8

3D Stereoscopic Rendering: An Overview of Implementation Issues

Anders Hast

UPPMAX, Uppsala University

Overview

In recent years, there has been an increasing interest in the field of 3D display technologies from the entertainment industry. Today, the movie industry is moving in at a wide front as thousands of 3D stereoscopic movie theaters are being installed worldwide and movie production companies produce their films also in 3D. In fact, a new job title has emerged, the *stereoscopist*, in charge of making sure that the scenes can be viewed without problems by the audience. Some of the things the stereoscopist must deal with are explained in this gem. At the same time, many products for the home audience have been developed at an affordable price range and sold to a growing number of customers, including stereo capable TV sets and computer screens. This gives the games industry a user base that will be familiar with watching 3D stereoscopic content and who, in the future, might also be expecting their favorite games to be released in stereoscopic 3D. This gem deals with the far most common type of stereoscopic display, the *plano-stereoscopic* display. In contrast to other types of stereoscopic displays, these are displays that work with two planar surfaces in order to achieve the impression of depth. We base the discussion around the importance of designing the content to fit for a stereoscopic display and the different kinds of viewing conditions that must be considered. Moreover, we discuss the mathematics that help us compute these viewing conditions in order to be able to view the content without any problems. And finally, we provide an overview of different types of display techniques.

We start by briefly familiarizing the reader with the field of stereoscopy and different depth cues, covering some implementation details. We then iterate over some issues that can arise when integrating stereoscopic display support into a game engine. Even though most terms are explained herein, it might be valuable for the reader to use the online glossary provided by [6].

8.1 Mechanisms of Plano-Stereoscopic Viewing

The concept behind plano-stereoscopic displays can be seen quite simply as the creation of two planar views of the game, one for the left eye and one for the right. Then, it is important to make sure that each eye sees only the view intended for that eye. The processes involved can be seen as coding and decoding processes. In scientific visualization, one says that stereoscopic images are aimed to help the viewer form a 3D mental image of the data set. In video games, on the other hand, the aim is to give the player a richer visual experience.

In order to create the sense of depth in a normal rendered game (i.e., a monoscopic rendering), monocular depth cues are used in contrast to the binocular depth cues discussed later. Some examples of *monocular* depth cues are the following:

- *Occlusion* occurs when objects closer to the viewer occlude objects that are further away, and this is handled by the depth buffer algorithm.
- *Parallax* is an effect caused by the motion of the observer, and it creates the illusion that objects close to the observer are moving by faster than objects further away. For

instance, to someone looking out the window of a moving train, trees in the foreground appear to move by much faster than a distant hillside.

- The *size* of an object varies depending on the distance from the viewer due to the perspective projection, where parallel lines converge at the horizon.
- *Texture detail levels* provide information about the distance to an object.
- *Atmospheric effects* such as scattering or haze make objects appear more gray in relation to the distance between them and the viewer.
- *Shading and shadows* tell us about curvature and inter-object relationships.
- The *proximity to the horizon* also provides a depth cue since we know that the horizon is far away, and objects close to the line of the horizon are thus perceived as being far away.

What a stereoscopic display adds to these cues are the three *binocular* depth cues known as *accommodation*, *convergence*, and *retinal disparity*. Convergence occurs when we focus our view at an object in real life by rotating our eyes so that their lines of sight intersect at the point of interest. At the same time, we apply pressure to the lenses in the eyes in order to focus, and this is called accommodation. Under normal natural viewing conditions, both accommodation and convergence correspond and are habitual, but can be voluntarily put out of function by crossing the eyes. The third cue is retinal disparity, and this pertains to the fact that we have two retinal images that fall on different points of the two retinas. These are then merged by the brain and perceived as a single image.

The virtual space that we define is divided by the screen into two regions called *view space* and *screen space*. The volume between the viewer and the display is the view space and the volume behind the display is called screen space. If we look at a point lying at the same depth as the display surface onto which it projects, as shown in the left image of Figure 8.1, then the homologous points on the display have zero parallax because they have no lateral displacement. The homologous points are identical features in the stereo pair; thus, the same point in space is located on different places in the left and right stereo image for nonzero parallax. Points that are lying in either view space or in screen space have a lateral displacement and are then said to have either negative or positive parallax. All three cases are shown in Figure 8.1. Converging at a point behind the display surface causes the homologous points on the display to have positive parallax. This point is said to be in screen space. Similarly, converging at a point in front of the display surface causes the homologous points on the display to have negative parallax, and this point is said to be in view space.

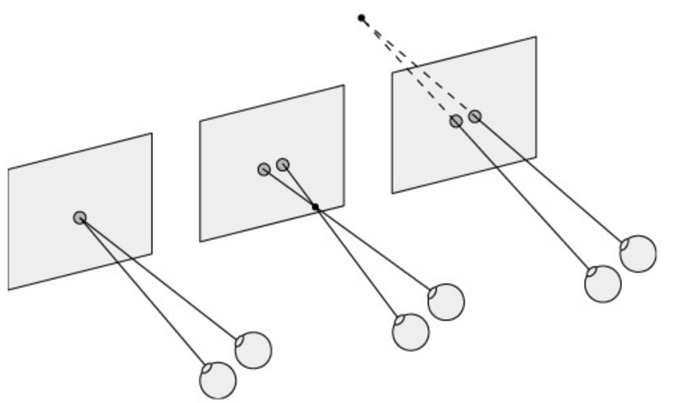

Figure 8.1: Three different types of parallax that can occur, from left to right— zero parallax, negative parallax, and positive parallax.

Converging the eye's axes upon a virtual point at distance D_c supports the fusion of the parallax image and stereopsis as shown in Figure 8.2, where stereopsis is the mental and psychological process in visual perception leading to the sensation of depth from two slightly different projections of the world onto each eye [1]. Keeping the visual structure on-screen in focus requires accommodation at screen distance D_s. Usually, accommodation is dominant, and for unaided viewing, we see one planar double image in focus. With some training (crossing the eyes) we can converge at D_c and see a fused 3D image out of focus. Stereographic devices such as stereo glasses greatly support image fusion and stereopsis at the cost of suppressed accommodation.

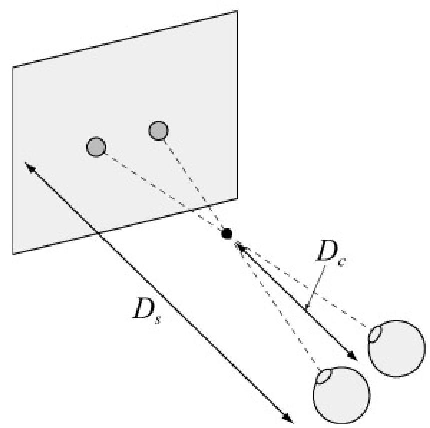

Figure 8.2: Convergence and accommodation in plano-stereoscopic displays.

Scale Considerations for 3D Stereo Images

To be able to create comfortable stereo images, we need to calculate the correct perspective for the given setup and make sure that we stay within those limits [8]. Some new situations arise from using a stereoscopic display. Mainly, scale considerations are important when modeling the virtual view volume and creating the actual stereo pair.

When converging on objects at some certain distance, a point at a different distance in the scene appears at the retina with some lateral disparity, independent of convergence, as shown in Figure 8.3. The inter-pupillary distance (*IPD*) is the distance between the eyes measured from the center of the pupils in each eye. The retinal disparity is, according to Kalawsky [12], computed as the difference of the two angles shown in the figure:

- *disp*=θ_1-θ_2.

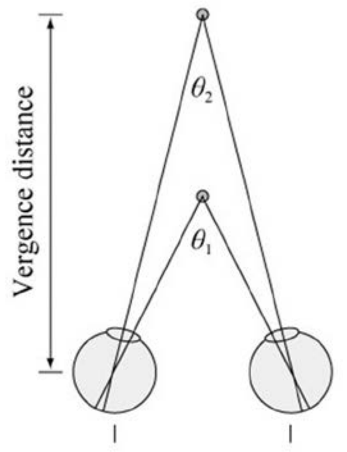

Figure 8.3: The two angles used to compute the retinal disparity.

A retinal disparity of more than 10° causes diplopia (double vision) and should, of course, be avoided at all times.

The projection in stereoscopic displays does not scale linearly. The projection of the two points that we fuse in our brain is very dependent on how far away we sit from the screen and how big the screen is. As this point is projected onto our eyes, the distance between our eyes is also a big impact factor. An average person has an eye separation around 6.5 cm. In a more uncommon case, we can find people with up to 7.5 cm between their eyes, and the largest audience or user group, youngsters, can have as few as 4.5 cm between their eyes. And it should be remembered that there is a great variation among people, particularly on different continents [4]. Some good advice would probably be for the IPD to be an adjustable variable set by the player in order to give a comfortable stereo depth.

The physical limit on our depth perception is around 200 yards. This comes from the fact that beyond this point, we don't get any convergence information at all since our line of sight is more or less parallel. This important fact must be considered when we create our virtual space, which we must design before we start to place objects in it.

Another practical issue in stereo graphics is that one should not exceed parallax values of more than 1.5° visual angle in order to not feel uncomfortable [13], as shown in Figure 8.4. The on-screen parallax (*osp*), measured in centimeters for different viewing distances *D*, is shown in Table 8.1.

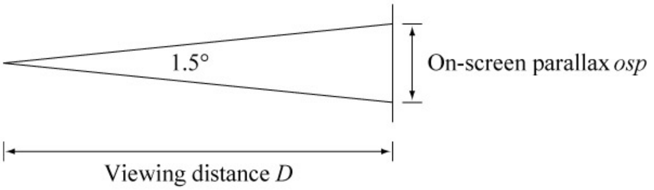

Figure 8.4: Parallax values greater than 1.5° visual angle should not be exceeded.

Table 8.1: Practical examples for on-screen parallax values.

***D* (cm)**	**On-screen-parallax (cm)**
50	1.31
75	1.96
100	2.62
200	5.24
300	7.86
400	10.47

Let us now look at some practical examples of how the virtual space depth is limited due to the distance to the observer and the *osp* for a negative parallax situation, as shown in Figure 8.5, where *d* is the virtual spatial depth. We have

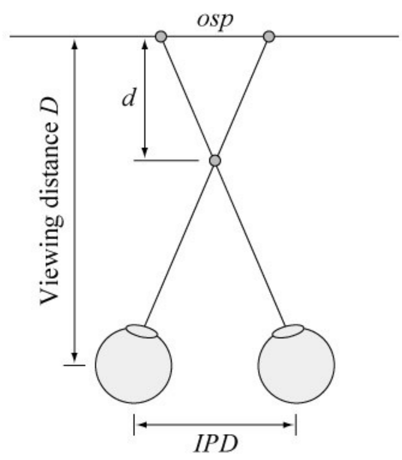

Figure 8.5: Negative parallax.

and solving for *d* gives us

$$d = \frac{D}{IPD/osp - 1}$$

Similarly, we can compute how the virtual space depth *d* is limited due to the distance *D* to the observer and the *osp* for a situation with positive parallax as shown in Figure 8.6. Tables 8.2 and 8.3 show some example values for the cases of negative and positive parallax.

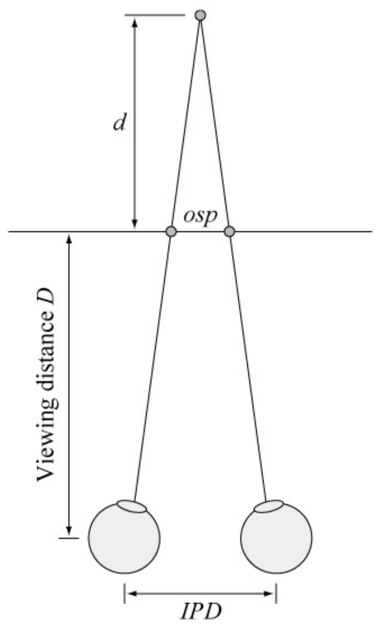

Figure 8.6: Positive parallax.

Table 8.2: Example for IPD=6.5 cm, for negative parallax.

D **(cm)**	***osp*** **(cm)**	***d***	***d/D***
50	1.31	8.38	0.17
75	1.96	17.40	0.23
100	2.62	28.72	0.29
200	5.24	89.24	0.45
300	7.86	164.17	0.55
400	10.47	246.83	0.62

Table 8.3: Example for IPD=6.5 cm, for positive parallax.

D **(cm)**	***osp*** **(cm)**	***d***	***d*/D**
50	1.31	12.61	0.25
75	1.96	32.47	0.43
100	2.62	67.47	0.67
200	5.24	829.45	4.15
300	5.76	1714.7	7.79
400	6.42	18612.4	75.97

Before you go ahead and work out the math for your game, there are some things worth mentioning. First, one should notice that values for maximum allowable disparity in the literature are expressed in different terms such as angular disparity, on-screen parallax, etc. These values sometimes apply only as a rule of thumb. Lipton's values appear to work for many VR applications, but they do not generally apply for all viewing conditions. According to Lipton, compositing stereoscopic images is an art, not a science. In the end what matters is that it looks and feels good!

The Basic Setup

Several propositions have been made on how to setup and render graphics for stereoscopic displays as efficiently as possible. One way is to let the graphics driver handle it. Some of the major graphics vendors have native support for stereoscopic displays in their drivers where it creates the stereo pair. This does require that the game is compatible with stereoscopic display in the form that the content is conformed for these kinds of systems. And this limits your game to be used only for these specific graphics card vendors in order to make your game work in stereo. The second, less common alternative in the context of the game industry is to make a graphics command interceptor [3]. This is an application that pretends to be the graphics driver and intercepts all the graphics sent from an application. This can be stored and then in theory be used to create the two stereo pairs. It is still required that the game create content that is suitable for stereoscopic displays.

Usually, we would use the following simple approach to create the game in stereo.

```
while (user wants to play)
  doGameLogic()
  setupProjectionForLeftEye()
  render()
  setupProjectionForRightEye()
  render()
```

In our game, we must set the virtual cameras to focus on the point of interest, as shown in Figure 8.7, using the previously described math in order to ensure that the parallax do not exceed the values for the *allowable* virtual space depth.

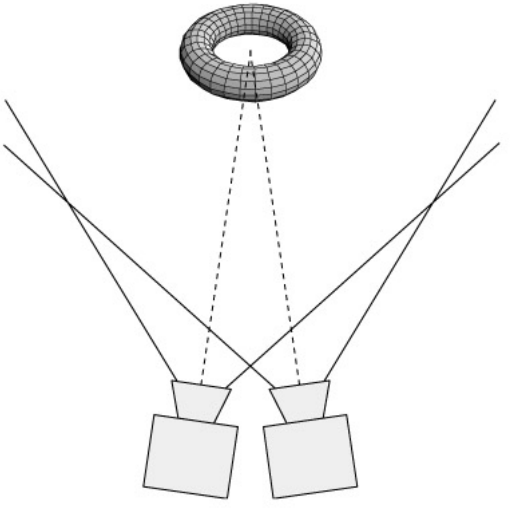

Figure 8.7: Camera setup for the 3D stereoscopic game.

8.2 Stereo Techniques

Today, there are three different popular techniques of viewing graphics in stereo on the market: anaglyph stereo (a.k.a. red/green stereo) [2], temporal multiplexing, and polarization. We give a brief overview of these techniques and mention some pros and cons as well as examine the coding and decoding process for each of them. An important concept discussed is *ghosting* [14], which should be avoided as much as possible. Ghosting means that one eye sees some of the content meant for the other eye.

Anaglyph Stereo

Anaglyph stereo is the simplest and cheapest way of delivering stereoscopic game content to the players. Here, the left and right views are separated using wavelength separation. The left view is simply encoded in the red channel and the right view is encoded with complementary colors such as the green channel or both blue and green (cyan). The player needs to wear the well-known red/green glasses in order to separate the two images so that each eye sees only the image meant for it. Hence, the color of the lenses in the glasses corresponds to the color channels that encode the picture, and this is the reason for its name. The source is encoded into one image, which means that this method can be used for a lot of different media, even in printed form.

The benefit of using this technique for games is that it does not require any special display system, just a pair of cheap red/green glasses in order to be able to view the effect. You've probably gotten a pair of cardboard glasses with acrylic lenses for free when you bought you favorite comic magazine that had a special 3D centerfold issue.

One pitfall using this technique is that the player must calibrate the screen so that the color matches the filtered glasses. Otherwise, the player sees a lot of ghosting with a bad quality stereo effect. There is also a noticeable loss of color, as a lot of the color information is being filtered out.

A more sophisticated version, sometimes called the super anaglyph technique, uses spectral multiplexing, more often referred to as Interference Light Technology (INFITEC) [9]. Spectral multiplexing works by dividing the visual spectra into six narrow bands, two bands for each of the primary colors red, green, and blue. These bands are then separated by filters and divided so that one band of each color reaches each eye. That is, half of the red spectra reaches the left eye, and the other half reaches the right eye, and so on. This system requires that two such filters of different type performing this spectral multiplexing are mounted onto two projectors for the display, as well as a pair of lightweight glasses that the player wears with one type for each eye. Hence, the viewer clearly sees, if he shuts one of his eyes, that the color information reaching the eye is slightly different compared to looking at the display using only the other eye. Nonetheless, the color differences are less than that seen when using the red/green glasses. Finally, the brain merges these two images into one without a problem, and the picture looks as expected.

Temporal Multiplexing

This technique encodes the stereo pair by interleaving them time-wise. Hence, one frame is being shown for the left eye while the right eye is being occluded by an active shutter. Similarly, the next frame is visible to the right eye while the left eye is being occluded. If this procedure is performed fast enough, the player perceives the interleaved images as one continuous stream of images and gets the stereopsis right. This technology cuts the effective frame rate in half, as it is necessary to render twice as many images to get the same update rate. The occluding is performed by a pair of active shutter glasses, usually a pair of LCD screens that is synchronized to the display. While the display system shows a new frame, it sends a signal to the glasses to make the shuttering. It is clear that this technique requires glasses that cost a lot more than the simple red/green glasses. Furthermore, it cannot be used for printed media since, clearly, the display system has to have two sources for the output.

Polarized Light

Image separation can also be achieved by encoding each image using polarization. The two images are superimposed onto the screen by the display system through a pair of orthogonal polarizing filters. The viewer also wears a pair of glasses with corresponding filters, which are relatively cheap compared to the active shutter glasses. The polarization directions in the glasses correspond to those on the source. Thus, no extra hardware is needed in the glasses, but once again it is necessary to have two sources of light, and this technique cannot be used for printed media. The viewer must not tilt his head when using this technique, as it results in severe ghosting. Alternatively, circular polarization can be used in an attempt to avoid this problem. In all cases, polarized filters dim the brightness of the source because the original content is filtered.

Summary

All three techniques previously mentioned (with the exception of the INFITEC variation of anaglyph stereo) have become mainstream for playing stereo games. A pair of anaglyph glasses has a very low price and does not require the player to invest in any additional hardware to enjoy stereo games, but it produces the worst color representation of the three mentioned techniques. However, the frame rate does not have to be increased as for temporal multiplexing. Temporal multiplexing requires that the player acquires the actual glasses and also has a source that can preferably emit 120 Hz, since the actual frame rate is cut in half with this technique. The current trend is that more of this type of display and TV are coming out. The polarized light technique is realized either as an actively polarized display or by having two projectors for the really luxurious players. Regardless, the glasses are passive and thus generally cheaper.

8.3 Design Considerations for 3D Scenes

What really differs when creating content for a stereoscopic game engine compared to a monoscopic one? There are some approximations that can look very bad or very flat when ported to a stereoscopic display. We discuss some of them in this section. It is also important to avoid some situations that can be unpleasant for the player [10].

Culling can be a problem in the sense that we have two frusta to clip against. Clipping against the monoscopic frustum would yield erroneous results, as this is smaller than the combination of the left and right frusta. This problem can be handled in at least two ways, either clip against the joint frusta from left and right at once or clip separately against both frusta.

Stereoscopic rendering in its naive form, by rendering the scene twice from two different perspectives, occurs at around twice the cost. Some effects do not need to be calculated twice, for example, those that are view point independent. Some of the shortcuts that are normally made in monoscopic rendering do not work in stereoscopic rendering. If these visual effects are not gameplay critical, like glows around objects, they can be turned off or replaced by similar object-based effects.

The offscreen buffers only need to be duplicated if they have some view-dependent effect. This applies to shadow mapping and to those variants where the shadow map is based on the setup of the view frustum. Many of the often used screen-space post processing effects do not look good in stereo. One thing to be careful with is high dynamic range (HDR) rendering. The contrast between left and right view is very important, and if we apply tone mapping separately to left and right eye, we can introduce a shift in contrast that induces strain to the player.

Billboarding is another commonly used effect that does not port well to stereoscopic displays. Billboards, which are 2D objects that always face the viewer, look under certain conditions like two planar objects and not like a volumetric object as they were intended. The same applies to impostors—they must be made eye-dependent to not look completely flat on a stereoscopic display. This also incurs a performance penalty since twice as many impostors must be rendered.

Backgrounds must, on a stereoscopic display, be placed at the proper depth so that they appear to be as far away as backgrounds usually are. This is especially important with skyboxes and skydomes, as they otherwise appear to be placed too close to the player as if the sky is part of the ceiling. This also relates to the depth range in your scene. If techniques are used that require different depth ranges, these also destroy the stereoscopic sensation. Different depth ranges imply different depth in stereo, and objects end up in unexpected places in the virtual room, causing them to look deformed.

The same effect can be seen with overlays such as graphical user interfaces (GUIs), which are normally rendered in screen space without depth information. However, in stereo this gives contradicting depth cues because the menu can be drawn over an object that lies in view space, but the occluding cue implies that we should see the object. The GUI should lie behind the object, but as consequence of the screen-space rendering, lies at the wrong depth. One solution to this problem is to always put the GUI closest to the player. Depending on how the virtual room is set up, this solution might shove the GUI into the player's face so it must be done with great caution.

Other effects that traditionally take place in screen space, such as text labels, must also be placed in the game world at the proper depth. This now means that they can also be covered by objects in front of them, which might change how the actual game plays. In combination with GUI and text labels, we often also have some kind of representation of input devices. The standard mouse cursor rendered by the operating system can often cause anomalies in different ways. If we are rendering the content to a side-by-side buffer, which is a common technique to render 3D content, we only see the cursor in one eye since both eyes share the same buffer. Seeing an object with only one eye that should be seen by both eyes is really annoying and should always be avoided!

Thus, we have the same problems as with the GUI when the pointer is rendered in screen space. It is therefore advisable to create a software cursor that is placed as an object with depth in the world.

It is really tiring for the eyes, and the viewer even loses the 3D effect for a short while, if the focus from one scene to another changes dramatically. If an object appears very close to the viewer in one scene, then it can be dragged back a bit before the scene changes, or rather, the focus needs to be pulled back to the place of focus for the next scene. This is something that traditional games usually never bother with, but will probably be a great challenge for stereo game developers. A similar problem occurs for stereo films with subtitles, as they will definitely make the viewer tired since the focus needs to be changed repeatedly. The viewer will probably end up concentrating on the film without even bothering to read the subtitles.

Situations with contradicting depth cues in relation to occlusion can also arise when objects move off the screen in view space. Now, the depth cue tells the player that the object is in front of the screen, but when it comes close to the right or left edge of the screen it disappears behind it. To make this so-called edge conflict a bit less apparent, we can move the object slowly off screen or introduce virtual borders on the screen. These would be two guard bands lying in depth closest to the viewer to prevent the contradicting depth cues. This solution is known as floating windows and was used as early as the 1950s [15]. It has so far not been used for computer games, but was recently used for the Pixar film *Up*.

We also see a need for a greater lower bound of the accepted frame rate in a game when migrating to stereo. People tend to experience jerkiness if the frame rate drops below 60 FPS when viewing it in stereo, a far greater number than with monoscopic displays. Depending on what type of viewing device the player uses, we can also experience problems with frame-sequential delivery of stereographical content.

8.4 Outlook

Using stereoscopic displays is a great way to increase the gameplay value in the form of immersion. This is still somewhat of an unexplored field even though stereoscopic displays have been used for a longer period of time in other fields such as scientific visualization. The film industry has more experience than the game industry with stereoscopic displays, and we are now starting to see more and more movies created for these types of displays. When will the game industry follow in those footsteps and release more games that are directly catered for stereo enabled devices? So far, the majority of the games that are stereo capable are ports from monoscopic games. The step from monoscopic to stereoscopic also gives us game developers a new dimension to create more interesting games design-wise. After all, 3D stereoscopy is regarded by the film makers as yet another important storytelling technique. The future will hold many new design approaches, both from a pure rendering perspective and also from a gameplay perspective. An interesting approach is to render the scene from one virtual camera position and use the information in the depth buffer to generate a stereo pair using Depth-Image-Based Rendering (DIBR) [7]. A simple example of a unique gameplay experience available only in stereo could be the sniper scope common in many games involving guns. The standard way of handling this is to render a circle in the middle of the screen where the player sees the zoomed-in piece of the world. The stereoscopic version could then, when the player zooms with the scope, black out one eye and the player would also loose the stereopsis, giving more realism and immersion.

The future also holds other types of stereoscopic displays called auto-stereoscopic displays. These displays function without glasses, but must render many more than two views [11, 5]. Even though the technique has been around for a long time, it still has not been widely accepted because it has some drawbacks such as ghosting. Making sure that your game engine is adopted for plano-stereoscopic displays is a step towards making them work smoothly with auto-stereoscopic displays, and the guidelines presented in this gem will help you to achieve this as a game engine designer.

Acknowledgements

The author wishes to thank Stefan Seipel, Uppsala University, and Martin Ericsson, UPPMAX, Uppsala University, for contributing their material for this gem.

References

[1] Akiyuki Anzai, Izumi Ohzawa, and Ralph D. Freeman,"Neural Mechanisms for Processing Binocular Information". *Journal of Neurophysiology*, Volume 82, Number 2 (August 1999), pp. 891–908.

[2] Anaglyph. http://en.wikipedia.org/wiki/Anaglyph_image

[3] Chromium. http://chromium.sourceforge.net/

[4] Neil A. Dodgson."Variation and extrema of human interpupillary distance". *Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems XI*, 2004, pp. 36–46.

[5] Neil A. Dodgson, J. R. Moore, and S. R. Lang."Multi-view autostereoscopic 3D display". *IBC (International Broadcasting Convention) 1999*, pp. 497–502.

[6] International Stereoscopic Union. "International Stereoscopic Union: A Glossary of Stereoscopic Terms". http://www.stereoscopy.com/isu/glossary-index.html

[7] Julien C. Flack, Hugh Sanderson, Steven I. Pegg, and Simon Kwok."Optimising 3D image quality and performance for stereoscopic game drivers". *Proceedings of SPIE Stereoscopic Displays and Applications XX*, 2009.

[8] Graham R. Jones, Delman Lee, Nicolas S. Holliman, and David Ezra,"Controlling perceived depth in stereoscopic images", *Proceedings of SPIE Stereoscopic Displays and Virtual Reality Systems VIII*, 2001, pp. 42–53.

[9] Helmut Jorke and Markus Fritz."INFITEC—A New Stereoscopic Visualisation Tool by Wavelength Multiplex Imaging". *Journal of Three Dimensional Images*, Volume 19, Number 3 (September, 2005), pp. 50–56.

[10] Jukka Häkkinen, Monika Pölönen, Jari Takatalo, and Göte Nyman."Simulator sickness in virtual display gaming: a comparison of stereoscopic and non-stereoscopic situations". *ACM International Conference Proceeding Series*, Volume 159 (2006), pp. 227–230.

[11] Ken Perlin, Salvatore Paxia, and Joel S. Kollin."An Autostereoscopic Display", *Proceedings of the 27th Annual Conference on Computer graphics and Interactive Techniques*, 2000, pp. 319–326.

[12] Roy S. Kalawsky.*The Science of Virtual Reality and Virtual Environments*. Addison-Wesley, 1993.

[13] Lenny Lipton.*The CrystalEyes Handbook*. StereoGraphics Corporation, 1991.

[14] Bernard Mendiburu. *3D Movie Making: Stereoscopic Digital Cinema from Script to Screen*. Focal Press, 2009.

[15] Raymond Spottiswoode and Nigel Spottiswoode.*The Theory of Stereoscopic Transmission and its Application to the Motion Picture*. University of California Press, 1953.