Interactive Stereoscopy Optimization for Head-Mounted Displays

Patrick Min and Hans Jense

February 1994

Abstract

In current Virtual Environment systems, the stereoscopic images presented in a Head-Mounted Display are far from optimal. The aim is to achieve orthostereoscopy, which roughly means images should "behave as in real life". A theoretical model developed by Robinett and Rolland [RR91] was used to implement a stereoscopy test system, allowing the independent setting of many stereoscopy related parameters. Tests aiming to optimize these parameters were devised. This paper reviews the theoretical model, then describes its implementation and the conducted tests. It ends with the test results and the conclusions that may be drawn from them.

1 Introduction

Especially in more serious VE applications such as simulation and training systems it is essential that the environment mimics its real counterpart as much as possible. This includes a realistic three-dimensional image of the environment. Ideally, the image is orthostereoscopic, or as Sutherland states:

> "The image presented by the three-dimensional display must change in exactly the way that the image of a real object would change for similar motions of the user's head [Sut68]"

In most current VE systems stereo images are produced without taking all of the factors that influence the stereoscopic quality into account. Instead of using an accurate computational model of the Head-Mounted Display (HMD), most HMD parameters are simply ignored, or set to default values. This results in an image that is not orthostereoscopic, i.e. not three-dimensionally realistic, and probably causes eyestrain. In the next section we discuss the errors that may occur for each parameter.

Robinett and Rolland presented a computational model for the stereoscopic optics in an HMD [RR91]. This model was used as the basis for the implementation of a system allowing independent manipulation of stereoscopy related parameters. A review of the model is given in section three.

Next an overview of the test system is provided, both the hardware used and the software functionality. Tests were devised aiming to optimize various parameters. Twelve people were tested in order to evaluate the tests. The results are presented and discussed. Finally a conclusion section is given, containing our suggestions for better stereoscopy, and for future research.

2 Possible errors

The errors that may occur when using incorrect parameters are classified into two categories:

- general, not HMD specific errors
- HMD specific errors

Errors of the first category result from an incorrect simulation of reality, in other words of the way eyes see in real life. Errors of the second category result from not, or incorrectly, incorporating the properties of the HMD itself in the rendering calculations.

2.1 General errors

2.1.1 Accomodation does not correspond with convergence

Eyes are accustomed to convergence and accomodation being correlated: for every distance there is an appropriate convergence angle (such that the eyes are turned towards an object at that distance), and accomodation (to bring the object into focus). A display screen normally is positioned at a fixed distance, hence the eyes have a constant accomodation. But the convergence of the eyes corresponds to the apparent distance of an object in the Virtual Environment. Veron calls this phenomenon an accomodation/convergence conflict [VSLC90]. Robinett and Rolland suggest the user must learn to decouple accomodation and convergence [RR91].

2.1.2 Incorrect projection

- *Off-axis projection*
  The off-axis projection assumes converging viewlines and two centers of projection, one for each

eye [Hod92]. It most closely corresponds with reality, because our eyes converge towards the object in focus. The only problem here is to find out on which object the viewer is focusing, because this determines the convergence angle.

- *On-axis projection*
  On-axis projection uses parallel viewlines and one center of projection. The necessary horizontal shift for the left- and right-eye is accomplished by translating the object data. Hodges describes an algorithm for on-axis projection [Hod92]. Roughly, the algorithm works as follows: for the right eye view:

  – translate the object data to the left by IPD/2

  – standard perspective projection

  – pan the resulting image back

  (and the other way round for the left eye view) The Field Of View of on-axis projection is the same as for a single perspective projection. Williams and Parrish show that for example for a 40 degree horizontal FOV per eye the binocular FOV is 35 % smaller than it would have been if an off-axis projection had been used [WP90].

2.1.3 Incorrect Inter-Pupillary Distance

The IPD of a viewer determines how much he must converge his eyes to focus on an object at a specific distance. If a standard IPD is assumed in the rendering computations (e.g. 65 mm), a viewer with a larger IPD would perceive the object at too large a distance, and someone with a smaller IPD would think the opposite.

2.2 HMD specific errors

Next the HMD specific errors are considered. Figure 1 illustrates the role of the HMD in a typical VE system.

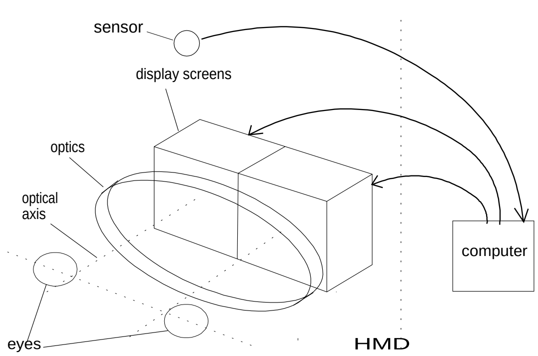

Figure 1: Role of the HMD in a typical VE system

2.2.1 Positional errors

If the optical axes were parallel, and passed through the center pixels of the screens and through the centers of the eyes, turning on the center pixels would show a dot positioned at infinity. But the axes may not be parallel, the screen centers may be offset from the axes, and the eyes may be offset with respect to the axes as well.

- *Failure to account for angle between optical axes*
  When the optical axes are not parallel (e.g. due to manufacturing tolerances/restrictions), this has to be corrected by a rotation of the left and right-eye image, such that it balances out the rotation.

- *Failure to incorporate position of screens*
  If the screen centers are offset from the optical axes, all displayed data are offset. In case of a horizontal offset, the eyes need a different (incorrect) convergence angle to focus on an object. A vertical offset results in a height error.

- *Failure to incorporate Inter-Pupillary Distance*
  In addition to using the correct IPD in the projection, it is also important with respect to the HMD. If the viewer has an IPD equal to the distance between the optical axes, the images are positioned correctly: each center of projection is located exactly in front of each eye. If the IPD differs from this distance, the images are in a horizontally incorrect position, resulting in a convergence error. In an HMD with mechanical IPD adjustment this problem does not occur, as the screens themselves are moved to get the centers positioned right.

- *Incorrect Field Of View*
  The Field Of View used in the projection computations should be the same as the FOV actually experienced by the viewer, i.e. the FOV actually subtended by the images of the display screens. If the computational FOV is too small, then the displayed object will appear too large, and vice versa.

2.2.2 Optics errors

- *Non-linear distortion*
  When a wide Field Of View is warped onto a flat plane, distortion is inevitable [How91]. The LEEP optics used in many HMDs use a fish-eye like transformation to be able to project a large FOV onto a plane. This means that the largest part of the image area is devoted to the central part of the FOV, and that the peripheral area is compressed into the side of the image. [1]

[1] Fortunately this *does* correspond with the relative importance of the various parts of the human FOV

When a flat plane is seen through the optics, the magnification is larger for points that are further from the optical axis. This is called a positive or pin-cushion distortion, and causes lines that are straight on the screen to be curved in the virtual image. In the next section a model for the distortion is discussed, as well as an approximate inverse distortion to correct the error (called a negative or barrel distortion).

- *Chromatic aberration*
  Differently coloured light rays diffract differently in the lens system, causing lateral chromatism, or "chromatic difference of magnification" [How91]. In the LEEP optics, blue is magnified about 1 % more than red, with green in between. This error is especially noticable in the peripheral part of the FOV.

3 The computational model

3.1 Model for one eye

In order to compute correct projections of the 3D image space, several HMD specific parameters need to be incorporated in the calculations. A computational model for the optics in an HMD given by Robinett and Rolland will aid in determining these parameters [RR91]. The general ideas involved are reviewed in this section. For a thorough discussion of the model the reader is referred to the original article.

The model relates the radial position of a pixel on a display screen inside the HMD (r_s) with the radial position of its corresponding pixel as perceived in the Virtual Environment (r_v). Both are normalized by dividing them by their maximum possible values, yielding an r_{sn} and an r_{vn}. If the optics had no distortion, then $r_{vn} = r_{sn}$. But they have, so an approximate correction term must be added: $r_{vn} = r_{sn} + k_{vs} * r_{sn}^3$. The coefficient k_{vs} is a measure for the amount of distortion present.

To remove the distortion, an image should be transformed by the inverse transformation. This is called predistortion. So to be able to predistort an image, we need to find the inverse of the above function. An exact closed-form expression is not possible, so we again use an approximation: $r_{sn} = r_{vn} + k_{sv} * r_{vn}^3$. Robinett and Rolland show that this approximation is at worst about 2 % off from the correct value [2].

3.2 Model for two eyes

So far we considered just one eye. We need to expand the model to include two eyes in order to:

- calculate a correct FOV
- incorporate the offset of the screen centers from the optical axes

3.2.1 Calculate a correct FOV

The simplest way would be to assume that the optics have linear magnification, but this obviously results in an incorrect computational FOV. We should account for the distortion. First we compute the radial positions in the virtual image of the top, bottom, left and right side of a screen. Using the distance of the virtual image we then get the angular position of each side, and consequently the horizontal and vertical FOV.

Another method is analytical ray tracing: from the exact optics specifications (for each lens in the lens system) the exact path of a ray passing each lens surface is calculated. This yields a slightly more accurate FOV.

3.2.2 Incorporate the offset of the screen centers from the optical axes

To correct this error, a perspective projection is necessary that has its computational center of projection at that offset.

4 The test system

4.1 Hardware

Figure 2 shows the hardware setup of the test system. The operator controls the setting of all parameters from his console. The resulting images are seen by someone wearing the HMD (the viewer).

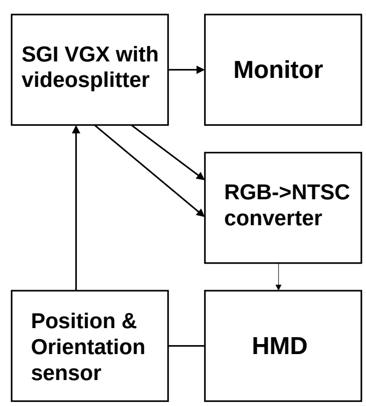

Figure 2: Hardware setup of test system

The equipment consisted of:

- computer: a Silicon Graphics 4D-240VGX graphics workstation with a "videosplitter", enabling output

[2] this is an estimated error, measured from a graph in the article

of up to four arbitrary quadrants of the operator screen

- monitor: having a screen resolution of 1280 by 1024 pixels

- converter: as the HMD requires two composite video signals with NTSC (RS170A) timing, and the videosplitter produces RGB with NTSC timing, a converter is needed

- HMD: a Virtual Research Flight Helmet, containing LEEP optics and two LCD screens with 320 by 200 LCD pixel resolution. Each pixel is part of a "colour triad" and is either red, green or blue, giving a colour resolution of approximately 185 by 139 pixels. The total (binocular) Field Of View is 99 degrees horizontal and 59 degrees vertical.

4.2 Software

The system was written in C++. The graphics routines use the Silicon Graphics GL library, the user interface routines the Forms Library [Ove92]. In Figure 3 a data transformation diagram of the display loop is shown.

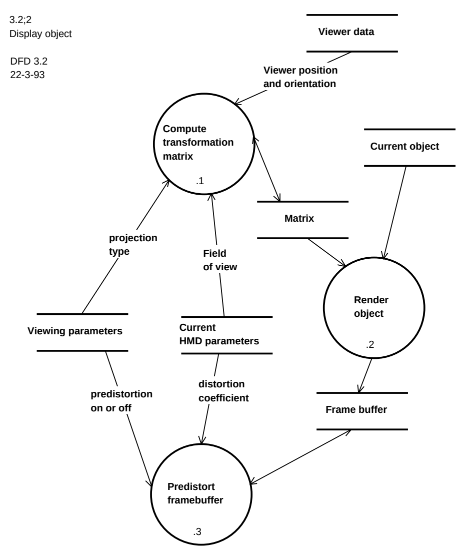

Figure 3: Data transformation diagram of display loop

The operator screen displays the images sent to each HMD screen, a menu and a status window showing the values of most parameters. An option is available that allows the viewer to switch between viewing the 3D image or the menu and status window inside the HMD, to enable one person to operate the entire system. This only makes sense when the resolution of the HMD screens is sufficiently high.

4.3 Correcting the errors

4.3.1 Moving the center of projection

Whenever a parameter affecting the horizontal position of the images changes (such as the IPD, or the distance between the optical axes), the computational center of projection has to be moved. In our system this is done by computing and displaying images that are larger than the ones seen in the HMD: only a part of each image is seen by the viewer. So if the center of projection has to be moved, we simply move the position of the parts that are sent to the HMD.

4.3.2 Projection type

Both off and on-axis projection are provided.

4.3.3 Account for the optical axes angle

If the optical axes are not parallel, the left and right image are rotated clockwise and counter-clockwise respectively (through the corresponding eye), both by half the angle between the axes.

4.3.4 Account for the FOV

From the optics specification and the position of the screens w.r.t. the optics the horizontal and vertical FOV are calculated by the method specified in paragraph 3.2.1. It is also possible to interactively change the horizontal and vertical FOV.

4.3.5 Predistortion

There is a finite number of pixels that have to be predistorted, i.e. moved to another location. Hence all destination coordinates can be precomputed and stored in a table. We use a table for both the left and right image, as the optical axis is in a different position in each screen [3].

In our configuration, with an image resolution (on the operator screen) of 640x485, the tables each use just under 1.2 MBytes of memory. Storage of images and z-buffers required an additional 8 tables, bringing the total memory requirements to just under 12 MBytes.

4.3.6 Chromatic aberration

Correction of the chromatic aberration is not implemented. A way to do this could be to render the red, green and blue components of each object in three separate frame buffers, scaled by the correct amount to compensate for the aberration, and then combine them.

[3] it is also possible to use one table, which is large enough to hold every possible offset from an optical axis

5 The tests

Our test procedure involved the following steps:

- stereoscopic viewing test
- IPD measurement
- IPD test in the VE, using a specially designed test object
- predistortion test using a regular grid

5.1 Stereoscopic viewing test

Subjects should be able to view stereoscopically. The test we used is the standard TNO test for stereoscopic vision [IT72]. The subject wears a pair of glasses with the left glass coloured red, and the right one green. A series of random dot stereograms is presented, containing pictures requiring a certain stereo acuity in order to be seen.

5.2 IPD test

First the subject's physical IPD was measured. Then the subject put on the HMD, in which a special test object was displayed, as shown in Figure 4.

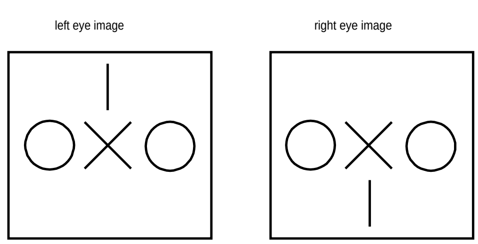

Figure 4: IPD test object

The fact that the left and right eye image are different is because the brain is very much able to correct for stereo images based on an incorrect IPD. The idea with this test object is that the brain will attempt to converge the central part of the images (the "OXO"), and probably succeed if the computational IPD is not too far off, and leave the vertical lines at their true position, as they are each conflicting with the data received by the other eye. So we assume that the brain behaves differently for different areas of a perceived image.

Initially the computational IPD was set to the measured IPD plus 10 mm. In this way we were sure that the subject perceives an incorrect image, because he cannot diverge his eyes. The subject would then change the computational IPD until the OXO converged and could be comfortably viewed. The vertical lines however were usually not aligned by then, which made a further, more precise adjustment of the IPD possible.

5.3 Predistortion test

The subject was positioned directly in front of a regular grid, and was asked if the grid lines in the edges of the image either:

- curved outward
- were straight
- curved inward

The predistortion coefficient was adjusted until the viewer was convinced that the grid lines appeared (approximately) straight.

6 Test results and discussion

6.1 Stereoscopic viewing test

Twelve persons were tested, all able to see stereoscopically. The average stereo acuity was $69'' \approx 1'$.

6.2 IPD test

The average measured IPD was 65 mm. The IPD test was conducted twice, once with the test object at a distance of two meters, and once at a distance of 0.5 meters. At the latter distance the test object would almost completely fill the FOV.

The average IPD that appeared most comfortable was 63 mm with the test object at 2 meters, and 66 mm with the test object at 0.5 meters.

The significance of these results is somewhat reduced because the precise mapping of the images on the operator screen to the display screens inside the HMD is not known. For this the HMD must be disassembled (at the time of writing the manufacturer did not have exact data either). By looking with the right eye through the left eye optics and vice versa it could be seen that:

- not all of the operator screen images (i.e. the part sent to the HMD) is visible on the HMD screens
- the loss of data is different for each side of a screen
- the loss of data is different for the left and right screen

This implies that all correctional translations are inaccurate: they are calculated in operator screen pixels, assuming a certain number of pixels map on a certain width in millimeters on an HMD screen. Compensation for this error (called *video overscan* or *image cropping*) is treated extensively in a recent report by Rolland and Hopkins [RH93].

6.3 Predistortion test

The theoretically optimal predistortion coefficient for the LEEP optics in our HMD is -0.18 [RR91]. The average of the coefficients chosen by our subjects to be optimal was -0.17. This coefficient results in a slightly less (barrel) predistorted grid.

The difference between our value and the theoretically optimal one may be explained by two reasons:

- as has been said, the exact operator screen image to HMD screen mapping is not known, causing pixel positions calculated in the precomputing stage to be slightly off
- before the grid is predistorted, the subject sees a pin-cushion distorted grid. This may influence the subject such that he sees a barrel-distorted grid after predistortion, even if this grid is actually straight.

6.4 Relative importance of the parameters

Using our brief experience with the test system, we can attempt to determine the relative importance of each parameter influencing stereoscopic quality.

- *correct convergence*
  This depends on the images' horizontal position, which in turn depends on whether or not the IPD, screen center offset, the distance and angle between the optical axes are incorporated in the calculations.
- *projection type*
  We found that on-axis projection is especially disturbing for viewing at close distances. If a VE application has just one object of interest (e.g. a tool) at a small distance, one may choose to render it using converging viewlines (= off-axis projection), and the surroundings using on-axis projection.
- *optics distortion*
  The optics distortion may also result in incorrect convergence, especially near the edges of the image. Apart from that it obviously causes the image to be incorrect. The importance of this error also very much depends on the type of VE application.

  The predistortion is computationally too expensive to be performed during real-time rendering. In our system, after optimization and parallel implementation, a frame rate of 4 Hz may be achieved (that's using four 25 MHz MIPS R3000 processors). The predistortion can easily be implemented in hardware, which should be done if real-time rendering is required.
- *field of view*
  As an incorrect FOV only results in a (relatively small) size error, we classify it as the least important error. Naturally this may not be the case, depending on the application requirements.

Note that the resolution of the HMD display screens determines whether a certain error correction is useful or not: if a positional error does not cause a shift of at least one pixel, it will not be visible anyway.

Concerning the special IPD test object, it will be interesting to see if after incorporation of the exact operator screen images to HMD mapping the test can be used to determine *one* comfortable IPD for all distances. In our opinion the test object should occupy a considerable part of the FOV, so it does not become too easy for the brain to combine both images.

7 Conclusion

From our brief experience with the test system, it has already become obvious that the incorporation of as much knowledge as possible that we have about the display system (i.e. the HMD) pays off: when set to the correct IPD, the system rendered convincing, solid three-dimensional objects. The ability to independently vary parameters influencing stereoscopy has proven to be very useful: we may now determine the quality vs. computational cost trade-off of several optimizations.

Future improvements of our system will include:

- accounting for the video overscan
- experimenting with mixed projection types
- experimenting with "IPD setting" test objects
- improvement of the predistortion performance, either through hardware or software

It follows that we recommend the following setup for a general VE system:

- incorporate all parameters that improve convergence
- use off-axis projection (converging viewlines) for nearby (say closer than 3 meters) objects, and on-axis projection for further objects
- use predistortion implemented in hardware

After taking these measures, we will come close to achieving orthostereoscopy.

References

[Hod92] Larry F. Hodges. Time multiplexed stereoscopic computer graphics. *IEEE Computer Graphics and Applications*, 12(2):20–30, march 1992.

[How91] Eric M. Howlett. Wide angle orthostereo. In *Proc. SPIE vol.1457, Stereoscopic Displays and Applications II*, pages 210–223, 1991.

[IT72] IZF-TNO. *TNO test for stereoscopic vision.* Laméris Ootech, ninth edition, 1972.

[Ove92] Mark H. Overmars. *Forms Library, A Graphical User Interface Toolkit for Silicon Graphics Workstations.* Utrecht University, the Netherlands, 2.1 edition, 1992.

[RH93] Jannick P. Rolland and Terry Hopkins. A method of computational correction for optical distortion in head-mounted displays. Technical Report TR93-045, Dept. of Computer Science, UNC at Chapel Hill, 1993.

[RR91] Warren Robinett and Jannick P. Rolland. A computational model for the stereoscopic optics of a head-mounted display. In *Proc. SPIE vol.1457, Stereoscopic Displays and Applications II*, pages 140–160, 1991.

[Sut68] Ivan E. Sutherland. A head-mounted three dimensional display. In *Proc. Fall Joint Computer Conference*, pages 757–764, 1968.

[VSLC90] Harry Veron, David A. Southard, Jeffrey R. Leger, and John L. Conway. Stereoscopic displays for terrain database visualization. In *Proc. SPIE vol.1256, Stereoscopic Displays and Applications*, pages 124–135, 1990.

[WP90] Steven P. Williams and Russell V. Parrish. New computational control techniques and increased understanding for stereo 3-d display. In *Proc. SPIE vol.1256, Stereoscopic Displays and Applications*, pages 73–82, 1990.