a non-degenerate viewing configuration, Δ is non-singular and hence Δ^{-1} exists. Like Δ, Δ^{-1} is a non-affine collineation so it has a plane, *P*, of affine points which are mapped to ideal points (points at infinity). This plane is called the vanishing plane since these points have no image in Euclidean space. Clearly, Δ being the inverse of Δ^{-1} maps these ideal points back to the affine plane *P*. These ideal points represent the points lying infinitely far beyond the projection plane that get mapped to the maximum depth plane. *P* then is precisely this maximum depth plane. Therefore, the equation for the maximum depth plane is the vanishing plane of Δ^{-1}. It is easy to find the vanishing plane of a perspective matrix [Gold92]. With this insight the maximum depth plane is:

$$z = \frac{r\left(d_z^{\,2} - I_z^{\,2}\right)}{I_z\left(1-r\right)} \qquad (6)$$

Equation (6) illustrates how the maximum depth plane position varies with the head position's z-component, I_z. This helps explain the head-position dependent squashing of displayed space illustrated in Figure 42. Here the displayed grid compresses as the head moves towards the projection plane. This motion also brings the maximum depth plane (the dash red line) closer in.

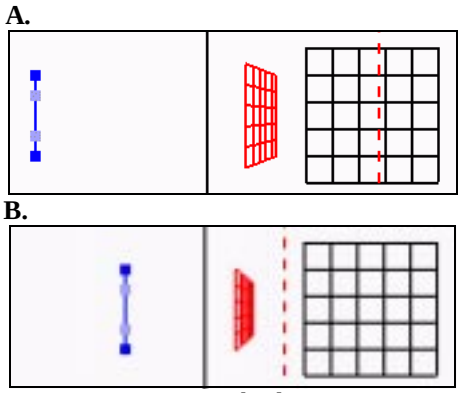


Figure 42: Effect of Head Position on Maximum Depth Plane -- Displayed grid (red) is squashed towards view plane as the maximum depth plane (dashed red line) moves inward.

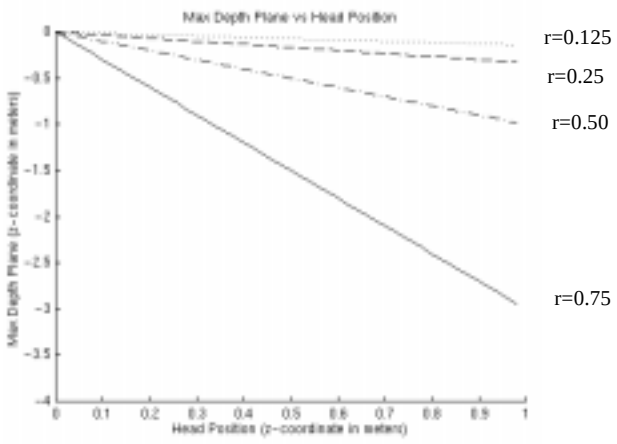


Figure 43: Maximum Depth Plane Versus User Head Position— *r* is the modeled to true eye separation ratio.

Figure 43 plots the position of the maximum depth plane as a function of viewer head position (I_z) for several eye separations ratios (r): 0.75 (solid), 0.5 (dash-dot), 0.25 (dash) and 0.125 (dot). Note, Figure 43 assumes the head is parallel to the projection plane (d_z=0); however, even for non-parallel case d_z is typically small compared to I_z. In Figure 43, the maximum depth plane position is linear with respect to the head position while it varies non-linearly with r. Smaller modeled eye separations produce a closer maximum depth plane and hence a greater compression of the displayed space.

2.4.3 Lateral Shearing

Figure 34 A and B illustrated the lateral shearing induced by false eye-separation. Here we examine this shifting more rigorously. We plot the x-coordinate difference of a modeled point, **E**, from its distorted point, **F**, as a function of head position. For simplicity, assume the eyes are parallel to the projection plane and are contained in the X-Z plane (d_z,d_y=0). Fix the central eye's (I) z-coordinate to 1 meter and then vary the central eye's x-coordinate so that the head moves side to side. In this case, F_x and hence F_x-E_x, varies linearly with I_x. This is illustrated pictorially in Figure 44. Lateral displacement of the head towards the left results in further lateral displacement of the displayed point to the left.

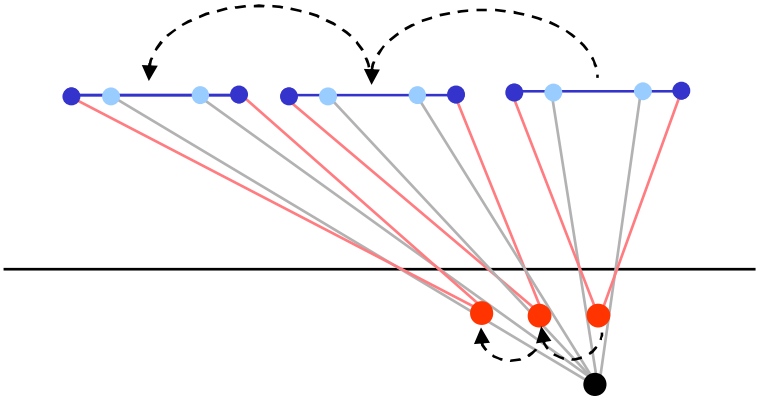


Figure 44: Linear Lateral displacement of Displayed Point

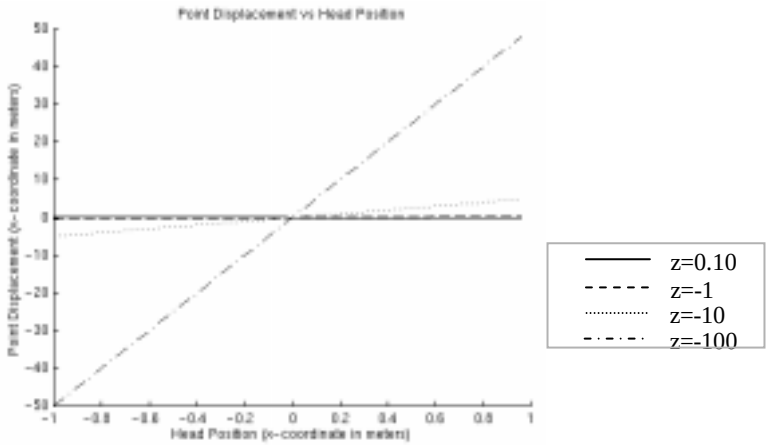


Figure 45: Displacement of a Displayed Point versus Head Position -- Head position, *Ix*, varies from –1 to 1; *r* is 0.5; eye-separation is 0.065. Plots are drawn for a model point at various z coordinates.

In Figure 45, F_x-E_x is plotted against I_x. I_x varies from –1 to 1; *r* is 0.5; eye-separation is 6.5cm. Plots are drawn for a model point a E_z=0.10 (solid), E_z =-1 (dashed), E_z =-10 (dotted) and E_z =-100 (dash-dot). Sensitivity to head position grows with object depth, with E_z =0.10m ranging up to 0.05m and E_z =-100 m ranging up to –50 m. Figure 46 illustrates this more intuitively. Here increasing the depth of the modeled point increases the lateral displacement of the displayed point.

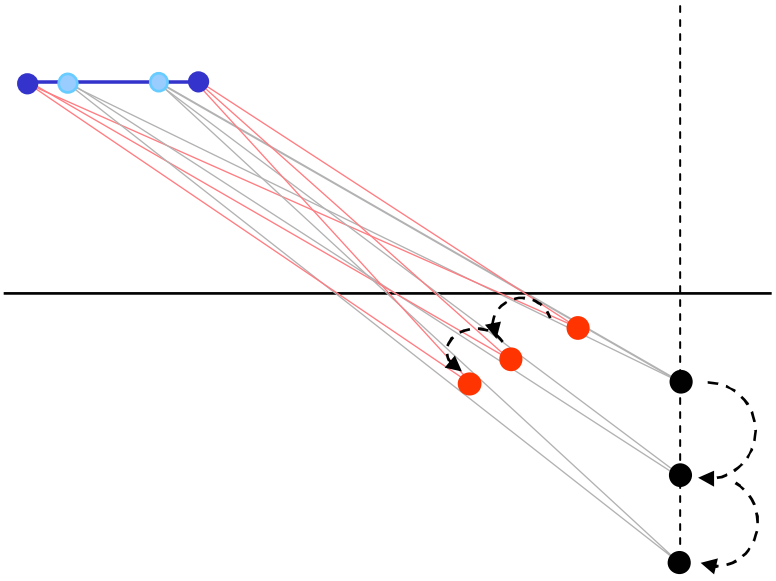


Figure 46: Effect of Increasing the Depth of the Modeled Point on Lateral Displacement the Displayed Point.

Figure 47 (page 110) shows the effect of different values for *r* for a model point at (0,0,-10). In Figure 47A, *r* is 0.75 (solid), 0.5 (dash-dot), 0.25 (dash) and 0.125 (dot). In Figure 47B, r is 1 (solid), 2 (dash-dot), 4 (dash) and 8 (dot). Generally, as we move away from using true eye separation, *r*=1, the shifting grows more sensitive to head movement. Figure 48 (page 111) show this pictorially. When changing the modeled eye separation from the outer modeled eye separation to the inner one, the lateral displacement of the displayed point increases. Note in Figure 47 also the change from positive to negative slope as *r* goes from less to greater than one. This represents a reversal in the direction of the shifting.

This discussion illustrates the behavior of the distortions shifting. The plots show the shift grows quite large especially for modeled eye-separations far from the true value (*r*=1).

2.5 Distortion Implications

Δ does not preserve distance, angles nor parallelism. From many applications this may be problematic. In command-and-control applications, for example, users often demand undistorted views of

terrain. In a CAD application, a user designing what she perceives to be as a cube may actually have designed a more general truncated pyramid. Equivalent to Woods et al. [Wood93] observations in

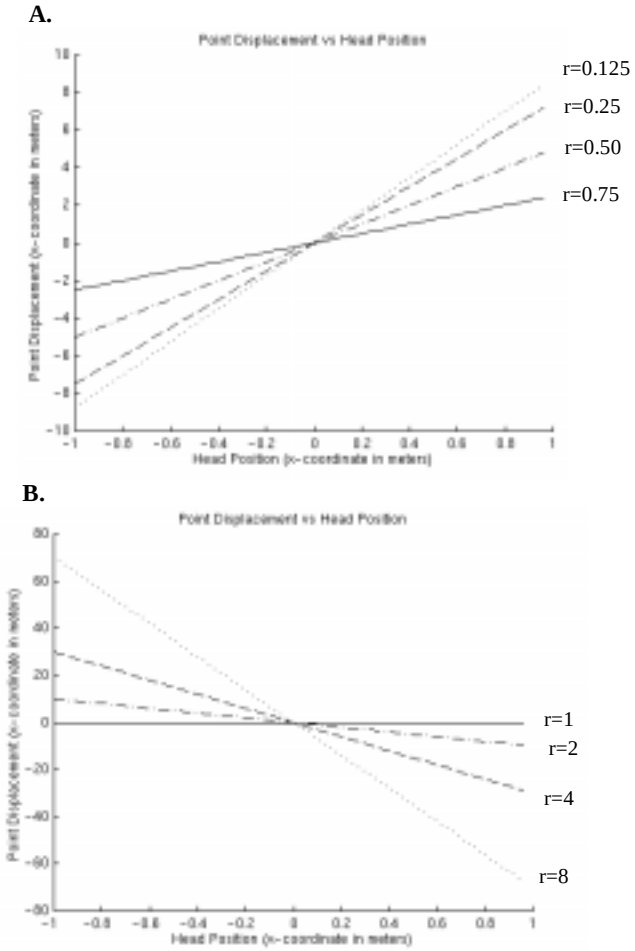


Figure 47: Displacement of a Displayed Point versus Head Position -- The modeled point is at (0,0,-10). *r*, the eye separation ratio, is varied over a range of values.

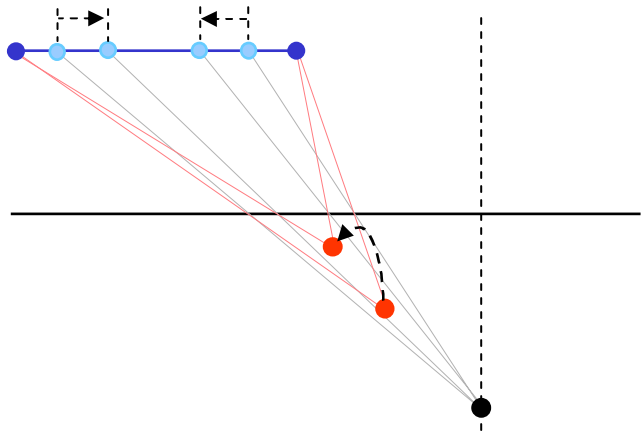


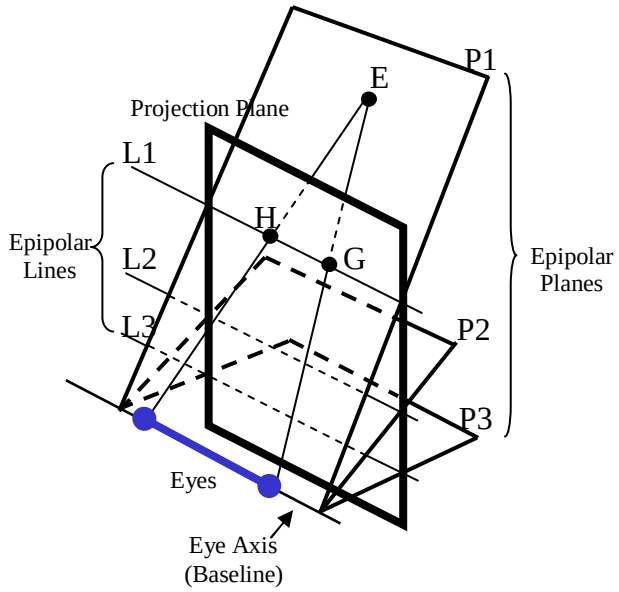
Figure 48: Effect of Modeled Eye Separation on Lateral Displacement of the Displayed Point.

teleoperator environments, perceptions of velocity through the environment could be distorted given this non-linear distortion. Multi-screen environments such as a CAVE [Cruz93] could be especially problematic. Since Δ is relative to a particular view plane, each screen would distort the world in a different manner. A virtual object which spans two adjacent screens would be distorted differently by each screen creating further visual anomalies. Finally in any stereoscopic HTD application that uses 6 DOF input devices, the distortions from false eye separation will ruin the correspondence between the physical device and its virtual representation.

3 Image Scaling and Image Shifting

This section examines image scaling and image shifting and compares them to false eye separation. We will show that image scaling and image shifting have a number of geometric side effects not present in either false eye separation nor view scale and view placement manipulations. This indicates that false eye separation combined with view scale and view placement are a better choice for image fusion control in stereo HTDs. Therefore we analyze these distortions in less detail than we analyzed false eye separation.

A



B

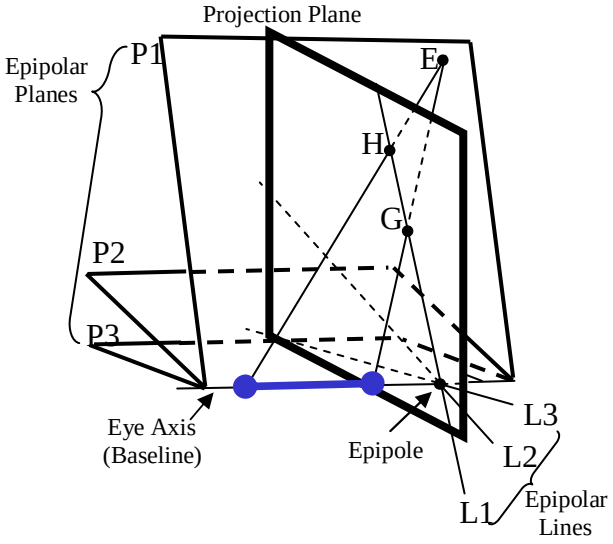


Figure 49: Epipolar Geometry of Planar-Coincident Stereo Display -- In (A) the baseline is parallel to the projection plane while in (B) it is not.

Epipolar geometry, a key concept in stereoscopic imaging [Milan99,page 458], is particularly useful in examining image scaling and image shifting. Figure 49 (pg 112) briefly reviews the terminology. The line through the eyes is called the baseline or eye axis. Planes through this line are the epipolar planes (e.g. planes **P1**, **P2** and **P3**). The intersections of the epipolar planes with the projection plane are the epipolar lines (e.g. lines **L1**, **L2** and **L3)**. Epipolar lines are parallel if and only if the baseline is parallel to the projection plane as in Figure 49A. Epipolar lines and the baseline intersect with the projection plane at the epipole. In Figure 49A, the epipole is the ideal point (or point at infinity) common to lines parallel to the baseline. In Figure 49B, the epipole is a regular point. Any point in three space has an associated epipolar plane and epipolar line. In Figure 49A and B, the point **E** has epipolar plane **P1** and epipolar line **L1**. The projected image points, **H** and **G,** of a 3D point, **E**, always lie on the 3D point's epipolar line, **L1**. This is called the epipolar constraint.

3.1 Image Scaling Geometric Construction

The suggested implementation for image scaling (called frame magnification in [Sou92]) is to compose the following matrices:

$$M = M_{scr}\, S_{mag}\, M_{proj}\, M_{view}\, M_{model} \qquad (7)$$

In our notation, $\boldsymbol{M_{model}}$ maps model coordinates to world coordinates. $\boldsymbol{M_{view}}$ maps world coordinates to view coordinates. $\boldsymbol{M_{proj}}$ maps view coordinates to the canonical projection coordinates and $\boldsymbol{S_{mag}}$ is the image scaling. Note, $\boldsymbol{S_{mag}}$ only scales the x and y coordinates by a common scale factor. A scale factor less than 1 scales down the x and y coordinates which is equivalent to enlarging the modeled projection window size. Hence the name "frame magnification." $\boldsymbol{M_{scr}}$ maps canonical projection coordinates to the device dependent screen coordinates. (Note, Southard's notation uses row vector notation so our presentation is the reverse order of his, and also he combines the image scaling scale, $\boldsymbol{S_{mag}}$, and $\boldsymbol{M_{proj}}$ into a single matrix which he labels $\boldsymbol{N_{proj}}$.)

$\boldsymbol{M_{scr}}$ contains scales and translations [Sou92, pg278] and is invertible. So:

$$M_{scr}\, S_{mag} = (M_{scr}\, S_{mag}\, M_{scr}^{-1})\, M_{scr} \qquad (8)$$

The right hand side equation is simply a scale about the center of the final window in screen coordinates. Assuming that all components of the viewing hierarchy are correctly measured, a scale about the screen window center is equivalent to a 2D scale of the projected image about the origin of Projection Plane Coordinate System (Figure 50, pg 115)).

We can describe the stereoscopic distortion induced by image scaling with the geometric construction illustrated in Figure 50. Figure 50 is an abstract diagram of a user viewing a stereoscopic display. The diagram is drawn from an overhead point of view looking down on the user. The diagrams uses the same color conventions established in Section 2.1 of this chapter. (Note, Figure 50 only shows a portion of the projection window so the window does not actually appear centered in the diagram). **I** is the eye axis center. **D** is the left eye displaced by vector **d**. **A** is the right eye displaced by vector –**d**. **E** is a modeled point on a virtual object. The modeled point is first projected on the projection window to points **H** and **G**. Image scaling then scales these points by factor *s* about the origin of the coordinate system. This yields points **H′**=*s*·**H** and **G′**=*s*·**G**. These scaled points are the points actually displayed to the user. Ideally **H′** and **G′** could also be generated by mapping modeled point **E** to displayed point **F** and then projecting**.** **F** would be at the intersection of the lines **AH′** and **DG′**. Unfortunately image scaling introduces a problem which is not evident in this 2D diagram: the red rays **AG′** and **DH′** do not generally intersect when the eye axis is not parallel to the projection plane. This is because the direct image manipulation violates the epipolar constraint. To deal with this complication, we analyze the parallel case and the more general, non-parallel case separately.

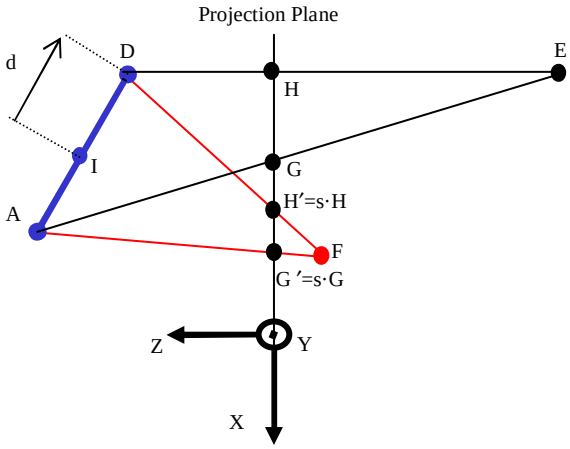


Figure 50: Effect of Image Scaling (Simplified).

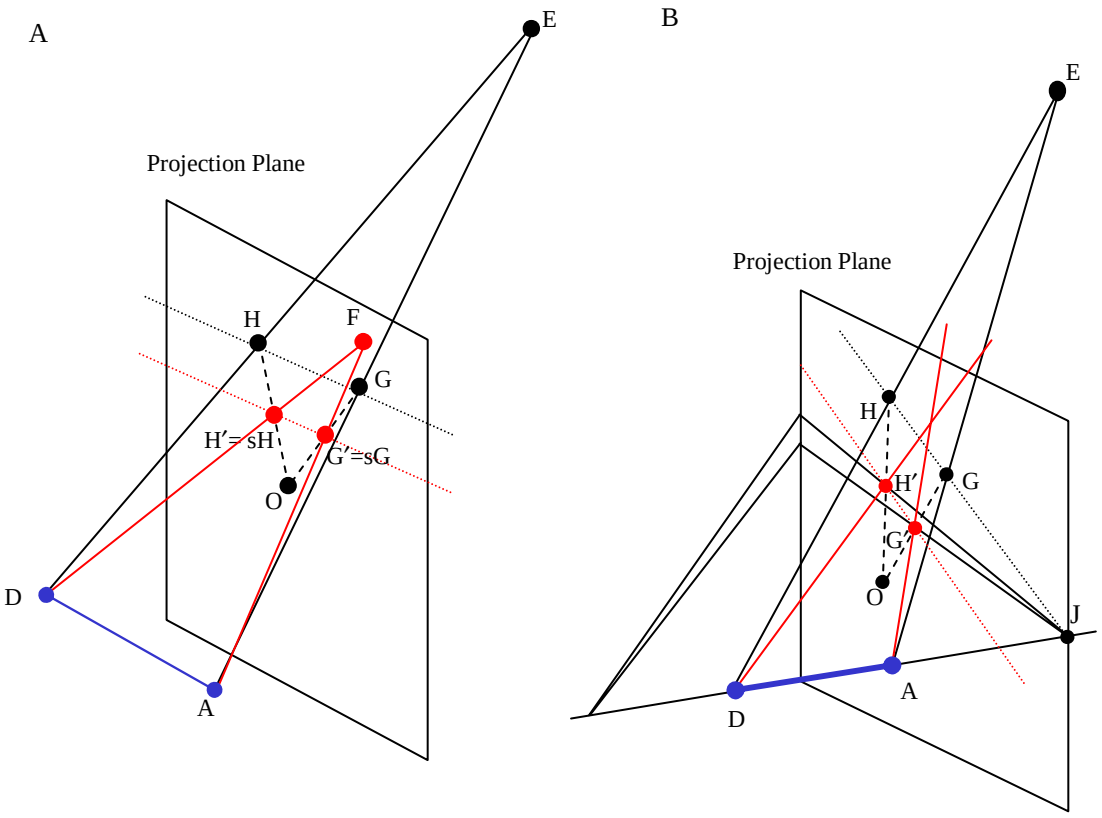


Figure 51: Effect of Image Scaling on Epipolar Geometry.

First we verify that in the parallel case the rays **AG′** and **DH′** do intersect. Figure 51A (page 115) illustrates the parallel case construction in 3D. **O** is the origin of the Projection Plane Coordinate System. **E** projects onto **H** and **G**. Since **AD** is parallel to the plane, **HG** is parallel to **AD**. (This occurs since for a line *l* (here **AD**) parallel to a plane *p* (here the projection plane), any plane *q* (here **ADGH**) containing *l* intersects plane *p* in another line parallel (here **HG**) to *l*.) Next, line **H′G′** is parallel to line **HG** because a uniform scale preserves angles. By transitivity **AD** is then parallel to **H′G′**. Hence there is a plane containing **AD** and **H′G′** and lines **AG′** and **DH′** are coplanar. Since coplanar lines intersect, **AG′** and **DH′** intersect. (Note, we are assuming a projective geometry where even parallel lines intersect at their ideal point). The construction defines a mapping on projective 3-space.

On the other hand in the *non*-parallel case the rays **AG′** and **DH′** typically do *not* intersect. Figure 51 B illustrates the construction when the projection plane and eye axis are not parallel. Assume **G** and **H** are not collinear with **O**. Scaling **G** and **H** to **G'** and **H'** yields a line **G'H'** which is parallel to **GH** since uniform scales preserve parallelism. Since lines **G'H'** and **GH** are parallel, they have no points in common; in particular **G'H'** cannot intersect the epipole **J**. So **G'H'** is not an epipolar line. Rather **G'** and **H'** lie on separate epipolar lines **JG'** and **JH'**. Hence **AG'** and **DH'** are in separate epipolar planes and are skew.

This lack of an intersection makes analysis of the complete 3D distortion more difficult. However, there is an atypical subcase where **AG′** and **DH′** will intersect even when the eye axis is oblique with respect to the projection plane. The subcase occurs whenever an epipolar line intersects the origin, **O**. When scaled, points on such an epipolar line move along the same epipolar line preserving the epipolar constraint. So 3D points on an epipolar plane that contains the origin will not yield intersection problems. Unless the epipole coincides with the origin, only one such epipolar line and plane will exist. Section 5.2 of this chapter analytically examines a case where one epipolar line intersects origin and will show that the planar geometric distortion does not preserve lines. Section 5.2 will also examine the vertical visual angles (VVA) between the epipolar planes of **G'** and **H'** and compare these values against VVA fusion limits from prior work.

3.2 Image Shifting Geometric Construction

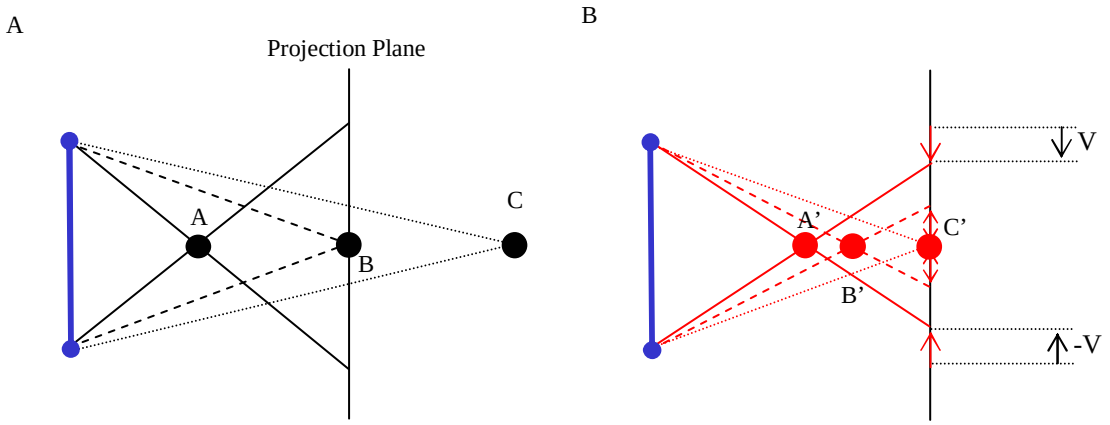


Figure 52: Image Shifting Reducing the Maximum Absolute Screen Parallax -- Figure A contains 3 points, **A**,**B** and **C**, at their modeled position with projectors indicating their left and right eye images. Figure B translates the points' left and right eye images by **V** and –**V** to yield different displayed points, **A′**, **B′** and **C′**. The maximum absolute screen parallax given by **A** is been reduced.

Next we examine image shifting. Image shifting translates the two projected stereo images toward each other. Image shifting is particularly effective when the scene only contains distant geometry in far space. The images can be translated so that the minimum positive parallax becomes zero. All other screen parallaxes in the scene will be reduced. If geometry exists in front and behind the screen, image shifting may both reduce screen parallax for some modeled points and increase screen parallax for other modeled points. If done carefully, however, this technique can be used to reduce overall *maximum absolute* screen parallax. Figure 52 illustrates this possibility. In Figure 52A two eye points view three virtual points **A**, **B**, and **C**. The projectors for these points are drawn as black lines distinguished by different line styles. In Figure 52B image shifting is applied to the projected images of these points. The translation vector is **V** for the left eye image and –**V** for the right eye image. In this example we choose **V** to equal half the screen parallax of point **C**. As a result the new point, **C′**, now has zero screen parallax. Figure 52B shows the effect of this image shift on all three points. Note that while point **B**'s screen parallax

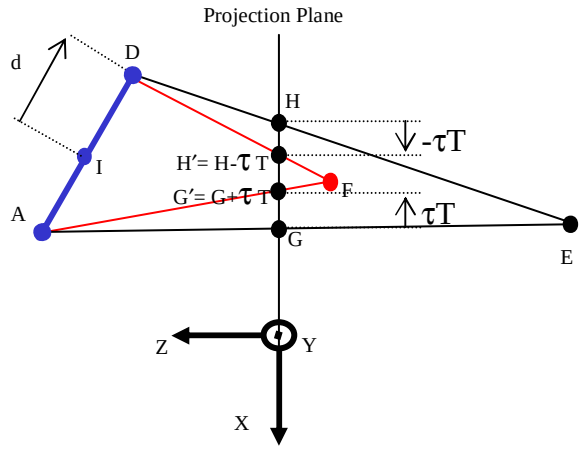


Figure 53: Effect of Image Shifting (Simplified)

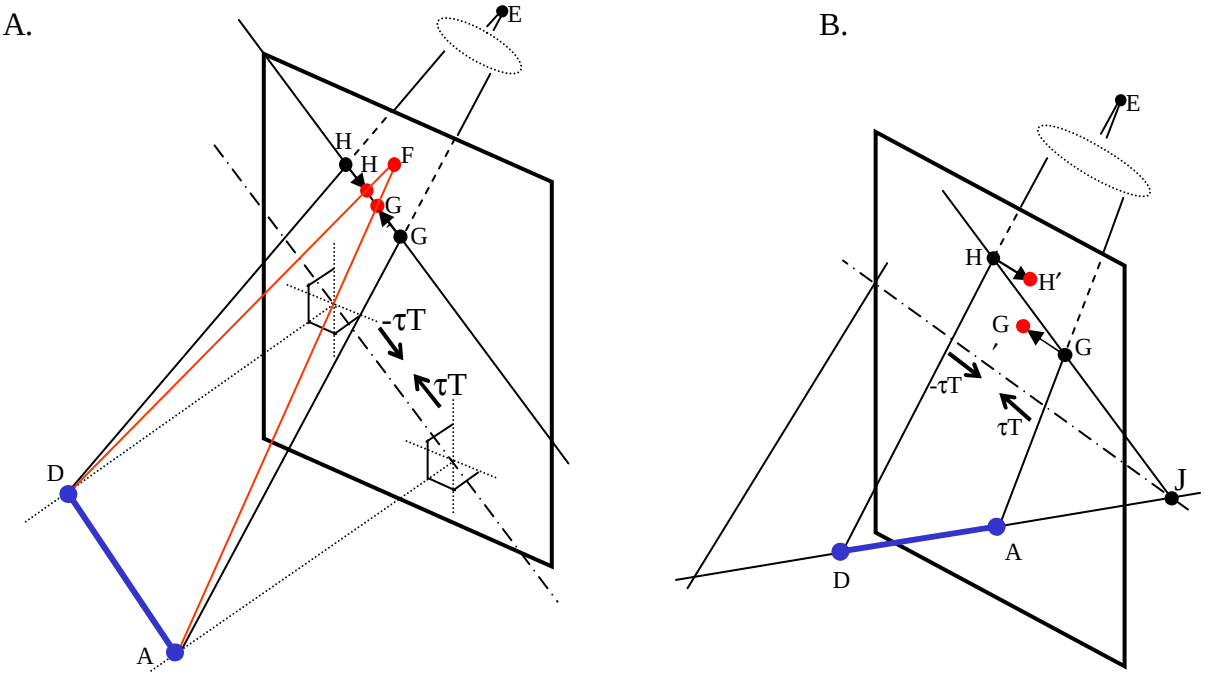


Figure 54: Effect of Image Shifting -- (A) Eye axis is parallel to the screen. (B) Eye axis is not parallel to the screen

increases, from zero to 2**V**, the overall maximum absolute screen parallax goes down. In Figure 52B the absolute screen parallax of **A'** is smaller than that of **A** in Figure 52A. Hence image shifting can potentially be used to reduce fusion problems in scenes with geometry on both sides of the projection plane.

To understand how image shifting distorts displayed 3D space for a stereo HTD we must be careful about the direction of the translation. An intuitive choice is to translate parallel to the perpendicular projection of the eye axis on the projection plane. Figure 54A (pg 118) shows this projected axis as a dashed-dotted line. Let **T** be the unit vector on this axis pointing in the direction of the left eye and let τ be the magnitude of the desired translation. The left eye image is translated by vector $-\tau\mathbf{T}$ and the right eye image by $\tau\mathbf{T}$.

We can describe the stereoscopic distortion induced by image shifting with the geometric construction illustrated in Figure 53. Figure 53 is a highly abstract diagram of a user viewing a stereoscopic display. The color and labeling follow earlier conventions. Modeled point **E** is projected onto the projection plane to points **H** and **G**. Image shifting translates these to points $\mathbf{H}'=\mathbf{H}-\tau\mathbf{T}$ and $\mathbf{G}'=\mathbf{G}+\tau\mathbf{T}$. These are the points displayed to the user. Ideally **H′** and **G′** could also be generated by mapping modeled point **E** to displayed point **F** and then projecting**.** **F** would be at the intersection of the lines **AH′** and **DG′**. Unfortunately, the lines **AG′** and **DH′** only generally intersect if the eye axis is parallel to the projection plane. Due to this complication we analytically examine the parallel case and the non-parallel case separately in Sections 3.3 and 3.4 of this chapter.

First, we verify that in the parallel case lines **AG′** and **DH′** generally intersect (Figure 54A). Since **AD** is parallel to the projection plane, all epipolar lines are parallel. So point **E**'s epipolar line **HG** is parallel to the projected eye axis (dashed-dot line). Because **HG** is parallel to the projected eye axis, when **H′** and **G′** are shifted along **T** they remain on the same epipolar line and plane. This guarantees the lines **AG′** and **DH′** intersect or are parallel. This construction defines a mapping on 3-space.

In the non-parallel case (Figure 54B), however, epipolar line **HG** is not generally parallel to the projected eye axis along which the 2D images are translated. Consequently, applying the translations $-\tau\mathbf{T}$

and τ**T** to the left and right eye images yields points not on the line **HG**. Typically, **G′** and **H′** will be on separate epipolar lines and planes which violates the epipolar constraint and leaves lines **AG′** and **DH′** skew. A rare exception occurs for the one epipolar line that happens to be parallel to the projected eye axis. Section 3.4 of this chapter shows that in this case where **AG′** and **DH′** do intersect, the resulting distortion does not preserve lines. Section 3.4 also examines the vertical visual angles (VVA) between the epipolar planes of **G'** and **H'** and compares these values against VVA fusion limits from prior work.

3.3 Analysis of Parallel Case

This section presents analytic descriptions of image scaling and image shifting based on the geometric constructions of the prior two sections.

Appendix B Section 1 uses Mathematica [Wolf96] to obtain the analytic distortion of image scaling for an eye axis which is parallel to the projection plane. The resulting equation is parameterized on the central eye position, **I**, the vector to the left eye, **d**, and the scale factor *s*, as shown in Figure 50 (pg 115). Using column vector notation the matrix in Projection Plane Coordinates is:

$$\Delta_{sc}^{p} = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & \frac{s-1}{I_z} & 1 \end{bmatrix} \tag{9}$$

Δ_{sc}^{p} is a non-affine collineation so it preserves straight lines but not parallelism. Δ_{sc}^{p} contains no translation nor dynamic shearing components. Δ_{sc}^{p} also varies with head to screen distance, I_z,. Δ_{sc}^{p} contains a static uniform scale component indicated by the repeating *s* along the diagonal. We observed that objects will appear not only to be compressed in depth, but also to change in overall size. For example a virtual box anchored to the display plane will shrink in all dimensions and appear as a tiny box. More generally, a modeled scene shown at true life-size may be shrunk and then appear to be a small toy model.

This also alters the effective field of view. For instance, imagine that in the original scene, a portion of the façade of a building fills up the screen. Image scaling may shrink the scene so that the user sees the entire building at a smaller scale and she may see around the sides of the building as well. Neither false eye separation nor image shifting has these effects. This empirical observation correlates with the fact that these other technique's distortion matrices either do not have a uniform scale component or the uniform scale component is not dependent on the technique's fusion parameter (such as modeled eye separation for false eye separation or the translation factor for image shifting). Therefore while the other techniques allow manipulation of the regular view scale factor independent of the fusion technique, image scaling does not. Note, that the presence or absence of a uniform scale is an affine transformation component. This is separate from the fact that when used for homogenous coordinate transformations, a 4 by 4 matrix is only unique up to a scale factor. The latter is true because when a 4 by 1 homogenous coordinate vector is multiplied by a matrix and then reduced to 3 space, the division by W causes any common multiplicative factor from the matrix to cancel out.

Appendix C Section 1 uses Mathematica [Wolf96] to find an expression for the distortion induced by image shifting when the eye axis and projection plane are parallel. The distortion is parameterized on the eye axis center, **I**, the vector to the left eye, **d**, and the translation distance τ (Figure 53, pg 118). Using column vector notation the matrix in Projection Plane Coordinates is:

$$\Delta_{sh}^{p} = \begin{bmatrix} Q & 0 & -\frac{I_x}{I_z}\tau & I_x\tau \\ 0 & Q & -\frac{I_y}{I_z}\tau & I_y\tau \\ 0 & 0 & Q-\tau & I_z\tau \\ 0 & 0 & -\tau/I_z & \tau+Q \end{bmatrix} \qquad (10)$$

where

$$Q = \sqrt{d_x^{\,2} + d_y^{\,2}}$$

Δ_{sh}^{p} is a non-affine collineation and hence does not preserve parallelism. Δ_{sh}^{p} contains a translation component in the fourth column. This is to be expected since Figure 52 showed that points in the projection plane are moved out of the plane. Hence in-screen geometry is not preserved. This can be detrimental for applications which utilize the physical plane of the screen as a work surface for two-dimensional interactions such as precise curve drawing or laying route points on a map. Finally there are dynamic, head position dependent shearing components $-I_x/I_z\ \tau$ and $-I_y/I_z\ \tau$. Neither of these artifacts occur in image scaling and the first does not occur in false eye separation. Also, note that while Δ_{sh}^{p} contains x,y and z scale factors only the z scale factor varies with the fusion parameter, τ, and the x,y factors only depend on head orientation. The authors observed that image shifting does not lead to the same overall scaling effect as does image scaling although the depth dimension is, of course, noticeably altered.

A pictorial comparison of the distortion artifacts of image scaling, image shifting and false eye separation is illustrated in Figure 55 and Figure 56. All these diagrams share the same color coding and format. Figure 55 illustrates how perpendicular head motion affects the displayed 3D image for each fusion control method, while Figure 56 illustrates how lateral head motion affects the image. The projection plane is the middle black line. The eyes are at the top in blue. The modeled grid is the square in black and the displayed grid is the trapezoidal in red. All images assume a true eye separation of 6.5 cm, a typical average value [Lipt82,p44]. In A and B, image scaling is applied with scale of 0.5. In C and D, false eye separation is used. The eye separation is underestimated by one-half its true value. In E and F, image shifting is used with τ=1.625 cm. All fusion method parameters are set to reduce the nominal, maximum positive parallax of 6.5 cm to 3.25 cm, one half of its value. (Recall that in a properly calibrated system the maximum possible positive parallax equals the modeled eye separation.)

Figure 55 illustrates that all three techniques exhibit dynamic artifacts under perpendicular head motion. In contrast, in Figure 56 image scaling (A,B) does not exhibit dynamic artifacts under lateral head motion but false eye separation (C,D) and image shifting (E,F) do exhibit dynamic artifacts under lateral head motion. Since these are 2D diagrams only a side to side head movement is shown, but similar results occur for head movement in any direction parallel to the screen.

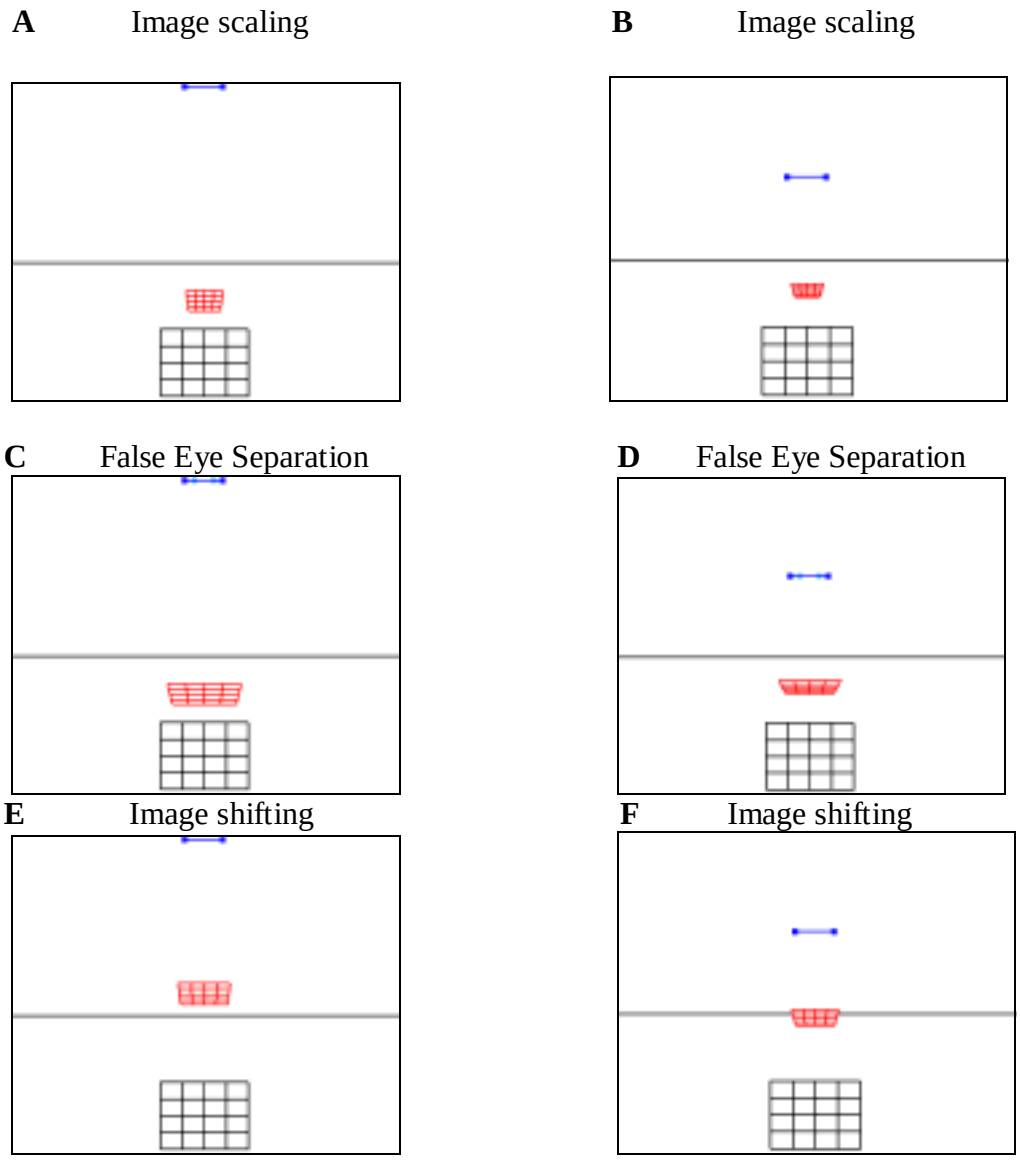


Figure 55: Distortion Due to Perpendicular Head Motion for Various Fusion Control Techniques -- (A) and (B) use image scaling with scale factor 0.5. (C) and (D) use false eye separation with eye separation ratio 0.5. (E) and (F) use image shifting (see Section 5) with translation magnitude 1.625 cm.

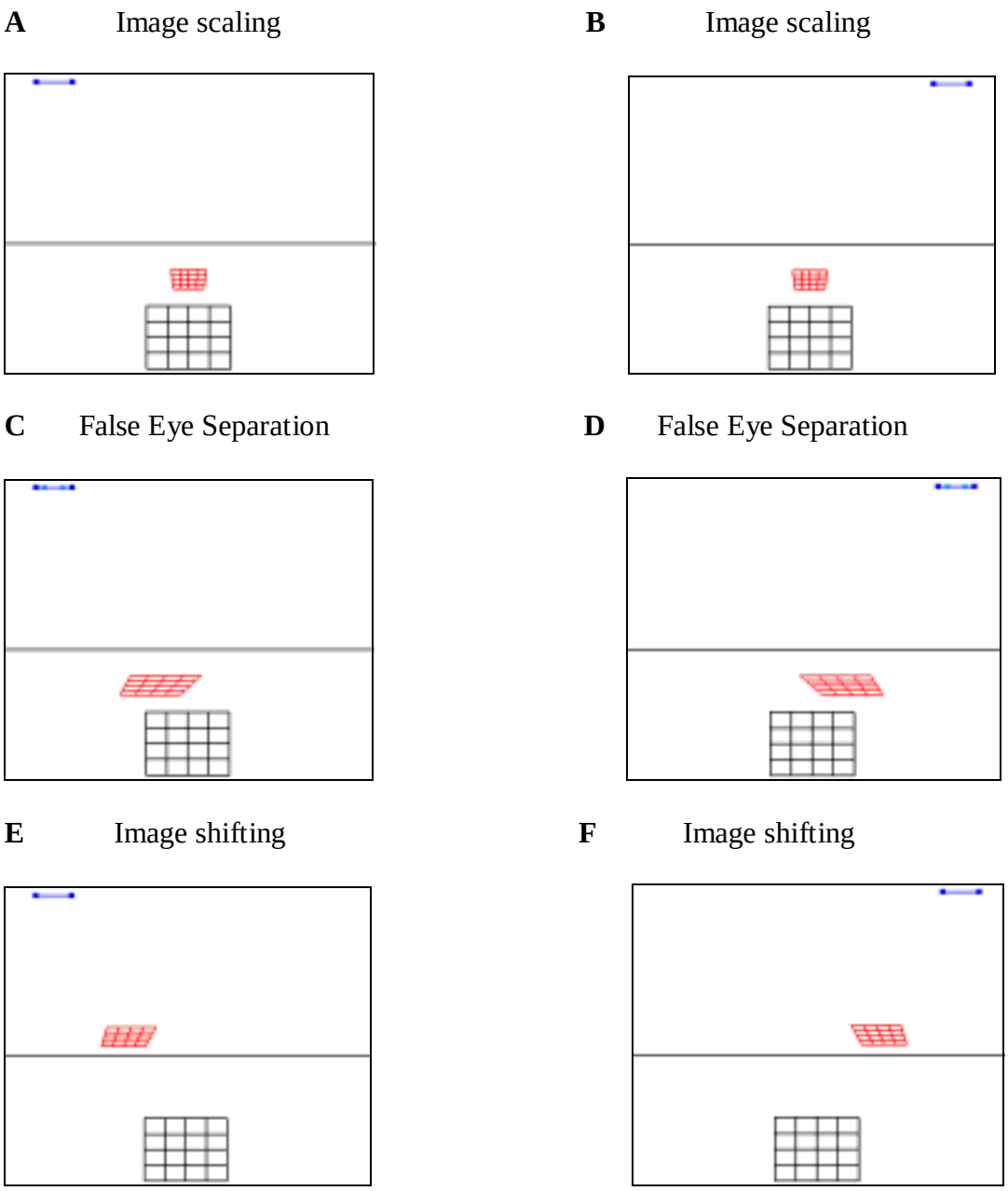


Figure 56: Distortion Due to Lateral Head Motion for Various Fusion Control Techniques --(A) and (B) use image scaling with scale factor 0.5. (C) and (D) use false eye separation with eye separation ratio 0.5. (E) and (F) use image shifting with translation magnitude 1.625 cm.

To summarize, in the parallel case image scaling has fewer dynamic distortion artifacts than image shifting and false eye separation. However, image scaling has a uniform scale component that can change the apparent size of the world and hence the amount of scene seen by the user. In contrast, false eye separation and image shifting have a much smaller effect on the relative field of view and does not change it as noticeably. If an application designer desires a fusion control method to alter the displayed scene as subtly as possible then false eye separation technique is a better choice than image scaling due to the lack of an overall scale. Additionally, if it is important to control the uniform scaling component independently of the fusion control adjustment such as in Ware et al. [Ware95a] or Chapter IV of this thesis, then false eye separation is also a better choice since it allows independent control of the perceived uniform scale factor through the Platform Coordinate System scale. The next section shows that if the eye axis is not parallel to the screen, image scaling and image shifting have some troublesome geometric artifacts not found in false eye separation.

3.4 Analysis of the Non-Parallel Case

This section considers the analytic distortions in the general case when the eye axis is not parallel to the screen. First we discuss image scaling. In the non-parallel case, image scaling in general violates the epipolar constraint. An exception occurs when an epipolar plane contains the origin of the scale. This section first examines such a special case when the modeled point and the eyes are in the XZ plane. This restriction is somewhat arbitrary but it allows simpler computation and serves to illustrate the induced curvature.

For the restricted XZ plane case, using Mathematica [Wolf96]] Appendix B, Section 2 finds the analytic distortion to be:

$$\Delta_{\mathbf{sc}}^{xz}: \qquad F_x = \frac{\left(E_x^2 K_1 + E_x K_2 + E_x E_z K_3 - E_z^2 K_4\right)}{w}$$

$$F_z = \frac{\left(E_x E_z K_5 \quad + E_z K_6 + E_z^2 K_7\right)}{w}$$

$$w = \left(E_x K_8 + E_z K_9 + E_z^2 K_{10} + K_{11}\right) \tag{11}$$

where K_1 through K_{11} are dependent on **I**,**d** and *s*

(see Appendix B, Section 2, page 191)

Recall we've restricted ourselves to the XZ plane so all y coordinates are just zero. **E** and **F** are the modeled and distorted points. **I** is the eye axis center and **d** the vector to the left eye. The scalar *s* is the image scaling factor. The coordinate equations are 2nd degree rational polynomials so image scaling maps lines in modeled space to curves in displayed space. This is shown in Figure 57. Figure 57 is another abstract overhead view looking down on the user. The eyes are at the top in blue. The projection plane is the black horizontal line. Figure 57 shows the modeled black linear grid (black) is mapped to a curved displayed grid (red).

While the curvature is quite noticeable in Figure 57 C and D is it less noticeable in Figures A and B. The author viewed a similar grid on a desktop-VR display using a similar scale factor. We did not observe curvature. Perhaps this is because in situations like Figures C and D, one is too close to the screen to see anything anyway. When farther from the screen, as in Figures A and B, the curvature is more subtle. Our inability to perceive the curvature from these larger distances is also less surprising if we consider the following. While image scaling distorts 3D space curvilinearly, image scaling only distorts the individual 2D images by a scale which of course preserves lines. So the 2D projected image of a straight line must remain a straight line. Any curvature imparted to the displayed 3D curve exists solely in the depth, or the Z component. For example in Figure 58, a line *l* in modeled space is distorted into a displayed curve *l'* by such a distortion; however, the curvature of l'_{xy} is always 0. Humans can perceive a

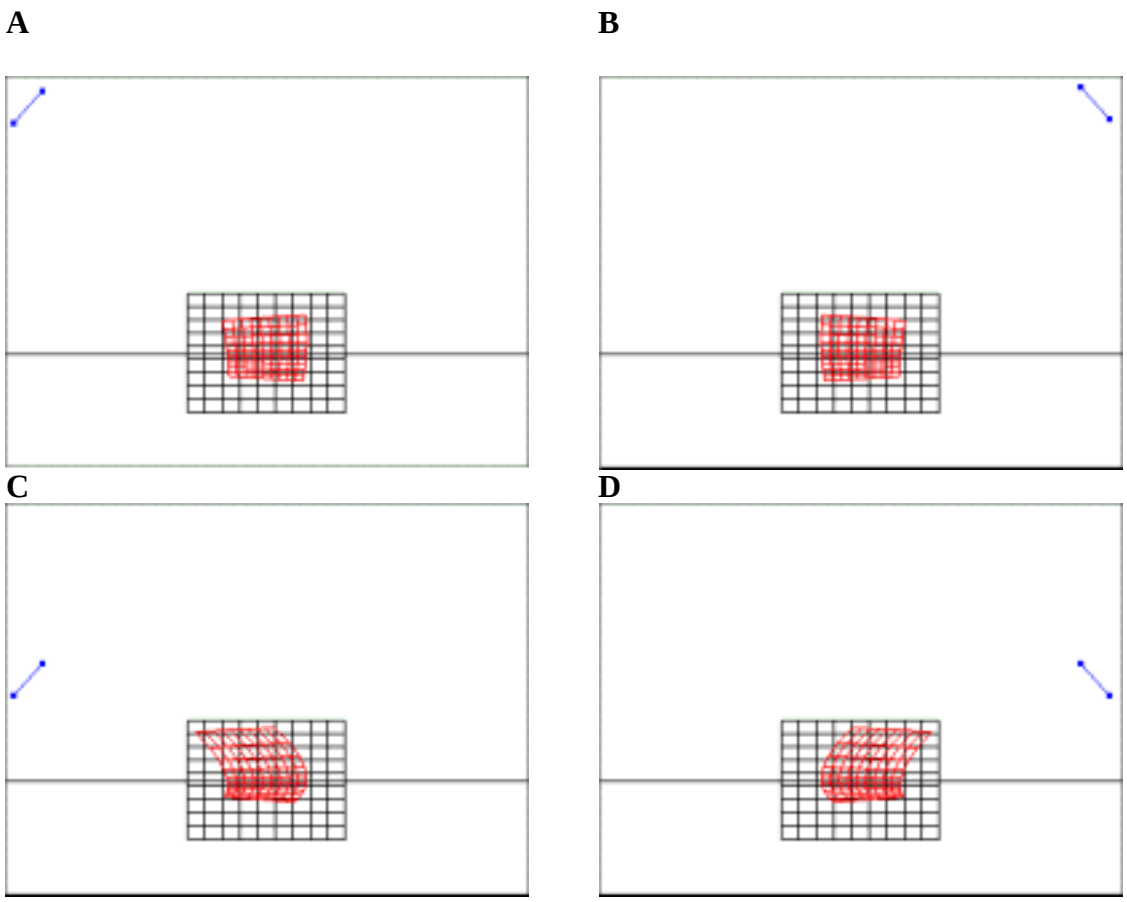


Figure 57: Curvature of Image Scaling

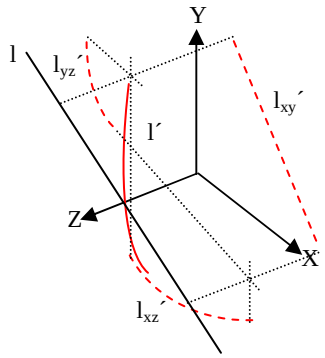


Figure 58: Curvature of Depth Dimension -- Some of stereo image techniques map a modeled line *l* to the equivalent displayed curve *l´*. However, the straightness of l_{XY}´, the XY projection of the displayed line *l´*, is preserved. It is the straightness of the XZ and YZ projection (i.e., l_{XZ}´ and l_{YZ}´) that is not preserved. The curvature exists only in depth.

curved surface whose curvature is indicated purely by stereopsis. Julesz [Jul71] illustrates a number of hyperbolic paraboloid and cosine surfaces whose curvature is indicated only by stereopsis cues. However perceiving this curvature probably involves a different mechanism than that used to distinguish curves from straight lines drawn on a piece of paper. Additionally variations in shading and texture provide strong cues to surface shape and in VR environments these cues are based on the modeled geometry not the stereoscopically distorted geometry. Since we could not observe the induced curvature in a simple wireframe scene when we were explicitly looking for such curvature, we suspect that in a more complex shaded scene a typical user would probably not notice these curvatures either. Clearly, however, only human factors studies over a wider variety of display configurations can more fully explore this issue.

We found similar results for image shifting. As discussed earlier in the non-parallel case, image shifting in general violates the epipolar constraint. An exception occurs for the epipolar line parallel to the direction of the image shift. One such case occurs when the eyes and modeled points are restricted to the XZ plane. This restriction is somewhat arbitrary but it allows simpler computation and serves to illustrate the theoretic induced curvature. Figure 59 illustrates the distortion. τ is set to 1.625 cm. There is a slight geometric curvature. Appendix C Section 2 contains the precise expression. Again, the coordinate

A B

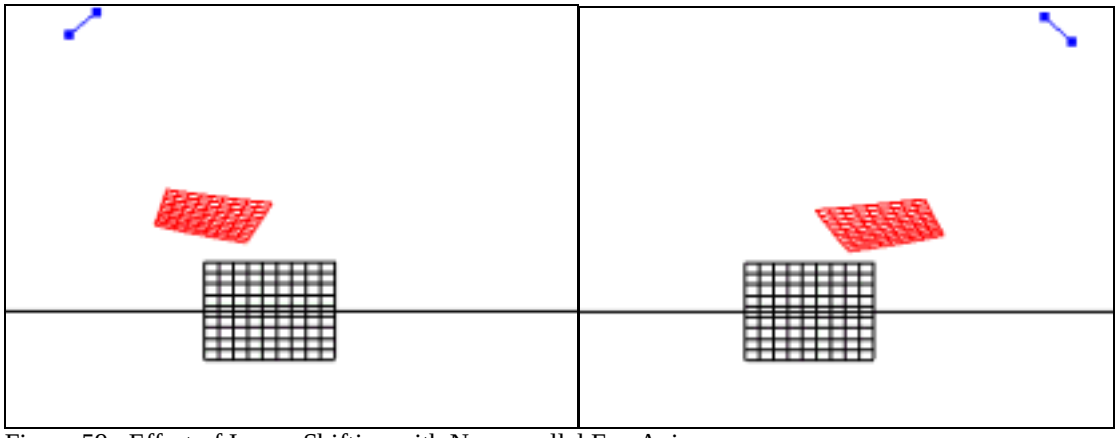


Figure 59: Effect of Image Shifting with Non-parallel Eye Axis.

equations are rational quadratic ones. The authors did not observe the curvature when viewing a similar grid on a desktop-VR setup. We suspect this occurred for the same reasons cited in the previous paragraph concerning image scaling.

Potentially more troublesome is the fact that image scaling and image shifting violate the epipolar constraint by displacing homologous points vertically with respect to the eyes. This situation, called vertical parallax, aggravates image fusion problems because humans can fuse only a very small range of vertical parallax. Experiments with random dot stereograms show that if vertical parallax is slowly increased until fusion breaks down, the average limit for vertical visual angle (VVA) is only 20 minutes of arc. Additionally, once breakdown does occur VVA must be reduced back to 6 minutes of arc for fusion to reoccur [Fend67][Jul71].

First we determine the VVA for image scaling as shown in Figure 51B (pg 115). **H** and **G** are the projections of a 3D modeled point onto the projection plane. **H'** and **G'** are **H** and **G** scaled about the window center, **O**. **J** is epipole. These points form epipolar planes **DJH** (same as **DJG**), **DJH'**, and **DJG'**. While **H** and **G** form one common plane, **H'** and **G'** lie in separate epipolar planes. The VVA separating **H'** and **G'** is the angle between epipolar planes **DJH'** and **DJG'** as measured about the line **AD**.

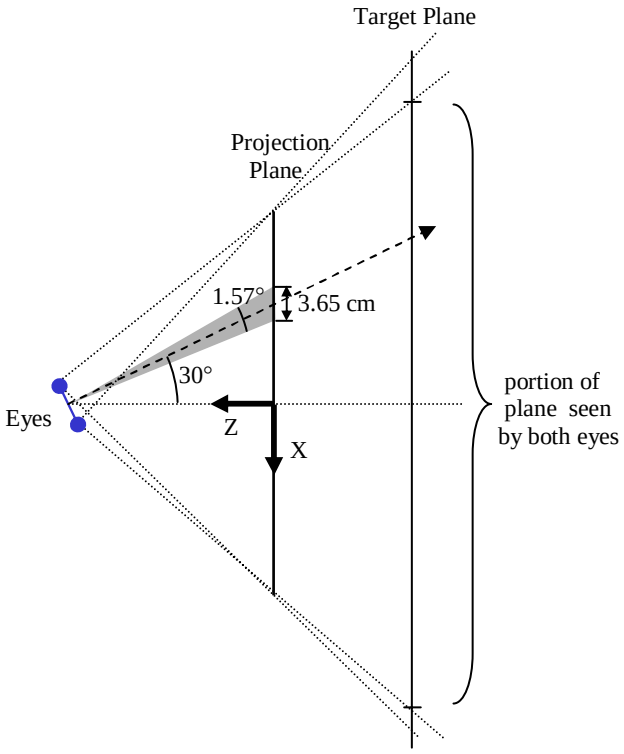


Figure 60**:** Example View Configuration used for VVA Plot —This is the view configuration used to create a 3D plot of VVA for a given plane in model space. The plane of the diagram is assumed to be parallel to the XZ plane but at the height of the eyes (see text). This diagram is not drawn to scale.

We will graph the VVA for image scaling using the following approach illustrated in Figure 60. We pick some specific viewing configuration that includes eye positions, screen position and screen size. Next we pick some target plane in space with a fixed z coordinate and we compute the x and y extents of the region in this plane that is viewable from both the left and right eye. In Figure 60, this area is delimited by horizontal marks on the target plane. The purpose of using a fixed z coordinate is to help show how VVA varies across the screen which is parallel to the target plane. We project points on the target plane onto the projection plane once for each eye (**H** and **G** in Figure 51B) and apply the fusion control technique to these projected points (yielding **H'** and **G'** in Figure 51B). We then compute the VVA for corresponding pairs of

altered projected points. This yields a VVA value for each point on the target plane. Rather than plotting the VVA against the X and Y coordinates of the points on the target plane, we plot VVA against the X, Y coordinates of the left eye's altered projected 2D image (**H'** in Figure 51B). This makes it easy to correlate VVA values with screen positions. The procedure for image shifting is quite similar.

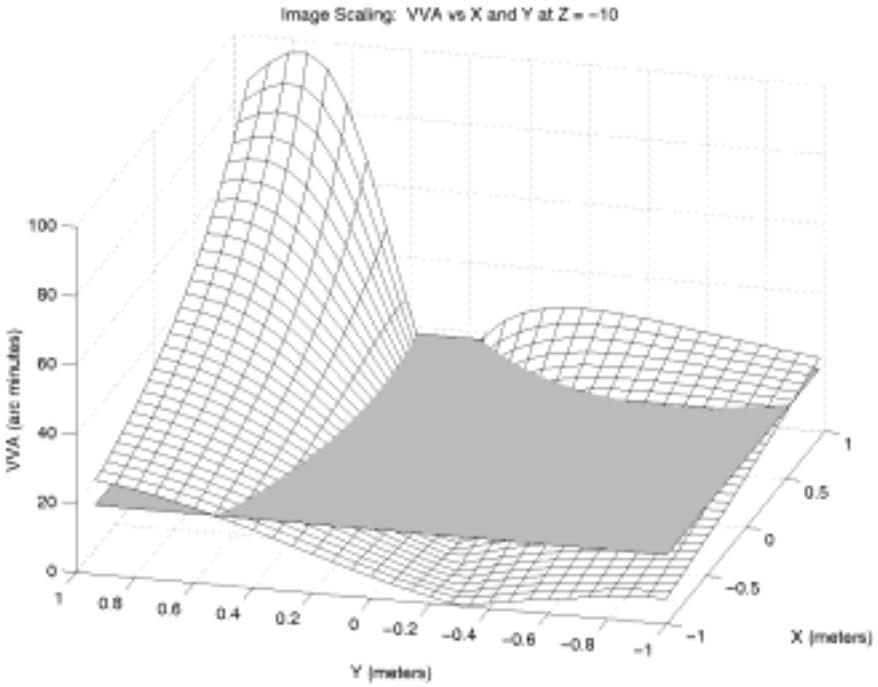
The following examples assume an upright 2 x 2 meter screen. The eye separation is 6.5 cm. The axis is twisted 30 degrees looking towards the left of the screen while remaining in the XZ plane. The axis center is located at (0,0.524,1) in projection plane coordinates. This accounts for a user standing 1 meter from the screen with eyes 1.524 meters (5 feet) off the ground. The target plane is at z = -10 (i.e. 10 meters behind the screen). Figure 60 shows an overhead view of the eye screen configuration. Note that distances are not to scale in order to make all relevant aspects legible. In order to approximate how much screen parallax may be generated beyond the recommended HVA limit, the traditional method assumes the eye axis is parallel to the screen and uses the perpendicular distance to the screen to map HVA to desirable screen parallax. In the above configuration, this yields a maximum possible positive screen parallax of 6.5 cm. An HVA limit of 1.57° then yields a desirable parallax of 2.74 cm. This implies an image scale factor of 0.4216 (desired/maximum) and image shift of 1.88 cm ((maximum-desired)/2). For arguments sake, we'll actually account for the rotated eye axis. Since the eye axis is not actually parallel to the screen it is no longer true that the maximum possible parallax equals 6.5 cm. By plotting parallax for very distant target planes ($z = -10^{13}$), we find the maximum possible parallax along the line of sight approaches 7.4 cm. As shown in Figure 60, at the given line of sight distance of 1.154 m (1/cos30°) to the screen, the recommended 1.57 degree HVA limit yields a desirable screen parallax of 3.65 cm. (HVA is measured within a given epipolar plane). This yields an image scale factor of 0.4938 and image shift of 1.87 cm. These values are more liberal than the values calculated with the traditional assumption since they are closer to the scale factor of 1.0 and a translation of 0. If these more liberal values yield VVA beyond the desirable VVA limits then so will the more stringent traditional values. We use the more liberal calculation.

Shortly we'll show that reducing screen parallax in the above viewing configuration using the calculated scale and shift values will create VVA values greater or equal to the VVA 20 arc minute limit. This is the acceptable threshold for vertical disparities found in random dot stereograms [Fend67][Jul71]. An important question, is whether such VVA values are displayable in pixels. Assume the screen is 1000 by 1000 pixels so that pixels are 2 mm tall. 20 arc minutes will subtend slightly different screen sizes depending what part of the screen is viewed because the eye to screen distance varies slightly. However, even at the closest distance of 1 meter, 20 arc minutes subtends 5.8 mm which is roughly 3 pixels. Hence, a VVA greater or equal to 20 arc minutes is significant pixel-wise.

Figure 61A (pg 133) shows the VVA graph for image scaling along with a horizontal plane at the 20 arc minute threshold. The other axes, X and Y, are the coordinates on the 2 x 2 meter screen. A scale factor value of 0.4938 achieves the desired reduction of screen parallax. The shape of the plot is typical. The surface touches the VVA=0 plane along a valley that runs diagonally across the screen. This corresponds to the epipolar line which intersects the origin of the scale. As discussed earlier this is a special case where the epipolar constraint is preserved and hence VVA equals zero. Moving away from the valley along Y, the VVA grows larger in either direction, but reaches a higher peak in the positive Y direction. VVA shrinks as the x-coordinate moves in the direction of head twist. In Figure 61A, the user's head is oriented to the left and the VVA shrinks as the x-coordinate moves to the left. This plot shows VVA meeting or exceeding the 20 arc minute limit for some y-position for every x-position. For the central view direction where x=0.5774, the VVA reaches 33 arc minutes at the top of the screen. The 6 arc minute re-fusion limit is far exceeded for an even larger portion of the space. Larger target plane distances such as 10000 km yield a very similar graph but with slightly steeper slopes and slightly higher VVA values; however, 10 m is sufficient to exceed the VVA threshold in the example configuration.

Figure 61B graphs VVA for the same eye-screen configuration while using image shifting. The horizontal gray plane shows the 20 arc minute threshold. An image shift value of 1.87 cm is used. The surface touches the VVA=0 plane along a valley that runs horizontally across the screen. This corresponds to the epipolar line which is parallel to the image translation direction. Moving away from the valley along

A



B

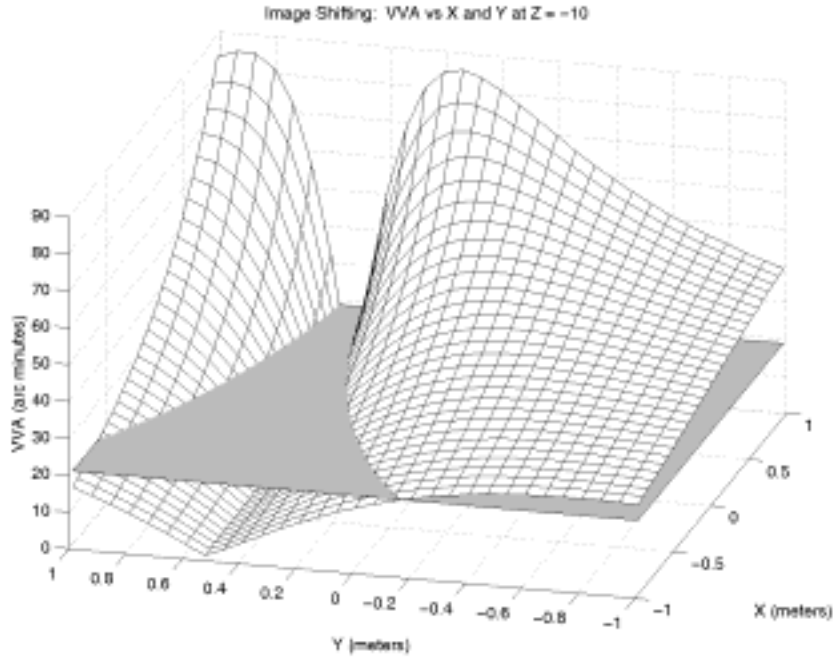


Figure 61: VVA for Image Scaling (A) and Image Shifting (B)—The chosen plane of points in modeled space is at Z=-10.

Y, the VVA grows larger rising at similar rates in either the positive or negative Y direction. VVA shrinks as the x-coordinate moves in the direction of head twist. For the central view direction where x=0.5774 (tan 30°), the VVA peaks at around 28 arc minutes.

Ideally geometric analysis would show either that image scaling and shifting yield acceptable VVA for all display configurations or that image scaling and shifting yield unacceptable VVA for all configurations. Unfortunately, this is not the case because VVA depends on many factors. There are a large number of independent variables including screen size, eye axis orientation, eye axis position, and target plane depth. As with any screen based visual angle, VVA grows smaller with farther viewing distances. VVA grows larger with larger eye axis angles and with greater target plane distances. Algorithmic variations further complicate geometric analysis. An image scaling algorithm can use either a fixed scale factor or a variable scale factor. A variable scale factor could be chosen to either avoid near or far fusion problems or both. Similarly an image shifting algorithm can use either a fixed translation factor or a variable translation factor. All these issues have to be considered in order to determine whether the image manipulation will geometrically exceed VVA limits for a particular application with a particular display configuration. These geometric results indicate that image shifting and image scaling could have detrimental effects in stereoscopic HTDs. Further empirical studies are needed in order to examine the subjective effects of such conditions on user viewing comfort.

In contrast to image shifting and image scaling, false eye separation does not induce VVA at all. False eye separation moves the centers of projection along the baseline (eye axis). This has the effect of moving all the projected image points along their original epipolar lines. Therefore, the epipolar constraint is always maintained.

CHAPTER VI

REMOVING DYNAMIC ASPECTS FROM FALSE EYE SEPARATION

1 Overview

The previous chapter showed that the distortions induced by image scaling, image shifting and false eye separation all dynamically change with head motion. This is vexing because one of the advantages of adding head-tracking to a stereoscopic display is to remove qualitatively similar dynamic distortions found in non-head-tracked displays. An interesting question is whether these dynamic components can be removed while retaining the desired reduction in screen parallax. Image scaling has qualitatively fewer dynamic components than image shifting or false eye separation since the former only changes with perpendicular head motion while the latter two change with both perpendicular and lateral head motion. Taken alone this indicates that image scaling might be an ideal starting point for investigating the removal of dynamic components. On the other hand, image scaling has a uniform scale factor that confounds the view scale factor. This makes it impossible to control the view scale independent of the fusion technique. False eye separation does not have this problem. Additionally, false eye separation does not have the geometric features of inducing curvature and creating vertical parallax. These considerations would indicate false eye separation would be a good starting point for further investigation. Historically, we investigated the removal of dynamic components in false eye separation prior to examining the distortion due to image scaling and image shifting. However, the above considerations indicate that attacking the dynamic components in false eye separation is at least as reasonable a first step as attacking those in image scaling. This chapter presents several methods we developed for removing the dynamic components of false eye separation. The methods are presented in the order of their historical development. Unfortunately while Section 2 and 3 show that it is possible to remove the dynamic components, empirically there is a tradeoff in doing so. This is discussed at the end of Section 3. There may be further useful compromises that are discussed in the conclusions and future work of the final chapter.

2 α-false eye separation

In the context of a rendering pipeline with a matrix stack such as

$$\boldsymbol{M}_{\boldsymbol{Model}}^{\boldsymbol{Screen}} = \boldsymbol{M}_{\boldsymbol{World}}^{\boldsymbol{Screen}} \cdot \boldsymbol{M}_{\boldsymbol{Model}}^{\boldsymbol{World}} \tag{12}$$

using false eye separation effectively induces the complete transformation:

$$\boldsymbol{M}_{\boldsymbol{Model}}^{\boldsymbol{*Screen}} = \boldsymbol{M}_{\boldsymbol{World}}^{\boldsymbol{Screen}} \cdot [\boldsymbol{\Delta}]_{\boldsymbol{World}} \cdot \boldsymbol{M}_{\boldsymbol{Model}}^{\boldsymbol{World}} \tag{13}$$

To remove some of the artifacts of false eye separation while maintaining the effect on screen parallax and depth, we can derive a predistortion transform $\boldsymbol{Q}$ to place on the matrix stack:

$$M_{World}^{Screen} \cdot [Q]_{World} \cdot M_{Model}^{World} \tag{14}$$

With false eye separation, this results in an effective matrix stack:

$$M'^{Screen}_{Model} = M_{World}^{Screen} \cdot [\Delta \cdot Q]_{World} \cdot M_{Model}^{World} \tag{15}$$

$\boldsymbol{Q}$ should cancel the undesirable aspects of $\boldsymbol{\Delta}$ while retaining the effect on screen parallax and displayed depth. $\boldsymbol{Q} = \boldsymbol{\Delta}^{-1}$ is not useful since it cancels all the effects of false eye separation including the desired ones. Next, the discussion of the maximum depth plane (Chapter V, Section 2.4.2) illustrated that the changes to displayed depth due to false eye separation are inherently non-affine in nature. Therefore, the non-affine aspect of $\boldsymbol{\Delta}$ is not removable. The lateral shearing effect, however, can be removed.

2.1 Δ_{Shear}^{-1} Predistortion

Predistorting the world by the inverse shear component of Δ will remove the lateral shearing. To extract this component, Δ is first decomposed into a shear of X and Y along Z, Δ_{shear}; a Z scale, Δ_{scale}; and a pure projection, $\Delta_{Project}$ (Appendix A, Section 2, pg 186). From the detailed decomposition, we can find the inverse of Δ_{shear}:

$$\Delta_{Shear}^{-1} = \begin{bmatrix} 1 & 0 & \frac{-(1-r)(I_x I_z + d_x\ d_z\ r)}{d_z^2 r^2 - I_z^2} & 0 \\ 0 & 1 & \frac{-(1-r)(I_y I_z + d_y\ d_z\ r)}{d_z^2 r^2 - I_z^2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

To predistort the image, the matrix stack we should build at run-time is:

$$M_{World}^{Screen} \cdot \left[\Delta_{Shear}^{-1} \right]_{World} \cdot M_{Model}^{World} \tag{17}$$

When using false eye separation, predistorting world space with Δ_{Shear}^{-1} cancels the shear component of Δ, yielding the complete effective transform:

$$M_{World}^{Screen} \cdot \left[\Delta_{Project} \cdot \Delta_{Scale} \right]_{World} \cdot M_{Model}^{World} \tag{18}$$

This technique removes the head dependent shearing. Figure 62 (page 138) illustrates this. Figures A and B show the original grid (black) and the displayed grid (red) without predistortion. Next C and D show the displayed mesh after the original mesh is predistorted into a new mesh (blue) by Δ_{Shear}^{-1}. Keep

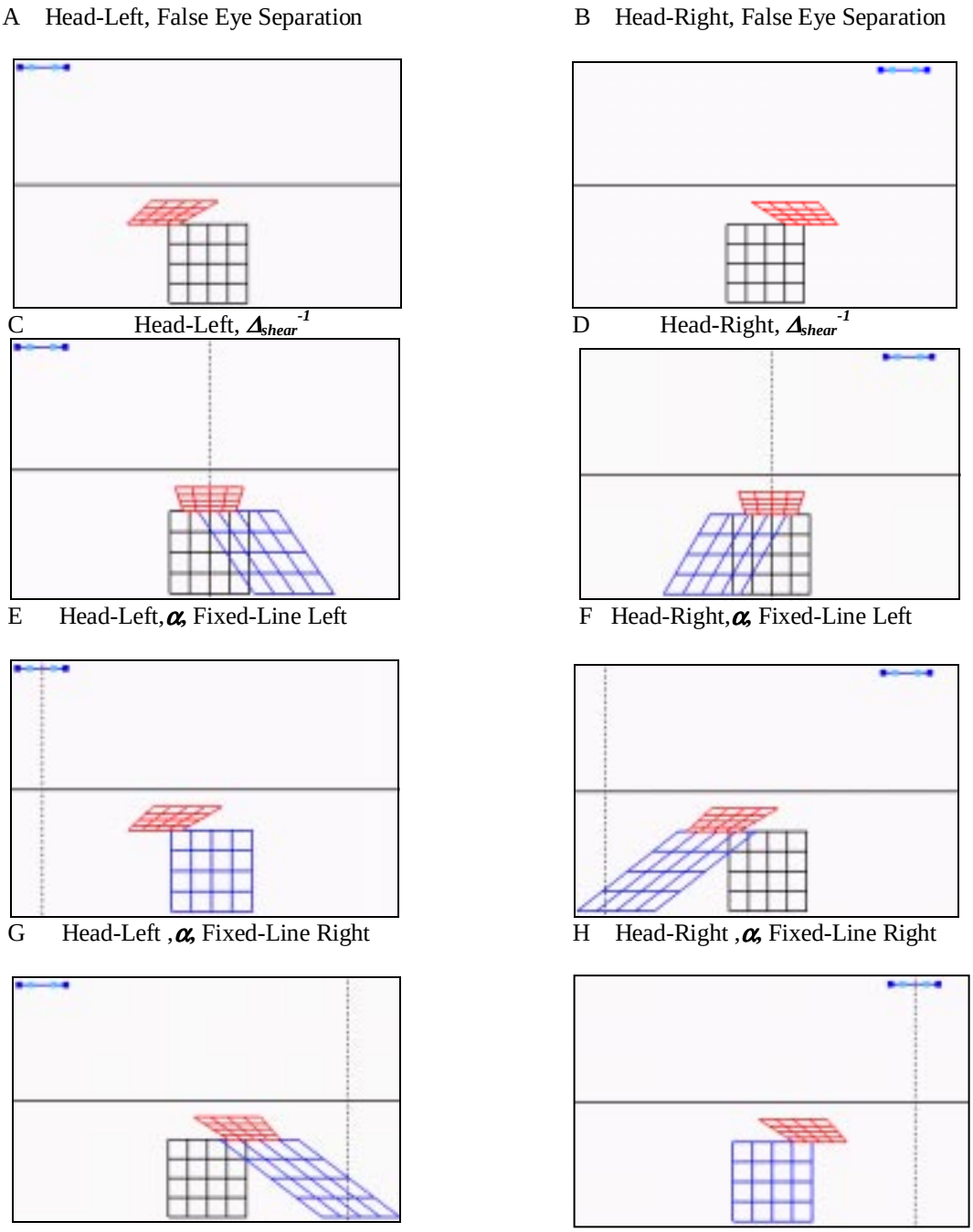


Figure 62: Comparison of False Eye Separation, Δ_{shear}^{-1} Predistortion, and α Predistortion

in mind, however, that it is the red mesh that is actually displayed. With the predistortion the displayed mesh is stationary even as the eye moves from left to right (C to D). Again this technique removes the shearing that occurs for head movement in any lateral direction not just the left and right direction illustrated in these 2D diagrams. (Note, figures E-H are related to a latter section).

Additionally, we investigated predistorting by $\Delta_{shear}^{-1} \bullet \Delta_{scale}^{-1}$. However, while this technique removed the lateral motion shearing component it made additional visible alterations to the displayed image. Figure 63 (page 140) illustrates this. Figures A and B repeat the images for false eye separation just for ease of comparison. Figures C and D show Δ_{shear}^{-1} predistortion while figure E and F show $\Delta_{shear}^{-1} \bullet \Delta_{scale}^{-1}$ predistortion. The key point is that while in both Δ_{shear}^{-1} predistortion (C/D) and $\Delta_{shear}^{-1} \bullet \Delta_{scale}^{-1}$ predistortion (E/F) the displayed grid (red) no longer shears side to side with lateral head motion, in $\Delta_{shear}^{-1} \bullet \Delta_{scale}^{-1}$ predistortion (E/F) the displayed grid (red) is also expanded relative to the displayed grid with regular false eye separation (A/B). So Δ_{shear}^{-1} predistortion does a better job of capturing the effect of false eye separation while only removing the lateral shearing. Based on this diagrammatic observation we conclude Δ_{shear}^{-1} is a better choice than $\Delta_{shear}^{-1} \bullet \Delta_{scale}^{-1}$ predistortion.

An alternative, more intuitive explanation of why Δ_{Shear}^{-1} predistortion works is as follows. Recall that for a 3D point at a given depth relative to the screen, the projected 2D image point for that 3D point will move due to lateral user head movement. The displacement of the 2D image point depends on both the magnitude of the head displacement and the depth of the point. Taking an image analysis point of view, a monoscopic visual system can invert this relation. By knowing how much the 2D image point moved due to a known amount of head motion, the system can extrapolate the depth of the original 3D point. This method of gaining depth information from head or camera motion is well-known in computer vision and human vision communities and is referred to as motion parallax. At the same time, a *stereoscopic* system gains depth information from the stereo parallax. False eye separation, however, creates a discrepancy between the depth extrapolated from motion parallax cues and the depth extrapolated from stereo parallax cues. A consistent way to resolve this discrepancy is for the vision system to interpret

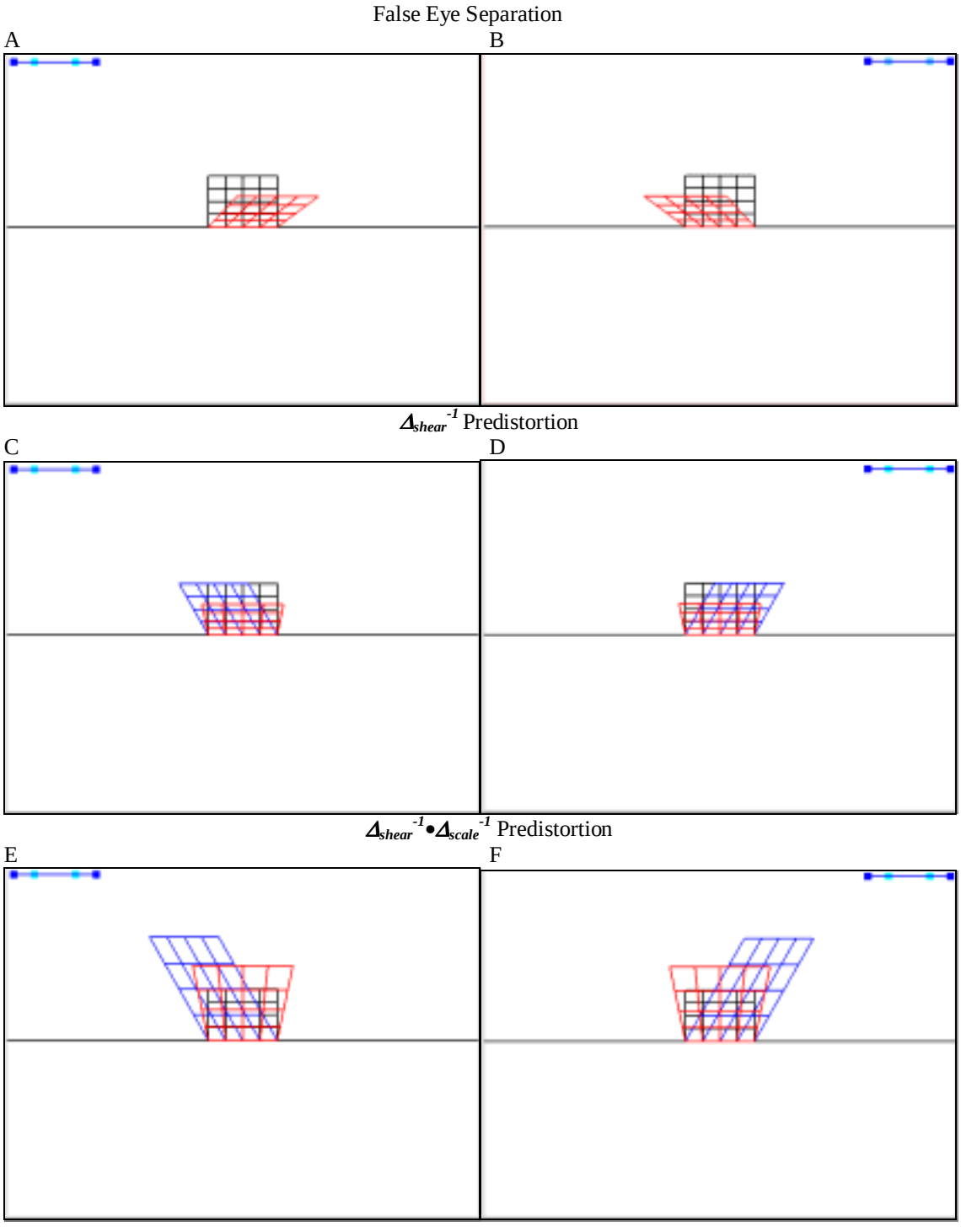


Figure 63: Δ_{shear}^{-1}Predistortion versus $\Delta_{shear}^{-1} \bullet \Delta_{scale}^{-1}$ Predistortion

the 3D scene as shearing with head (or camera) motion. Δ_{Shear}^{-1} predistortion forces the depth information from motion parallax and the depth information from stereo parallax partially back into sync.

2.2 α-Predistortion

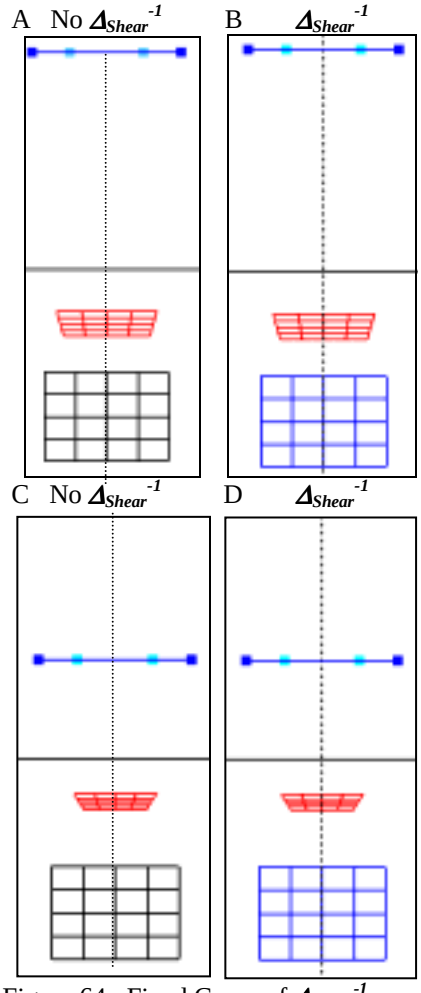


Figure 64: Fixed Curve of Δ_{Shear}^{-1}

Figure 64 hints at an interesting aspect of the Δ_{Shear}^{-1} technique. In Figure 64 A and C only false eye separation is used while in Figure 64 B and D Δ_{Shear}^{-1} is added. In B and D the blue predistorted matrix exactly overlaps the modeled matrix. Hence Δ_{Shear}^{-1} happens to be the identity matrix for the head

positions shown. In general for an eye axis parallel to the view plane there is a line perpendicular to the view plane such that when **I** is on this line Δ_{Shear}^{-1} is the identity matrix. Therefore this line is a locus of points which define view positions for which the displayed space is not altered by Δ_{Shear}^{-1}. In Figure 64 this line is the vertical dashed line.

For an arbitrarily oriented eye axis this special line becomes a curve which is the intersection of two hyperbolic surfaces. In the general case, for fixed **d** and *r*, Δ_{Shear}^{-1} is the identity matrix when *(Ix Iz + dx dz r)* and *(Iy Iz + dy dz r)* both equal zero. This occurs along the curve defined by: $Ix = -(dx\ dz\ r) / Iz$, $Iy = -(dy\ dz\ r) / Iz$. As observed in the previous paragraph this degenerates to the line, (Ix=0,Iy=0), when the eye axis is parallel to the view plane. Call this special curve the fixed curve of the predistortion.

Experimentally, we find it useful to move this fixed curve to adapt the Δ_{Shear}^{-1} technique. It is useful to position this curve so that the user's resting head position for a given stereo HTD lies on the curve. This locks the displayed objects in place as seen from this resting view position. In practice for purposes of moving the fixed curve it is suitable to assume that d_z=0 and that the curve is a straight line. Henceforth, we simply use the term "fixed-line". With a vertical display system, the fixed line could be centered horizontally on the display and then positioned vertically to coincide with the average user's eye level. For a horizontal display, such as the virtual workbench, the fixed line could be centered horizontally on the display and then translated forward perhaps half a meter in front of the workbench.

To shift this fixed line we derive the following predistortion matrix, α, to replace Δ_{Shear}^{-1}:

$$\alpha = \begin{bmatrix} 1 & 0 & \dfrac{-(1-r)(I_x I_z + d_x\ d_z\ r - I_z\ F_x)}{d_z^2 r^2 - I_z^2} & 0 \\ 0 & 1 & \dfrac{-(1-r)(I_y I_z + d_y\ d_z\ r - I_z\ F_y)}{d_z^2 r^2 - I_z^2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (19)$$

where

F_x *and* F_y *are the x and y coordinates of the fixed line*

The fact that α moves the fixed line as desired is shown below.

$$I_x I_z + d_x d_z r - I_z F_x = 0, \quad I_y I_z + d_y d_z r - I_z F_y = 0$$

$$\Rightarrow$$

$$I_x = -\frac{d_x d_z r}{I_z} + F_x, \quad I_y = -\frac{d_y d_z r}{I_z} + F_y$$

$$\Rightarrow$$

$$I_x = F_x, \quad I_y = F_y \quad \text{in order to make } \boldsymbol{\alpha} \text{ the identity assuming } d_z = 0$$

Figure 62 (page 138) illustrates the use of the predistortion α with different fixed lines (dash gray). The left column (A,C,E,G) illustrates eyes on the left while the right column (B,D,F,H) illustrates eyes on the right. In the first row (A,B), α is not used. In the next 3 rows (C,D/E,F/G,H) α is used with the fixed line in at center, on the left, and on the right. Using α with the fixed line on the left, the displayed grid for the left head position *is not altered* by α (compare E to A), while the displayed grid for the right head position *is altered* by α (compare F to B). On the other hand, using α with a fixed line on the right, the displayed grid for the left head position *is altered* by α (compare G to A), while the displayed grid for the right head position *is not altered* by α (compare H to B). Hence changing the position of the fixed line determines which viewpoints are locked in place. This fixed line can be positioned to contain the typical viewing position in a given head-tracked stereo display system.

In OpenGL [Woo97], α-predistortion can be implemented as follows. At every frame compute α. Next compute $[\alpha]_{view}$. $[\alpha]_{view}$ is α relative to the OpenGL view coordinate system. Recall OpenGL combines the affine model and view transforms on a single "ModelView" matrix stack while the projective frustum transform goes on the "Projection" matrix stack. Typically, the first transform placed on the ModelView stack is that which maps world coordinates to view coordinates. This transform consists of a rotation and a translation accounting for the location of the eye point and the orientation of the view plane. Since the orientation of the view coordinate system and the projection plane coordinate system are the same, α is mapped from projection plane coordinates to view coordinates as follows: $[\alpha]_{view} = S^{-1} \bullet T^{-1} \bullet \alpha$

$\bullet \boldsymbol{T} \bullet \boldsymbol{S}$ where $\boldsymbol{T}$ is a translation by the coordinates of the eye in projection plane coordinates and $\boldsymbol{S}$ is the Platform scale factor. $[\alpha]_{view}$, must be placed between the last transform on the Projection stack and the first transform on the ModelView stack. To avoid affecting lighting, place $[\alpha]_{view}$, on the Projection matrix stack as the last transform.

Two issues remain. For applications with bounding-box to view-volume culling, the α transform must be taken into account. Second virtual representations of 6 DOF devices should probably be rendered without α and with true eye separation to maintain physical to virtual correspondence. Collision detection between these device representations and other virtual geometry must account for the resulting discrepancies by carefully transforming the device geometry by $(\Delta \bullet \alpha)^{-1}$.

2.3 Applications of α-predistortion

False eye separation is used either to control fusion problems by underestimating the eye separation or to enhance stereoscopic depth by overestimating eye separation. For head-tracked displays, this induces a distortion Δ. While we tested and verified that α-enhanced false eye separation removes the shearing for both underestimated and overestimated eye separation, for overestimated eye separation we find some reasons why it may be less useful.

With overestimated eye separation two choices exist:

- produce a distorted image that shears with lateral head-movement, compresses/expands with perpendicular head movement and does not preserve parallelism

or

- produce a distorted image that does not shear but still compresses/expands with perpendicular head movement and does not preserve parallelism (i.e. α-predistortion)

In the best case overestimated eye separation still distorts the image by a non-affine collineation about the projection plane. Alternatively, one could just use a perpendicular scale and preserve parallelism. Anecdotally, we observed that when viewing flat surfaces that are parallel to the screen the perpendicular scale method is preferable. For other flat scenes this may not be the case, however.

As discussed in Chapter III Section 5.2 , perpendicular scaling can be undesirable for reducing screen parallax. Additionally, for underestimated eye separation Δ has an interesting property. Because Δ is a non-affine collineation it has the effect of bringing points at infinity to some affine fixed plane beyond the projection plane. So by underestimating user eye separation we can map the *entirety* of space beyond the projection plane to a finite region between the projection plane and some maximum depth plane (Chapter V, Section 2.4.2). (No affine transform can do this). Now we can set this maximum depth plane to the farthest fusible depth plane. For example we can rewrite Southard's [Sou95] equation (see (1), pg 32) for the maximum fusible depth plane using our nomenclature. Now from equation (6) we can solve for the eye separation ratio, *r*, that will bring all points infinitely far beyond the view plane into the comfortably fusible region delimited by *far_fusible*:

$$r = \begin{cases} \dfrac{far\_fusible \cdot I_z}{d_z^2 - I_z^2 + far\_fusible \cdot I_z}, & \text{if } 2|\mathbf{d}| - I_z\theta_{max} > 0 \\ 1, & \text{otherwise} \end{cases} \qquad (20)$$

The r=1 case occurs when the user is far enough from the projection plane so that the maximum depth plane is at infinity, i.e. all far space is fusible. In this case we use the true eye separation (r=1). Finally, applying α-predistortion removes the left/right shearing.

3 γ-distortion

In applying α-false eye separation to wider variety of scenes and display sizes, we observed that the removal of the lateral shearing effect can make the perpendicular expansion/expression effect more noticeable. This was more significant for displays that encourage perpendicular head movement. For instance in our virtual workbench whose display is laid horizontally we anecdotally observe that most head motion is lateral to the display. When moving to our desktop-VR or CAVE like system, we tended to have a fair of amount perpendicular head motion too. Based on this observation and simple qualitative

considerations, we next pursued the removal of the compression/expansion effect. This section first discusses quantitative aspects of the remaining expansion/compression and then presents a set of four idealistic properties that a fusion technique would satisfy. While mathematically we can satisfy three of these properties with a new transform, $\boldsymbol{\beta}$, this transform empirically fails the fourth property. This result leads us to a final third solution named γ-distortion.

3.1 Expansion/Compression in α-False Eye Separation

It will be useful to augment our distortion notation to be "$\boldsymbol{Q}$ ($a \mid b$)" where $\boldsymbol{Q}$ is the distortion name, a are the fusion parameters deliberately manipulated and b are the head position variables such as **I** or **d** that affects Q. In this scheme regular false eye separation is denoted $\boldsymbol{\Delta}(r \mid \mathbf{I},\mathbf{d})$. From Appendix D, Section 1, for α-false eye separation we have:

$$\Delta_{\alpha}(r, F_x, F_y \mid d_z, I_z) = \begin{bmatrix} 1 & 0 & (1-r)(I_z F_x)/\Omega & 0 \\ 0 & 1 & (1-r)(I_z F_y)/\Omega & 0 \\ 0 & 0 & r(d_z^2 - I_z^2)/\Omega & 0 \\ 0 & 0 & I_z(1-r)/\Omega & 1 \end{bmatrix}$$

Recall that the parameters (F_x,F_y) are the X,Y coordinates of the "fixed line" which allows the adjustment of the technique to different display arrangements. α-false eye separation dynamically varies with head-screen distance, I_z, and head orientation, d_z. To examine these dynamic artifacts we look at characteristic planes of the distortion transform. Assuming $r < 1$, the maximum depth plane is the plane of ordinary affine points to which the ideal plane is mapped. Hence all of space beyond the projection plane is mapped to the region between the projection plane and the maximum depth plane. Recall this plane is given by:

$$z = \frac{r\left(d_z^{\,2} - I_z^{\,2}\right)}{I_z\left(1-r\right)}$$

The second plane of interest is the plane whose point's z-coordinates remain fixed while their x,y coordinates generally change. Call this the fixed-depth plane. Do not confuse this fixed-depth plane with the fixed-plane whose point's x, y, and z coordinate all remain fixed. (The fixed-plane is just the projection plane itself). From Appendix D, Section 2 the fixed-depth plane is:

$$z = \frac{r(d_z^{\,2} - I_z^{\,2}) - d_z^{\,2} r^2 + I_z^{\,2}}{I_z(1-r)} \qquad (21)$$

or

$$z = I_z \qquad \text{when } d_z = 0$$

Both plane positions vary with the head distance from the projection plane (I_z), the head orientation (d_z), and eye separation ratio (r). However, Section 2.3 showed how to dynamically alter α-false eye separation's parameter, *r*, so as to keep the maximum depth plane at a fixed location. Called this "adjusted α-false eye separation." It's controllable parameter is then, *mdp*, the maximum depth plane. This is useful for implementing dynamic or static fusion control implementations since we can set the maximum depth plane to the far fusible plane. Importantly, adjusted α-false eye separation still has dynamic artifacts since the depth-fixed plane still varies with head motion.

3.2 Is there a perfect technique?

We first seek improvements not by further adjustments to false eye separation, but rather through direct derivation of a new transform. In this derivation we use characteristics of prior image fusion control methods that we've analyzed. We begin with the following claim:

At the very least a fusion control technique for a stereo HTD should:

(1) map lines to lines

(2) not generate vertical parallax

A fusion control technique should distort geometry to the minimum degree possible. Mapping lines to curves represents an additional distortion beyond line preserving algorithms. Another reason to aim for line preservation is that standard graphics pipelines only support line preserving transformations, or collineations, via a 4x4 matrix. This implies it would be highly inefficient to directly generate a 3D curvilinear distortion. Moreover, analysis of non-line preserving fusion control techniques, in particular image scaling and image shifting, found such distortions to be associated with vertical parallax which one would like to avoid.

A fusion technique can either directly apply a 3D transformation to the scene (such as perpendicular scaling or view scaling) or it can manipulate displayed space indirectly (false eye separation, image scaling and image shifting). Since direct techniques apply a 3D transformation to the scene prior to projection onto the image plane, the direct techniques cannot create vertical parallax. Therefore when trying to derive a new 3D transformation, property 2 is automatically satisfied. This yields the following suggestive properties for an idealistic fusion control technique for near-far or far space applications of arbitrary depth:

(1) Preserve lines

(2) Map the near geometry point and far geometry point onto the near and far fusible range

(3) Preserve visual angles

(4) Preserve in-screen geometry

The most general possible line preserving 3D transform is a non-affine collineation which need not preserve parallelism. This is represented by a general 4x4 matrix with 15 degrees of freedom. One degree of freedom is lost because in projective geometry a point's coordinate is unique up to a scale factor. We will assume element (4,4) is set to 1. Prior experience helps define the new transform. The 3 translation components are best handled by manipulating the view position so we remove these 3 degrees of freedom

from the new transform. Next there is no need for adding rotations, and X or Y scales would only serve to visibly shrink in-screen and near geometry. Shears generally are a problem as seen in false eye separation. This removes another 8 degrees of freedom and 4 remain: the Z scale factor and the first three elements of the bottom row of the matrix. Recall the bottom row is normally [0 0 0 1] for affine transforms while non-zero' s occur in this row for non-affine transforms. We can also leave the first two elements as zeros. This is because the transform should map all of far space to the farthest fusible depth, *ff*, and map the nearest geometry point, *np*, to the nearest fusible depth, *nf*. Appendix E shows that the resulting matrix in projection plane coordinates is:

$$\beta(np,nf,fp,ff) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & A & 0 \\ 0 & 0 & A & (1/f) & 1 \end{bmatrix} \qquad (22)$$

$$A = \frac{nf}{np(1-nf/f)}$$

$$f = \begin{cases} fp/(fp/ff-1) & , fp \neq \infty \\ ff & , fp = \infty \end{cases}$$

β preserves lines (Property 1) and since the fixed plane of β is the screen itself it preserves in-screen geometry (Property 4). Next, β allows exact mapping of both *nf* to *np* and *ff* to *fp*. In contrast all other non-affine techniques fundamentally lack enough degrees of freedom to do so. Unfortunately, empirically β's visual angle distortion of far geometry is only slightly better than perpendicular scaling and noticeably worse than false eye separation. The problem occurs when the near point is manipulated. Figure 65 shows a sequence of stereo pairs of the scene from Chapter III, Section 5.2 with β distortion applied in order to only manipulate the far point. Under these circumstances the results are reasonable. However, Figure 66 shows the results once the near point is manipulated. In the sequence B through D we see the increasing visual angle of the distant cubes and the distorted aspect ratio of the nearby cubes. Interestingly,

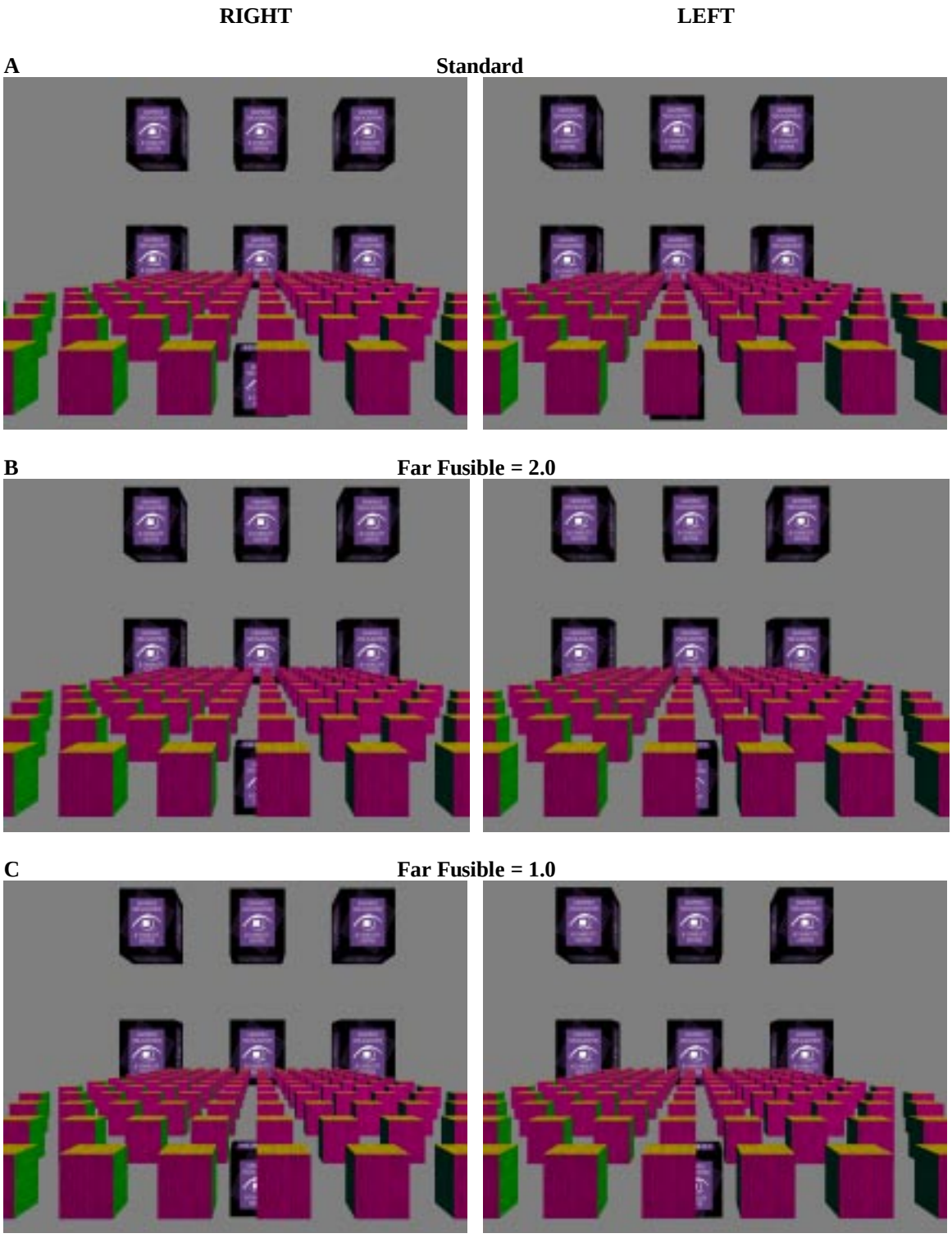


Figure 65: Beta Distortion Effects (figure continues on next page)

D **Far Fusible = 0.45**

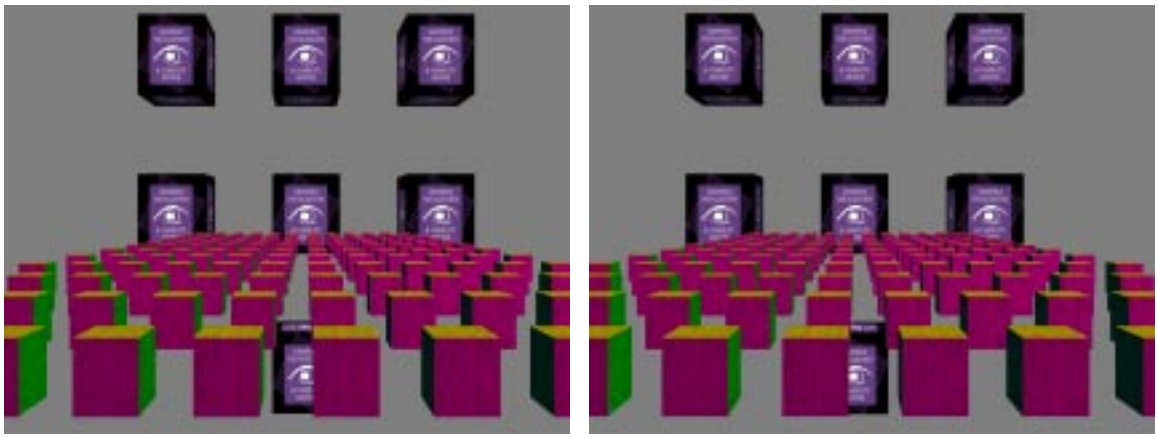
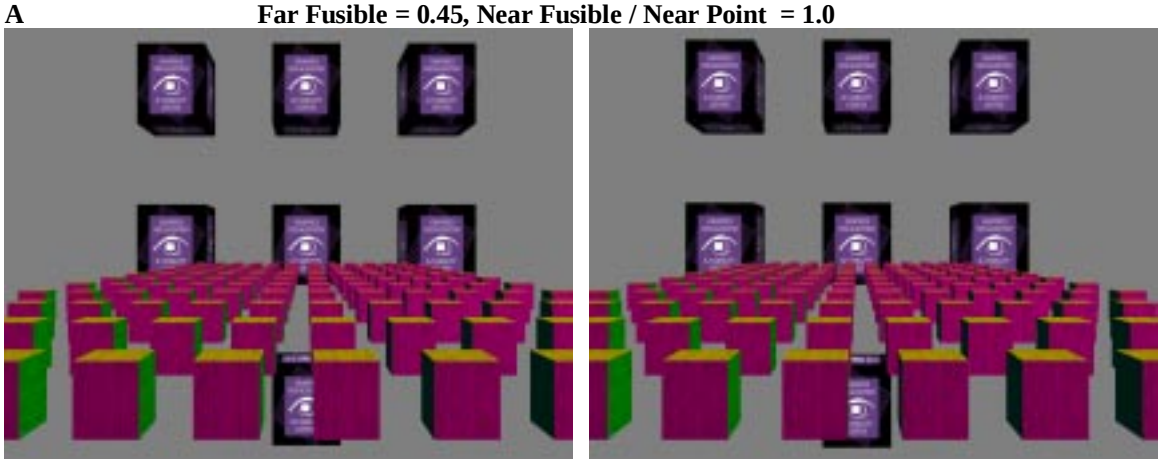
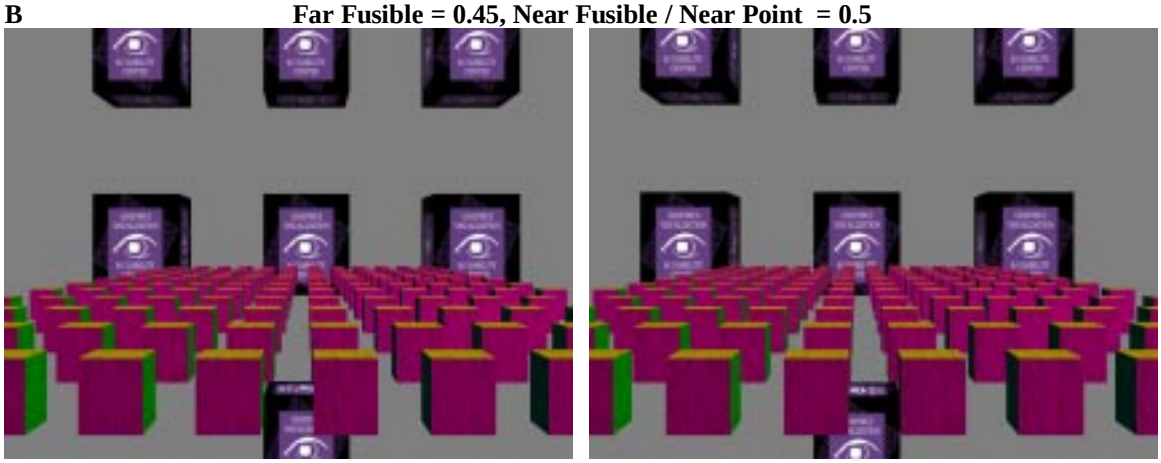


Figure 65: Beta Distortion Effects with No Near Space Fusion Compression

A **Far Fusible = 0.45, Near Fusible / Near Point = 1.0**



B **Far Fusible = 0.45, Near Fusible / Near Point = 0.5**



C **Far Fusible = 2.00 m, Near Fusible / Near Point = 0.5**

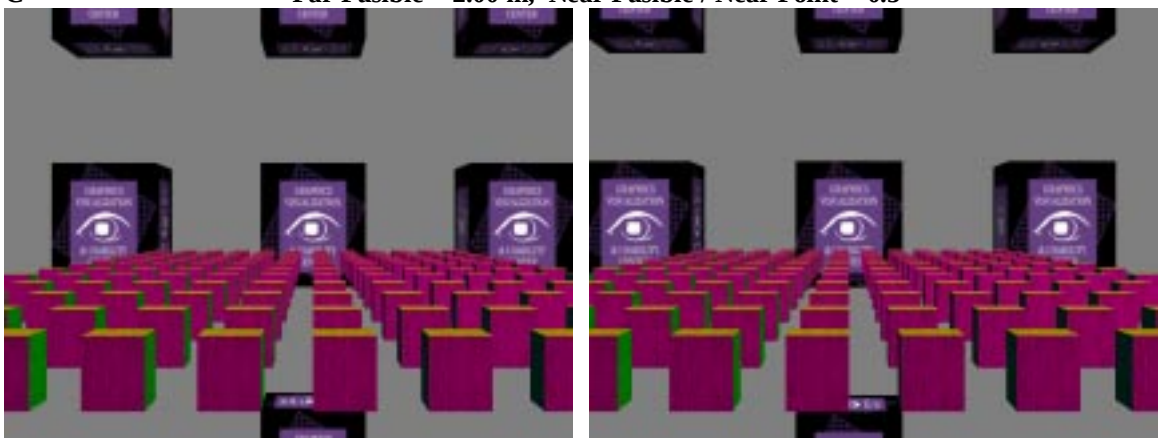
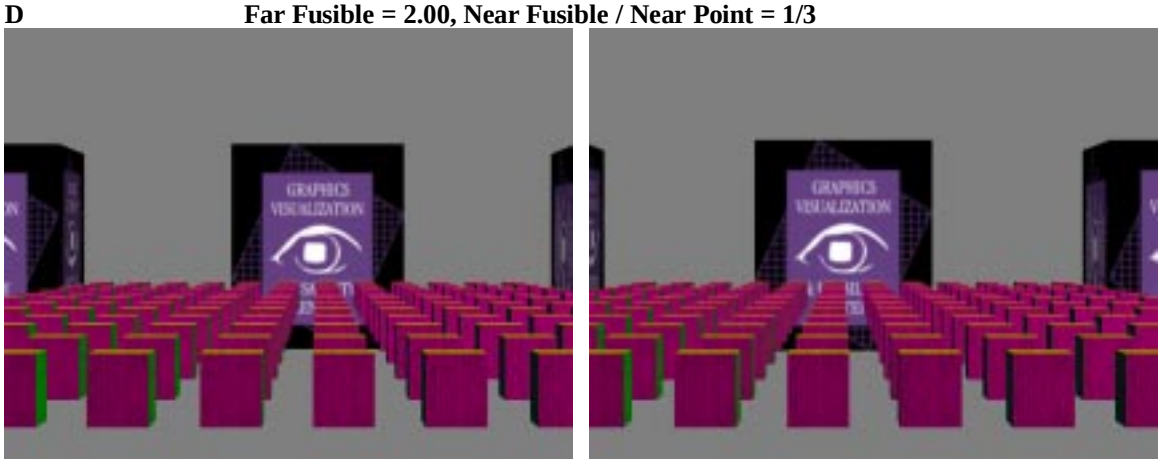


Figure 66: Beta Distortion Effects (figure continues on next page)

D **Far Fusible = 2.00, Near Fusible / Near Point = 1/3**



E **Far Fusible = 0.15, Near Fusible / Near Point = 1/3**

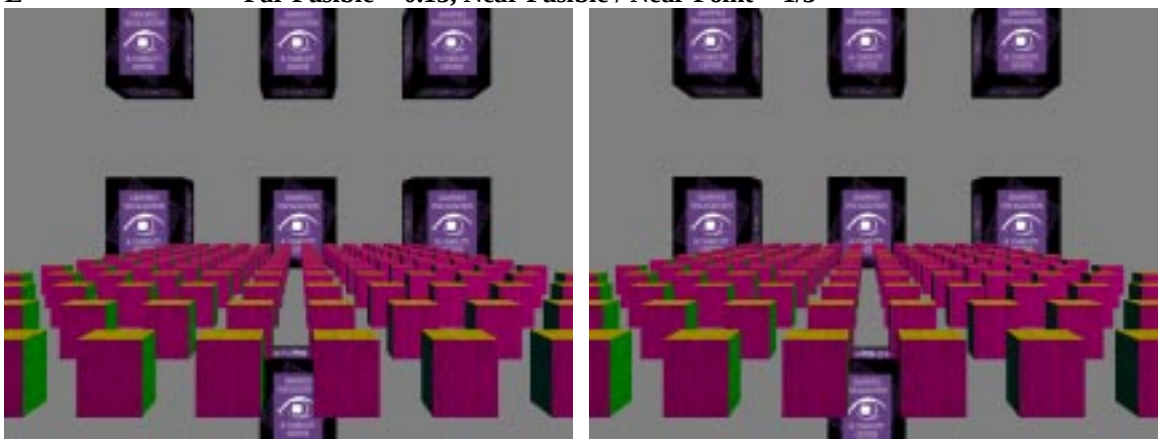


Figure 66: Beta Distortion Effects when Near Fusible / Near Point Ratio Is Not 1.0

if we keep the same near space compression (1/3) but alter the far fusible point from 2.0 m to 0.15 m we get a much more desirable image shown in Figure 66E. The trouble is that we cannot control the near and far fusible points independently and simultaneously avoid noticeable loss of aspect ratios, etc. We might counteract the visual angle distortion by adding z-translation components or x,y scales; but, this of course violates the preservation of in-screen geometry and confounds independent view scale control Since $\boldsymbol{\beta}$ is the theoretical result of satisfying properties (1), (2) and (4) and since it empirically fails (3), it appears satisfying all four goals is not possible.

3.3 Revisiting False Eye Separation

If we must drop a property, Property 4, preserving in-screen geometry, is the best choice. Preserving in-screen geometry is primarily useful for applications that use the screen as a physical work plane or prop. In this mode, the user is manipulating near-space geometry. Such manipulations would probably be fairly rare in near-far or far space applications. So while it would be ideal if we could preserve in-screen geometry, dropping this property perhaps will not hurt the majority of applications. Having dropped Property 4, we can find a more satisfactory solution that does preserve visual angles. We found little value in trying to reformulate $\boldsymbol{\beta}$ because one ends up trying to mimic the static properties of false eye separation. Instead we use the distortion of false eye separation, $\boldsymbol{\Delta}(r \mid \mathbf{I},\mathbf{d})$, directly. We replace parameter's **I** and **d** with a fixed resting head position in order to "freeze" the distortion at this point. Let **C** be the resting head position. To simplify the arithmetic we assume that at the resting position, the head is parallel to the screen (i.e. d_z=0). Additionally we solve for r in order to map the farthest point (*fp*) to the farthest fusible point (*ff*). Note, the formula simplifies slightly if an application designer wishes to set *fp* to infinity. This yields:

$$\boldsymbol{\gamma}(fp, ff, \mathbf{C}) = \begin{bmatrix} 1 & 0 & (r-1)C_x / C_z & 0 \\ 0 & 1 & (r-1)C_y / C_z & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & (r-1)/C_z & 1 \end{bmatrix} \qquad (23)$$

$$r = \frac{ff - \dfrac{C_z \quad ff}{fp}}{ff - C_z}$$

While it is obvious from the formulation that $\boldsymbol{\gamma}$ is not dependent on head position (**I** or **d**), it aids intuition to compare illustrations of γ-distortion with that of false eye separation and adjusted α-false eye separation. In all diagrams of Figure 67 (pg 156)the central horizontal black line is the projection plane. The black grids are objects in model space. The red grids are the distorted objects in displayed space. The dark blue dots are the left and right eye points and the blue line segment is the eye axis. The upper dotted red line is the fixed-depth plane. The lower dashed red line is the maximum depth plane. (Note this line is "off-screen" in the first column). The first column contains the results of false eye separation. In this column, the modeled eye points are shown in light blue. The second column contains the results of adjusted α-false eye separation. Again the modeled eye points are shown in light blue. Recall adjusted α-false eye separation varies the modeled eye separation in order to maintain a fixed maximum depth plane. For adjusted α-false eye separation, the fixed line is shown as the dashed, vertical black line. γ-distortion is shown in the third column. A red 'X' shows the location of the center, **C**, of $\boldsymbol{\gamma}$. Note the maximum depth plane is set to the same depth for both adjusted α-false eye separation and γ-distortion.

The first two rows illustrate the effect of lateral head motion. In the first row, the head is on the left and the second row the head is on the right. In false eye separation (first column) lateral head motion induces a dynamic shearing artifact. Both adjusted α-false eye separation (second column) and γ-distortion (third column) remove this effect. The last two rows illustrate the effect of perpendicular head motion.

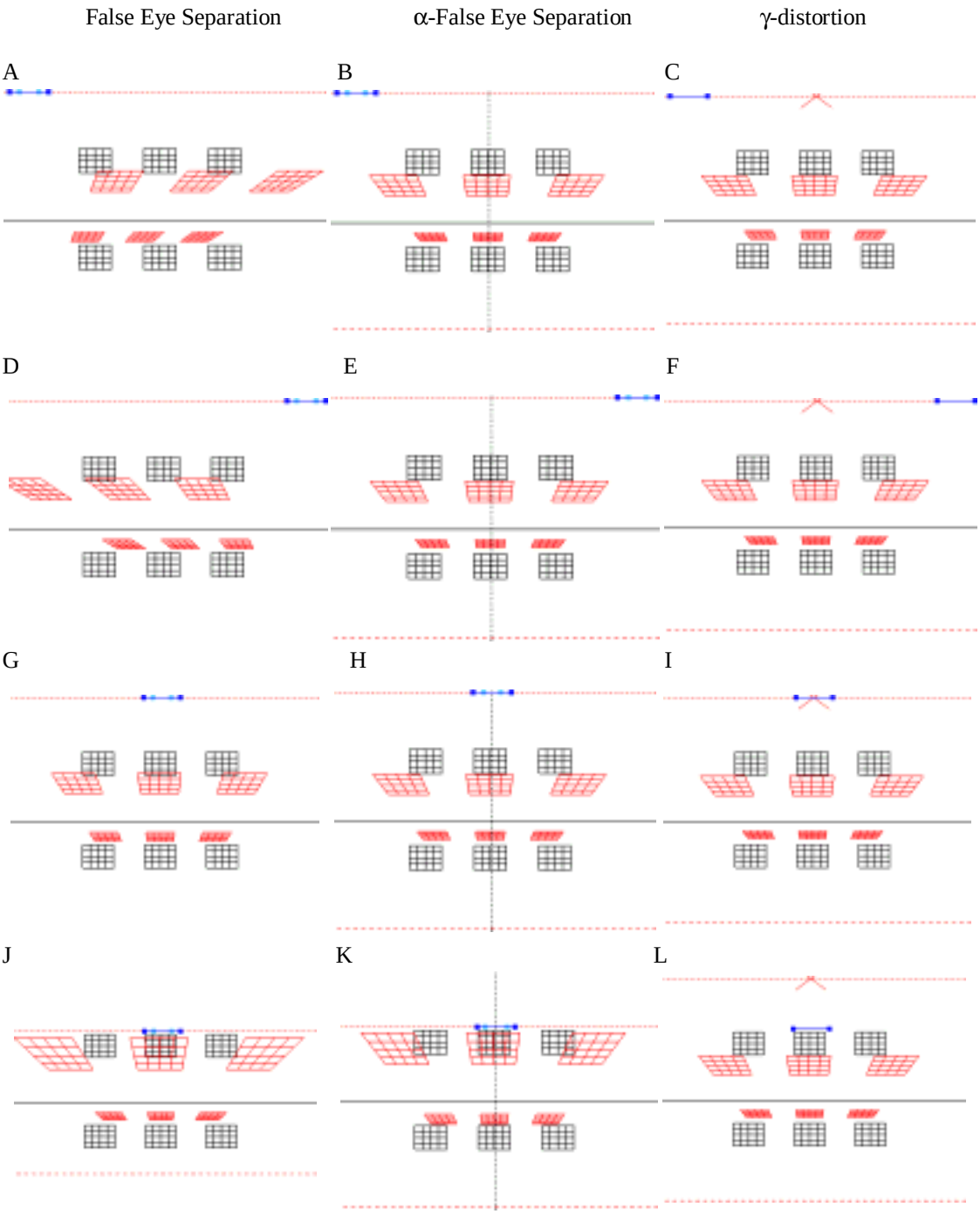


Figure 67: Effects of Head Motion on Display Space for False Eye Separation, Adjusted α-False Eye Separation and γ-Distortion.

Both false eye separation (first column) and adjusted α-false eye separation (second column) exhibit dynamic compression/expansion. Only γ-distortion (third column) avoids this.

For Case II of the generic algorithm (pg 60), the case of far fusion problems, use $\boldsymbol{\gamma}(fp,ff,\mathbf{C})$. For Case III, the case of near fusion problems, use $\boldsymbol{\gamma}(np,nf,\mathbf{C})$. For Case I, the case of near and far fusion problems, more work is needed since $\boldsymbol{\gamma}$ lacks the necessary degrees of freedom. Therefore we combine $\boldsymbol{\gamma}$ with a translation perpendicular to the projection plane. This requires solving the following pair of equations for *r*, the raw $\boldsymbol{\gamma}$ parameter, and *t*, the z-translation in projection plane coordinates:

$$ff = r\,(fp+t)/((fp+t)(r\text{-}1)/C_z+1)$$
$$nf = r\,(np+t)\,/\,((np+t)(r\text{-}1)/C_z+1)$$

We also have the following constraints: $nf > 0$, $np > 0$, $ff < 0$, $fp < 0$, $Cz > 0$, $fp <$ ff, and $np > nf$. The fusion algorithm only computes solutions to the two equations if both near and far geometry exist ($np > 0$ and $fp < 0$) and both are unfusible ($fp < ff$) and ($np > nf$). Generally, users can always fuse some range of space so $ff < 0$ and $nf > 0$ and from the derivation of γ, *Cz* should be greater than 0. Using a commercial analysis tool [Wolf96], the following solution is found:

$$t = \frac{1}{2Cz(ff-nf)}\left(\Phi + Cz^2(ff-nf) - Cz(ff-nf)(fp+np)\right)$$
$$r = -\frac{1}{2(Cz-ff)(Cz-nf)(fp-np)}\begin{pmatrix}\Phi + Cz^2(-ff+nf) + Cz(ff+nf)(fp-np)\\ +2\,ff\quad nf\quad(-fp+np)\end{pmatrix} \qquad (24)$$

$$\Phi = \sqrt{\left(\mathrm{Cz}^2(ff-nf)+2\,ff\quad nf\quad(fp-np) - Cz(ff+nf)(fp-np)\right)^2 - 4(Cz-ff)\,ff\,(Cz-nf)\,nf\,(fp-np)^2}$$

At first glance, the solution seems highly complex and rife with possible degeneracies. However, coding equation (24) is trivial and empirically the degeneracies are not a problem. First, Appendix F, Section 1 proves that *r* and *t* are always real under the mentioned constraints. Next, the constraints rule out undefined results with the exception of when $C_z = nf$. In theory, the asymptotic behavior as C_z approaches *nf* can yield uselessly high *r* values. In practice, however, the spike in *r* values occurs for a very narrow range of C_z around *nf* . *r* has nearly identical values and trends on both sides of the spike and empirically *r* is always positive. We have never actually seen this spike occur in a live implementation and only through very careful plotting is it noticeable. Nonetheless, to be safe if $|Cz - nf| < 0.0001$ (in meters), then we set C_z to be *nf* +/- 0.0001. While not a degeneracy, over certain ranges of input, *t* is positive which pushes the scene forward. Visually we find this result is unsatisfactory. Therefore if $t > 0$ set t = 0 and set *r* to the smaller of the two values produced by $\chi(fp,ff,C)$ and $\chi(np,nf,C)$. This compresses displayed space a bit more than necessary, but since *nf* and *ff* are already psychophysical approximations, this is a reasonable solution for the $t > 0$ subcase.

3.4 Implementing γ-distortion

α-false eye separation uses false eye separation and a pre-distortion of the virtual world by the affine transform, $\boldsymbol{\alpha}$. γ-distortion use correct model eye separation and applies a non-affine transform to the matrix stack. When applying either an affine or non-affine matrix we must take these matrices into account when handling bounding volumes and mapping screen space back into world space for picking or similar operations. Since it is generally simpler to deal with an additional affine matrix than with an additional non-affine one, it would be ideal if we could achieve the end result of γ-distortion by using false eye separation and some pre-distortion matrix, $\boldsymbol{M}$, where $\boldsymbol{M}$ is affine. This requires: $\boldsymbol{\gamma}=\boldsymbol{\Delta M}$ or $\boldsymbol{M}=\boldsymbol{\Delta}^{-1}\boldsymbol{\gamma}$, where $\boldsymbol{\Delta}$ is the distortion due to false eye separation. However, if we compute $\boldsymbol{M}$ (Appendix F, Section 2), the last row in $\boldsymbol{M}$ is:

$$\begin{bmatrix} 0 & 0 & (r-1)\left[\dfrac{I_z}{(d_z^{\,2} - I_z^{\,2})} + \dfrac{1}{C_z}\right] & 1 \end{bmatrix} \tag{25}$$

This generally is not equal to [0 0 0 1] as is necessary for $\boldsymbol{M}$ to be affine. Therefore, while it is possible to implement γ-distortion as a pre-distorted false eye separation technique, there is no advantage in doing so. Hence, we apply γ directly with correct eye separation.

In OpenGL [Woo97], γ-distortion can be implemented as follows. At every frame compute $\boldsymbol{\gamma}$. Next compute $\boldsymbol{[\gamma]_{view}}$. $\boldsymbol{[\gamma]_{view}}$ is $\boldsymbol{\gamma}$ relative to the OpenGL view coordinate system. Recall OpenGL combines the affine model and view transforms on a single "ModelView" matrix stack while the projective frustum transform goes on the "Projection" matrix stack. Typically, the first transform placed on the ModelView stack is that which maps world coordinates to view coordinates. This transform consists of a rotation and a translation accounting for the location of the eye point and the orientation of the view plane. Since the orientation of the view coordinate system and the projection plane coordinate system are the same, γ is mapped from projection plane coordinates to view coordinates as follows: $\boldsymbol{[\gamma]_{view} = S^{-1} \bullet T^{-1} \bullet \gamma \bullet T \bullet S}$ where **T** is a translation by the coordinates of the eye in projection plane coordinates and **S** is the Platform scale. $\boldsymbol{[\gamma]_{view}}$ must be placed between the last transform on the Projection stack and the first transform on the ModelView stack. To avoid affecting lighting, place $\boldsymbol{[\gamma]_{view}}$ on the Projection matrix stack as the last transform.

3.5 Fusibility, Rigidity, Reactivity: Pick Two

Initially we implemented γ-distortion as described above in a simple application using SVE [Kess00], on a desktop VR setup. Using dialog boxes we can turn on and off various groups of virtual boxes at various distances. Some objects exist in front of the screen, some in the plane of the screen, and others beyond or far beyond the screen. Additionally, by adjusting two sliders we could control the input value for the nearest fusible point and the farthest fusible point. A third button set the center of the γ-distortion to

the current tracked location of the central eye point, **I**. This setup allows us to verify the correct working of the algorithm. We informally experimented with all three cases of Algorithm 1 (pg 60) using equation (24). In Case I objects only breached the recommended near fusible depth and the translation step moved the view in order to push the objects back to the screen. In Case II objects only breached the far fusible depth and the γ matrix was adjusted to compress the farthest objects to the far fusible depth. In Case III objects breached both the near and far fusible depths and both parameters r and t were automatically adjusted to put the nearest and farthest object in the desired depth range. The effect of deliberately altering the center of γ was quite noticeable. Our software also let us rapidly switch between γ-distortion, α-false eye separation and regular false eye separation while keeping the visible object set and the near/far fusible depths the same. When switching between these methods and then moving one's head laterally and perpendicularly, the observed differences between the techniques were qualitatively consistent with the diagrams of Figure 67 (pg 156). These observations were made by the author and a few other graduate students in the lab. This demonstration verified that the algorithms were behaving as expected.

Next we used a virtual environment consisting of a large set of objects and displayed it on a stereo HTD with a screen size of 8 by 6 ft. We used just a single screen of a three screen display called the NAVE at Georgia Tech. Unlike in the desktop VR environment where the user is sitting, in this display the user is standing and walking around. She can crouch down, stand on her toes, and walk left/right and forward and back throughout a space of roughly 8 by 8 ft. In this display environment with a broader set of objects, a rather disappointing anomaly becomes evident.

The scene consisted of a variety of near by objects arranged in a grid like placement at a virtual distance of 1000 meters. Additionally a number of other objects were arranged similarly at a distance near to the screen. The author was investigating the possibility of a virtual environment for a informal experiment in which the subject would be asked to read a random number placed on a randomly distant object. The distant objects could be occluded by nearby objects and therefore the subject would have to step side to side or perhaps crouch down in order to see around nearby objects. This task would encourage large head motions and provide a stimulus for comparing subjective impressions of scene rigidity between

false eye separation and γ-distortion. The results with γ-distortion were immediately troublesome, however. If, with false eye separation, the author had to step roughly 1 ft over in order to see around a nearby object, with γ-distortion the author had to step roughly 3 ft over. If, with false eye separation, the author had to crouch down a little bit order to see under a nearby object, with γ-distortion the author had to crouch very far down. While there was a strong subjective sense of increased in rigidity with γ-distortion the need to move through larger distances to see around near objects was objectionable. This result is quite disappointing.

Analytically, false eye separation creates a discrepancy between the depth information provided by stereo parallax and the depth information provided by head motion parallax. This discrepancy is analytically equivalent to a 3D distortion of displayed space that warps and shears with head motion. γ-distortion essentially forces the stereo parallax and motion parallax depth information back into sync, restoring rigidity to the displayed scene. However, effectively this is done at the cost of reducing the motion parallax of a given displayed point. In the investigation discussed above, the effective reduction of motion parallax was so large so as to create a new problem completely. While we are not throwing away heading-tracking information when using γ-distortion, the end result is to reduce the "reactivity" of the scene to head movement. The reduction in motion parallax could have been analytically anticipated; however, not until the algorithm was fully implemented in a system where head motion was quite large could its downside be truly appreciated. Having derived γ-distortion and examined its effect empirically, we are left with the following conclusion. In the limit, we can only achieve two of the three following outcomes: fusibility, rigidity, or reactivity. If we ignore fusion issues for scenes of arbitrary depth users will tend to suffer from eye strain but we can retain rigidity (ignoring latency) and full reactivity. If we tweak the stereo image in order to reduce eye strain, then we must choose between rigidity and reactivity.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

Chapter III presented various geometric characteristics of fusion control algorithms for stereo HTDs. Future work should include usability and perceptual studies regarding the tradeoffs between scene-depth sensitive versus insensitive implementations and active head versus resting head implementations. Our anecdotal observation has also been that more accurate models of the fusible depth range are needed to get the most out of active-head implementations. Chapter III also clarified the difference between the geometric constructions used for false eye separation and those used for hyper/hypo stereoscopy. Finally, Chapter III presented a geometric demonstration of why and when techniques such as false eye separation, which have a more complex 3D distortion, can be preferable to view scaling and perpendicular scaling which have very simple 3D distortions. While for very deep scenes perpendicular scaling yields undesirable results, there is perhaps a range of "slightly deep" scenes for which the effect on visual angle is less noticeable. The limits of this small depth range will probably vary with user preferences and across application domains. Investigating this would require usability and perceptual studies.

Chapter IV describes an orbital or exo-centric travel technique for viewing a whole-planet terrain database. The technique addresses a wide variety of stereoscopic viewing issues discussed in Chapter II. The key geometric characteristic of this environment is that the local views of the terrain surface tend to involve shallow depth ranges while the planet surface as a whole can obviously occupy a large depth range. The solution is to carefully manipulate view position, orientation and scale when transitioning between the locally shallow views. The described travel technique should be a good starting point for exo-centric travel over high-detail surfaces in other applications.

Chapter V analyzed the equivalent 3D distortions induced by false eye separation, image scaling and image shifting. The image shifting and image scaling distortions have a translation and uniform scale component respectively which interfere with the independent control of view location and view scaling. Image shifting and image scaling can also induce a geometric curvature into the displayed space. On our

desktop VR display, we anecdotally observed, however, that the curvature appears too subtle to be perceived. We suspect that perceptual studies across a wide variety of display configurations would probably show the curvature to be negligible for many, but perhaps not all, applications. Image shifting and image scaling also introduce vertical parallax. Computationally, we illustrate that under certain eye-screen configurations the vertical parallax exceeds limits mentioned in the psychophysics literature. Other geometric manipulations for fusion control, such as view placement, scale and false eye separation, do not have this property. Given the flexibility in virtual reality stereo HTDs for manipulating view placement, scale and false eye separation, all at viewing time, these results suggest that image shifting and image scaling are less appropriate for stereo HTDs. For other stereo media such as stereo photography, cinematography, and tele-operator systems, image shifting and image scaling are often one of the few manipulations possible at viewing time since these systems view the physical world, not a malleable virtual one. Also for most of these systems, head-tracking is lacking, so subtle issues of vertical parallax can arise even without image shifting and image scaling. For these two reasons, the vertical parallax of image shifting and image scaling described in Chapter V is perhaps a mute point with respect to these other stereo media.

Chapter VI investigated geometric techniques for removing the dynamic distortion components from false eye separation. A key motivation is that one advantage of adding head-tracking to a stereo display is to remove qualitatively similar dynamic distortions that occur in non-head-tracked stereo systems. The outcome of removing the dynamic components of false eye separation in stereo HTDs does not yield the unambiguously positive outcome that we hoped for. We show that geometrically it is possible to remove the dynamic components and we anecdotally observe that these geometric methods have qualitatively the expected perceptual outcome. However, the increased rigidity comes at a cost. The author observed that α-false eye separation's removal of the lateral shearing tends to make the perpendicular motion components more obvious. This is particularly noticeable in display systems that encourage perpendicular head-motion such as desktop VR and surround screen VR. It is perhaps less of an issue in horizontal displays like the virtual workbench. We then pursued the development of γ-distortion, a transformation

that is static under all head motion. We showed how to combine γ-distortion with a translation and simultaneously solve for the translation and *r* parameter of γ in order to map both the scene near point to the comfortably fusible near point and the scene far point to the comfortably fusible far point. However, once we implemented this technique, we observe that γ-distortion, in bringing the motion and stereo parallax completely into sync, may reduce the reactivity of the image to head motion by too much. This can make it difficult to perform certain tasks such as looking around nearby objects in order to see farther ones in a cluttered environment. The degree of this reduction, of course, increases with depth compression since the more we reduce stereo parallax the more we must "reduce" motion parallax to re-sync it with the stereo parallax.

The question remains whether and when the rigidity/reactivity tradeoff is worthwhile. Geometric analysis alone cannot answer this question. Hence, future formal user studies are needed to further explore this issue. Some reasonable dependent variables to investigate are subjects' subjective rating of the rigidity, realism, and motion of the perceived scene [Rund00]. One independent variable that might be useful to investigate is the effect of display size since larger displays encourage a wider range of head motion. Another independent variable might be the degree of clutter in the environment. Perhaps if the environment is less cluttered the need to look around nearby occluding objects will be less of an issue. Finally, one might expect individual differences between the preference for the rigidity and reactivity tradeoff.

Another possibility is to investigate/develop a technique that interpolates between the fully rigid result of γ-distortion and the full "head-motion reactive" result of false eye separation. Such a method would have a 'rigidity' attribute that could be altered to suit user preferences. This possibility is particularly interesting in light of Runde's work [Rund00]. Runde used a clever mechanical device that effectively alters the motion parallax of a grid of real, physical objects as the user moves side to side when viewing the objects with both eyes. He found a wide range gain factors in the motion parallax that users preferred and rated as most realistic. The average gain factor that user's choose is 0.75 and not 1.0. He found similar

results in a stereo HTD. Perhaps then a “variable rigidity” version of γ-distortion that effectively reduces motion parallax to a controllable degree would be more desirable.

APPENDIX A

FALSE EYE SEPARATION DISTORTION **Δ**

A1 Derivation of Distortion

The following figure illustrates the distortion induced by false eye separation for a head at an arbitrary position and orientation. The eye points are on the left, the projection plane is the X-Y plane, and the modeled and displayed object points, **E** and **F**, are on the right. The user's central eye point is at **I**. The left eye, **D**, is displaced by *d* and the right eye, **A**, is displaced by **-d**. 2|**d**| is the true eye separation. The scalar *r* is the ratio of the modeled eye separation to the true separation. Hence the left and right modeled eyes, **C** and **B**, are displaced by *r****d** and –*r****d** respectively, and 2*r*|**d**| is the modeled eye separation. **E** is a point on a virtual object and **H** and **G** are **E's** left and right projected images. **F** is the displayed point.

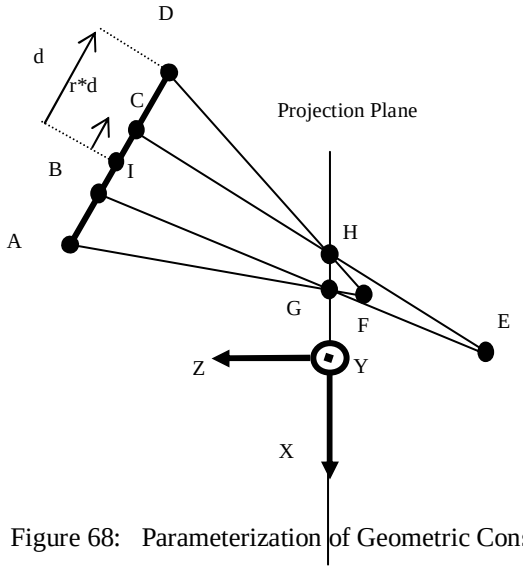


Figure 68: Parameterization of Geometric Construction

Numerous hand drawings of this construction indicated the induced transform preserved lines and was projective. We therefore developed a software program to distort a mesh of points by the construction by

computing appropriate line intersections. Chapter V Section 2 contains numerous examples. The visual results convinced us the transform was projective. Rather than pursuing a rigorous geometric proof that this construction defined a non-affine collineation, we directly pursued an analytic proof. The goal is to produce three rational linear expressions, one for each coordinate of **F,** in terms of the three coordinates of **E** such that the denominator in each of the three expressions is the same sub-expression while the terms of the three numerators are unique for each of the three coordinate expressions. So we need:

$$F_\theta = \frac{M_\theta\, Ex + N_\theta Ey + O_\theta\, Ez + P_\theta}{Q\, Ex + R\, Ey + S\, Ez + T} \qquad (A.1-1)$$

where x, y and z are symbolically subsituted for θ

A1.1 From the figure:

A = **I** – **d**
B = **I** – r***d**
C = **I** + r***d**
D = **I**+**d**

*A1.2 Solve for **H**:*

Equation of line **CH** is:

P = (**E**-**C**)*t* + **C**

At $z = 0$:

$$0 = (E_z - C_z)t + C_z \quad \Rightarrow \quad t = \frac{-C_z}{E_z - C_z} = \frac{C_z}{C_z - E_z}$$

So:

$$\mathbf{H} = (\mathbf{E} - \mathbf{C})\frac{C_z}{C_z - E_z} + \mathbf{C}$$

Or from A1.1:

$$\mathbf{H} = (\mathbf{E} - \mathbf{I} - r\,\mathbf{d})\frac{I_z + rd_z}{I_z - E_z + rd_z} + \mathbf{I} + r\,\mathbf{d}$$

*A1.3 Solve for **G**:*

Using arguments similar to A1.2:

$$\mathbf{G} = (\mathbf{E} - \mathbf{B})\frac{B_z}{B_z - E_z} + \mathbf{B} = (\mathbf{E} - \mathbf{I} + r\,\mathbf{d})\frac{I_z - rd_z}{I_z - E_z - rd_z} + \mathbf{I} - r\,\mathbf{d}$$

A1.4 Solve for Fx*:*

To begin:

$$\mathbf{F} = \overline{\mathbf{AG}} \cap \overline{\mathbf{DH}}$$

So use the following equation to find two unknowns t_a and t_d:

$$(\mathbf{G} - \mathbf{A})t_a + \mathbf{A} = (\mathbf{H} - \mathbf{D})t_d + \mathbf{D}$$

Solve for t_a for z-component:

$$(G_z - A_z)t_a + A_z = (H_z - D_z)t_d + D_z \quad \Rightarrow \quad t_a = \frac{(D_z - A_z) \quad + \quad (H_z - D_z)t_d}{(G_z - A_z)}$$

Substitute t_a in original equation's x-component:

$$(G_x - A_x)\left(\frac{D_z - A_z + (H_z - D_z)td}{G_z - A_z}\right) + A_x = (H_x - D_x)t_d + D_x$$

Solve for t_d:

$$(G_x - A_x)\left(\frac{D_z - A_z + (H_z - D_z)t_d}{G_z - A_z}\right) - (H_x - D_x)t_d = D_x - A_x$$

$$\Rightarrow \frac{(G_x - A_x)(D_z - A_z)}{(G_z - A_z)} + td\,\frac{(G_x - A_x)(H_z - D_z)}{(G_z - A_z)} - (H_x - D_x)t_d = D_x - A_x \quad \| \text{multiply through by } (G_x - A_x)$$

$$\Rightarrow t_d\left(\frac{(G_x - A_x)(H_z - D_z)}{(G_z - A_z)} - (H_x - D_x)\right) = D_x - A_x - \frac{(G_x - A_x)(D_z - A_z)}{(G_z - A_z)} \quad \| \text{factor out } t_d\text{, move non-}t_d \text{ terms to right hand side}$$

$$\Rightarrow t_d = \frac{(D_x - A_x) - \dfrac{(G_x - A_x)(D_z - A_z)}{(G_z - A_z)}}{\left(\dfrac{(G_x - A_x)(H_z - D_z)}{(G_z - A_z)} - (H_x - D_x)\right)} \quad \| \text{divide to make } t_d \text{ sole term in left hand side}$$

$$\Rightarrow t_d = \frac{(G_z - A_z)(D_x - A_x) - (G_x - A_x)(D_z - A_z)}{(G_x - A_x)(H_z - D_z) - (G_z - A_z)(H_x - D_x)} \quad \| \text{multiply numerator and denominator by } (G_z\text{-}A_z)$$

$$\Rightarrow t_d = \frac{-A_z(D_x - A_x) - (G_x - A_x)(D_z - A_z)}{(G_x - A_x)(-D_z) + A_z(H_x - D_x)} \quad \| \text{use } H_z = G_z = 0$$

$$\Rightarrow t_d = \frac{-A_z D_x + \|A_z A_x\| - G_x D_z + G_x A_z + A_x D_z - \|A_x A_z\|}{-D_z G_x + A_x D_z + A_z H_x - A_z D_x} \quad \| \text{expand, identify like terms}$$

$$\Rightarrow t_d = \frac{A_x D_z - G_x D_z + A_z G_x - A_z D_x}{-A_z D_x + A_x D_z - D_z G_x + A_z H_x} \quad \| \text{remove like terms, and reorder}$$

Now substitute in t_d in for F_x:

$$F_x = t_d(H_x - D_x) + D_x = \left(\frac{A_x D_z - G_x D_z + A_z G_x - A_z D_x}{-A_z D_x + A_x D_z - D_z G_x + A_z H_x} \right)(H_x - D_x) + D_x$$

Put F_x over common denominator, expand, identify like terms and simplify:

$$F_x = \frac{\left(A_x D_z - G_x D_z + A_z G_x - A_z D_x\right)(H_x - D_x) + D_x\left(-A_z D_x + A_x D_z - D_z G_x + A_z H_x\right)}{-A_z D_x + A_x D_z - D_z G_x + A_z H_x}$$

$$= \frac{\begin{array}{l} A_x D_z H_x - G_x D_z H_x + A_z G_x H_x - \|A_z D_x H_x\|_4 \quad -\|A_x D_z D_x\|_3 + \|G_x D_z D_x\|_1 - A_z G_x D_x \\ + \|A_z D_x D_x\|_2 - \|A_z D_x D_x\|_2 + \|A_x D_z D_x\|_3 - \|D_z G_x D_x\|_1 + \|A_z H_x D_x\|_4 \end{array}}{-A_z D_x + A_x D_z - D_z G_x + A_z H_x}$$

$\|$expand and identify like terms

$$= \frac{A_x D_z H_x - G_x D_z H_x + A_z G_x H_x - A_z G_x D_x}{-A_z D_x + A_x D_z - D_z G_x + A_z H_x} \quad \| \text{remove like terms}$$

$$= \frac{-\left(-A_x D_z H_x + G_x D_z H_x - A_z G_x H_x + A_z G_x D_x\right)}{-\left(A_z D_x - A_x D_z + D_z G_x - A_z H_x\right)} \quad \| \text{factor out -1}$$

$$= \frac{-A_x D_z H_x + G_x D_z H_x - A_z G_x H_x + A_z G_x D_x}{A_z D_x - A_x D_z + D_z G_x - A_z H_x} \quad \| \text{cancel -1}$$

$$= \frac{A_z D_x G_x - A_x D_z H_x + (D_z - A_z) G_x H_x}{A_z D_x - A_x D_z + D_z G_x - A_z H_x} \quad \| \text{reorder terms, and factor out } GxHx \qquad (26)$$

Now substitute expressions for H_x and G_x from A1.2 and A1.3 into this definition of F_x and use D_z-A_x=2d_z. This yields:

$$F_x = \frac{\begin{array}{l} A_z D_x \left(I_x - rd_x + \dfrac{(E_x - I_x + rd_x)(I_z - rd_z)}{I_z - E_z - rd_z} \right) \\ -A_x D_z \left(I_x + rd_x + \dfrac{(E_x - I_x - rd_x)(I_z + rd_z)}{I_z - E_z + rd_z} \right) \\ +(2 \quad d_z) \left(I_x - rd_x + \dfrac{(E_x - I_x + rd_x)(I_z - rd_z)}{I_z - E_z - rd_z} \right) \left(I_x + rd_x + \dfrac{(E_x - I_x - rd_x)(I_z + rd_z)}{I_z - E_z + rd_z} \right) \end{array}}{\begin{array}{l} A_z D_x \\ -A_x D_z \\ +D_z \left(I_x - rd_x + \dfrac{(E_x - I_x + rd_x)(I_z - rd_z)}{I_z - E_z - rd_z} \right) \\ -A_z \left(I_x + rd_x + \dfrac{(E_x - I_x - rd_x)(I_z + rd_z)}{I_z - E_z + rd_z} \right) \end{array}}$$

Next multiply the complete expression by:

$$\frac{(I_z - E_z + rd_z)(I_z - E_z - rd_z)}{(I_z - E_z + rd_z)(I_z - E_z - rd_z)}$$

When doing so, however, treat the numerator and denominator separately in sections 1.4.1 and 1.4.2.

A1.4.1 Solve For F_x Denominator

Begin by solving for the denominator of F_x multiplied by $(I_z\text{-}E_z\text{+}rd_z)(I_z\text{-}E_z\text{-}rd_z)$:

$$(I_z - E_z + rd_z)(I_z - E_z - rd_z)\begin{pmatrix} A_z D_x \\ -A_x D_z \\ +D_z\left(I_x - rd_x + \dfrac{\left(E_x - I_x + rd_x\right)\left(I_z - rd_z\right)}{I_z - E_z - rd_z}\right) \\ -A_z\left(I_x + rd_x + \dfrac{\left(E_x - I_x - rd_x\right)\left(I_z + rd_z\right)}{I_z - E_z + rd_z}\right)\end{pmatrix}$$

$$= (I_z - E_z + rd_z)(I_z - E_z - rd_z)\begin{pmatrix} A_z D_x \\ -A_x D_z \\ +D_z I_x - rd_x D_z + D_z\dfrac{\left(E_x - I_x + rd_x\right)\left(I_z - rd_z\right)}{I_z - E_z - rd_z} \\ -A_z I_x - rd_x A_z - A_z\dfrac{\left(E_x - I_x - rd_x\right)\left(I_z + rd_z\right)}{I_z - E_z + rd_z}\end{pmatrix} \quad \Big\| \text{ mulitply through by } D_z, A_z$$

$$\begin{aligned} &= (I_z - E_z + rd_z)(I_z - E_z - rd_z)\left(A_z D_x - A_x D_z + D_z I_x - rd_x D_z - A_z I_x - rd_x A_z\right) \\ &\quad +(I_z - E_z + rd_z)D_z\left(E_x - I_x + rd_x\right)\left(I_z - rd_z\right) \\ &\quad -(I_z - E_z - rd_z)A_z\left(E_x - I_x - rd_x\right)\left(I_z + rd_z\right) \end{aligned}$$

$\Big\|$ multiply through by $(I_z - E_z + rd_z)(I_z - E_z - rd_z)$

$$\begin{aligned} &= (I_z - E_z + rd_z)(I_z - E_z - rd_z)\left(A_z D_x - A_x D_z + I_x\left(D_z - A_z\right) - rd_x\left(A_z + D_z\right)\right) \\ &\quad +D_z(I_z - E_z + rd_z)\ \left(E_x - I_x + rd_x\right)\left(I_z - rd_z\right) \\ &\quad -A_z(I_z - E_z - rd_z)\left(E_x - I_x - rd_x\right)\left(I_z + rd_z\right) \end{aligned}$$

$\Big\|$ reassociate and commute

$$\begin{aligned} &= \begin{pmatrix} (I_z - E_z + rd_z)(I_z - E_z - rd_z) \\ \left(\left\|\left(I_z - d_z\right)\left(I_x + d_x\right) - \left(I_x - d_x\right)\left(I_z + d_z\right)\right\|_1 + \left\|I_x\left(\left(I_z + d_z\right) - \left(I_z - d_z\right)\right) - rd_x\left(\left(I_z - d_z\right) + \left(I_z + d_z\right)\right)\right\|_2\right) \end{pmatrix} \\ &\quad +\left(I_z + d_z\right)(I_z - E_z + rd_z)\ \left(E_x - I_x + rd_x\right)\left(I_z - rd_z\right) \\ &\quad -\left(I_z - d_z\right)(I_z - E_z - rd_z)\left(E_x - I_x - rd_x\right)\left(I_z + rd_z\right) \end{aligned}$$

$\Big\|$ substitute A_x, A_z, D_x, D_z from A1.1 and identify interesting patterns (1 and 2) for next step

$$= \Big((I_z - E_z + rd_z)(I_z - E_z - rd_z)\big(2I_z d_x - \|2d_z I_x\| + \|2I_x d_z\| - 2rd_x I_z\big)\Big)$$
$$+\big(I_z + d_z\big)(I_z - E_z + rd_z)\ \ \big(E_x - I_x + rd_x\big)\big(I_z - rd_z\big)$$
$$-\big(I_z - d_z\big)(I_z - E_z - rd_z)\big(E_x - I_x - rd_x\big)\big(I_z + rd_z\big)$$

$\Big\|$ Apply (A-B)(C + D)-(C-D)(A + B) = 2AD-2BC to part 1
where A = I_z,B = d_z,C = I_x,D = d_x.
Combine like terms in part 2. Identify like terms.

$$= \big\|(I_z - E_z + rd_z)(I_z - E_z - rd_z)\big\|_1\big(2I_z d_x - 2rd_x I_z\big)$$
$$\left\|\begin{array}{l}+\big(I_z + d_z\big)(I_z - E_z + rd_z)\ \ \big(E_x - I_x + rd_x\big)\big(I_z - rd_z\big)\\ -\big(I_z - d_z\big)(I_z - E_z - rd_z)\big(E_x - I_x - rd_x\big)\big(I_z + rd_z\big)\end{array}\right\|_2$$

$\|$ Combine like terms. Identify parts for next step.

$$= \Big(\big(I_z - E_z\big)^2 - r^2 d_z{}^2\Big)\big(2I_z d_x - 2rd_x I_z\big)$$
$$+2\begin{pmatrix} I_z d_z\big(I_z - E_z\big)\big(E_x - I_x\big) + I_z{}^2 rd_z\big(E_x - I_x\big) - I_z\big(I_z - E_z\big)rd_z\big(E_x - I_x\big) - d_z r^2 d_z{}^2\big(E_x - I_x\big)\\ +I_z{}^2\big(I_z - E_z\big)rd_x + I_z d_z rd_z rd_x - d_z\big(I_z - E_z\big)rd_z rd_x - I_z r^2 d_z{}^2 rd_x\end{pmatrix}$$

$\left\|\begin{array}{l}$ Part 1 : $(A+B)(A-B) = A^2 - B^2$
Part 2: (A + B)(C + D)(E + F)(A-D)-(A-B)(C-D)(E-F)(A + D) =
2(ABCE + A^2DE-ACDE-BD^2E + A^2CF + AB*DF* - *BCDF* - *A*D^2*F)*
where A = I_z, B = d_z, C = (I_z-E_z), D = r d_z, E = (E_x-I_x), F = r d_x\end{array}\right.$

$$= 2\begin{pmatrix}\Big(I_z{}^2 - 2E_z I_z + E_z{}^2 - r^2 d_z{}^2\Big)\big(I_z d_x - rd_x I_z\big)\\ +I_z d_z\big(I_z E_x - I_z I_x - E_z E_x + E_z I_x\big) + E_x I_z{}^2 rd_z - I_x I_z{}^2 rd_z - I_z rd_z\big(I_z E_x - I_z I_x - E_z E_x + E_z I_x\big)\\ -E_x d_z r^2 d_z{}^2 + I_x d_z r^2 d_z{}^2\\ +I_z I_z{}^2 rd_x - E_z I_z{}^2 rd_x + I_z d_z rd_z rd_x - I_z d_z rd_z rd_x + E_z d_z rd_z rd_x - I_z r^2 d_z{}^2 rd_x\end{pmatrix}$$

$\|$ Expand.

$$= 2 \begin{pmatrix} I_z d_x I_z^2 - 2 I_z d_x E_z I_z + I_z d_x E_z^2 - I_z d_x r^2 d_z^2 & - r d_x I_z d_z I_z^2 + 2 r d_x I_z E_z I_z - r d_x I_z E_z^2 + r d_x I_z r^2 d_z^2 \\ + I_z d_z I_z E_x - I_z d_z I_z I_x - I_z d_z E_z E_x + I_z d_z E_z I_x + E_x I_z^2 r d_z - I_x I_z^2 r d_z & \\ - I_z r d_z I_z E_x + I_z r d_z I_z I_x + I_z r d_z E_z E_x - I_z r d_z E_z I_x & \\ - E_x d_z r^2 d_z^2 + I_x d_z r^2 d_z^2 & \\ + I_z I_z^2 r d_x - E_z I_z^2 r d_x + I_z d_z r d_z r d_x - I_z d_z r d_z r d_x + E_z d_z r d_z r d_x - I_z r^2 d_z^2 r d_x & \end{pmatrix}$$

‖Expand.

$$= 2 \begin{pmatrix} I_z d_x I_z^2 - 2 I_z d_x E_z I_z + I_z d_x E_z^2 - I_z d_x r^2 d_z^2 - \| r d_x I_z I_z^2 \|_1 + 2 \| r d_x I_z E_z I_z \|_2 - r d_x I_z E_z^2 + \| r d_x I_z r^2 d_z^2 \|_3 \\ + I_z d_z I_z E_x - I_z d_z I_z I_x - I_z d_z E_z E_x + I_z d_z E_z I_x + \| E_x I_z^2 r d_z \|_4 - \| I_x I_z^2 r d_z \|_5 \\ - \| I_z r d_z I_z E_x \|_4 + \| I_z r d_z I_z I_x \|_5 + I_z r d_z E_z E_x - I_z r d_z E_z I_x \\ - E_x d_z r^2 d_z^2 + I_x d_z r^2 d_z^2 \\ + \| I_z I_z^2 r d_x \|_1 - \| E_z I_z^2 r d_x \|_2 + \| I_z d_z r d_z r d_x \|_6 - \| I_z d_z r d_z r d_x \|_6 + E_z d_z r d_z r d_x - \| I_z r^2 d_z^2 r d_x \|_3 \end{pmatrix}$$

‖Expand. Identify like terms.

$$= 2 \begin{pmatrix} d_x I_z^3 - 2 d_x E_z I_z^2 + d_x E_z^2 I_z - d_x d_z^2 I_z r^2 & & & + d_x E_z I_z^2 r - d_x E_z^2 I_z r \\ + d_z E_x I_z^2 - d_z I_x I_z^2 - d_z E_x E_z I_z + d_z E_z I_x I_z & & & \\ & + d_z E_x E_z I_z r - d_z E_z I_x I_z r & & \\ - d_z^3 E_x r^2 + d_z^3 I_x r^2 & & & \\ & & + d_x d_z^2 E_z r^2 & \end{pmatrix}$$

‖Combine like terms. Alphabetize variables.

$$= 2 \begin{pmatrix} d_x I_z^3 - 2 d_x E_z I_z^2 + d_x E_z^2 I_z - d_x d_z^2 I_z r^2 & + d_x E_z I_z^2 r - d_x E_z^2 I_z r + d_z E_x I_z^2 - d_z I_x I_z^2 \\ - d_z E_x E_z I_z + d_z E_z I_x I_z + d_z E_x E_z I_z & - d_z E_z I_x I_z r - d_z^3 E_x r^2 + d_z^3 I_x r^2 + d_x d_z^2 E_z r^2 \end{pmatrix}$$

‖ Rewrite.

$$= -2(d_z E_x - d_x E_z - d_z I_x + d_x I_z)(E_z I_z - I_z^2 - E_z I_z r + d_z^2 r^2) \qquad \text{(A1.4.1-1)}$$

‖factor by inspection. This is motivated by the need to end up with a form like A1-1.

A1.4.2 Solve for F_x Numerator:

Continue with the numerator multiplied by $(I_z-E_z+rd_z)(I_z-E_z-rd_z)$ and proceed as follows:

$$(I_z-E_z+rd_z)(I_z-E_z-rd_z)\begin{bmatrix} 2\,d_z\left(I_x-rd_x+\frac{(E_x-I_x+rd_x)(I_z-rd_z)}{I_z-E_z-rd_z}\right) \\ \left(I_x+rd_x+\frac{(E_x-I_x-rd_x)(I_z+rd_z)}{I_z-E_z+rd_z}\right) \\ -A_xD_z\left(I_x+rd_x+\frac{(E_x-I_x-rd_x)(I_z+rd_z)}{I_z-E_z+rd_z}\right) \\ +A_zD_x\left(I_x-rd_x+\frac{(E_x-I_x+rd_x)(I_z-rd_z)}{I_z-E_z-rd_z}\right) \\ \end{bmatrix}$$

$$= 2\;\; d_z\left[(I_z-E_z-rd_z)(I_x-rd_x)+(E_x-I_x+rd_x)(I_z-rd_z)\right]$$
$$\left[(I_z-E_z+rd_z)(I_x+rd_x)+(E_x-I_x-rd_x)(I_z+rd_z)\right]-$$
$$A_xD_z(I_z-E_z-rd_z)\left[(I_z-E_z+rd_z)(I_x+rd_x)+\left(E_x-I_x-rd_x\right)\left(I_z+rd_z\right)\right]+$$
$$A_zD_x(I_z-E_z+rd_z)\left[(I_z-E_z-rd_z)(I_x-rd_x)+(E_x-I_x+rd_x)(I_z-rd_z)\right]$$
$$\|\text{multiply through by } (I_z\text{-}E_z+rd_z)(I_z\text{-}E_z\text{-}rd_z)$$

$$= 2\;\; d_z\begin{bmatrix} \|I_xI_z\|_1-I_x\;E_z-\|I_x\;r\;d_z\|_2 \;\; -\|r\;d_x\;I_z\;\|_3+r\,d_x\;E_z+\|r\,d_x\;r\;d_z\;\|_4 \\ +\;I_z\;E_x\;-\|Iz\;I_x\|_1+\|I_z\;r\,d_x\|_3 \;\; -r\;d_z\;E_x+\|r\,d_z\;I_x\|_2-\|r\;d_z\;r\;d_x\|_4 \end{bmatrix}$$
$$\begin{bmatrix} \|I_xI_z\|_5-I_xE_z+\|I_x\;r\;d_z\|_6 \;\; +\|r\,d_x\;I_z\|_7-rd_x\;E_z+\|rd_x\;rd_z\|_8 \\ +\;I_zE_x-\|I_zI_x\|_5-\|I_z\;rd_x\|_7 \;\; +rd_z\;E_x-\|rd_z\;I_x\|_6-\|rd_z\;rd_x\|_8 \end{bmatrix}-$$
$$A_xD_z(I_z-E_z-rd_z)\begin{bmatrix} \|I_x\,I_z\|_9-I_x\,E_z+\|I_x\,r\,d_z\|_{10}+\|r\,d_x\;I_z\|_{11}-r\,d_x\,E_z+\|\,r\,d_x\,r\,d_z\|_{12} \\ +E_xI_z+E_xrd_z \;\; -\|I_x\;I_z\|_9-\|I_x\;rd_z\;\|_{10}-\|rd_x\;I_z\;\|_{11}-\|rd_x\;rd_z\|_{12} \end{bmatrix}+$$
$$A_zD_x(I_z-E_z+rd_z)\begin{bmatrix} \|\;I_x\;I_z\|_{13}-I_x\;E_z-\|I_x\;rd_z\|_{14} \;\; -\|rd_x\;I_z\|_{15}+rd_x\;E_z+\|rd_x\;rd_z\|_{16} \\ +E_x\;I_z-E_x\;rd_z-\|I_x\;I_z\|13+\|\;I_x\;rd_z\;\|_{14}+\|rd_x\;I_z\;\|_{15}-\|\;rd_x\;rd_z\|_{16} \end{bmatrix}$$
$$\|\text{ expand and identify like terms}$$

$$= 2\ \ d_z\left[(E_xI_z-E_zI_x)-(E_xrd_z-E_zrd_x)\right]\left[(E_xI_z-E_zI_x)+(E_xrd_z-E_zrd_x)\right]$$
$$-A_xD_z(I_z-E_z-rd_z)\left[E_xI_z+E_xrd_z\ \ -I_xE_z\ \ -rd_xE_z\ \right]$$
$$+A_zD_x(I_z-E_z+rd_z)\left[E_xI_z-E_xrd_z-I_xE_z+rd_xE_z\right]$$

$\|$ Cancel like terms. Expand.

$$= 2\ \ d_z\left[(E_xI_z-E_zI_x)^2-(E_xrd_z\ -E_zrd_x)^2\right]$$
$$-A_xD_z(I_z-E_z-rd_z)\left[E_xI_z+E_xrd_z\ \ -I_xE_z\ \ -\ rd_xE_z\ \right]$$
$$+A_zD_x(I_z-E_z+rd_z)\left[E_x\ I_z-E_xrd_z-I_xE_z+rd_xE_z\right]$$

$\|$ use: (A+B)(A-B)=A^2-B^2

$$= 2\ \ d_z\left[E_x^2I_z^2-2E_xE_zI_xI_z+E_z^{\,2}I_x^2-E_x^2r^2d_z^2+2\,E_xE_zrd_xrd_z-d_x^2E_z^2\ r^2\right]$$
$$-A_xD_z\begin{bmatrix} I_zE_xI_z+\|I_zE_xrd_z\|_1-I_zI_xE_z-I_zrd_xE_z & -E_zE_xI_z-E_zE_xrd_z+E_zI_xE_z+E_zrd_xE_z \\ -\|r\ d_zE_xI_z\|_1-rd_zE_xrd_z & +rd_zI_xE_z\ +\ rd_zrd_xE_z \end{bmatrix}$$
$$+A_zD_x\begin{bmatrix} I_zE_xI_z-\|I_zE_xrd_z\|_2-I_zI_xE_z+I_zrd_xE_z & -E_zE_xI_z+E_zE_xrd_z+E_zI_xE_z-E_zrd_xE_z \\ +\|rd_zE_xI_z\|_2-rd_zE_xrd_z-rd_zI_xE_z+rd_zrd_xE_z & \end{bmatrix}$$

$\|$ Expand. Identify like terms

$$= 2\ \ d_z\left[E_x^2I_z^2-2E_xE_zI_xI_z+E_z^{\,2}I_x^2-E_x^2r^2d_z^2+2\,d_xd_zE_xE_zr^2-d_x^2E_z^2\ r^2\right]$$
$$-A_xD_z\begin{bmatrix} E_xI_z^{\,2} & -E_zI_xI_z-d_xE_zI_zr-E_xE_zI_z-d_zE_xE_zr+E_z^{\,2}I_x+d_xE_z^{\,2}r \\ -\ d_z^{\,2}E_xr^2\ +\ d_zE_zI_xr\ +\ d_xd_zE_zr^2 & \end{bmatrix}$$
$$+\ A_zD_x\begin{bmatrix} E_xI_z^{\,2} & -E_zI_xI_z\ +\ d_xE_zI_zr\ -E_xE_zI_z+d_zE_xE_zr\ +E_z^{\,2}I_x\ -d_xE_z^{\,2}\ r \\ -d_z^{\,2}E_xr^2\ -d_zE_zI_xr\ +d_xd_zE_zr^2 & \end{bmatrix}$$

$\|$ Cancel like terms. Alphabetize variables.

$$= 2\ \ d_z\left[E_x{}^2\ \ I_z{}^2 - 2E_xE_zI_xI_z + E_z{}^2I_x{}^2 - E_x{}^2\ \ r^2d_z{}^2\ \ + 2\ \ d_xd_zE_xE_zr^2\ \ - d_x{}^2E_z{}^2\ \ r^2\right]$$

$$-A_xD_z\begin{bmatrix} E_xI_z{}^2 - \ \ E_zI_xI_z\ \ - E_xE_zI_z + E_z{}^2I_x - d_z{}^2E_xr^2\ \ + d_xd_zE_zr^2 \\ +\left(-\ \ d_xE_zI_zr - d_zE_xE_zr + d_xE_z{}^2r\ \ +\ \ d_zE_zI_xr\right)\end{bmatrix}$$

$$+A_zD_x\begin{bmatrix} E_xI_z{}^2 - \ \ E_zI_xI_z\ \ - E_xE_zI_z + E_z{}^2I_x - d_z{}^2E_xr^2\ \ + d_xd_zE_zr^2\ \ \\ -\ \ \left(-\ \ d_xE_zI_zr - d_zE_xE_zr + d_xE_z{}^2r\ \ +\ \ d_zE_zI_xr\right)\ \ \ \ \ \ \ \ \end{bmatrix}$$

$\|$ Reassociate

$$= 2\ \ d_z\left[E_x{}^2I_z{}^2 - 2E_xE_zI_xI_z + E_z{}^2I_x{}^2 - E_x{}^2r^2d_z{}^2\ \ + 2d_xd_zE_xE_zr^2\ \ - d_x{}^2E_z{}^2r^2\right]\ \ -$$

$$-\left(I_x - d_x\right)\left(I_z + d_z\right)$$

$$\left[E_xI_z{}^2 - E_zI_xI_z - E_xE_zI_z + E_z{}^2I_x - d_z{}^2E_xr^2 + d_xd_zE_zr^2 + \left(-d_xE_zI_zr - d_zE_xE_zr + d_xE_z{}^2r + d_zE_zI_xr\right)\right]$$

$$+\ \ \left(I_z - d_z\right)\left(I_x + d_x\right)$$

$$\left[E_xI_z{}^2 - E_zI_xI_z - E_xE_zI_z + E_z{}^2I_x - d_z{}^2E_xr^2 + d_xd_zE_zr^2 - \left(-d_xE_zI_zr - d_zE_xE_zr + d_xE_z{}^2r + d_zE_zI_xr\right)\right]$$

$\|$ Substitute for A_x, A_z, D_x, D_z.

$$= 2\ \ d_z\left[E_x{}^2I_z{}^2 - 2E_xE_zI_xI_z + E_z{}^2I_x{}^2 - E_x{}^2r^2d_z{}^2\ \ + 2d_xd_zE_xE_zr^2\ \ - d_x{}^2E_z{}^2r^2\right]\ \ -$$

$$\left(\overbrace{\left(I_xI_z - d_xd_z\right)}^{A} + \overbrace{\left(d_zI_x - d_xI_z\right)}^{B}\right)$$

$$\left[\overbrace{E_xI_z{}^2 - E_zI_xI_z - E_xE_zI_z + E_z{}^2I_x - d_z{}^2E_xr^2 + d_xd_zE_zr^2}^{C} + \overbrace{\left(-d_xE_zI_zr - d_zE_xE_zr + d_xE_z{}^2r + d_zE_zI_xr\right)}^{D}\right] +$$

$$\left(\overbrace{\left(I_xI_z - d_xd_z\right)}^{A} - \overbrace{\left(d_zI_x - d_xI_z\right)}^{B}\right)$$

$$\left[\overbrace{E_xI_z{}^2 - E_zI_xI_z - E_xE_zI_z + E_z{}^2I_x - d_z{}^2E_xr^2 + d_xd_zE_zr^2}^{C} - \overbrace{\left(-d_xE_zI_zr - d_zE_xE_zr + d_xE_z{}^2r + d_zE_zI_xr\right)}^{D}\right]$$

$\|$ Expand and then label like components.

$$= 2\ \ d_z\left[E_x{}^2 I_z{}^2 - 2E_x E_z I_x I_z + E_z{}^2 I_x{}^2 - E_x{}^2 r^2 d_z{}^2\ \ + 2d_x d_z E_x E_z r^2\ \ - d_x{}^2 E_z{}^2\ \ r^2\right]$$

$$-2\left(d_z I_x - d_x I_z\right)\left(E_x I_z{}^2 -\ \ E_z I_x I_z\ \ - E_x E_z I_z + E_z{}^2 I_x - d_z{}^2 E_x r^2\ \ + d_x d_z E_z r^2\right)$$

$$-2\left(I_x I_z - d_x d_z\right)\left(-\ \ d_x E_z I_z r - d_z E_x E_z r + d_x E_z{}^2 r\ \ +\ \ d_z E_z I_x r\right)$$

$\Big\|$ Use: $-(A+B)(C+D)+(A-B)(C-D) = -2BC - 2AD$
where A,B,C,D are as labeled in preceding expression (Note these labels have nothing to do with the points A,B,C,D). .

$$= 2\ \ \left(\begin{array}{l} d_z E_x{}^2 I_z{}^2 - 2d_z E_x E_z I_x I_z + d_z E_z{}^2 I_x{}^2 - d_z E_x{}^2 r^2 d_z{}^2\ \ + 2d_z d_x d_z E_x E_z r^2\ \ - d_z d_x{}^2 E_z{}^2 r^2 \\ -\left(\begin{array}{l} d_z I_x E_x I_z{}^2 -\ \ d_z I_x E_z I_x I_z\ \ - d_z I_x E_x E_z I_z + d_z I_x E_z{}^2 I_x - d_z I_x d_z{}^2 E_x r^2\ \ + d_z I_x d_x d_z E_z r^2 \\ -d_x I_z E_x I_z{}^2 + d_x I_z E_z I_x I_z\ \ + d_x I_z E_x E_z I_z - d_x I_z E_z{}^2 I_x + d_x I_z d_z{}^2 E_x r^2\ \ - d_x I_z d_x d_z E_z r^2 \end{array}\right) \\ -\left(\begin{array}{l} -I_x I_z d_x E_z I_z r - I_x I_z d_z E_x E_z r + I_x I_z d_x E_z{}^2 r\ \ + I_x I_z d_z E_z I_x r \\ +d_x d_z d_x E_z I_z r + d_x d_z d_z E_x E_z r - d_x d_z d_x E_z{}^2 r\ \ - d_x d_z d_z E_z I_x r \end{array}\right) \end{array}\right)$$

$\Big\|$ Factor out 2. Expand.

$$= 2\ \ \left(\begin{array}{l} d_z E_x{}^2 I_z{}^2 - 2d_z E_x E_z I_x I_z + d_z E_z{}^2 I_x{}^2 - d_z{}^3 E_x{}^2 r^2\ \ + 2d_x d_z{}^2 E_x E_z r^2\ \ - d_x{}^2 d_z E_z{}^2 r^2 \\ -\left(\begin{array}{l} d_z E_x I_x I_z{}^2 -\ \ d_z E_z I_x{}^2 I_z\ \ - d_z E_x E_z I_x I_z + d_z E_z{}^2 I_x{}^2 -\ \ d_z{}^3 E_x I_x r^2\ \ +\ \ d_x d_z{}^2 E_z I_x r^2 \\ -d_x E_x I_z{}^3 + d_x E_z I_x I_z{}^2\ \ + d_x E_x E_z I_z{}^2 - d_x E_z{}^2 I_x I_z\ \ + d_x d_z{}^2 E_x I_z r^2\ \ - d_x{}^2 d_z E_z I_z r^2 \end{array}\right) \\ -\left(\begin{array}{l} -\ \ d_x E_z I_x I_z{}^2 r -\ \ d_z E_x E_z I_x I_z r + d_x E_z{}^2 I_x I_z r\ \ +\ \ d_z E_z I_x{}^2 I_z r \\ +d_x{}^2 d_z E_z I_z r + d_x d_z{}^2 E_x E_z r - d_x{}^2 d_z E_z{}^2 r\ \ - d_x d_z{}^2 E_z I_x r \end{array}\right) \end{array}\right)$$

$\Big\|$ Alphabetize terms.

$$= 2 \begin{pmatrix} d_z E_x{}^2 I_z{}^2 - 2\|d_z E_x E_z I_x I_z\|_1 + \left\|d_z E_z{}^2 I_x{}^2\right\|_2 - d_z{}^3 E_x{}^2 r^2 + 2 d_x d_z{}^2 E_x E_z r^2 - d_x{}^2 d_z E_z{}^2 r^2 \\ -d_z E_x I_x I_z{}^2 + d_z E_z I_x{}^2 I_z + \|d_z E_x E_z I_x I_z\|_1 - \left\|d_z E_z{}^2 I_x{}^2\right\|_2 + d_z{}^3 E_x I_x r^2 - d_x d_z{}^2 E_z I_x r^2 \\ +d_x E_x I_z{}^3 - d_x E_z I_x I_z{}^2 - d_x E_x E_z I_z{}^2 + d_x E_z{}^2 I_x I_z - d_x d_z{}^2 E_x I_z r^2 + d_x{}^2 d_z E_z I_z r^2 \\ + d_x E_z I_x I_z{}^2 r + d_z E_x E_z I_x I_z r - d_x E_z{}^2 I_x I_z r - d_z E_z I_x{}^2 I_z r \\ -dx^2\ dz\ Ez\ Iz\ r - dx\ dz^2\ Ex\ Ez\ r + dx^2\ dz\ Ez^2\ r + dx\ dz^2 Ez\ Ix\ r \end{pmatrix}$$

‖ Expand out −1. Identify like terms.

$$= 2 \begin{pmatrix} d_z E_x{}^2 I_z{}^2 - d_z E_x E_z I_x I_z & -d_z{}^3 E_x{}^2 r^2 + 2 d_x d_z{}^2 E_x E_z r^2 & -d_x{}^2 d_z E_z{}^2 r^2 \\ -d_z E_x I_x I_z{}^2 + d_z E_z I_x{}^2 I_z & + d_z{}^3 E_x I_x r^2 & - d_x d_z{}^2 E_z I_x r^2 \\ +d_x E_x I_z{}^3 - d_x E_z I_x I_z{}^2 & -d_x E_x E_z I_z{}^2 + d_x E_z{}^2 I_x I_z & -d_x d_z{}^2 E_x I_z r^2 + d_x{}^2 d_z E_z I_z r^2 \\ + d_x E_z I_x I_z{}^2 r + d_z E_x E_z I_x I_z r - d_x E_z{}^2 I_x I_z r - d_z E_z I_x{}^2 I_z r & & \\ -d_x{}^2 d_z E_z I_z r - d_x d_z{}^2 E_x E_z r + d_x{}^2 d_z E_z{}^2 r & + d_x d_z{}^2 E_z I_x r & \end{pmatrix} \quad A1.4.2-1$$

‖ Combine like terms.

.

THIS SPACE IS INTENTIONALLY LEFT BLANK

PLEASE TURN TO THE NEXT PAGE

$$= 2\left(d_z E_x - d_x E_z - d_z I_x + d_x I_z\right)\left(-\left\|E_z I_x I_z\right\|_2 + \left\|E_x {I_z}^2\right\|_1 - \left\|d_x d_z E_z r\right\|_6 + \left\|E_z I_x I_z r\right\|_5 - \left\|d_z{}^2 E_x r^2\right\|_3 + \left\|d_x d_z E_z r^2\right\|_4\right)$$

Factor out $(d_z E_x - d_x E_z - d_z I_x + d_x I_z)$. This is motivated by the need to get the final result for F_x in form A1-1. Since the F_x denominator (A1.4.1-1) has factor $\alpha = (d_z E_x - d_x E_z - d_z I_x + d_x I_z)$, we need to extract this same factor from the previous equation (A1.4.2-1) which is the numerator of F_x. This is necessary so that in the complete fraction for F_x, α cancels out leaving a rational linear equation. To factor (A1.4.2-1), note that (A1.4.2-1) has 24 terms, counting the $2 d_x d_z{}^2 E_x E_z r^2$ twice. So we must factor (A1.4.2-1) into $\alpha \bullet \beta$ where α has the known 4 terms and β has 6 unknown terms. To find the terms of β make a table of the 4 terms of α versus the 24 terms of (A1.4.2-1) as in the table below. Divide each of (A1.4.2-1)'s 24 terms by each of α's terms, if possible, to yield a quotient term. We must now find 6 quotient terms each of which occurs in every column Additionally, when we choose these six common terms no choosen quotient term appears in the same row as another quotient term. This is why we reject the terms marked *4X and *3X. Clearly multiplying α by this choosen set of 6 quotient terms will yield β. Hence these choosen quotients terms are exactly the terms of β. These choosen terms are labeled with an *<number> in the table and then labeled in the final exression of $\alpha \bullet \beta$ given above.

	dzEx		**-dxEz**		**-dzIx**		**dxIz**	
dzEx^{2}Iz^2	ExIz^2	***1**						
-dzExEzIxIz	-Ez IxIz	***2**			ExEzIz			
-dz^3 Ex^{2}r^2	-dz^2 Ex r^2	***3**						
dxdz^{2}ExEz r^2	dx dzEz r^2	***4X**	-dz^{2}Ex r^2	***3**				
dxdz^{2}ExEz r^2	dx dzEz r^2	***4**	-dz^{2}Ex r^2	***3X**				
-dx^{2}dzEz^2 r^2			dxdzEzr^2	***4**				
-dzExIxIz^2	-IxIz^2				ExIz^2	***1**		
dzEzIx^{2}Iz					-EzIxIz	***2**		
dz^{3}ExIxr^2	dz^{2}Ixr^2				-dz^{2}Exr^2	***3**		
-dxdz^{2}EzIx r^2			-dz^{2}Ix r^2		dxdzEz r^2	***4**		
dxExIz^3							ExIz^2	***1**
-dxEzIxIz^2			IxIz^2				-EzIxIz	***2**
-dxExEzIz^2			ExIz^2	***1**			-ExEzIz	
dxEz^{2}IxIz			-EzIxIz	***2**			Ez^{2}Ix	
-dxdz^{2}ExIz r^2	-dxdzIz r^2						-dz^{2}Ex r^2	***3**
dx^{2}dzEzIz r^2			-dxdzIz r^2				dxdzEzr^2	***4**
dxEzIxIz^{2}r			-IxIz^{2}r				EzIxIz r	***5**
dzExEzIxIz r	EzIxIz r	***5**			-ExEzIz r			
-dxEz^{2}IxIz r			EzIxIz r	***5**			-Ez^{2}Ix r	
-dzEzIx^{2}Iz r					EzIxIz r	***5**		
-dx^{2}dzEzIz r			dxdzIz r				-dxdzEz r	***6**
-dxdz^{2}ExEz r	-dxdzEz r	***6**	dz^{2}Exr					
dx^{2}dzEz^2 r			-dxdzEz r	***6**				
dxdz^{2}EzIx r			-dz^{2}Ix r		-dxdzEz r	***6**		

Table 4: Factoring terms from (A1.4.2-1) by terms of α and labeling common results.

$$= -2\left(dzEx - dxEz - dzIx + dxIz\right)\left(Ez\,Ix\,Iz - Ex\,Iz^2 + dx\,dz\,Ez\,r - Ez\,Ix\;Iz\,r + dz^2\;Ex\,r^2 - dx\,dz\,Ez\,r^2\right)$$

$\|$factor out -1

A1.4.3 Solve for F_x Complete Fraction:

Now return to the complete fraction of F_x. Take the numerator from the last page of A1.4.2, page 179 and the denominator from the last page of A1.4.1, page 173. Then cancel common factors from the complete expression for F_x and then collect like terms:

$$F_x = \frac{-2\left(d_z E_x - d_x E_z - d_z I_x + d_x I_z\right)\left(E_z I_x I_z - E_x I_z{}^2 + d_x d_z E_z r - E_z I_x I_z r + d_z{}^2 E_x r^2 - d_x d_z E_z r^2\right)}{-2(d_z E_x - d_x E_z - d_z I_x + d_x I_z)(E_z I_z - I_z{}^2 - E_z I_z r + d_z{}^2 r^2)}$$

$$= \frac{\left(E_z I_x I_z - E_x I_z{}^2 + d_x d_z E_z r - E_z I_x I_z r + d_z{}^2 E_x r^2 - d_x d_z E_z r^2\right)}{(E_z I_z - I_z{}^2 - E_z I_z r + d_z{}^2 r^2)}$$

$$= \frac{E_x\left(d_z{}^2{}_r{}^2 - I_z{}^2\right) + E_z\left(I_x I_z + d_x d_z r - I_x I_z r - d_x d_z r^2\right)}{E_z\left(I_z(1-r)\right) + \left(d_z{}^2 r^2 - I_z{}^2\right)}$$

$$= \frac{E_x\left(d_z{}^2 r^2 - I_z{}^2\right) + E_z(1-r)\left(I_x I_z + d_x d_z r\right)}{E_z\left(I_z(1-r)\right) + \left(d_z{}^2 r^2 - I_z{}^2\right)}$$

A1.5 Solve for F_y

Using a parallel derivation as in A1.4:

$$F_y = \frac{E_y\left(d_z{}^2 r^2 - I_z{}^2\right) + E_z(1-r)\left(I_y I_z + d_y d_z r\right)}{E_z\left(I_z(1-r)\right) + \left(d_z{}^2 r^2 - I_z{}^2\right)}$$

A1.6 Solve for F_z

Using the initial results from A1.4:

$$F_z = t_d(H_z - D_z) + D_z = \left(\frac{D_z A_x - D_z G_x + A_z G_x - A_z D_x}{-A_z D_x + A_x D_z - D_z G_x + A_z H_x}\right)(H_z - D_z) + D_z$$

$$= \left(\frac{D_z A_x - D_z G_x + A_z G_x - A_z D_x}{-A_z D_x + A_x D_z - D_z G_x + A_z H_x}\right)(-D_z) + D_z \qquad \| H_z = 0$$

Please turn to the next page.

Rewrite the expression over a common denominator and simplify using H_z=0:

$$F_z = \frac{(D_z A_x - D_z G_x)(-D_z) + (A_z G_x - A_z D_x)(-D_z) + D_z\left(-A_z D_x + A_x D_z - D_z G_x + A_z H_x\right)}{-A_z D_x + A_x D_z - D_z G_x + A_z H_x}$$

$$= \frac{\left\|D_z A_x D_z\right\|_1 + \left\|D_z G_x D_z\right\|_2 - A_z G_x D_z + \left\|A_z D_x D_z\right\|_4 - \left\|D_z A_z D_x\right\|_4 + \left\|D_z A_x D_z\right\|_1 - \left\|D_z D_z G_x\right\|_2 + D_z A_z H_x}{-A_z D_x + A_x D_z - D_z G_x + A_z H_x}$$

$$= \frac{A_z D_z H_x - A_z D_z G_x}{-A_z D_x + A_x D_z - D_z G_x + A_z H_x} \qquad \| \text{Cancel common terms.} \qquad (27)$$

As in A1.4 proceed by treating the numerator and denominator separately and multiply both by "$(I_z$-E_z-$rd_z)(I_z$-E_z+$rd_z)$".

A1.6.1 Solve for F_z Numerator

Begin with the numerator multiplied by $(I_z$-E_z-$rd_z)(I_z$-E_z+$rd_z)$ as follows:

$$(I_z - Ed + rd_z)(I_z - E_z - rd_z)\left(-A_z D_z G_x + A_z D_z H_x\right)$$

$$= A_z D_z (I_z - E_z + rd_z)(I_z - E_z - rd_z)(H_x - G_x)$$

$$= A_z D_z (I_z - E_z + rd_z)(I_z - E_z - rd_z)$$

$$\left(\left(I_x + rd_x + \frac{(E_x - I_x - rd_x)(I_z + rd_z)}{I_z - E_z + rd_z}\right) - \left(I_x - rd_x + \frac{(E_x - I_x + rd_x)(I_z - rd_z)}{I_z - E_z - rd_z}\right)\right)$$

$\|$ Substitute for H_x (A1.2), G_x (A1.3)

$$= A_z D_z (I_z - E_z + rd_z)(I_z - E_z - rd_z)\left(2\ \ rd_x + \frac{(E_x - I_x - rd_x)(I_z + rd_z)}{I_z - E_z + rd_z} - \frac{(E_x - I_x + rd_x)(I_z - rd_z)}{I_z - E_z - rd_z}\right)$$

$\|$Combine like terms.

$$= A_z D_z \begin{pmatrix} 2\ \ rd_x \left\|(I_z - E_z + rd_z)(I_z - E_z - rd_z)\right\|_1 \\ \left\|+(I_z - E_z - rd_z)(E_x - I_x - rd_x)(I_z + rd_z)\right\| \\ -(I_z - E_z + rd_z)(E_x - I_x + rd_x)(I_z - rd_z)\Big\|_2 \end{pmatrix}$$

$\|$Multiply through by $(I_z - E_z + rd_z)(I_z - E_z - rd_z)$. Identify parts for next step.

$$= A_z D_z \begin{pmatrix} 2 \;\; rd_x\left((I_z - E_z)^2 - r^2 d_z{}^2\right) + \\ 2\left((I_z - E_z) rd_z (E_x - I_x) + \;\; r^2 d_z{}^2 rd_x - (I_z - E_z) rd_x I_z - rd_z (E_x - I_x) I_z\right) \end{pmatrix}$$

$\Big\|$ Part 1: Use (A-B)(A + B) = A^2-B^2

Part 2: Use (A-B)(C-D)(E + B)-(A + B)(C + D)(E-B) = 2(ABC + B^2D − ADE − BCE)

A = I_z-E_z, B = rd_z, C = E_x-I_x, D = rd_x, E = I_z, B = rd_z

$$= A_z D_z \begin{pmatrix} 2 \;\; rd_x\left((I_z - E_z)^2 - r^2 d_z{}^2\right) + \\ 2\left((I_z E_x - I_z I_x - E_z E_x + E_z I_x) rd_z + \;\; d_x d_z{}^2 r^3 \;\; - \;\; (I_z rd_x I_z \;\; - E_z rd_x I_z) \;\; - rd_z I_z E_x \;\; + rd_z I_z I_x\right) \end{pmatrix}$$

$\|$ Expand.

$$= A_z D_z \begin{pmatrix} 2 \;\; \left(\left\| d_x I_z{}^2 r \right\|_1 - 2d_x E_z I_z r + d_x E_z{}^2 r - \left\| d_x d_z{}^2 r^3 \right\|_2\right) + \\ 2\begin{pmatrix} \left\| I_z E_x rd_z \right\|_4 - \left\| I_z I_x rd_z \right\|_3 - E_z E_x rd_z + E_z I_x rd_z + \left\| \;\; d_x d_z{}^2 r^3 \right\|_2 - \left\| I_z rd_x I_z \right\|_1 + E_z rd_x I_z \\ - \left\| rd_z I_z E_x \right\|_4 + \left\| rd_z I_z I_x \right\|_3 \end{pmatrix} \end{pmatrix}$$

$\|$ Expand and identify like terms.

$$= 2A_z D_z \left(\;\; -2d_x E_z I_z r + d_x E_z{}^2 r - d_z E_x E_z r + d_z E_z I_x r \;\; + \;\; d_x E_z I_z r \;\; \right)$$

$\|$ Factor out 2. Combine like terms and alphabetize variables.

$$= 2\left(I_z - d_z\right)\left(I_z + d_z\right)\left(-2d_x E_z I_z r + d_x E_z{}^2 r \;\; + d_x E_z I_z r \;\; - d_z E_x E_z r + d_z E_z I_x r\right)$$

$\|$Substitute for A,D from (A1.1).

$$= 2\left(I_z{}^2 - d_z{}^2\right)\left(-2d_x E_z I_z r + d_x E_z{}^2 r \;\; + d_x E_z I_z r \;\; - d_z E_x E_z r + d_z E_z I_x r\right)$$

$\|$Expand.

$$= 2\begin{pmatrix} -2I_z{}^2 d_x E_z I_z r + I_z{}^2 d_x E_z{}^2 r \;\; + I_z{}^2 d_x E_z I_z r \;\; - I_z{}^2 d_z E_x E_z r + I_z{}^2 d_z E_z I_x r \\ --d_z{}^2 2d_x E_z I_z r + -d_z{}^2 d_x E_z{}^2 r \;\; + \;\; -d_z{}^2 d_x E_z I_z r \;\; --d_z{}^2 d_z E_x E_z r + -d_z{}^2 d_z E_z I_x r \end{pmatrix}$$

$\|$Expand.

$$= 2\begin{pmatrix} -2\left\|d_x E_z I_z{}^3 r\right\|_1 + d_x E_z{}^2 I_z{}^2 r & +\left\|d_x E_z I_z{}^3 r\right\|_1 & -d_z E_x E_z I_z{}^2 r + d_z E_z I_x I_z{}^2 r \\ +2\left\|d_x d_z{}^2 E_z I_z r\right\|_2 - d_x d_z{}^2 E_z{}^2 r & -\left\|d_x d_z{}^2 E_z I_z r\right\|_2 & +d_z{}^3 E_x E_z r - d_z{}^3 E_z I_x r \end{pmatrix}$$

$\|$Alphabetize and identify like terms.

$$= 2\begin{pmatrix} -d_x E_z I_z{}^3 r + d_x E_z{}^2 I_z{}^2 r & -d_z E_x E_z I_z{}^2 r + d_z E_z I_x I_z{}^2 r \\ +d_x d_z{}^2 E_z I_z r - d_x d_z{}^2 E_z{}^2 r & +d_z{}^3 E_x E_z r - d_z{}^3 E_z I_x r \end{pmatrix}$$

$\|$Combine like terms.

$$= 2E_z r\begin{pmatrix} -d_x Iz^3 & +d_x E_z I_z{}^2 & -d_z E_x I_z{}^2 & +d_z I_x I_z{}^2 \\ +d_x d_z{}^2 I_z - d_x d_z{}^2 E_z & +d_z{}^3 E_x & -d_z{}^3 I_x & \end{pmatrix}$$

$\|$Factor out E_z and r.

$$= 2\,E_z\,r(d_z{}^2 - I_z{}^2)\,(d_z\,E_x - d_x E_z - d_z I_x + d_x I_z)$$

$\|$Factor.

A1.6.2 Solve For F_z Denominator

Simplifying the denominator is practically completed from derivation A1.4.1:

$$Den = (I_z - E_z - rd_z)(I_z - E_z + rd_z)\left(-A_z D_x + A_x D_z - D_z G_x + A_z H_x\right)$$

$$= -\left(I_z - E_z - rd_z\right)\left(I_z - E_z + rd_z\right)\left(A_z D_x - A_x D_z + D_z G_x - A_z H_x\right)$$

$$= -\left(-2\left(d_z E_x - d_x E_z - d_z I_x + d_x I_z\right)\left(E_z I_z - I_z{}^2 - E_z I_z r + d_z{}^2 r^2\right)\right)$$

$\|$This from the result of A1.4.1 which found a nearly identical expression for the denominator of F_x multiplied by $(I_z - E_z - rd_z)(I_z - E_z + rd_z)$. We just need to fiddle with the sign.

$$= 2\left(d_z E_x - d_x E_z - d_z I_x + d_x I_z\right)\left(E_z I_z - I_z{}^2 - E_z I_z r + d_z{}^2 r^2\right)$$

A1.6.3 Solve For F_z Fraction

Now return to the complete fraction, cancel common expression and collect like terms:

$$F_z = \frac{2\,E_z\,r(d_z^{\,2} - I_z^{\,2})(d_z\,E_x - d_x\,E_z - d_z I_x + d_x\,I_z)}{2\left(d_z E_x - d_x E_z - d_z I_x + d_x I_z\right)\left(E_z I_z - I_z^{\,2} - E_z I_z r + d_z^{\,2} r^2\right)}$$

$$= \frac{E_z\,r\,(d_z^{\,2} - I_z^{\,2})}{E_z\left(I_z(1-r)\right) + \left(d_z^{\,2} r^2 - I_z^{\,2}\right)} \qquad \| \text{Cancel common factors and factor denominator.}$$

A2 Rewrite in matrix form

Rewriting the coordinate equations in matrix from A1.4-A1.6 yields Δ':

$$\mathbf{\Delta}' = \begin{bmatrix} \left(d_z^{\,2} r^2 - I_z^{\,2}\right) & 0 & (1-r)\left(I_x I_z + d_x d_z r\right) & 0 \\ 0 & \left(d_z^{\,2} r^2 - I_z^{\,2}\right) & (1-r)\left(I_y I_z + d_y d_z r\right) & 0 \\ 0 & 0 & r(d_z^{\,2} - I_z^{\,2}) & 0 \\ 0 & 0 & I_z(1-r) & d_z^{\,2} r^2 - I_z^{\,2} \end{bmatrix} \qquad (28)$$

Note this will degenerate to a singular transform if any of the 4 true or false eye points become embedded in the view plane, but in practice this should not happen. Hence we ignore this case. Given this assumption and the fact that scalar multiples of a projective transformation matrix are equivalent rewrite Δ:

$$\mathbf{\Delta} = \begin{bmatrix} 1 & 0 & \dfrac{(1-r)\left(I_x I_z + d_x d_z r\right)}{d_z^{\,2} r^2 - I_z^{\,2}} & 0 \\ 0 & 1 & \dfrac{(1-r)\left(I_y I_z + d_y d_z r\right)}{d_z^{\,2} r^2 - I_z^{\,2}} & 0 \\ 0 & 0 & \dfrac{r(d_z^{\,2} - I_z^{\,2})}{d_z^{\,2} r^2 - I_z^{\,2}} & 0 \\ 0 & 0 & \dfrac{I_z(1-r)}{d_z^{\,2} r^2 - I_z^{\,2}} & 1 \end{bmatrix}$$

Finally Δ can be decomposed as follows:

$$\Delta = \Delta_{Project} \bullet \Delta_{Scale} \bullet \Delta_{Shear}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{I_z(1-r)}{r(d_z^2 - I_z^2)} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{r(d_z^2 - I_z^2)}{d_z^2 r^2 - I_z^2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{(1-r)(I_x I_z + d_x d_z r)}{d_z^2 r^2 - I_z^2} & 0 \\ 0 & 1 & \frac{(1-r)(I_y I_z + d_y d_z r)}{d_z^2 r^2 - I_z^2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (29)$$

For completeness Δ^{-1} can be found component wise:

$$\Delta^{-1} = \Delta_{Shear}^{-1} \bullet \Delta_{Scale}^{-1} \bullet \Delta_{Project}^{-1}$$

$$= \begin{bmatrix} 1 & 0 & \frac{-(1-r)(I_x I_z + d_x d_z r)}{d_z^2 r^2 - I_z^2} & 0 \\ 0 & 1 & \frac{-(1-r)(I_y I_z + d_y d_z r)}{d_z^2 r^2 - I_z^2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{d_z^2 r^2 - I_z^2}{r(d_z^2 - I_z^2)} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{-I_z(1-r)}{r(d_z^2 - I_z^2)} & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & \frac{-(1-r)(I_x I_z + d_x d_z r)}{r(d_z^2 - I_z^2)} & 0 \\ 0 & 1 & \frac{-(1-r)(I_y I_z + d_y d_z r)}{r(d_z^2 - I_z^2)} & 0 \\ 0 & 0 & \frac{d_z^2 r^2 - I_z^2}{r(d_z^2 - I_z^2)} & 0 \\ 0 & 0 & \frac{-I_z(1-r)}{r(d_z^2 - I_z^2)} & 1 \end{bmatrix}$$

APPENDIX B

IMAGE SCALING DISTORTION Δ_{SC}

B1 Parallel Case

Figure 50, page 115 illustrates the geometric construction for image scaling. To briefly review, the eye points are on the left in blue and the projection plane is in the X-Y plane. The projection window is centered about the origin. (Note, Figure 50 only shows a portion of the projection window so the window does not actually appear centered in the diagram). **E** is the modeled object point and **F** is the displayed point. The user's central eye point is at **I**. The left eye, **D**, is displaced by **d** from **I** and the right eye, **A**, is displaced by **-d**. 2|**d**| is the true eye separation. We assume correct modeling of the eye separation. **E** is projected onto the points **H** and **G** on the projection plane. Image scaling by scalar factor *s* scales points **H** and **G** into points $s\cdot\mathbf{H}$ and $s\cdot\mathbf{G}$. These scaled points are those the user sees. This image manipulation has the same effect as if the 3D point **E** were mapped to **F**. Now in this section we assume the eye axis is parallel to the projection window so d_z=0. As discussed in Chapter V, Section 3.1 this yields a well defined transformation of 3 space.

B1.1 From Figure 50:

$$\mathbf{A} = \mathbf{I} - \mathbf{d}$$
$$\mathbf{D} = \mathbf{I} + \mathbf{d}$$
$$d_z = 0$$

*B1.2 Solve for **H**:*

Equation of line **DH** is:

$$\mathbf{P} = (\mathbf{E}-\mathbf{D})t + \mathbf{D}$$

At $z = 0$:

$$t = \frac{D_z}{D_z - E_z}$$

So:

$$\mathbf{H} = (\mathbf{E} - \mathbf{D})\frac{D_z}{D_z - E_z} + \mathbf{D}$$

Or from B1.1:

$$\mathbf{H} = (\mathbf{E} - \mathbf{I} - \mathbf{d})\frac{I_z + d_z}{I_z + d_z - E_z} + \mathbf{I} + \mathbf{d}$$

*B1.3 Solve for **G**:*

Using arguments similar to B1.2:

$$\mathbf{G} = (\mathbf{E} - \mathbf{A})\frac{A_z}{A_z - E_z} + \mathbf{A} = (\mathbf{E} - \mathbf{I} + \mathbf{d})\frac{I_z - d_z}{I_z - d_z - E_z} + \mathbf{I} - \mathbf{d}$$

B1.4 Solve for F_x*:*

To begin:

$$\mathbf{F} = \overline{\mathbf{A} s \mathbf{G}} \cap \overline{\mathbf{D} s \mathbf{H}}$$

In Appendix A1.4 we derived a similar result for distortions due to false eye separation. By substituting $s \cdot \mathbf{G}$ for the **G** and $s \cdot \mathbf{H}$ for the **H** for in equation (26), page 169, we have

$$F_x = \frac{A_z D_x \, s G_x - A_x D_z \, s H_x + (D_z - A_z) \, s G_x \, s H_x}{A_z D_x - A_x D_z + D_z \, s G_x - A_z \, s H_x}$$

Unlike Appendix A, the author, having "discovered" Mathematica [Wolf96], will use this mathematics tool for the remainder of this derivation. To find F_x, we use Mathematica with the following input file:

```
Hx=(Ex-Ix-dx)(Iz+dz)/(Iz-Ez+dz)+Ix+dx
Gx=(Ex-Ix+dx)(Iz-dz)/(Iz-Ez-dz)+Ix-dx

Dx = Ix + dx
Dz = Iz + dz
Ax = Ix - dx
Az = Iz - dz
```

```
dz = 0

(* FX *)
Num=Expand[Az Dx s Gx - Ax Dz s Hx + (Dz-Az)s Gx s Hx]
Num = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Num]
Num = Factor[Num]

Den = Expand[Az Dx - Ax Dz + Dz s Gx - Az s Hx]
Den = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Den]
Den = Factor[Den]

Fx = Cancel[Num/Den]
Fx = Collect[Numerator[Fx],{Ez,Ex}]/Collect[Denominator[Fx],{Ex,Ez}]
```

The final result is:

$$F_x = \frac{E_x I_z s}{E_z(s-1)+I_z}$$

B1.5 Solve for F_y*:*

Solving for F_y uses a parallel derivation to F_x, yielding:

$$F_y = \frac{E_y I_z s}{E_z(s-1)+I_z}$$

B1.6 Solve for F_z*:*

Using the initial results from Appendix A1.6 (see equation (27), page 182) and substituting s **H** for **H** and s **G** for **G** in (27):

$$F_z = \frac{A_z D_z sH_x - A_z D_z sG_x}{-A_z D_x + A_x D_z - D_z sG_x + A_z sH_x}$$

Expression for **H** and **G** are in the above sections B1.2 and B1.3. To find F_z, we use Mathematica with the following input file:

```
Hx=(Ex-Ix-dx)(Iz+dz)/(Iz-Ez+dz)+Ix+dx
Gx=(Ex-Ix+dx)(Iz-dz)/(Iz-Ez-dz)+Ix-dx

Dx = Ix + dx
Dz = Iz + dz
Ax = Ix - dx
Az = Iz - dz
dz = 0
```

```
(* FZ *)
Num = Expand[Az Dz s Hx - Az Dz s Gx]
Num = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Num]
Num = Factor[Num]

Den = Expand[-Az Dx + Ax Dz - Dz s Gx + Az s Hx]
Den = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Den]
Den = Factor[Den]

Fz = Cancel[Num/Den]
Fz = Collect[Numerator[Fz],{Ez}]/Collect[Denominator[Fz],{Ez}]
```

This yields:

$$F_z = \frac{E_z I_z s}{E_z (s-1) + I_z}$$

B1.7 Rewrite in Matrix Form:

Combining the coordinate equations from B1.4-B1.6 and using column vector notation the distortion matrix is:

$$\Delta_{sc}^{p} = \begin{bmatrix} I_z s & 0 & 0 & 0 \\ 0 & I_z s & 0 & 0 \\ 0 & 0 & I_z s & 0 \\ 0 & 0 & s-1 & I_z \end{bmatrix} = \begin{bmatrix} s & 0 & 0 & 0 \\ 0 & s & 0 & 0 \\ 0 & 0 & s & 0 \\ 0 & 0 & \frac{s-1}{I_z} & 1 \end{bmatrix}$$

The last simplification is possible since such matrices are only unique up to a scale factor and we assume the eye axis center is not embedded in the projection plane.

B2 Non-Parallel Case

If we assume the eye axis is not parallel to the projection plane as shown in Figure 50, then as discussed in Chapter V, Section 3.1 the epipolar constraint is generally violated. An exception occurs for 3D points whose epipolar plane contains the origin of the scale. For simplicity, assume **A**,**D**,**d** and **E** are restricted to be in the XZ plane (see Figure 50). Since A_y,D_y,d_y,E_y equal zero and the y coordinates of all dependent points, **H**, **G**, etc., are also zero, this yields a well-defined point at location **F** for points in the y=0 plane. The derivation for this planar distortion is parallel to Section 1 of this Appendix. When

computing the expressions for F_x and F_z just remove the line 'dz=0' in the respective Mathematica files (Section B1.4 and B1.6). F_y is simply 0. The resulting expression from Mathematica is:

$$\Delta_{sc}^{xz}: \quad F_x = \frac{\left(\begin{array}{l} E_x\left(d_z I_x - d_x I_z\right)\left(d_z{}^2 - I_z{}^2\right)s + E_x{}^2\left(d_z \quad s^2\right)\left(-d_z{}^2 + I_z{}^2\right) + \\ E_x E_z \quad s\left(-2d_z I_x I_z\left(-1+s\right) + d_x\left(-d_z{}^2 - I_z{}^2 + 2d_z{}^2 s\right)\right) - E_z{}^2\left(d_z\right)\left(d_x{}^2 - I_x{}^2\right)\left(-1+s\right)s \end{array}\right)}{w}$$

$$F_z = \left(E_x E_z\left(-d_z s\left(d_z{}^2 - I_z{}^2\right)\right) \quad + E_z\left(d_z{}^2 - I_z{}^2\right)\left(d_z I_x - d_x I_z\right)s + E_z{}^2\left(d_x \quad s \quad \left(d_z{}^2 - I_z{}^2\right)\right)\right) / w$$

$$w = \left(\begin{array}{l} E_x\left(d_z \quad s \quad \left(-d_z{}^2 + I_z{}^2\right)\right) + E_z\left(d_x I_z{}^2\left(-2+s\right) - 2d_z I_x I_z\left(-1+s\right) + d_x d_z{}^2 s\right) + \\ E_z{}^2\left(d_z I_x - d_x I_z\right)\left(-1+s\right) + \left(d_z I_x - d_x I_z\right)\left(d_z{}^2 - I_z{}^2\right) \end{array}\right)$$

Or more compactly:

$$\Delta_{sc}^{xz}: \quad F_x = \frac{\left(E_x{}^2\left(-L\,O\,s\right) + E_x\left(N\,L\,s\right) + E_x E_z\, s\left(P + d_x\left(-d_z{}^2 - I_z{}^2 + 2O d_z\right)\right) - E_z{}^2\left(d_x{}^2 - I_x{}^2\right)M\,O\right)}{w}$$

$$F_z = \frac{\left(E_x E_z\left(-O\,L\right) \quad + E_z\, L\,N\,s + E_z{}^2\left(d_x \quad s \quad L\right)\right)}{w}$$

$$w = \left(E_x\left(O\left(-L\right)\right) + E_z\left(d_x I_z{}^2\left(-2+s\right) + P + d_x O\, d_z\right) + E_z{}^2 N\,M + N\,L\right)$$

where

$$L = \left(d_z{}^2 - I_z{}^2\right)$$
$$M = \left(-1+s\right)$$
$$N = \left(d_z I_x - d_x I_z\right)$$
$$O = d_z\, s$$
$$P = -2d_z I_x I_z\, M$$

APPENDIX C

IMAGE SHIFTING DISTORTION Δ_{SH}

C1 Parallel Case

Figure 53, page 118 illustrates the geometric construction for image shifting. To briefly review, the eye points are on the left in blue and the projection plane is in the X-Y plane. The projection window is centered about the origin. (Figure 53 only shows a portion of the projection window so the window does not actually appear centered in the diagram). **E** is the modeled object point and **F** is the displayed point. The user's central eye point is at **I**. The left eye, **D**, is displaced by **d** from **I** and the right eye, **A**, is displaced by **-d**. 2|**d**| is the true/modeled eye separation. **E** is projected onto the points **H** and **G** on the projection plane. Image shifting by shift factor τ translates points **H** and **G** into points $\mathbf{H'}=\mathbf{H}-\tau\mathbf{T}$ and $\mathbf{G'}=\mathbf{G}+\tau\mathbf{T}$. **T** is the direction of translation. This section assumes the eye axis is parallel to the projection window so d_z=0. As discussed in Chapter V, Section 3.2 this yields a well defined transformation of 3 space.

C1.1 From Figure 53:

$\mathbf{A} = \mathbf{I} - \mathbf{d}$
$\mathbf{D} = \mathbf{I} + \mathbf{d}$
$d_z=0$
$\mathbf{T} = (d_x/\text{sqrt}(d_x^2+ d_y^2),\ d_y/\text{sqrt}(d_x^2+ d_y^2), 0)$

C1.2 Solve for F_x*:*

To begin:

$$\mathbf{F} = \overline{\mathbf{AG}'} \cap \overline{\mathbf{DH}'}$$

where

$$\mathbf{G}' = \mathbf{G} + \tau\mathbf{T}$$

$$\mathbf{H}' = \mathbf{H} - \tau\mathbf{T}$$

Expressions for **H** and **G** the same as in Appendix B, Section 1.2 and Appendix B, Section 1.3. By substituting **G′** for the **G** and **H′** for the **H** in equation (26), page 169, we have

$$F_x = \frac{A_z D_x\ G'_x - A_x D_z\ H'_x + (D_z - A_z)\ G'_x\ H'_x}{A_z D_x - A_x D_z + D_z\ G'_x - A_z\ H'_x}$$

We use the following Mathematica [Wolf96] program:

```
Clear["*"]

Hx=(Ex-Ix-dx)(Iz+dz)/(Iz-Ez+dz)+Ix+dx
Gx=(Ex-Ix+dx)(Iz-dz)/(Iz-Ez-dz)+Ix-dx

Tx = dx / Sqrt[dx*dx + dy*dy]

Hx = Hx - tau Tx
Gx = Gx + tau Tx

Dx = Ix + dx
Dz = Iz + dz
Ax = Ix - dx
Az = Iz - dz
dz = 0

(* FX *)
Num=Expand[Az Dx Gx - Ax Dz Hx + (Dz-Az) Gx Hx]
Num = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Num]
Num = Factor[Num]

Den = Expand[Az Dx - Ax Dz + Dz  Gx - Az Hx]
Den = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Den]
Den = Factor[Den]

Fx = Cancel[Num/Den]
Fx = Collect[Numerator[Fx],{Ex,Ez}]/Collect[Denominator[Fx],{Ex,Ez}]
```

The final result is:

$$F_x = \frac{E_x I_z \sqrt{d_x{}^2 + d_y{}^2} - E_z I_x \tau + I_x I_z \tau}{-E_z \tau + I_z \left(\tau + \sqrt{d_x{}^2 + d_y{}^2} \right)}$$

C1.3 Solve for F_y*:*

Solving for F_y uses a parallel derivation to F_x (B1.2), yielding:

$$F_y = \frac{E_y I_z \sqrt{d_x{}^2 + d_y{}^2} - E_z I_y \tau + I_y I_z \tau}{-E_z \tau + I_z \left(\tau + \sqrt{d_x{}^2 + d_y{}^2} \right)}$$

C1.4 Solve for F_z*:*

Using the initial results from Appendix A1.6 (see equation (27), page 182) and substituting **H′**for **H** and *s* **G′** for **G** in (27):

$$F_z = \frac{A_z D_z H'_x - A_z D_z G'_x}{-A_z D_x + A_x D_z - D_z G'_x + A_z H'_x}$$

Next we use Mathematica with the following input file:

```
Clear["*"]

Hx=(Ex-Ix-dx)(Iz+dz)/(Iz-Ez+dz)+Ix+dx
Gx=(Ex-Ix+dx)(Iz-dz)/(Iz-Ez-dz)+Ix-dx

Tx = dx / Sqrt[dx*dx + dy*dy]

Hx = Hx - tau Tx
Gx = Gx + tau Tx

Dx = Ix + dx
Dz = Iz + dz
Ax = Ix - dx
Az = Iz - dz
dz = 0


(* FZ *)
Num = Expand[Az Dz  Hx - Az Dz Gx]
Num = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Num]
Num = Factor[Num]

Den = Expand[-Az Dx + Ax Dz - Dz  Gx + Az  Hx]
Den = Cancel[(Iz-Ez+dz)(Iz-Ez-dz)Den]
```

```
Den = Factor[Den]

Fz = Cancel[Num/Den]
Fz = Collect[Numerator[Fz],{Ez}]/Collect[Denominator[Fz],{Ez}]
```

This yields:

$$F_z = \frac{E_z I_z\left(\sqrt{{d_x}^2 + {d_y}^2} - \tau\right) + {I_z}^2\tau}{-E_z\tau + I_z\left(\tau + \sqrt{{d_x}^2 + {d_y}^2}\right)}$$

C1.5 Rewrite in Matrix Form:

Combining the coordinate equations from sections C1.2-C1.4 and using a column vector notation yields the matrix:

$$\Delta_{sh}^{p} = \begin{bmatrix} I_z Q & 0 & -I_x\tau & I_x I_z\tau \\ 0 & I_z Q & -I_y\tau & I_y I_z\tau \\ 0 & 0 & I_z Q - I_z\tau & {I_z}^2\tau \\ 0 & 0 & -\tau & I_z(\tau + Q) \end{bmatrix} = \begin{bmatrix} Q & 0 & -\frac{I_x}{I_z}\tau & I_x\tau \\ 0 & Q & -\frac{I_y}{I_z}\tau & I_y\tau \\ 0 & 0 & Q-\tau & I_z\tau \\ 0 & 0 & -\tau/I_z & \tau + Q \end{bmatrix}$$

where

$$Q = \sqrt{{d_x}^2 + {d_y}^2}$$

The final simplification is possible since collineation matrices are only unique up to a scale factor and we assume the eye axis center is not embedded in the projection plane.

C2 Non-Parallel Case

If we assume the eye axis is not parallel to the projection plane as shown in Figure 53, then as discussed in Chapter V, Section 3.2 the epipolar constraint is generally violated. An exception occurs for 3D points whose epipolar plane's epipolar line is parallel to **T** the direction of image translation. For simplicity, assume the eyes (**A**,**D**,**I**,**d**) and the modeled point, **E,** are restricted to be in the XZ plane (see Figure 53). Since A_y,D_y,E_y equal zero and the y coordinates of all dependent projected points, **H**, **G**, etc.,

are also zero, this yields a well-defined displayed point, **F**. The derivation for this planar distortion is parallel to section C1. When computing the expressions for F_x and F_z just remove the line 'dz=0' in the respective Mathematica files (section C1.2 and C1.4). F_y is simply 0. The resulting expression from Mathematica is:

$$\Delta_{sh}^{xz} : F_x = \begin{pmatrix} E_x\left(d_x \quad d_z{}^3 I_x - d_x{}^2 d_z{}^2 I_z - d_x d_z I_x I_z{}^2 + d_x{}^2 I_z{}^3\right) + E_x{}^2\left(-d_x d_z{}^3 + d_x \quad d_z \quad I_z{}^2\right) \\ +E_x E_z\left(d_x{}^2 d_z{}^2 - d_x{}^2 I_z{}^2 - 2|d_x| d_z{}^2 \tau\right) + E_z\left(2|d_x| d_z{}^2 I_x \tau - 2|d_x| I_x I_z{}^2 \tau + 2 d_x d_z I_z \tau^2\right) \\ +E_z{}^2\left(d_x |d_x| d_z \tau + |d_x| I_x I_z \tau - d_x d_z \tau^2\right) \\ +\left(d_x |d_x| d_z{}^3 \tau - |d_x| d_z{}^2 I_x I_z \tau - d_x |d_x| d_z I_z{}^2 \tau + |d_x| I_x I_z{}^3 \tau + d_x d_z{}^3 \tau^2 - d_x d_z I_z{}^2 \tau^2\right) \end{pmatrix} / w$$

$$F_z = -\begin{pmatrix} E_x E_z\left(-d_x d_z\left(I_z{}^2 - d_z{}^2\right)\right) \\ +E_z\left(I_z{}^2 - d_z{}^2\right)\left(d_x d_z I_x - d_x{}^2 I_z + 2|d_x| I_z \tau\right) \\ +E_z{}^2\left(I_z{}^2 - d_z{}^2\right)\left(d_x{}^2 - |d_x|\tau\right) \\ +\left(I_z{}^2 - d_z{}^2\right)\left(|d_x| d_z{}^2 \tau - |d_x| I_z{}^2 \tau\right) \end{pmatrix} / w$$

$$w = E_x\left(-d_x d_z{}^3 + d_x d_z I_z{}^2\right) + E_z\left(d_x{}^2 d_z{}^2 - d_x{}^2 I_z{}^2 - 2|d_x| I_z{}^2 \tau\right) + E_z{}^2\left(|d_x| I_z \tau\right) + \left(d_x d_z{}^3 I_x - d_x{}^2 d_z{}^2 I_z - d_x d_z I_x I_z{}^2 + d_x{}^2 I_z{}^3 - |d_x| d_z{}^2 I_z \tau + |d_x| I_z{}^3 \tau\right)$$

APPENDIX D

ASPECTS OF α-FALSE EYE SEPARATION

D1 Distortion of α-false eye separation

From equation (29) (pg 186) and equation (19) (pg 142) the distortion due to α-false eye separation is as follows:

$$\Delta_{\alpha} = \Delta_{Project} \bullet \Delta_{Scale} \bullet \Delta_{Shear} \bullet \alpha$$

$$= \Delta_{Project} \bullet \Delta_{Scale} \bullet \begin{bmatrix} 1 & 0 & \frac{(1-r)(IxIz + dx\,dz\,r)}{dz^2 r^2 - Iz^2} & 0 \\ 0 & 1 & \frac{(1-r)(IyIz + dy\,dz\,r)}{dz^2 r^2 - Iz^2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{-(1-r)(IxIz + dx\,dz\,r - Iz\,Fx)}{dz^2 r^2 - Iz^2} & 0 \\ 0 & 1 & \frac{-(1-r)(IyIz + dy\,dz\,r - Iz\,Fy)}{dz^2 r^2 - Iz^2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \Delta_{Project} \bullet \Delta_{Scale} \bullet \begin{bmatrix} 1 & 0 & \frac{(1-r)(IxIz + dx \quad dz \quad r)}{dz^2 r^2 - Iz^2} - \frac{(1-r)(IxIz + dx \quad dz \quad r - Iz \quad Fx)}{dz^2 r^2 - Iz^2} & 0 \\ 0 & 1 & \frac{(1-r)(IyIz + dy \quad dz \quad r)}{dz^2 r^2 - Iz^2} - \frac{(1-r)(IyIz + dy \quad dz \quad r - Iz \quad Fy)}{dz^2 r^2 - Iz^2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \Delta_{Project} \bullet \Delta_{Scale} \bullet \begin{bmatrix} 1 & 0 & \frac{(1-r)(Iz \quad Fx)}{dz^2 r^2 - Iz^2} & 0 \\ 0 & 1 & \frac{(1-r)(IzFy)}{dz^2 r^2 - Iz^2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \dfrac{Iz(1-r)}{r(dz^2\text{-}Iz^2)} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{r(dz^2\text{-}Iz^2)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \dfrac{(1-r)(Iz\,Fx)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 1 & \dfrac{(1-r)(IzFy)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \dfrac{Iz(1-r)}{r(dz^2\text{-}Iz^2)} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \dfrac{(1-r)(Iz\,Fx)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 1 & \dfrac{(1-r)(IzFy)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 0 & \dfrac{r(dz^2\text{-}Iz^2)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dfrac{(1-r)(Iz\,Fx)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 1 & \dfrac{(1-r)(IzFy)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 0 & \dfrac{r(dz^2\text{-}Iz^2)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 0 & \dfrac{Iz(1-r)}{dz^2r^2 - Iz^2} & 1 \end{bmatrix}$$

D2 Fixed Depth Plane

There are several characteristics planes of such a collineation. One such plane is the plane parallel to the fixed plane (here the projection plane) and through the transformations center. The points on a plane through the center map to other points on the same plane. Hence the mentioned plane is a fixed-depth plane where depth is the z coordinate in projection plane coordinates. Then we can find the fixed-depth plane as follows. Let **P** be some arbitrary point and let **P′** be its image:

$$P' = \Delta_{\alpha} P$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} xw' \\ yw' \\ zw' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dfrac{(1-r)(Iz\quad Fx)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 1 & \dfrac{(1-r)(IzFy)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 0 & \dfrac{r(dz^2 - Iz^2)}{dz^2r^2 - Iz^2} & 0 \\ 0 & 0 & \dfrac{Iz(1-r)}{dz^2r^2 - Iz^2} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Hence:

$$z' = \frac{z\dfrac{r(dz^2 - Iz^2)}{dz^2 r^2 - Iz^2}}{z\dfrac{Iz(1-r)}{dz^2 r^2 - Iz^2} + 1}$$

So for a fixed-depth plane z maps to itself and we have:

$$z = \frac{z\dfrac{r(dz^2 - Iz^2)}{dz^2 r^2 - Iz^2}}{z\dfrac{Iz(1-r)}{dz^2 r^2 - Iz^2} + 1} = \frac{z\ \ r(dz^2 - Iz^2)}{z\ \ Iz(1-r) + dz^2 r^2 - Iz^2}$$

$$\Rightarrow z^2\ \ Iz(1-r) + z\ \ dz^2 r^2 - z\ \ Iz^2 = z\ \ r(dz^2 - Iz^2)$$

$$\Rightarrow Z\left(z\ \ Iz(1-r) + dz^2 r^2 - Iz^2 - r(dz^2 - Iz^2)\ \right) = 0$$

$$\Rightarrow z\ \ Iz(1-r) + dz^2 r^2 - Iz^2 - r(dz^2 - Iz^2) = 0$$

$z = 0$ solution represents the fixed plane but we are interested in this other solution.

$$\Rightarrow z\ \ Iz(1-r) = r(dz^2 - Iz^2) - dz^2 r^2 + Iz^2$$

$$\Rightarrow z\ = \frac{r(dz^2 - Iz^2) - dz^2 r^2 + Iz^2}{Iz(1-r)}$$

APPENDIX E

DERIVATION OF β

$\boldsymbol{\beta}$ must map all of far space to within the farthest fusible depth, *ff*, and all of near space to within the nearest fusible depth, *nf*. We will first construct $\boldsymbol{\beta}_F$ to handle far space, followed by $\boldsymbol{\beta}_N$ to handle near space and then $\boldsymbol{\beta}$ will be the composition of these matrices.

$\boldsymbol{\beta}_F$ must map all of far space to the farthest fusible depth, *ff*. There are several ways to derive this component. Let *f* be the maximum depth plane that satisfactorily maps *ff* to *fp*. We begin by deriving the inverse $\boldsymbol{\beta}_F^{-1}$. $\boldsymbol{\beta}_F^{-1}$ maps the plane Z=*f* to the plane of ideal points. So for all points **P**=(x,y,*f*,1), **P'** = $\boldsymbol{\beta}_F^{-1}$ **P** where **P'**=(x',y',z',0).

$$\begin{bmatrix} x' \\ y' \\ z' \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & A & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ f \\ 1 \end{bmatrix}$$

$$\Rightarrow A f + 1 = 0$$
$$\Rightarrow A = -1/f$$
$$\Rightarrow \beta_F^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix}$$

By inspection it is easy to see that $\boldsymbol{\beta}_F$, the inverse of $\boldsymbol{\beta}_F^{-1}$, is:

$$\beta_F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{bmatrix}$$

We choose *f* in order to map *fp* to *ff*:

$$\begin{bmatrix} x' \\ y' \\ ff \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ fp \\ 1 \end{bmatrix}$$

$$\Rightarrow ff = fp/(fp/f + 1)$$

$$\Rightarrow f = \begin{cases} fp/(fp/ff - 1) & , fp \neq \infty \\ ff & , fp = \infty \end{cases}$$

To handle near space we must map a near plane, *np*, to the nearest fusible plane, *nf*. Ignoring for the moment $\boldsymbol{\beta}_F$, we could treat this as a perpendicular scale (a scale along Z). The scale factor would be *nf*/*np*. The scale must be applied before $\boldsymbol{\beta}_F$ to avoid affecting the maximum depth plane. We want to map *np* to *nf*' where *nf*' is the pre-image of *nf* under $\boldsymbol{\beta}_F$. Examine the effect of $\boldsymbol{\beta}_F$ on *nf*'. Let **P** be an arbitrary point on the near fusible plane's pre-image, **P**=(*x*,*y*,*nf*',1). Then:

$$P' = \beta_F \cdot P$$

$$\Rightarrow \begin{bmatrix} x' \\ y' \\ n \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/f & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ nf' \\ 1 \end{bmatrix}$$

$$\Rightarrow nf = nf'/(nf'/f + 1)$$

$$\Rightarrow nf' = nf/(1 - nf/f)$$

So our correct scale factor is *nf*'/*np* =*(nf/(1-nf/f)) / np* yielding:

$$\beta_N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{nf}{np(1 - nf/f)} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now in combination we get $\boldsymbol{\beta}$:

$$\beta(np, nf, fp, ff) = \beta_F \beta_N$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \mathrm{A} & 0 \\ 0 & 0 & \mathrm{A}(1/f) & 1 \end{bmatrix}$$

$$\mathrm{A} = \frac{nf}{np(1 - nf/f)}$$

$$f = \begin{cases} fp/(fp/ff - 1) & , fp \neq \infty \\ ff & , fp = \infty \end{cases}$$

APPENDIX F

γ-DISTORTION

F1 Solution for *t* and *r*

We need to solve for *t* and *r* in:

ff = r (fp+t)/((fp+t)(r-1)/Cz+1)

nf = r (np+t) / ((np+t)(r-1)/Cz + 1)

We also have the following constraints:

C1: nf > 0
C2: np > 0
C3: ff < 0
C4: fp < 0
C5: Cz > 0
C6: fp < ff
C7: np > nf

The fusion algorithm only computes solutions to the two equations if near and far geometry exist (np > 0 and fp < 0) and both are unfusible (fp < ff) and (np > nf). Generally, users can always fuse some range of space so *ff < 0* and *nf* > 0 and from the derivation of γ, ***Cz*** should be greater than 0. Using commercial analysis tools [Wolf96], we find two solutions:

$$t = \pm \frac{1}{2Cz(ff - nf)} \left(\Phi \mp \left(Cz^2 (ff - nf) + -Cz(ff - nf)(fp - np) \right) \right)$$

$$r = \mp \frac{1}{2(Cz - ff)(Cz - nf)(fp - np)} \left(\Phi \pm \left(Cz^2 (- ff + nf) + Cz(ff + nf)(fp - np) + 2\, ff\, nf\, (- fp + np) \right) \right)$$

$$\Phi = \sqrt{\left(Cz^2 (ff - nf) + 2\, ff\, nf\, (fp - np) - Cz(ff + nf)(fp - np) \right)^2 - 4(Cz - ff)\, ff\, (Cz - nf)\, nf\, (fp - np)^2}$$

(30)

An immediate question is whether *t* and *r* are always real under our constraints. This depends on the sign of the term in **Φ** under the radical. To see whether this term is negative we first examine when the

square-rooted expression of Φ is zero. Each of the following steps were performed with Mathematica. This expression can be factored as follows:

$$Cz^2\,(ff-nf)\,(Cz^2\,ff-2\,Cz\,ff\,fp+ff\,fp^2-Cz^2\,nf-2\,Cz\,fp\,nf+4\,ff\,fp\,nf-fp^2\,nf+2\,Cz\,ff\,np-2\,ff\,fp\,np+2\,Cz\,nf\,np-4\,ff\,nf\,np+2\,fp\,nf\,np+ff\,np^2-nf\,np^2) \tag{31}$$

This expression is 0 if any factor is 0. *Cz*=0 and *ff=nf* both violate the constraints. So only third factor can make (31) zero. This third factor is quadratic in *Cz*. It equals 0 if and only if:

$$Cz \to \frac{1}{ff-nf}\left(ff\,fp+fp\,nf-ff\,np-nf\,np-2\sqrt{-ff^2\,fp\,nf+ff\,fp^2\,nf+ff\,fp\,nf^2+ff^2\,nf\,np-2\,ff\,fp\,nf\,np-ff\,nf^2\,np+ff\,nf\,np^2}\right)$$

$$Cz \to \frac{1}{ff-nf}\left(ff\,fp+fp\,nf-ff\,np-nf\,np+2\sqrt{-ff^2\,fp\,nf+ff\,fp^2\,nf+ff\,fp\,nf^2+ff^2\,nf\,np-2\,ff\,fp\,nf\,np-ff\,nf^2\,np+ff\,nf\,np^2}\right) \tag{32}$$

Do these yield a C_z within our constraints? By constraint, C_z must be real. This occurs only if square-rooted expression in (32) is positive or zero. Factoring this new expression yields:

$$-ff\,nf\,(fp-np)\,(ff-fp-nf+np) \tag{33}$$

From our constraints *-ff nf* is positive and *(fp – np)* is negative. Next, *ff-fp-nf + np* > 0 is equivalent to *ff* > *fp+nf-np*. But *nf – np* is negative from C7: np > nf and C6: ff > fp . Hence *ff* > *fp+nf-np* is true and the third factor is positive. Therefore (33) is negative and the evaluation of (32) yields complex C_z. This in turn implies (31) is only zero within C1-C7 when *Cz* is complex which is physically meaningless.

So (31) is not zero within constraints, but perhaps it's negative. However, plugging in a value (ff=-1, fp = -1000, np = 0.75, nf = 0.5, Cz = 1.0) into (31) yields a positive 2.25×10^6. Equation (31) is continuous since it is a polynomial with positive integer exponents. Because (31) is continuous, never zero within our constraints and has at least one positive point, (31) must be positive for all points within our constraints.

Hence Φ in (30) is real and the solutions for *t* and *r* are real numbers. Other possible degeneracies in this solution for *t* and *r* are discussed in Chapter VI, Section 3.3

F2 Predistortion implementation of γ

If we want to implement the γ-technique as a pre-distortion matrix with false eye separation we need an $\boldsymbol{M}$ such that:

$$\boldsymbol{\gamma} = \boldsymbol{\Delta M}$$

$$\Rightarrow \boldsymbol{M} = \boldsymbol{\Delta}^{-1}\boldsymbol{\gamma}$$

$$= \begin{bmatrix} 1 & 0 & \dfrac{-(1-r)(IxIz + dx \quad dz \quad r)}{r(dz^2 - Iz^2)} & 0 \\ 0 & 1 & \dfrac{-(1-r)(IyIz + dy \quad dz \quad r)}{r(dz^2 - Iz^2)} & 0 \\ 0 & 0 & \dfrac{dz^2r^2 - Iz^2}{r(dz^2 - Iz^2)} & 0 \\ 0 & 0 & \dfrac{-Iz(1-r)}{r(dz^2 - Iz^2)} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & (r-1)Cx/Cz & 0 \\ 0 & 1 & (r-1)Cy/Cz & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & (r-1)/Cz & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & X_1 & 0 \\ 0 & 1 & X_2 & 0 \\ 0 & 0 & X_3 & 0 \\ 0 & 0 & (r-1)\left[\dfrac{Iz}{(dz^2 - Iz^2)} + \dfrac{1}{Cz}\right] & 1 \end{bmatrix}$$

REFERENCES

[Akka99] Robert Akka, Converting Existing Applications to Support High Quality Stereoscopy, *IS&T/SPIE Conference on Stereoscopic Displays and Applications X*, (January 1999, San Jose, California), pg 290-298.

[Bara97] Gregory Baratoff, Distortion of Stereoscopic Visual Space, Technical Report CAR-TR-861, Center for Automation Research, University of Maryland, College Park, USA, May 1997.

[Barf98] Woodrow Barfield, Blake Hannaford, John M. Hollerbach, Thomas B. Sheridan, David Zeltzer, Michael Zyda, *Presence: Teleoperators and Virtual Environments*, April, 1998.

[Bart95] Lynn Bartram, Albert Ho, John Dill, Frane Henigman, The Continuous Zoom: A Constrained Fisheye Technique for Viewing and Navigating Large Information Spaces, *Proceedings of ACM UIST'95*, (1995, Pittsburgh, PA), ACM Press, pg 207-215.

[Bed94] Benjamin B. Bederson, James D. Hollan, Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics, *Proceedings of ACM UIST'94*, (1994, Marina Del Ray, CA), ACM Press, pg 15-16.

[Berc98] John Bercovitz, Image-side Perspective and Stereoscopy, *IS&T/SPIE's Stereoscopic Displays and Applications IX*, (January 1998, San Jose, California, USA), vol.3295, pg 288-298.

[Bow97] Doug A. Bowman, David Koller. Larry F. Hodges, Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques, *Proceedings of IEEE 1997 Virtual Reality Annual International Symposium*, (1-5 March 1997, Albuquerque, NM, USA), pg 45-52.

[Bow99] Doug A. Bowman, Interaction Techniques for Common Tasks in Virtual Environments: Design, Evaluation and Application, Doctoral Dissertation, Georgia Institute of Technology, 1999.

[Chen88] Michael Chen, S. Joy Mountford and Abigail Sellen, A study in interactive 3-d rotation using 2-d control devices, *Computer Graphics (SIGGRAPH '88 Proceedings)*, vol. 22, no. 4, August 1988, pg 121-129.

[Cruz93] C. Cruz-Neira, D.J. Sandin, and T.A. DeFanti, Surround-screen projection-based virtual reality: the design and implementation of the CAVE, *SIGGRAPH 93 Conference Proceedings*, Annual Conference Series, ACM SIGGRAPH, Addison Wesley, August 1993, pg 135-142.

[Cutl97] Lawrence D. Cutler, Bernd Fröhlich and Pat Hanrahan, Two-handed direct manipulation on the Responsive Workbench, *Proceedings 1997 Symposium on Interactive 3D Graphics*, Providence, RI, USA 27-30 April 1997, pg 107-114/191.

[Cutt97] James E Cutting, How the eye measures reality and virtual reality, *Behavioral Research Methods, Instruments & Computers*, vol. 29, no. 1, Feb 1997, pg 27-36.

[Dav95] Elizabeth Thorpe Davis and Larry F. Hodges, "Human Stereopsis, Fusion, and Stereoscopic Virtual Environments," in Woodrow Barfield and Thomas A. Furness III editors, *Virtual Environments and Advanced Interface Design*, Oxford University Press, New York, Oxford, 1995.

[Deer92] Michael Deering, High Resolution Virtual Reality, *Computer Graphics (SIGGRAPH 92 Conference Proceedings)*, vol. 26, July 1992, pg 195-202

[Dew54] H. Dewhurst, *Introduction to 3-D; three dimensional photography in motion pictures. With chapters on wide-screen, Cinemascope, Cinerama, and stereo television***,** Chapman & Hall, London, 1954.

[Durb98] Jim Durbin, J. Edward Swan II, Brad Colbert, John Crowe, Rob King, Tony King, Chris Scannell, Zachary Wartell and Terry Welsh, Battlefield Visualization on the Responsive Workbench, *Proceedings IEEE Visualization '98*, (October 18-23, Research Triangle Park, North Carolina), IEEE Computer Society Press, 1998, pg 463-466.

[Ellis97] Stephen R. Ellis and Brian M. Menges, Judgements of the Distance to Nearby Virtual Objects: Interaction of Viewing Conditions and Accommodative Demand, *Presence*, vol. 6, no. 1, August 1992, pg 452-460.

[Fend67] D. Fender, B. Julesz, Extension of Panum's Fusional Area in Binocularly Stabilized Vision, *Journal of the Optical Society of America*, vol. 57, no. 6, June 1967, pg 819-831.

[Fol92] James D. Foley, Andries Van Dam, Steven K. Feiner and John F. Huges, *Computer Graphics: Principles and Practice*, Addison-Wesley Publishing Company, 1992.

[Froh95] Bernd Fröhlich, Berthold Krish, Wolfgang Krüger and Gerold Wesche, Further Development of the Responsive Workbench, Virtual Environments '95, Selected Papers of the *Eurographics Workshops*, pg 237-246.

[Furn95] George W. Furnas and Benjamin Bederson, Space-scale Diagrams: Understanding Multiscale Interfaces, *Proceedings CHI'95 Mosaic of Creativity*, (1995, Denver, Colorado), pg 234-240.

[Gold92] Ronald N Goldman, Decomposing Projective Transformations, in David Kirk, editor, *Computer Graphics Gems III*, Boston: Harcourt Brace Jovanovich, 1992.

[Gold96] E. Bruce Goldstein, *Sensation and Perception*, Pacific Grove, CA: Brooks/Cole Publishers, 1996.

[Gobl95] John C. Goble, Ken Hinckley, Randy Pausch, John W. Snell, Neal F. Kassel, Two-Handed Spatial Interface Tools for Neurosurgical Planning, *IEEE Computer*, vol. 28, no.7, July 1995, pg 20-6.

[Guan98] Chua Gim Guan, Luis Serra, Ralf A. Kockro, Ng Hern, Wieslaw L. Nowinski, Chumpon Chan, Volume-Based Tumor Neurosurgery Planning in the Virtual Workbench, *Proceedings of IEEE Virtual Reality Annual Internation Symposium*. (Atlanta, GA. 14-18, March 1998), pg 167-173.

[Hall97] Michael Halle, Autostereoscopic displays and computer graphics, *Computer Graphics*, vol. 31, no.2, May 1997, pg 58-62.

[Helm62] Herman von Helmholtz, *Helmholt'z Treatise on Physiological Optics*, Translated from the Third German Edition, Edited by James P. C. Southall, Volume III, New York, Dover Publications, 1962.

[Hersh99] Maurice Hershenson, *Visual space perception; a Primer*, Cambridge, Mass., MIT Press, 1999.

[Hix95] Deborah Hix, James N. Templeman and Robert J.K. Jacob, Pre-Screen Projection: From Concept to Testing of a New Interaction Technique, *Proceedings of CHI'95 Mosaic of Creativity*, (1995 Denver, Colorado), pg 226-232.

[Hodg85] Larry F. Hodges and David McAllister, Stereo and Alternating-Pair Techniques for Display of Computer-Generated Images, *IEEE Computer Graphics and Applications*, vol.5, no.9, September 1985, pg 38-45.

[Hodg90] Larry F. Hodges and David F. McAllister, Rotation algorithm artifacts in stereoscopic images. *Optical Engineering*, vol. 29, no. 8, August 1990, pg 973-976.

[Hodg92] Larry F. Hodges, Tutorial: Time-Multiplexed Stereoscopic Computer Graphics, *IEEE Computer Graphics and Applications*, vol.12, no.2, March 1992, pg 20-30.

[Hodg93a] Larry F. Hodges and Elizabeth Thorpe Davis, Geometric Considerations for Stereoscopic Virtual Environments, *Presence*, vol. 2, no. 1, 1993, pg 34-43.

[Hodg93b] Larry F. Hodges and David McAllister, "Computing Stereoscopic Views", in David McAllister, editor, *Stereo Computer Graphics and Other True 3D Technologies*, Princeton University Press-Princeton, New Jersey, 1993.

[John95] Jeff. A. Johnson, A Comparison of User Interfaces for Panning on a Touch-Controlled Display, *Proceedings of CHI'95 Mosaic of Creativity*, (1995 Denver, Colorado), pg 218-225.

[Jule71] Bela Julesz, *Foundations of Cyclopean Perception*, The University of Chicago Press, Chicago and London, 1971.

[Kess00] G. Drew Kessler, *SVE 2.1: The Simple Virtual Environment Toolkit*, Department of Computer and Information Science, Lehigh University, 2000.

[Krug94] W. Krüger and B. Fröhlich, The Responsive Workbench (virtual work environment), *IEEE Computer Graphics and Applications*, vol. 14, no. 3, May 1994, pg 12-15.

[Krug95] Wolfgang Krüger, Christian-A. Bohn, Bernd Fröhlich, Henrich Schüth, Wolfgang Strauss, Gerold Wesche, The Responsive Workbench: A Virtual Work Environment, *IEEE Computer*, vol. 28, no. 7, July 1995, pg 42-48.

[Leig96] J. Leigh, A.E. Johnson, C.A. Vasilakis, T.A DeFanti, Multi-perspective collaborative design in persistent networked virtual, *Proceedings of the IEEE 1996 Virtual Reality Annual International*, Santa Clara, CA, USA, 30 March-3 April 1996, pg 253-260/271-272.

[Lieb94] Henry Lieberman, Power of Ten Thousand: Navigating in Large Information Spaces, *Proceedings of ACM UIST'94*, (1994, Marina Del Ray, CA), ACM Press, pg 15-16.

[Lind96] Peter Lindstrom, David Koller, William Ribarsky, Larry Hodges, Nick Faust, and Gregory Turner, Real-Time Continuous Level of Detail Rendering of Height Fields, *Computer Graphics (SIGGRAPH 96)*, 1996, pg 109-118.

[Lipt82] Lenny Lipton, *Foundations of the Stereoscopic Cinema: A Study in Depth*, Van Nostrand Reinhold, 1982.

[Lipt93] Lenny Lipton, "Composition for Electrostereoscopic Displays," *Stereo Computer Graphics and Other True 3D Technologies*, David McAllister, ed., Princeton University Press-Princeton, New Jersey, 1993.

[MacA54] D. L. MacAdam, Stereoscopic Perceptions of Size, Shape, Distance, and Direction, *Journal of the SMPTE*, vol. 62, pg 271-293.

[Mack90] Jock D. Mackinlay, Stuart K. Card, and George G. Robertson, Rapid Controlled Movement Through a Virtual 3D Workspace, *(SIGGRAPH 90) Computer Graphics*, vol. 24, no. 4, August 1990, pg 171-176.

[Mart96] William L. Martens, Physiological approach to optimal stereographic game programming: a technical guide, *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 2653, 1996, pg 261-70.

[McAl93] David McAllister, "Introduction", in David McAllister, editor, *Stereo Computer Graphics and Other True 3D Technologies*, Princeton University Press-Princeton, New Jersey, 1993.

[McKe92] Michael McKenna, Interactive Viewpoint Control and Three-Dimensional Operations, *ACM 1992 Symposium on Interactive 3D Graphics, Special Issue of Computer Graphics*, pg 53-56.

[Mey92] Kenneth Meyer, Hugh L. Applewhite, Frank A. Biocca, A Survey of Position Trackers, *Presence*, vol. 1, no. 2, Spring 1992, pg 173-200.

[Milan99] Milan Sonka, Vaclav Hlavac, Roger Boyle, *Image Processing, Analysis, and Machine Vision*, Brooks/Cole Publishing Company, Pacific Grove, California, 1999.

[Mine97] Mark. R. Mine, Frederick P. Brooks Jr., Carlo H. Sequin, Moving Objects In Space: Exploiting Proprioception In Virtual Environment Interaction, *Computer Graphics Proceedings, Annual Conference Series*, 1997, pg 19-26.

[Mon95] M. Mon-Williams, J.P. Wann, and S. Rushton, Design factors in stereoscopic virtual-reality displays. *Journal of SID*, vol. 3, no. 4, 1995, pg 207-210.

[Naga96] Shojiro Nagata, The binocular fusion of human stereoscopic displays-fields of view and enviroment effects, *Ergonomics*, vol. 39, 1996, pg 1273-1284.

[Norl39] J. A. Norling, Three-Dimensional Motion Pictures, *Journal of Society of Motion Pictures and Entertainment*, Dec. 1939, pg 612-634.

[Norm90] Donald A. Norman, *The Design of Everyday Things*, Doubleday, New York, 1990.

[Obey96] Upul Obeysekare, Chas Williams, Jim Durbin, Larry Rosenblum, Robert Rosenberg, Frenando Grinstein, Ravi Ramamurti, Alexandra Landsberg and William Sandberg, *Proceedings IEEE Visualization '96*, pg 345-9/500.

[Omura96] K. Omura, S. Shiwa and F. Kishino, 3-D display with Accommodative Compensation (3DDAC) Employing Real-Time Gaze Detection, *SID 96 Digest*, pg 889-892.

[Robi92] Warren Robinett and Jannick P. Rolland, A Computational Model for the Stereoscopic Optics of a Head-Mounted Display, *Presence*, vol. 1, no. 1, 1992, pg 45-62.

[Robi95] Warren Robinett and Richard Holloway, The Visual Display Transformation for Virtual Reality, *Presence*, vol. 4, no. 1, Winter 1995, pg 1-23.

[Rose93] Louis B. Rosenberg, The Effect of Interocular Distance upon Operator Performance using Stereoscopic Displays to Perform Virtual Depth Tasks, Proceedings of *IEEE Virtual Reality Annual International Symposium 93*, Sept. 1993, pg 27-32.

[Rose97] Larry Rosenblum, (editor), Applications of the Responsive Workbench, *IEEE Computer Graphics and Applications*, vol. 17, no. 4. July-August 1997, pg 10-15.

[Rule41a] John T. Rule, The Geometry of Stereoscopic Projection, *Journal of the Optical Society of America*, vol. 31, pg 325-334.

[Rule41b] John T. Rule, The Shape of Stereoscopic Images, *Journal of the Optical Society of America*, vol. 28, pg 313-322.

[Schm83] Christopher Schmandt, Spatial Input/Display Correspondence in a Stereoscopic Computer Graphic Workstation. *Computer Graphics*, vol. 17, no. 3, July 1983, pg 253-61.

[Serr95] Luis Serra, Tim Poston, Ng Hern, Heng Pheng Ann and Chua Beng Choon, Virtual Space Editing of Tagged MRI Heart Data, *Computer Vision, Virtual Reality and Robotics in Medicine, First International Conference, CRMed '95,* 1995, pg 70-76.

[Sieg00] Mel Siegel, Shojiro Nagata, Just Enough Reality: Comfortable 3D Viewing via Microstereopsis, *IEEE Transactions on Circuits and Systems for Video Technology*, April 2000, vol.10, no.3, pg 18-27.

[Sou92] David A. Southard, Transformations for Stereoscopic Visual Simulation, *Computer & Graphics*, vol. 16, no. 4, 1992, pg 401-410.

[Sou95] David A. Southard, Viewing Model for Virtual Environment Displays, *Journal of Electronic Imaging*, October 1995, vol. 4, no. 4, pg 413–420.

[Spot53] Raymond Spottiswoode, *The theory of stereoscopic transmission & its application to the motion picture***,** Berkeley, University of California Press, 1953.

[Stoa95] Richard Stoakley, Matthew J. Conway, Randy Pausch, Virtual Reality on a WIM: Interactive Worlds in Miniature, *Proceedings of CHI'95 Mosaic of Creativity*, (1995 Denver, Colorado), pg 266-272.

[Sugi98] T. Sughihara, T. Miyasato, A Lightweight 3-D HMD with Accommodative Compensation, *SID 98 Digest*, pg 927-930.

[Surd97] R. Troy Surdick, Elizabeth T. Davis, Robert A. King, Larry F. Hodges, The Perception of Distance in Simulated Displays, *Presence*, vol. 6, no. 5, October 1997, pg 513-531.

[Suth68] Ivan E. Sutherland, A head-mounted three dimensional display, *Proceedings of the Fall Joint Computer Conference*, vol. 33, pg 757-764.

[Vero90] Harry Veron, David A. Southard, Jeffrey R. Leger, John L. Conway, Stereoscopic Displays for Terrain Database Visualization, *SPIE: Stereoscopic Displays and Applications*, vol. 1256, 1990, pg 124-130.

[Valy66] N. A.Valyus, *Stereoscopy*, The Focal Press, London and New York, 1966.

[Wagn85] Mark Wagner, The metric of visual space, *Perception & Psychophysics*, 1985, vol. 38, no. 6, pg 483-495.

[Ware90] Colin Ware and Steven Osborne, Exploration and Virtual Camera Control in Virtual Three Dimensional Environments, *Proceedings of the 1990 Symposium on Interactive 3D graphics* (Snowbird, Utah, March), *Computer Graphics*, vol. 24, no.2, pg 175-183.

[Ware93] Colin Ware, Kevin Arthur and Kellogg S. Booth, Fish Tank Virtual Reality, in proceedings of *InterChi '93*, April 1993, pg 37-41.

[Ware95a] Colin Ware, Cyril Gobrecht and Mark Paton, Algorithm for dynamic disparity Adjustment, *Proceedings of the SPIE - The International Society for Optical Engineering, Stereoscopic Displays and Virtual Reality Systems II*, vol. 2409, Feb. 1995, pg 150-156.

[Ware95b] Colin Ware. Dynamic Stereo Displays, in proceedings of *CHI'95 Mosaic of Creativity*, (1995 Denver, Colorado), pg 310-316.

[Ware97] Colin Ware and Daniel Fleet, Integrating Flying and Fish Tank Metaphors with Cyclopean Scale, *Proceedings. Computer Graphics International*, Hasselt and Diepenbeek, Belgium 23-27 June 1997, pg 39-46.

[Ware98] Colin Ware, Cyril Gobrecht, and Mark Andrew Paton, Dynamic Adjustment of Stereo Display Parameters, *IEEE Transactions on Systems, Man and Cybernetics—Part A: Systems and Humans*, vol. 28, no. 1, January 1998, pg 56-65.

[WarZ98a] Zachary Wartell, William Ribarsky Larry Hodges, Efficient Ray Intersection with Global Terrain using Spheroidal Height-Augmented Quadtrees, GVU Technical Report 98-38, College of Computing, Georgia Institute of Technology, 1998.

[WarZ99a] Zachary Wartell, Larry Hodges, William Ribarsky, The Analytic Distortion Induced by False-Eye Separation in Head-Tracked Stereoscopic Displays, GVU Technical Report 99-01, College of Computing, Georgia Institute of Technology, 1999.

[WarZ99b] Zachary Wartell, William Ribarsky, Larry Hodges, Third-Person Navigation of Whole-Planet Terrain in a Head-Tracked Stereoscopic Environment, *Proceedings of IEEE Virtual Reality '99 Conference*, (March 13-17, Houston, Texas), IEEE Computer Society Press, 1999, pg 141-148.

[WarZ99c] Zachary Wartell, William Ribarsky, Larry Hodges, Efficient Ray Intersection for Visualization and Navigation of Global Terrain using Spheroidal Height-Augmented Quadtrees, *Data Visualization '99, Proceedings of the Joint EUROGRAPHICS and IEEE TCVG Symposium on Visualization*, (May 26-28, 1999, Vienna, Austria), pg 213-223.

[WarZ99d] Zachary Wartell, Larry F. Hodges, William Ribarsky, Balancing Fusion, Image Depth, and Distortion in Stereoscopic Head-Tracked Displays, *SIGGRAPH 99 Conference Proceedings, Annual Conference Series*, ACM SIGGRAPH, Addison Wesley, August 1999, pg 351-357.

[Wats95] B.A. Watson, L. F. Hodges, Using texture maps to correct for optical distortion in head-mounted displays, proceedings of *IEEE Virtual Reality Annual International Symposium 95* (VRAIS '95), pg 172-178.

[Will90] Steven P. Williams and Russel V. Parrish, New computational control techniques and increased understanding for stereo 3-D displays, *Stereoscopic Displays and Applications*, SPIE vol. 1256, 1990, pg 73-82.

[Wolf96] Mathematica, Wolfram Research Inc., 1988-1996.

[Woo97] Mason Woo, Jackie Neider and Tom Davis, *OpenGL Program Guide*, Addison-Wesley Developers Press, Reading, Massachusetts, 1997.

[Wood93] Andrew Woods, Tom Docherty and Rolf Koch, Image Distortion in Stereoscopic Video Systems, in *Proceedings of the SPIE – The International Society for Optical Engineering, Stereoscopic Displays and Applications IV*, vol. 1915, 1993, pg 36 – 48.

[Wyli70] Clarence Raymond Wylie, *Introduction to projective geometry*, New York, Mcgraw-hill, 1970.

[Yeh90] Yei-Yu Yeh and Louis D. Silverstein, Limits of Fusion and Depth Judgments in Stereoscopic Color Displays, *Human Factors*, vol. 32, no. 1, Feb. 1990, pg 45-60.

[Yeh93] Yei-Yu Yeh, "Visual and Perceptual Issues in Stereoscopic Color Displays," *Stereo Computer Graphics and Other True 3D Technologies*, David McAllister, ed., Princeton University Press-Princeton, New Jersey, 1993.

[Yos99] Shunsuke Yoshida, Shin-ya Miyazaki, Toshihito Hoshino, Jun-ichi Hasegawa, Toru Ozeki and Takami Yasuda, A technique for precise depth representation in stereoscopic display, *Proceedings of Computer Graphics International 1999*, (7-11 June 1999, Canmore, Alta., Canada), pg 80-84.

[Zhai95] Shumin Zhai, Human Performance in Six Degree of Freedom Input Control, Doctoral Dissertation, Department of Industrial Engineering, University of Toronto, 1995.