Warren Robinett and Jannick P. Rolland
Computer Science Department,
University of North Carolina,
Chapel Hill, North Carolina
27599-3175

A Computational Model for the Stereoscopic Optics of a Head-Mounted Display

Abstract

For stereoscopic photography or telepresence, orthostereoscopy occurs when the perceived size, shape, and relative position of objects in the three-dimensional scene being viewed match those of the physical objects in front of the camera. In virtual reality, the simulated scene has no physical counterpart, so orthostereoscopy must be defined in this case as constancy, as the head moves around, of the perceived size, shape, and relative positions of the simulated objects.

Achieving this constancy requires that the computational model used to generate the graphics matches the physical geometry of the head-mounted display being used. This geometry includes the optics used to image the displays and the placement of the displays with respect to the eyes. The model may fail to match the geometry because model parameters are difficult to measure accurately, or because the model itself is in error. Two common modeling errors are ignoring the distortion caused by the optics and ignoring the variation in interpupillary distance across different users.

A computational model for the geometry of a head-mounted display is presented, and the parameters of this model for the VPL EyePhone are calculated.

1 Introduction

1.1 The Problem: Computing the Correct Stereoscopic Images in Virtual Reality

As you move through the world, images of the objects that surround you fall onto your retinas. As you move past a fixed object, seeing it from various angles, the size and shape of the images on your retinas change, yet you effortlessly and unconsciously perceive the object to have a stable position, shape, and size. This innate perceptual ability, honed by your daily experience ever since infancy, is so fundamental and habitual that it seems almost absurd to talk about objects that could change their position or shape or size depending on how you moved your head.

Yet the current state of the art in virtual reality (VR) gives us simulated objects that change their position, size, and shape as the head moves. The location of these objects appears to change as the head moves around, and their size and shape appear to change depending on whether they are being viewed directly in front of the user's head or off to the side.

In virtual reality, a head-mounted display (HMD) and a head tracker are used to rapidly measure head position and create an image for each eye appro-

Presence, *Volume 1, Number 1, Winter 1992*

priate to its instantaneous viewpoint. The HMD user can then see simulated objects from different points of view as the head moves. However, it is difficult to correctly calculate the images to be painted onto the display screens of the HMD. The user's eyes and brain (and vestibular system) are very sensitive to inconsistencies.

The computational problem of calculating the correct stereoscopic images in VR—getting the perceived objects to have the right position, size, and shape—is the same problem that faces the designers of stereoscopic photography and telepresence systems. For these systems, orthostereoscopy occurs when the perceived size, shape, and relative position of objects in the three-dimensional scene being viewed match those of the physical objects in front of the camera. In virtual reality, the simulated scene has no physical counterpart, so orthostereoscopy must be defined in this case as constancy, as the head moves around, of the perceived size, shape, and relative positions of the simulated objects. To calculate orthostereoscopic images, the display code must precisely model the geometry of the HMD system on which the images will be viewed. This includes the relative positions of the display screens, the optics, and the eyes. The relationship between the screen and the virtual image of it must also be modeled.

This paper addresses only the static image generation problem. To simulate objects that are spatially stable, temporal problems must also be solved, but those problems are outside the scope of this paper.

1.2 Prior Work

Since Ivan Sutherland built the first HMD in 1968, several HMD systems have been built. The display code for each system defined an implicit model for the particular geometry of each HMD. The authors of the display code for each system structured the code as they judged appropriate, and it is difficult to know the precise details of their display code from what has been published. It appears that most HMD systems treated their optics simply as a magnifier, ignoring distortion introduced by the optics.

In Sutherland's HMD, tiny half-inch monochrome CRTs were the display devices, and the virtual images seen through the optics subtended an angle of 40° and appeared to be at a distance of 18 in. in front of the eyes (Sutherland, 1965, 1968). Half-silvered mirrors superimposed the computer graphics onto the user's direct view of the real world. Later versions of the HMD were stereoscopic. The stereoscopic HMD had both a mechanical adjustment for interpupillary distance (IPD) and a software adjustment for the virtual eye separation. This HMD system was moved to the University of Utah, and essentially the same system was used by several students there (Vickers, 1974; Clark, 1976).

In 1983, Mark Callahan at MIT built a see-through HMD similar to Sutherland's (Callahan, 1983). It used half-silvered mirrors mounted on eyeglass frames, 2-in. monochrome CRTs, and a bicycle helmet. An optical disk was used to rapidly display prerecorded images in response to head movements.

In 1985 at NASA Ames Research Center, Mike McGreevy and Jim Humphries built a non-see-through HMD from monochrome LCD pocket television displays, a motorcycle helmet, and the LEEP wide-angle stereoscopic optics (Howlett, 1983). This HMD was later improved by Scott Fisher, Warren Robinett, and others (Fisher, McGreevy, Humphries, & Robinett, 1986). The display code for this system treated the LEEP optics as a simple magnifier. The LEEP optics system has very large exit pupils and therefore no mechanical IPD adjustment.

At Wright-Patterson Air Force Base in the 1970s and 1980s, Tom Furness directed a program that developed prototype HMDs for use in military aircraft (Glines, 1986). The system was called Visually Coupled Airborne Systems Simulator (VCASS), and several prototype see-through HMDs with custom-designed optics were developed there (Buchroeder, Seeley, & Vukobradatovich, 1981; Kocian, 1986).

CAE Electronics of Quebec has developed a fiberoptic helmet-mounted display system (FOHMD), intended for flight simulators (CAE, 1986; Henderson, 1989). Four light valves drive the HMD through fiberoptic cables and pancake optics allow the user to see-through to the flight simulator's control panel. There is a mechanical adjustment for IPD. The binocular field of view

(FOV) is 135° horizontally by 64° vertically, and it also has a 25 × 19° high-resolution inset field.

Several prototype HMDs have been constructed here at the University of North Carolina at Chapel Hill (Holloway, 1987; Chung et al., 1989). In 1985, a see-through HMD was made from color LCD displays, half-silvered mirrors, magnifying lenses, and a pilot's instrument training hood. The FOV was approximately 25° horizontally. A later model, built at the Air Force Institute of Technology (AFIT) with UNC collaboration, was made from color LCDs, very strong reading glasses, and a bicycle helmet. Its FOV is about 55° horizontally and it is not a see-through HMD. We are currently designing a see-through HMD for medical imaging applications. It will incorporate custom-designed optics.

In 1989, Eric Howlett, the inventor of the LEEP optics, put together a commercial HMD, the LEEPvideo System I. It used monochrome LCD displays, the LEEP optics, and a head-mounting apparatus designed by Howlett. Howlett subsequently introduced improved models that use color LCDs and have a wider FOV (LEEP, 1990).

Later in 1989, VPL Research of Redwood City, California, began selling the EyePhone (VPL, 1989). It uses color LCD displays and the LEEP optics. It attaches to the head with a rubber diving mask and fabric straps. It is not see-through.

In 1989, Reflection Technologies of Waltham, Massachusetts, produced a product called Private Eye, a single eye monochrome HMD (Times, 1989). It uses a vibrating mirror and an LED linear array to produce a two-dimensional (2-D) image. The horizontal FOV is about 25°.

1.3 Remaining Problems

Generating correct stereoscopic images for an HMD is a difficult task. The display code for each HMD system embodies an implicit computational model of the geometry of the HMD, and there are many sources of error that must be compensated for. In current practice, most of these models are inadequate because they ignore certain sources of error. Also, since these models are embodied only in the display code of the HMD systems, they are difficult to comprehend, and are not accessible to most people. It is difficult to compare the computational models of different HMD systems.

The display software often ignores the system optics. But because the optics actually do affect the images seen by the eyes, some of the parameters in the display software are tweaked to get the convergence and FOV of the HMD to be roughly correct. This type of measurement of the parameters of the HMD system by subjective calibration by the users is inaccurate compared with calculating the model parameters from the specifications of the optics and display screens.

Another problem is that most current HMD systems ignore the variation in IPD across different users. In this case, wide and narrow-eyed users will have different size perceptions of the same simulated objects.

This paper presents an explicit computational model for generating orthostereoscopically correct images. Implementing display software that follows this model will produce stereoscopic images that are orthostereoscopic—simulated objects will be perceived as three-dimensional and will be undistorted and correctly sized.

We first survey the various sources of error that cause incorrect stereoscopic images to be generated. We then introduce a computational model of the geometry of an HMD that models the optics, the distance between the user's eyes, and the relative positions of the eyes, optics, and display screens. This model allows correct orthostereoscopic images to be calculated. Finally, we calculate the model parameters for the VPL EyePhone.

2 Sources of Error

There is a very precise correlation between the movements of one's head and the images of an object that are formed on the retinas from moment to moment. This correlation can be described by simple geometry: the object's images are projected onto the retinas, and the retinal images depend only on the object's shape and the relative position of the two eyes with respect to the object. An HMD system attempts to mimic this geometry, painting images onto display screens in front of the

eyes to fool the eyes and brain into perceiving three-dimensional (3-D) objects. If the wrong images are painted onto the screens, the user is not able to perceive the simulated object correctly. The object will either be distorted, or there will be no perception of a 3-D object at all.

The wrong images are painted onto the screens either because of errors and inaccuracies in the head tracking, or because of errors in the software that controls the image generation. While the tracking hardware can introduce significant error, it is the display software that is the subject of this paper.

We will introduce a computational model for a head-mounted display, and say that the display software implements this model. For the display software to generate the images required to give the HMD user the perception of undistorted objects, the software must take into account the physical geometry of each hardware component that affects the final image seen by the eyes. This geometry includes the display screens, the optics used to image the displays, and the placement of the displays with respect to the eyes. Before introducing the computational model, we discuss some common errors in the display code for HMDs.

2.1 Incorrect Convergence

Both eyes are necessary for stereoscopic vision. When the eyes are focused on a distant object, the lines of sight are roughly parallel, and when focused on a near object, the lines of sight converge. The nearer the object, the greater the convergence.

A stereoscopic HMD has, for each eye, a display screen and an optical system through which the screen is viewed. If the optical axes were parallel for the two optical systems and if the optical axis passed through the center pixel of each screen, then by illuminating those two center pixels, the user would see a stereoscopic image of a point of light at infinity. This would be more or less like looking at a single star in the night sky. However, many HMDs do not satisfy those two conditions: turning on the center pixels would either produce a percept of a not-so-distant point of light in front of the user, or else be too divergent to fuse at all into a stereoscopic percept.

In addition to the horizontal alignment problem related to convergence and divergence, there can also be a vertical misalignment between the two eyes. This is called dipvergence.

Creating stereoscopic images with the correct convergence, when the optical axes are not parallel or the centers of the displays are offset from the optical axes, requires corrective transformations in the computational model. Neither of these properties is a mistake in the design of an HMD, they just make the computational model a little more complicated. In fact, many current HMDs have nonparallel optical axes and off-center screens.

2.2 Accommodation Not Linked to Convergence

The eyes converge on a nearby object by rotating inward; the lenses of the eyes simultaneously accommodate to the distance of the object to bring it into focus. Thus convergence and accommodation are normally linked. However, in an HMD system, each eye sees the virtual image of a display screen. With respect to focus, the entire virtual image appears at a fixed distance from the eye. (In a physical scene, different parts of the scene will be in focus at different accommodation depths.) Hence, the HMD user must learn to decouple accommodation and convergence. This problem cannot be overcome with any currently used display device. Until a display device with variable focus is developed, this problem must be accepted as a limitation of HMDs.

2.3 Incorrect Field-of-View

The display for a single eye in an HMD has an FOV, which is the angle subtended by the virtual image of the display screen, as seen by the eye. There is a horizontal and a vertical FOV corresponding to the left-to-right and top-to-bottom angular sweep across the virtual image of the display. Let us call these angles the physical FOV. To be accurate in our definition of the FOV, we shall consider that a point in object space contributes to

the physical FOV if the chief ray defined as the ray passing through that point and the center of the exit pupil of the viewing system (here the pupil of the eye) is not obstructed. We are somewhat conservative therefore in our definition of the FOV since a point could be said to belong to the FOV as long as at least one ray reaches the image plane. This FOV is usually referred to in the optics literature as the total FOV.

An FOV is also specified in the display code. This computational FOV determines how far away the center of projection is from the screen rectangle in the perspective transformation that is used to project the 3-D virtual world onto the 2-D screen. There is both a horizontal and vertical FOV in this case, also. These are specified in many graphics systems by giving the aspect ratio, the ratio of horizontal to vertical FOV angles, and then giving the vertical angle. Another term that is sometimes used instead of "computational FOV" is "geometric FOV."

Unfortunately, the computational FOV angles may not match the physical FOV angles. As with convergence, the software designer may not know the optical specifications and may be forced to measure the FOV empirically. To get it right, the position of the center of projection with respect to the screen should be exactly at the entrance pupil of the eye.

The optics, the position of the display relative to the optics, and the FOV used to compute the images jointly determine a point in the exit pupil of the optics where the pupil of the eye is expected to be. If the pupil of the eye is not at this expected point in space (too near or too far, to the left or right, too high or too low), the lines of sight to a simulated object will not be correct.

2.4 Failure to Use Off-Center Projection When Required

If the display screen is perpendicular to the optical axis and off-center from the axis, then the eye is off-center with respect to the screen, and the computational center of projection should be off-center, too. This situation requires an off-center perspective projection, in which the left, right, top, and bottom edges of the screen rectangle are specified independently of one another, rather than having left–right and top–bottom be symmetrical as usual. An off-center perspective projection transformation can be set up with the standard computer graphics hardware using the standard 4×4 homogeneous matrix representation of the transformation. Many people even in the computer graphics profession are unfamiliar with the off-center perspective projection—it is never needed in normal computer graphics because, since the physical eye position is unknown, a convenient one directly in front of the screen may as well be assumed.

Another mistake is to simply rotate the whole scene left or right to produce images for the two eyes. In general, the computational model needs to include transformations that take into account the position and orientation of the screens with respect to the user's eyes. This is likely to be a combination of translation, rotation, and off-center perspective projection, and very unlikely to be just a pure rotation (Saunders, 1968; Hodges & McAllister, 1990).

2.5 Interpupillary Distance Ignored

Among male and female adults, there is a fairly wide variation in the distance between the eyes, called the interpupillary distance (IPD). The range is roughly 53 to 73 mm, with the average IPD being about 63 mm. Children have even smaller IPDs.

The variation in IPD imposes a requirement on the HMD optics and display hardware. Either the exit pupil of the optics must be large enough to be used by the widest-eyed and the narrowest-eyed people, or else there must be a mechanical adjustment such as is found on binoculars. Both of these approaches have been used. For example, the LEEP optics, used in many current HMDs, has a very large exit pupil. The CAE fiberoptic HMD used for flight simulation requires a mechanical adjustment for each new user.

The distance between the eyes is the baseline from which a person judges the distance to objects in the physical world. The convergence of the lines of sight from the eyes to a point of fixation can be measured as an angle. As Figure 1 shows, a narrow-eyed and a wide-eyed person will have different convergence angles when

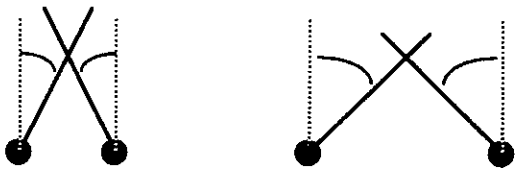

Figure 1. *Eyes with narrow versus wide IPDs looking at an object from the same distance.*

looking at an object at the same distance, say, half a meter away.

These two people have different convergence angles yet both perceive the object to be half a meter away. Each person is calibrated to his or her own IPD.

With the mechanical IPD adjustment, the images get piped into the user's eyes and his or her physical IPD has no effect on the images seen. With no mechanical adjustment but a large exit pupil, if the virtual images are at optical infinity, then a lateral change in eye position has no effect on the angles at which an object appears—in other words, in this case, too, the user gets the same one-size-fits-all images regardless of the physical distance between the eyes. If the virtual images are not at optical infinity, the situation is more complicated, and people of varying IPD are still not going to see images matched to their own IPDs.

The solution to this problem is to measure the IPD for each user, and have the IPD as a user-specific constant in the computational model. If this is done correctly, then each user can see a simulated object half a meter away with a convergence matched to his or her own IPD.

2.6 Optical Distortion Ignored

The display screens in an HMD are too close to the eyes to focus on directly, so an optical system is interposed between the eye and the screen. The main purpose of the optics is to provide an image to the user at a comfortable distance of accommodation and with as large a magnification as possible without altering the image. The eye, looking into the optics, sees a virtual image of the display screen. The virtual image is distant enough from the eye to focus on easily, and large enough to cover a large swath of the user's FOV. But the optics also distort the image nonlinearly, causing lines that were straight on the display screen to appear as curved in the virtual image.

Optical aberrations are defects of the image. They may be described in terms of the amount by which a geometrically traced ray misses a specified location in the image plane formed by the optical system. The displacement of the ray is referred to as the transverse ray aberration. Most often, the specified location for a ray in the image plane is that inferred from first-order laws of image formation (Hecht, 1974; Longhurst, 1973). Rays that do propagate not only near the optical axis but also at shallow angles with respect to the optical axis are known as paraxial rays. Under the paraxial approximation, the formation of images is referred to as first-order, paraxial, or Gaussian optics. The most common aberrations are spherical aberration (SA), coma, astigmatism (AST), field curvature (FC), distortion, and chromatic aberrations. It should be noted that, while SA, coma, AST, FC, and chromatic aberrations all affect the sharpness of the image points being formed, distortion distinguishes itself from the others since it causes the image points to be displaced transversally in a nonlinear fashion across the FOV but does not alter the sharpness of the image.

Transversal aberrations can be expressed mathematically as a polynomial expansion of third-order and higher order terms in both the image height and the height of strike of the ray on a reference sphere centered on the ideal image point and passing through the exit pupil of the optical system. The ideal image point is often chosen to be the paraxial image point. The sum of the exponents of the aperture and field terms indicates the order of the aberration represented by that term. Depending on how open the optical system is and how large the angles of incidence of the rays on the different surfaces of the optical elements are, a system is best described by a third-order or a higher order approximation. The complexity of the optics used is usually such that the sharpness of the images formed through the optical system is good enough for the display resolution available on the market today. The distortion of the images, however, is often disturbing if it has not been corrected for optically. The first-order polynomial is linear and describes an ideal magnifier with no distortion, and

since there are no even terms appearing in the expansion, the third-order polynomial is the simplest model of distortion.

Nonlinear field distortion causes straight lines on the screen to appear curved. This can be corrected for in the graphics system by predistorting the images written onto the screens. A straight line in the virtual image would be created by writing a curved line onto the screen, such that its curvature exactly balanced out the optical distortion. This would require that the inverse of the screen-to-virtual-image distortion function be stored in the graphics system, and that each pixel be remapped to a new location on the screen. This is computationally expensive.

Most current HMD systems just ignore the optical distortion, because they may not have access to the distortion function for the optics, and because of the performance penalty even if they did do the correction. This is not an unreasonable choice in the early stages of development of an HMD, because the system is usable even with the optical distortion.

However, one side-effect of nonlinear field distortion is that there is no single correct value for the FOV. Nonlinear distortion causes the magnification to vary across the FOV. If the computational FOV is set to match the physical FOV, then objects in the center of the field will appear to be the wrong size. But if the computational FOV is set to make small central objects appear to be the right size, the objects in the peripheral field will be positioned wrong. The only way to avoid this unpleasant choice is to predistort the image to correct the optical distortion.

If the optics are designed specially for an HMD, then specific types of aberrations can be minimized. But if optics designed for another purpose are used, then you take what you can get. The LEEP optics, used in many current HMD systems, were designed for stereoscopic photography, and purposely incorporate substantial field distortion and chromatic aberrations.

Our experience has been that although the LEEP optics in the VPL EyePhone introduce distortion, we have been able to use it very effectively without correcting the image to compensate for the distortion. However, our experience with shading and radiosity models has also shown us that we are not able to appreciate what we are missing until we have seen it done right.

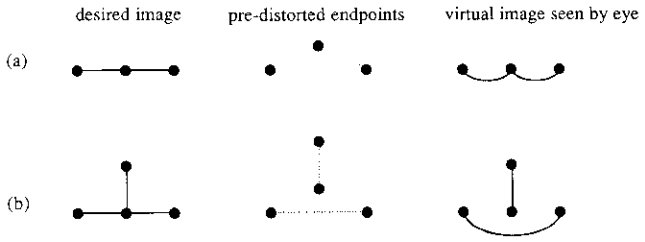

Figure 2. *Problems with predistorting only vertices.*

2.7 Transforming Only Polygon Vertices in the Presence of Nonlinearities

Computer graphics has traditionally gained much efficiency by representing graphics primitives such as lines and polygons as collections of points, running only the points through the transformation pipeline, and then drawing the lines or polygons between the transformed points. This works only if all the transformations in the pipeline are linear. It is tempting to run only the polygon vertices through the predistortion function and let the very efficient hardware fill in the polygons. But then only the vertices would be in the right place—the polygon edges would still be curved by the optical distortion. Figure 2a shows a simple case of this. Edges of polygons that cross a large fraction of the screen would be most noticeably curved.

Another problem with this approach is that continuity would be lost. A vertex that touched an edge before predistortion and scan-conversion would not be guaranteed to do so in the final virtual image. Gaps and holes would open up everywhere. Figure 2b shows this.

3 Optics Model for a Head-Mounted Display

The purpose of the optics model is to specify the computation necessary to create orthostereoscopically correct images for an HMD and indicate the parameters of that system that need to be measured and incorporated into the model.

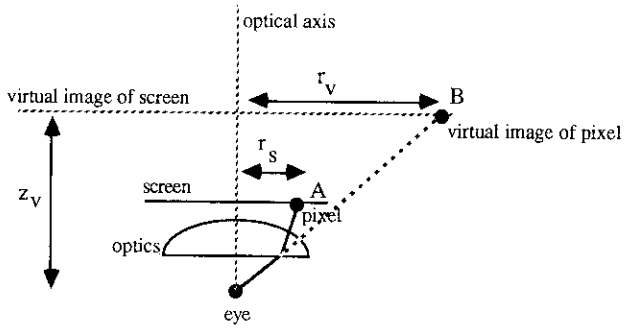

Figure 3. *Single-eye optics model.*

3.1 Single-Eye Optics Model

To achieve orthostereoscopy, the nonlinear optical distortion must be corrected by remapping all the pixels on the screen with a predistortion function. Linear graphics primitives such as lines and polygons are written into a virtual screen image buffer, and then all the pixels are shifted according to the predistortion function and written to the screen image buffer for display. The predistortion function is the inverse of the field distortion function for the optics, so that the virtual image seen by the eye matches the image in the virtual screen buffer. A straight line in the virtual image buffer is predistorted into a curved line on the display screen, which is distorted by the optics into a line that is seen as straight. Figure 3 shows the optics model for a single eye.

The mathematical representation of the optical distortion will depend on the nature of the optical system. For optics such as LEEP, a third-order polynomial approximation is adequate. We want to relate the radial position r_s of a pixel on the screen to the radial position r_v of the virtual image of that pixel. These two quantities are measured in millimeters with respect to the optical axis. Dividing r_s and r_v by the object field width w_s and the image field width w_v, respectively, we get the normalized position of the pixel on the screen

$$r_{sn} = r_s/w_s$$

and the normalized position of the virtual image of the pixel

$$r_{vn} = r_v/w_v$$

(The paraxial magnification of the system is thus $m_{sv} = w_v/w_s$.) The distortion is modeled with a third-order polynomial approximation

$$r_{vn} = r_{sn} + k_{vs} r_{sn}^3$$

in which the coefficient k_{vs} describes the amount of distortion present. This can be rearranged algebraically to

$$r_{vn} = (1 + k_{vs} r_{sn}^2)\, r_{sn}$$

and then expanded to rectangular coordinates using

$$r_{sn}^2 = x_{sn}^2 + y_{sn}^2$$

$$r_{vn}^2 = x_{vn}^2 + y_{vn}^2$$

to give the position of the virtual image of the pixel

$$(x_{vn}, y_{vn}) = \{[1 + k_{vs}(x_{sn}^2 + y_{sn}^2)]\, x_{sn},\ [1 + k_{vs}(x_{sn}^2 + y_{sn}^2)]\, y_{sn}\}$$

from the position (x_{sn}, y_{sn}) of the pixel on the screen. These positions are measured in the screen plane and virtual image plane, respectively, with respect to the optical axis.

The distance from the eye to the virtual image plane z_v is a constant. By combining the above equations, we have a function that gives the three-dimensional position of the virtual image of a pixel on the screen (x_v, y_v, z_v) in terms of its screen coordinates (x_s, y_s) and some constants. We will ignore z_v from here on since it is constant.

The expressions for x_{vn} and y_{vn} can be thought of as single-valued functions of two variables. If we name the distortion function D, then

$$(x_{vn}, y_{vn}) = D(x_{sn}, y_{sn})$$

and what we need to predistort the image on the screen is the inverse D^{-1}. There are various ways this inverse function could be represented in the computer. An exact closed-form expression is not feasible, but a polynomial approximation of the inverse is possible.

$$r_{sn} = r_{vn} + k_{sv} r_{vn}^3$$

$$(x_{sn}, y_{sn}) = D^{-1}(x_{vn}, y_{vn})$$

$$(x_{sn}, y_{sn}) = \{[1 + k_{sv}(x_{vn}^2 + y_{vn}^2)]\, x_{vn},\ [1 + k_{sv}(x_{vn}^2 + y_{vn}^2)]\, y_{vn}\}$$

Note that these two functions D and D^{-1} are each third-

order polynomial approximations and are not exact inverses of one another. The coefficients k_{sv} and k_{vs} will be opposite in sign.

Another possibility for representing D^{-1} on the computer is a two-dimensional table lookup for each of the output variables x_{sn} and y_{sn}. Using this approach, limits on table size would probably make it necessary to interpolate between table entries.

Distortion causes nonuniform magnification across the field, which causes the brightness also to vary across the field. This could be compensated for on a pixel-by-pixel basis with a brightness correction function $B(x_s, y_s)$, but limitations of space prevent us from going into this here.

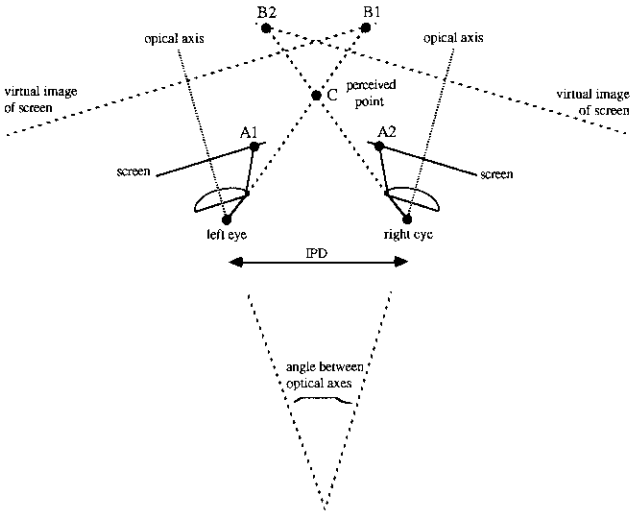

Figure 4. *Stereoscopic optics model.*

3.2 Stereoscopic Optics Model

Figure 4 shows the stereoscopic optics model. One pixel is illuminated on each screen (points A1 and A2) and a line of sight is drawn from each eye to the virtual image of its corresponding pixel (points B1 and B2). These two lines of sight intersect at the three-dimensional point perceived by the user (point C). The IPD is the baseline from which the user makes distance judgments based on the convergence angles to perceived points.

If the specifications for the optics are known, and the relative positions of the display screens, the optics, and the eyes are also known, then it is possible to accurately calculate several important parameters needed in the computational model. The horizontal and vertical FOVs can be calculated by starting from the known positions of the left, right, top, and bottom edges of the screen with respect to the optics and tracing the rays through the optics system back to the eye.

The screen centers may be offset from the optical axes by a certain distance. The two optical axes may be rotated with respect to one another. The screen center offsets and axis divergence angles can be used to set the convergence properly with no need for subjective calibrations. From the known position and orientation of the virtual image of the screen relative to the eye, the eye-to-virtual-image transformation can be calculated to be the correct mix of translation, rotation, and off-center perspective projection. Translation is needed because the eyes are in different positions in space, rotation is needed if the optical axes are not parallel, and the projection is off-center if the screens are off-center from the optical axes.

Table 1 shows the sequence of starting from the specifications of the optics and displays, measuring certain parameters of the HMD, and then calculating other parameters needed by the computational model. The left and right halves of the optical system are assumed to be symmetrical. The calculations are done for the right side, so the left edge of the screen is on the inside beside the nose and the right edge is on the outside.

Except for the IPD, which varies among users, every other parameter necessary to specify the head-to-eye, eye-to-virtual-image, and virtual-image-to-screen transformations for the left and right eyes can be derived from the specifications of the optics and the relative positions of the eyes, optics, and screens. Calculating these parameters is much more accurate than relying on subjective calibration procedures in which the parameters are adjusted until the image looks right. However, the subjective measurements provide a nice check against mistakes in the model, the measurements, or the calculations.

We have defined a computational model for the graphics computation without reference to a particular HMD system. Having done that, we now turn to calcu-

Table 1. *Calculating Parameters in the Optics Model*

Parameter	Symbol	Where it comes from
Screen resolution	$\text{Res}_H \times \text{Res}_V$	From display spec
Angle between optical axes	ϕ_{axes}	From optics spec or measure
Distance between optical axes (at front surface of optics)	d_{axes}	From optics spec or measure
Eye relief	d_{er}	Measure
Maximum field of view	ϕ_{max}	Calculate from d_{er} and optics spec
Object plane distance	d_{ob}	Measure
Distance from eye to virtual image plane	z_v	Calculate from d_{ob} and optics spec
Transversal magnification	m_{vs}	Calculate from d_{ob} and optics spec
Coefficient of optical distortion	k_{vs}	From optics spec
Maximum object field radius	w_s	From optics spec
Maximum virtual image field radius	w_v	$w_v = m_{vs} w_s$
Interpupillary distance of user	IPD	Measure user
Screen center offset from optical axis	(Cx_s, Cy_s)	Measure
Position of left screen edge	(Lx_s, Ly_s)	Measure
Position of right screen edge	(Rx_s, Ry_s)	Measure
Position of top screen edge	(Tx_s, Ty_s)	Measure
Position of bottom screen edge	(Bx_s, By_s)	Measure
Object height (of point on screen)	r_s	$r_s = (x_s + y_s)^{1/2}$
Normalized object height	r_{sn}	$r_{sn} = r_s / w_s$
Normalized image height (third-order approximation of D)	r_{vn}	$r_{vn} = r_{sn} + k_{vs} r_{sn}^3$
Image height (of point in virtual image)	r_v	$r_v = r_{vn} w_v$
Angular position of point in virtual image	ϕ	$\phi = \tan^{-1}(r_v / z_v)$
Angular position of left edge (inner edge of screen)	ϕ_L	From formula for ϕ
Angular position of right edge (outer edge of screen)	ϕ_R	From formula for ϕ
Angular position of top edge	ϕ_T	From formula for ϕ
Angular position of bottom edge	ϕ_B	From formula for ϕ
Single eye vertical field of view	FOV_v	$\text{FOV}_v = \phi_T + \phi_B$
Single eye horizontal field of view	FOV_h	$\text{FOV}_h = \phi_L + \phi_R$
Overlapped field of view	FOV_{ov}	$\text{FOV}_{ov} = 2\,\phi_L - \phi_{axes}$
Binocular field of view	FOV_{bin}	$\text{FOV}_{bin} = 2\,\phi_R + \phi_{axes}$
Translation part of viewing transformation	M_{trans}	$(\pm\text{IPD}/2, 0, 0)$
Rotation part of viewing transformation	M_{rot}	$(\pm\phi_{axes}/2)$ around Y-axis
Perspective projection	$M_{perspec}$	Use FOV_v, FOV_h, and offset $[\pm Cx_s/(Rx_s - Lx_s), Cy_s/(Ty_s - By_s)]$

lating the model parameters for a specific HMD, the VPL EyePhone.

4 Calculating the Model Parameters for the VPL Eyephone

4.1 Description of EyePhone Components

We have used the optical specifications of the LEEP optics and size and positioning of the LCD screens inside the EyePhone to calculate the model parameters for the VPL EyePhone, Model 1 (VPL, 1989). Model 2 of the EyePhone has identical optics, LCD displays, and positioning of the parts.

The LEEP optics (Howlett, 1983) are wide-angle, stereoscopic viewing lenses. It consists of a three-lens system in front of each eye, encased in a molded plastic mount, with a cut-out for the nose. It was designed for a single transparency in the object plane, upon which are two side-by-side stereoscopic photographs of a scene, each one a square approximately 64 mm on a side. For an eye relief distance of 29.4 mm, the FOVs for each eye are approximately +45 to −45° horizontally, and +45 to −45° vertically. The distance from the center of the rearmost lens surface to the object plane is approximately 16 mm. The optical axes for the two eyes are parallel and are 64 mm apart. The two optical systems are bilaterally symmetrical with respect to each other, and each optical system is radially symmetrical around the optical axis, except for the cut-outs for the user's nose in the front lenses.

The construction of the EyePhone is rigid enough to keep the eyes, optics, and display screens in a fixed relationship to one another, so it is possible to make accurate measurements of these relative positions.

The EyePhone uses the LEEP optics with two color LCD display screens positioned in the LEEP object plane. The two displays are positioned symmetrically with respect to the left and right eye optical axes. Figure 5 is a diagram of the positions of the LCD screens in the object plane.

Six important points are labeled in Figure 5. Only one side need be analyzed since the LCDs and optics are symmetrical.

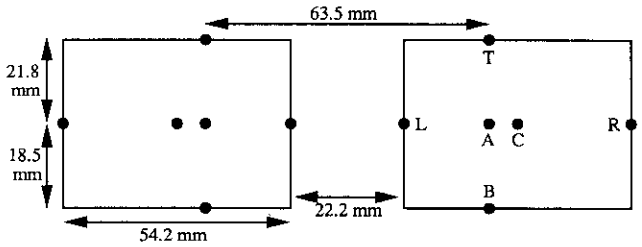

Figure 5. *Position of the EyePhone's LCD displays in the LEEP object plane.*

4.2 Calculation of EyePhone Field-of-View

We use two different methods to compute the FOVs for the EyePhone. First, we use the optics model with the parameters specific to the EyePhone to calculate the angles at which the points in Figure 5 are seen. Then, for comparison and validation, we compute these same angles by tracing rays through the LEEP optics. We do the ray tracing with Code V, a commercial optical analysis software package (ORA, 1991). To do the ray tracing, we use the detailed optical specifications of the LEEP optics, including the position, curvature, and index of refraction for each lens surface in the optical path.

First, we use the optics model. The dimensions given in Figure 5 are sufficient to determine the coordinates of the points A, L, R, T, and B in the LEEP object plane (with respect to the optical axis A). From these coordinates, the value of r_s for each point can be calculated. We feed the known positions of the edges of the LCD screens into the optics model to predict the positions of the screen edges in the virtual image, and from these positions calculate the FOVs for the EyePhone. We shall see that the chief rays corresponding to some of the object points on the LCDs are obstructed for $d_{er} = 29.4$ mm. In this case the FOV covers only part of the LCDs. The unseen part of the LCD is said to be vignetted. The model parameters for the LEEP optics as used in the EyePhone are

Table 2. *EyePhone FOV Calculation, Assuming d_{er} = 29.4 mm*

Point	x_s (mm)	y_s (mm)	r_s (mm)	r_{sn}	r_{vn} third order	r_v (mm)	Angle ϕ from model (degrees)	Angle ϕ from ray tracing (degrees)
A Optical axis	0	0	0	0	0	0	0	0
R Right edge of screen	33.3	0	28.1 (33.3 vignetted)	1.000	1.32	358.5	42.0	45.0
L Left edge of screen	−20.9	0	20.9	0.744	0.876	237.9	30.9	30.3
T Top edge of screen	0	21.8	21.8	0.776	0.926	251.5	32.3	31.8
B Bottom edge of screen	0	−18.5	18.5	0.658	0.749	203.4	27.1	26.6
C Center of screen	6.2	1.65						

$$d_{er} = 29.4 \text{ mm}$$

$$w_s = 28.1 \text{ mm}$$

$$w_v = 271.5 \text{ mm}$$

$$z_v = 398.2 \text{ mm}$$

$$k_{vs} = 0.32$$

and the equations from the optics model

$$r_{sn} = r_s / w_s$$

$$r_{vn} = r_{sn} + k_{vs} r_{sn}^3$$

$$r_v = r_{vn} w_v$$

$$\phi = \tan^{-1}(r_v / z_v)$$

can be used to calculate the angle ϕ for each point, as shown in Table 2.

The second method of computing the angles is ray tracing. Figure 6 shows a horizontal cross section of the right-eye LEEP optical system, with the rays from the points L, A, and R traced back to the eye. The ray tracing was done for each of the points A, L, R, T, and B in Figure 5.

Table 2 shows the angular positions of the virtual images of the edges of the LCD screen as predicted by the optics model, and as predicted by ray tracing. The comparison of the angles calculated by the two methods shows that the third-order approximation used in the optics model is adequate for the LEEP optics.

From these calculations, we can see that, for a single eye, the horizontal FOV is 75.3° (45.0 + 30.3) and the vertical FOV is 58.4° (31.8 + 26.6). These are the physical FOVs for the EyePhone—the physical angles at which the virtual images of the edges of the LCD screen are seen by the eye. To make the graphics calculation match the physical FOV, these angles must be incorporated into the calculation. Here at UNC, we are using the Pixel-Planes graphics engine (Eyles, Austin, Fuchs, Greer, & Poulton, 1988) to generate images for the EyePhone. Like many graphics systems, the Pixel-Planes graphics software accepts an angle for the vertical FOV, here 58.4°, and an aspect ratio, here 1.289.

The graphics calculation must also take into account the fact that in the EyePhone, the center of the LCD screen (point C) is off the optical axis (point A). How this off-center perspective projection is specified to the graphics software varies somewhat among graphics systems. For the Pixel-Planes graphics software, off-center projection is specified as a horizontal and vertical offset in pixels from the screen center. For the horizontal offset, 512 pixels across a 54.2 mm screen gives 0.106 mm/pixel, so the 6.2 mm horizontal offset is 58.5 pixels.

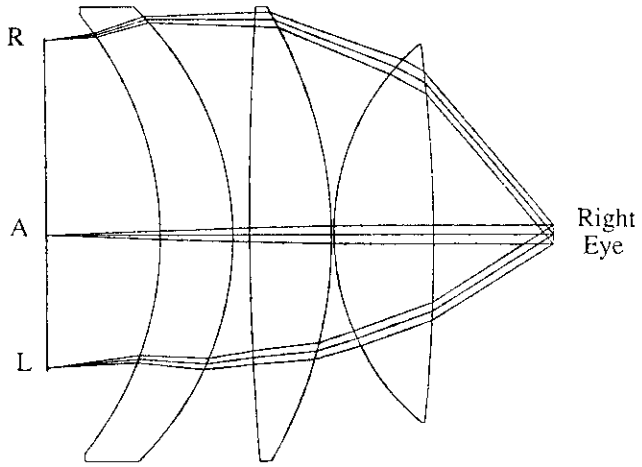

Figure 6. *Tracing rays from the EyePhone's LCD screen through the LEEP optics.*

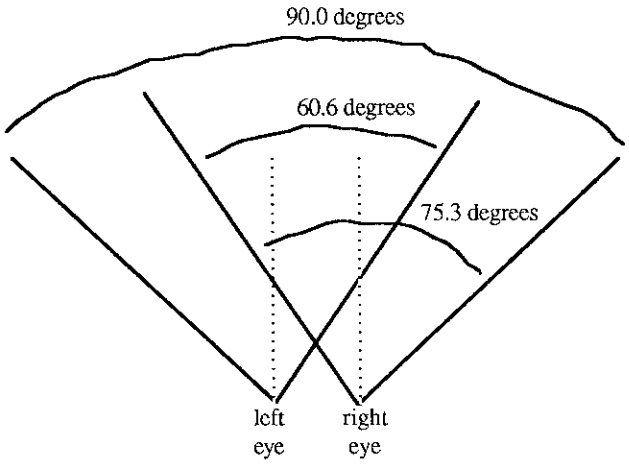

Figure 7. *Binocular, overlapped, and single-eye FOVs for the EyePhone.*

For the vertical offset, 512 pixels across a 40.3 mm screen height gives 0.079 mm/pixel, so the 1.65 mm vertical offset is 21.0 pixels.

These calculated values for FOV and screen-center offset together with the IPD of the user is enough to specify precisely the perspective projection for the two eyes. Using these projections, the convergence and FOV are guaranteed to be correct, with no need for adjustment or calibration by the user. Our experience with the EyePhone is that using the calculated parameters gives very solid stereo perception for all users who can see in stereo, with no need for tweaking these parameters. Getting this all to work depends on getting several subordinate things right—the specifications for the optics must be correct, the analysis of the optics must be correct, the measurements of the LCD positions and dimensions must be correct, and the graphics software for setting up the projections must be correct.

Before these calculations were done, the FOV and off-center parameters had been tweaked, through a long process of trial and error, trying to get the image to look right. The most effective test was to try to get the image of a 5.75-cm red sphere to be the right size and stay on top of a physical 5.75-cm red 3-ball (from the game of pool) with a Polhemus sensor inside as the 3-ball was moved around. This 3-ball, which has two pushbuttons on it, is one of our manual input devices. The FOVs for a single eye which were finally arrived at through these subjective tests were a horizontal FOV of 80° and a vertical FOV of 60°. Since the EyePhone does not allow the user to see through to the real world, the subjective tests had to be done by repeatedly raising the EyePhone to see how the position of the physical 3-ball compared with the remembered position of the simulated 3-ball. We estimated the accuracy to be about 2°. For a see-through HMD, superimposing a 3-D stereoscopic image onto a physical object of the same known shape is an extremely accurate test, because the eye can simultaneously compare the two scenes and detect tiny discrepancies. The acuity of the human eye is considered to be roughly one minute of arc.

Figure 7 shows that the horizontal FOVs for the two eyes in the EyePhone partially overlap. The region of overlap (60.6°) is wide enough for strong stereoscopic perception, and the binocular FOV (90.0°) is wide enough to provide a feeling of immersion within the scene. We believe, however, that a wider FOV will make the feeling of immersion stronger.

4.3 Correction of EyePhone Optical Distortion

To correct the image on the screen from optical distortion, the screen image must be predistorted as specified by the function D^{-1}. This would radially shift all the pixels in the image by some amount. This has not

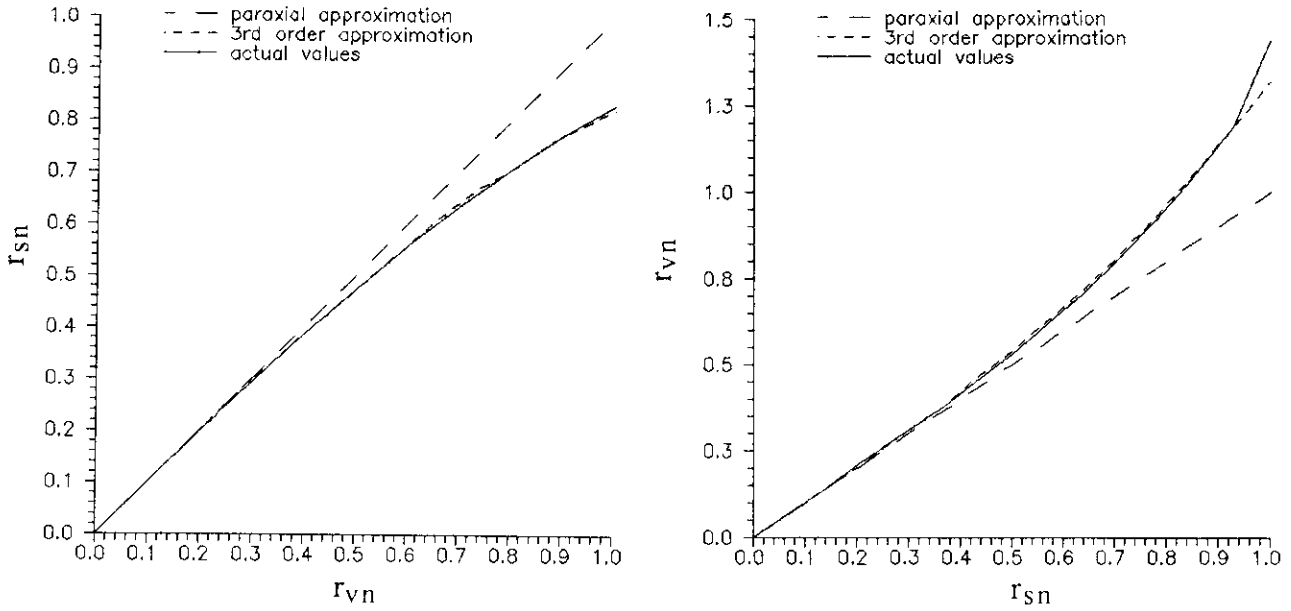

Figure 8. *Graph of r_{vn} vs. r_{sn} for D ($k_{sv} = 0.32$) and ray tracing for the LEEP optics; graph of r_{sn} vs. r_{vn} for D^{-1} ($k_{sv} = -0.18$) and ray tracing.*

yet been implemented on the UNC HMD system, but we plan to use a pair of two-dimensional tables $T_X[x_v, y_v]$ and $T_Y[x_v, y_v]$ in this implementation.

We have a 512 × 512 pixel screen to cover, but a table size of 512 × 512 is impractical. We expect to use a reduced table size, such as 64 × 64, and interpolate bilinearly between table entries. The table values will be computed off-line using the formula for D^{-1}.

The optics projects object points of height r_s to image points of height r_v. This mapping is monotonic and so its inverse exists. We approximate the mapping with a third-degree polynomial D and approximate its inverse with another third-degree polynomial D^{-1}

$$r_{sn} = r_{vn} + k_{sv} r_{vn}^3$$

in which, for the LEEP optics, k_{sv} is -0.18.

Figure 8 shows the graph of the normalized virtual image position r_{vn} versus the normalized screen position r_{sn} for the LEEP optics. It shows that the third-degree polynomial approximation of the distortion function D is quite close to the more accurate graph of D calculated by ray tracing through the optics. The second graph compares the third-degree polynomial approximation of the inverse D^{-1} with the values calculated by ray tracing.

Figure 9 shows a grid which has been predistorted by the function D^{-1}. When viewed through the LEEP optics, the lines of the grid appear straight.

Although the predistortion function D^{-1}, also known as the virtual-image-to-screen transformation, has not yet been implemented on the UNC HMD system, we have looked through the LEEP optics at the predistorted grid printed on paper to verify that the grid lines do appear straight. The predistorted grid in Figure 9 is full sized (79 mm square) and may be viewed through the LEEP optics. The lines will appear straight when the grid object is in close contact with the lens. Note that in the EyePhone, the LCD screen is separated by a gap of a few millimeters from the LEEP lens.

The straightforward method to correct computationally for the optical distortion would be to compute the image normally and then in a second step warp this 2-D image using the predistortion function D^{-1}. This would require moving every pixel in the image to a new location and is thus quite expensive computationally. Since a straight line such as the edge of a polygon is mapped by D^{-1} to a gently curved line, and our graphics engine Pixel-Planes 5 has the ability to evaluate quadratics in parallel, we have considered the idea of approximating

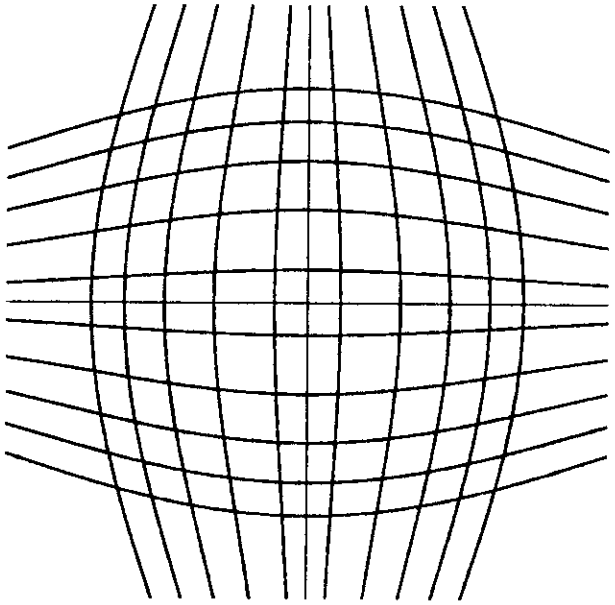

Figure 9. *Grid predistorted for the LEEP optics.*

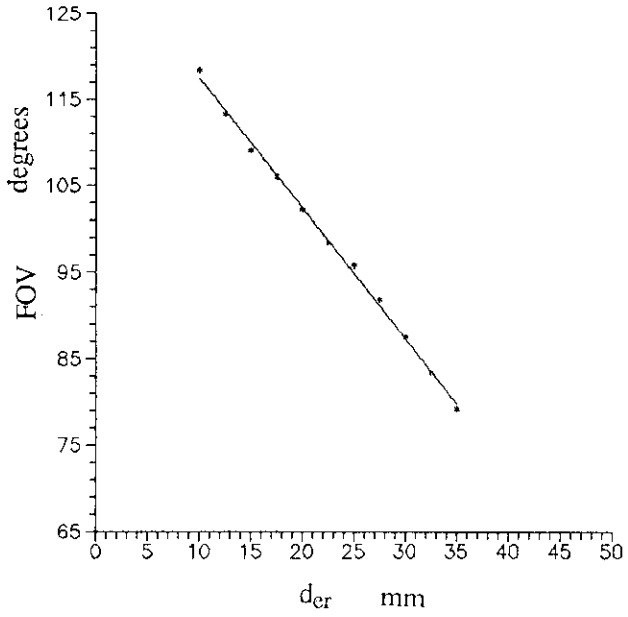

Figure 10. *Graph of FOV vs.* d_{er}.

these curved lines with quadratic polynomials, but we have not yet evaluated this idea in detail.

Another approach that has been used (Pang & Durlach, 1990) is to implement the distortion correction optically by displaying the undistorted image on a monitor and aiming a video camera with a LEEP inverse lens at the monitor.

4.4 Other EyePhone and LEEP Parameters

The graph of Figure 10 shows how the LEEP optics FOV varies with the eye relief distance d_{er}, the distance between the eye and the nearest lens surface. With the nominal eye relief of 29.4 mm, the FOV for the LEEP optics is +45 to −45°. (The EyePhone's FOV is less than this because the EyePhone's LCD screen does not fill the LEEP object field.) If the eye was able to get closer, the FOV would increase, but an eye relief of 25 to 30 mm is necessary to allow people with spectacles to use the system.

When moving the eye closer to the lens, two factors contribute to an increase of the FOV: first, any point on the virtual image is seen over a larger angle, and second, more of the LCD screens can be perceived. Especially for the EyePhone, moving the eye closer to the lens does increase the FOV because some of the LCD display screen is vignetted for a pupil distance of 29.4 mm.

The virtual image of the screen is formed at some distance z_v from the eye, and the eye must accommodate to this distance. The graph of Figure 11 shows that z_v is very sensitive to changes in the distance d_{ob} of the object plane to the LEEP optics. The EyePhone positions the LCD screen at d_{ob} = 16.4 mm from the nearest lens surface (measured along the optical axis). This value of d_{ob} results in an image distance of z_v = 398.2 mm and a magnification of m_{vs} = 9.66. As the object approaches the object focal point of the lens (d_{ob} = 20.7 mm) the image distance goes to infinity. However, such a positioning seems undesirable because of the conflict between convergence and accommodation.

The discussion of the resolution of the LCD display screens in the EyePhone is complicated by two competing ways of specifying resolution in a color display—by color pixels or by the RGB component cells of the pixels. The LCD used in the EyePhone has a monochrome array of 360 × 240 individually controllable light-producing cells, with red, green, and blue filters overlaid to divide the cells into three equal-sized groups. A triad of one red, one green, and one blue cell makes up a color

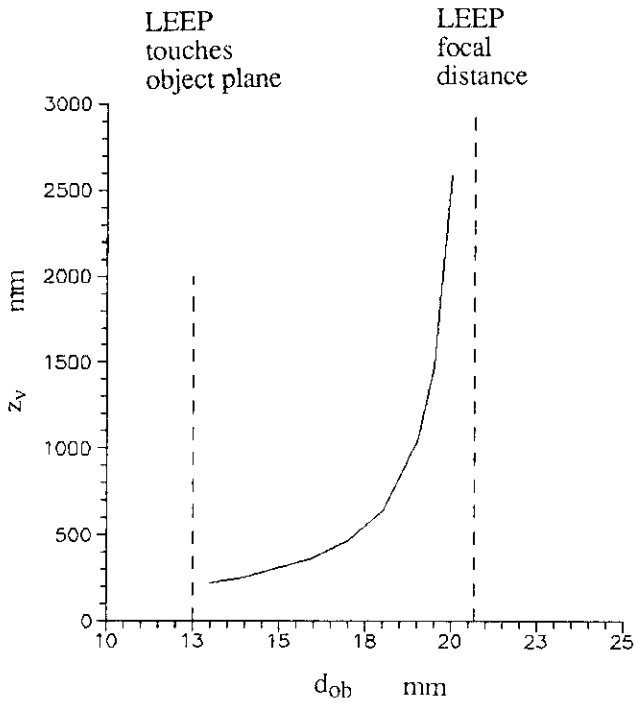

Figure 11. *Graph of* z_v *vs.* d_{ob}.

pixel. There are 86,400 cells (360 × 240), and therefore 28,800 color pixels. The resolution of the EyePhone in terms of color pixels is thus approximately 207.8 × 138.6.

Table 3 lists all the parameters for the EyePhone with respect to the model developed in this paper. All the angles (ϕ and FOV parameters) were calculated from the other parameters as described above.

The LEEP optics are used in other HMDs besides the EyePhone, and a new model of EyePhone with a different LCD display is also being developed. Some of the parameters in Table 3 depend only on the LEEP optics (ϕ_{axes}, d_{axes}). Other parameters with specific values in the table describe the specific configuration in the EyePhone (Models 1 and 2) of the LCD screen's size and position in front of the LEEP optics.

The rubber diving mask of the EyePhone holds the face and eyes in a fairly constant position with respect to the LEEP optics, although some individuals have deeper set eyes than others. Decreasing d_{er} by moving the eyes closer to the LEEP lens would cause more of the object field to be seen and the value of w_s that was chosen to be the highest value of the field would then have to be increased accordingly. If the same nominal eye relief used in this paper (29.4 mm) is assumed, then the distortion model can be applied to an HMD using the LEEP optics and a different screen using the coefficients calculated ($k_{vs} = 0.32, k_{sv} = -0.18$).

To determine the model parameters for other HMDs that use the LEEP optics, the distance d_{ob} from the LEEP optics to the display screen must first be known. This will determine the distance to the virtual image z_v and the magnification m_{sv}. An eye relief distance d_{er} must also be measured or assumed. This will determine the radius of the object and image fields w_s and w_v. The object and image fields describe what could be seen if the object field was completely filled, regardless of how completely the display screen does fill the object field. To compute the FOVs for the virtual image of the display screen, the positions of the edges of the display screen (points L, R, T, and B) must be measured in the object plane with the optical axis as the origin. Cranking these measurements through the model will give the angular positions of the edge points (ϕ_R, ϕ_L, ϕ_T, ϕ_B), and thus the horizontal and vertical FOVs (FOV_h, FOV_v) for a single eye. To find the binocular field of view FOV_{bin} for both eyes, the angles between the optical axes ϕ_{axes} must be taken into account. The position of the display screen's center with respect to the optical axis must also be measured to properly set up the perspective projection.

Summary and Conclusions

The optics model presented in this paper, if implemented correctly, will generate undistorted orthostereoscopic images for the user's two eyes.

To calculate the display parameters needed by the model for the particular HMD being used, it is necessary to know or measure the specifications of the optics, and the relative positions of the eyes, optics, display screens, and head position sensor. The construction of the HMD

Table 3. *Parameters for the VPL EyePhone, Models 1 and 2*

Parameter	Symbol	Value for EyePhone
Maximum object field radius	w_s	28.1 mm
Maximum virtual image field radius	w_v	271.55 mm
Transversal magnification	m_{sv}	9.66
Eye relief	d_{er}	29.4 mm (nominal)
Object plane distance (LCD screen to LEEP lens surface)	d_{ob}	16.4 mm
Distance from eye to virtual image plane	z_v	398.2 mm
Coefficient of optical distortion for D	k_{vs}	0.32
Coefficient of optical distortion for D^{-1}	k_{sv}	−0.18
Angle between optical axes	ϕ_{axes}	0°
Distance between optical axes (at front surface of optics)	d_{axes}	64 mm
Screen center offset from optical axis	(Cx_s, Cy_s)	(6.4 mm, 1.6 mm)
Screen resolution	$Res_H \times Res_V$	208 × 139 pixels (color triads)
Interpupillary distance of user	IPD	Varies across users
Angular position of virtual image of right edge of LCD	ϕ_R	45.0°
Angular position of virtual image of left edge of LCD	ϕ_L	30.3°
Angular position of virtual image of top edge of LCD	ϕ_T	31.8°
Angular position of virtual image of bottom edge of LCD	ϕ_B	26.6°
Single eye vertical field-of-view	FOV_v	58.4°
Single eye horizontal field-of-view	FOV_h	75.3°
Overlapped field-of-view	FOV_{ov}	60.6°
Binocular field-of-view	FOV_{bin}	90.0°

must be rigid enough that these values will not vary from day to day. If these parameters for the HMD are known, then several important derived parameters can be calculated—the FOVs, the screen-center offset for the perspective projection, the angle between the optical axes, and the coefficients for the optical field distortion function. Calculating the values of these parameters is much more accurate than attempting to measure them subjectively with users.

Acknowledgments

This paper was originally published in *Stereoscopic Displays and Applications II*, John O. Merritt, Scott S. Fisher, Editors, Proc. SPIE 1457, 140–160 (1991). Reprinted with permission of The International Society for Optical Engineering (SPIE).

We would like to thank many people for their contributions to this work. Eric Howlett, designer of the LEEP optics, kindly

provided the optical specifications. Mike Teitel of VPL provided needed measurements of the EyePhone. Jack Goldfeather helped with the mathematical representation of the distortion. Fred Brooks, Henry Fuchs, and Steve Pizer helped to initiate and guide the HMD project at UNC, and various parts of the UNC HMD system were built by each of the team members: Ron Azuma, Bill Brown, Jim Chung, Drew Davidson, Erik Erikson, Rich Holloway, Jack Kite, and Mark Ward. Fay Ward and Linda Houseman helped to hold it all together and Vern Chi, David Harrison, and John Hughes provided essential technical know how. The Pixel-Planes team provided us with the high-powered graphics engine needed to drive the HMD. This work builds upon earlier work by one of us (Robinett) at NASA Ames Research Center, and we would like to acknowledge the contributions of Scott Fisher, Jim Humphries, Doug Kerr, and Mike McGreevy. We would like to thank Fred Brooks for his critique of the paper, Julius Smith for his sketch of the ideal structure of a technical paper, and the anonymous reviewers for their suggestions. This research was supported in part by the Defense Advanced Project Research Agency, contract DAEA18-90-C-0044 and also by the Office of Naval Research, contract N00014-86-K-0680.

References

Buchroeder, R. A., Seeley, G. W., & Vukobradatovich, D. (1981). *Design of a Catadioptric VCASS Helmet-Mounted Display.* Optical Sciences Center, University of Arizona, under contract to the U.S. Air Force Armstrong Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, Dayton, Ohio, AFAMRL-TR-81-133.

CAE (1986). *Introducing the visual display system that you wear.* CAE Electronics, Ltd., C.P. 1800 Saint-Laurent, Quebec, Canada H4L 4X4.

Callahan, M. A. (1983). *A 3-D display headset for personalized computing.* M.S. thesis, Dept. of Architecture, Massachusetts Institute of Technology.

Chung, J. C., Harris, M. R., Brooks, F. P., Jr., Fuchs, H., Kelley, M. T., Hughes, J., Ouh-Young, M., Cheung, C., Holloway, R. L., & Pique, M. (1989). Exploring virtual worlds with head-mounted displays. *Non-Holographic True 3-Dimensional Display Technologies,* SPIE Proc. Vol. 1083.

Clark, J. H. (1976). Designing surfaces in 3-D. *Communications of the ACM, 19*(8), 454–460.

Eyles, J., Austin, J., Fuchs, H., Greer, T., & Poulton, J. (1988). Pixel-Planes 4: A summary. In A. A. M. Kuijk and W. Strasser (Eds.), *Advances in computer graphics hardware II* (pp. 183–208). Eurographics Seminars. Berlin: Springer-Verlag.

Fisher, S. S., McGreevy, M., Humphries, J., & Robinett, W. (1986). Virtual environment display system. *Proc. 1986 Workshop on Interactive 3D Graphics,* 77–87.

Glines, C. V. (1986). Brain buckets. *Air Force Magazine, 69*(8), 86–90.

Hecht, E., & Zajac, A. (1974). *Optics.* Reading, MA: Addison-Wesley.

Henderson, B. W. (1989). Simulators play key role in LHX contractor selection. *Aviation Week,* November 27.

Hodges, L. F., & McAllister, D. F. (1990). Rotation algorithm artifacts in stereoscopic images. *Optical Engineering 29*(8).

Holloway, R. L. (1987). *Head-mounted display.* Tech. Rep. TR87-015, Dept. of Computer Science, University of North Carolina at Chapel Hill.

Howlett, E. M. (1983). Wide angle color photography method and system. U.S. Patent Number 4,406,532.

Kocian, D. F. (1988). *Design considerations for virtual panoramic display (VPD) helmet systems.* Armstrong Aerospace Medical Research Laboratory, Visual Display Systems Branch, Wright-Patterson Air Force Base, Dayton, Ohio 45433-6573.

LEEP (1990). *Cyberface II applications note.* LEEP Systems/Pop-Optix Labs, 241 Crescent St., Waltham, Massachusetts 02154.

Longhurst, R. S. (1973). *Geometrical and physical optics.* New York: Longman.

ORA (1991). *Code V Reference Manual, Version 7.40.* Optical Research Associates, 550 North Rosemead Blvd., Pasadena, California 91107.

Pang, X. D., & Durlach, N. (1990). Personal communication.

Saunders, B. G. (1968). Stereoscopic drawing by computer—Is It Orthoscopic? *Applied Optics, 7*(8), 1499–1504.

Sutherland, I. E. (1965). The ultimate display. *Proceedings of the IFIPS Congress, 2,* 506–508.

Sutherland, I. E. (1968). A head-mounted three-dimensional display. *Fall Joint Computer Conference, AFIPS Conference Proceedings, 33,* 757–764.

Times (1989). Mini screen is a real eye opener. *London Times,* May 28.

Vickers, D. L. (1974). *Sorceror's Apprentice: head mounted display and wand.* Ph.D. dissertation, Dept. of Computer Science, Univ. of Utah, Salt Lake City.

VPL (1989). *VPL EyePhone Operations Manual.* VPL Research, 656 Bair Island Rd., Suite 304, Redwood City, California 94063, p. B-4.