

# An Introduction to 3D Computer Graphics, Stereoscopic Image, and Animation in OpenGL and C/C++

Fore June

## Chapter 9 Principles of 3D Stereoscopic Graphics

### 9.1 Anaglyph

We discussed in the previous chapter that retinal disparity enables us to perceive depth. Therefore, if we could present to our left eye a 2D image of an object slightly different from that presented to the right eye, we can deceive our brain to sense depth and think that we are seeing a real 3D object. There are many different ways of creating and viewing stereoscopic 3D images. They all rely on independently presenting different images to the left and right eye. Besides the method of using polarized glasses mentioned in the previous chapter to view stereoscopic images, one can also use LCD shutter glasses to view frame-sequential stereo, where the left and right eye images are presented alternatively and are synchronized with the LCD shutter. Figure 9-1 shows an example of this kind of glasses. LCD shutter glasses are expensive, ranging from around \$60 to \$800 per pair. A system of two pairs of glasses and an emitter range from \$1000 to \$2000.

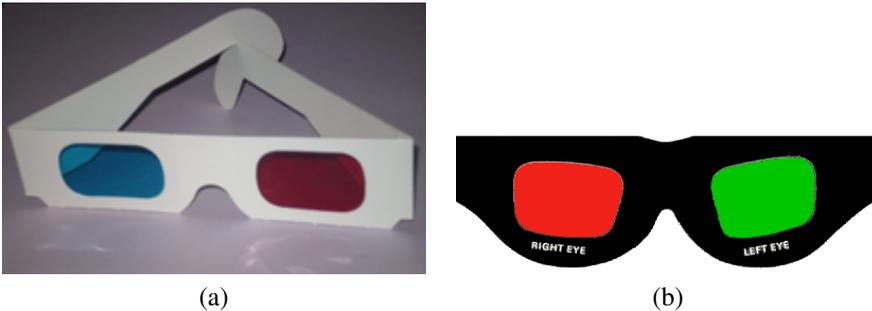


**Figure 9-1** LCD Shutter Glasses

A simple and inexpensive way to experience stereoscopic 3D effects is to use **anaglyphs**. An anaglyph is a stereoscopic picture where the left and right eye images are superimposed, but in different colors. A color filter over each eye only transmits the image component suitable for that eye. For example, the lenses of a pair of anaglyph glasses consists of chromatically opposite colors such as “red and cyan” or “red and blue”. An anaglyph image is made up of two color layers, superimposed, but offset with respect to each other to produce a depth effect. In general the main subject is at the center, while the foreground and background are shifted laterally in opposite directions. The picture contains two differently filtered colored images, one for each eye. When we view superimposed image through a pair of color-coded anaglyph glasses, we will see a 3D object or scene that looks realistic. Anaglyph glasses are usually cheap. Paper anaglyph filter may cost less than 50 cents and better glasses cost about \$3. Figure 9-2a shows a low-cost paper anaglyph filters and Figure 9-2b shows a red-green anaglyph glasses.

Many stereoscopic photographers do not like anaglyphs because of the poor color, ghosting and retinal rivalry they produce. However, most Internet 3D stereoscopic images are presented in anaglyph format as using anaglyphs is the cheapest way to create and project

stereoscopic 3D images using digital cameras and a single digital projector. NASA publishes anaglyphs extensively. Some children's books also often use anaglyphs to show 3D images. Figure 9-3 shows an example of anaglyph where the left eye image is in red and the right eye image is in cyan; they are superimposed to form one single image.



**Figure 9-2** Anaglyph Glasses



**Figure 9-3** An Anaglyph (red on left, cyan on right)

Despite its popularity, there are a few disadvantages of using anaglyphs to present 3D scenes, which include the following.

1. **Ghosting.** The viewers may see a faintly-colored ghost image if some color from the left image gets into the right eye or that from the right gets into the left. Increasing the parallax (stereo depth) separates the two images further and makes the 3D effect more pronounced but it also makes the ghosting effect worse. Ideally, a computer color display would show red, green and blue without any overlap between the colors. However, researchers have shown from spectral analysis that practical screens have “cross-talk” between the colors. For example, a green pixel component may

also emit some red light. If each RGB component does not emit its own colour exclusively, it is not possible for a pair of anaglyph glasses to filter the colors precisely; those monitors will produce ghosting effects. In general LCD screens have less color cross-talk than CRT monitors.

2. **Retinal rivalry.** We explained in the previous chapter that retinal rivalry may occur if image attributes such as brightness or size of the two images are different. Very often, the brightness of the two images of an anaglyph is not the same in each eye, leading to retinal rivalry which gives an unpleasant viewing effect. Some anaglyph glasses have one filter darker than the other, which will routinely cause retinal rivalry.

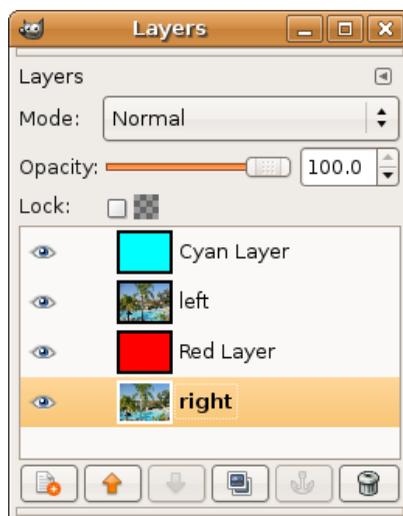
## 9.2 Creating Anaglyphs with GIMP

We can create an anaglyph using the open-source graphics package *GIMP*, the *GNU Image Manipulation Program*, which is a freely distributed piece of software for such tasks as *photo retouching*, *image composition* and *image authoring*. It works on many operating systems, and in many languages. You can download it from its official web site at <http://www.gimp.org/>. The following steps show you how to make an anaglyph using GIMP.

1. Take two photos of an object using a digital camera located at two different positions. The distance between two locations of the camera should be about 70 mm. Note that our cameras are mimicking our eyes, which always focus on something; at the two camera positions, the camera should not view straight ahead separately. Rather, we should point the camera to focus at the same focal point from the two camera locations.
2. Transfer the two images to your computer.
3. Start GIMP and open the right image. Note that objects in the right image are more to the left when overlaid with the left eye image.
4. From the color selector, choose color “red” (ff0000). (**Dialogs > Colors > Enter “ff0000”** in the text box)
5. Create a layer above the right image and fill it with this color (**Layer > New Layer > Select ‘Foreground Color’**).
6. Duplicate this layer (**Layer > Duplicate Layer**) and click on **Colors > Invert**. This will change the color of the duplicated red layer to a cyan color (direct opposite). The resulted RGB color will have  $R = 000$ ,  $G = 255$ , and  $B = 255$ .
7. Select the red layer. Click on **File > Open as Layers** and open the left image. (You can use Ctrl-Tab to cycle through all layers in an image if your window manager doesn’t trap those keys. You can always access the layers through **Dialog > Layer**.)
8. The above step will put the left image just above the red layer and you should now have 4 layers in the following order, from bottom to top: the right image, the red

layer, the left image and the cyan layer as shown in the figure below.

9. You can check the alignment of the image pair by turning off the red and cyan layers and adjusting the opacity of the left image to 50%. Make sure that both images are on the same vertical plane and shift up/down appropriately. When finished checking the alignment, change the opacity of the left image back to 100%. (A layer can be turned off by clicking the eye icon at the left of the Layers dialog box; you can turn it on again by clicking the position again.)
10. Select the cyan layer and change the transfer mode to *screen* by clicking on **Dialogs** > **Layers** > **Mode: Screen**.
11. Turn off the lower layer (the right image) and the red layer by clicking on the small eye icon, which is at the left side of the **Layers** menu.
12. Merge the cyan and left image layers together by clicking on **Image** > **Merge Visible Layers** > **Merge**. You should now see a faint image of your photo. Turn off this new cyan layer by clicking on the eye icon of the Layer dialog box.
13. Turn the red and right image layers back on and change the red layer transfer mode to *screen*. Then merge them in the same way you merged the cyan and left image layers in above. Again you will see a faint red image similar to the cyan image you saw before.
14. Turn on the new merged cyan/left image layer and select it. Change the transfer mode to multiply. You should now see the superimposed 3D image. Put on your anaglyph glasses (red on left, cyan on right) and you should see the realistic 3D effect. The anaglyph shown in Figure 9-3 was created in this way.



Layers of Step 8

### 9.3 Horizontal Parallax

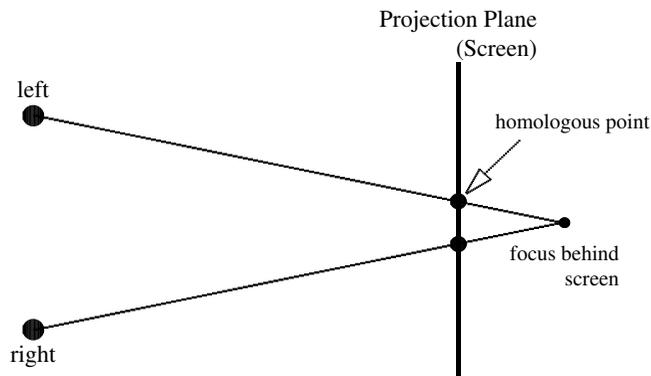
As we have discussed in the previous chapter, when we view an object along two different lines of sight, we will see the object appear at different positions; the displacement of the object (i.e. the difference of the apparent positions of the object) is referred to as **parallax**. Sometimes people define parallax as the angle subtended by the two lines of sight. Therefore, we may measure parallax by the angle or semi-angle of inclination between the two lines of sight. Nearby objects have a larger parallax than distant objects when observed from different positions. As a result we can use parallax to determine depth.

In astronomy, parallax is the apparent difference in position or angle of a body such as the sun or a star, as seen from some point on the earth's surface. Astronomers use the principle of parallax to measure distances to celestial objects, including the distances to the Moon, the Sun, and to stars beyond the Solar System. For example, the Hipparcos satellite took measurements for over 100,000 nearby stars, which provide a basis to pinpoint other distant stars. In this case, *parallax* is the angle or semi-angle of inclination between two lines of sight to the star.

When we look at a stereoscopic picture, our brain extracts and computes the size of the disparities to access the distance of the objects. Figure 9-4 shows the projections of a single model point on a plane for two different eye positions. Our left and right eyes see two different projections of the model point; the projection for the left eye is on the left and the projection for the right eye is on the right. The two projected points are called **homologous points**.

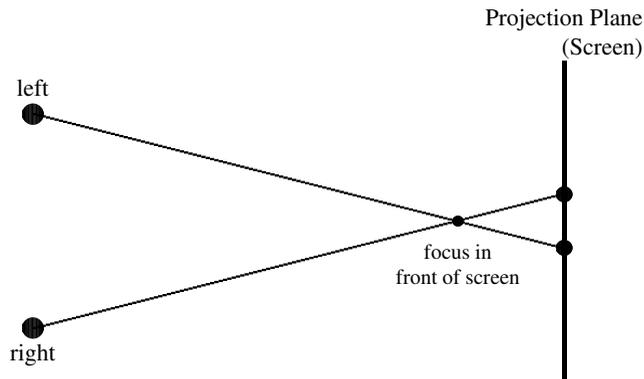
In general, two projected points (one for the left eye and one for the right eye) are **homologous** if they are projected from a common model point. The horizontal distance between the two homologous points is called the **horizontal parallax**. We may also define horizontal parallax as the angle subtended by the horizontal distance between the two homologous points. Similarly, the vertical distance between two homologous points is called the **vertical parallax**. Since our eyes are horizontal, when we talk about parallax, we refer to horizontal parallax unless otherwise stated.

In Figure 9-4, since the projections are on the same side as the respective eyes, we call the parallax a positive parallax. Note that the maximum positive parallax occurs when the object (model point) is at infinity; at this point the horizontal parallax is equal to the interocular (i.o.) distance, which is the separation between the two eyes.



**Figure 9-4** Positive Parallax with two homologous points on the plane

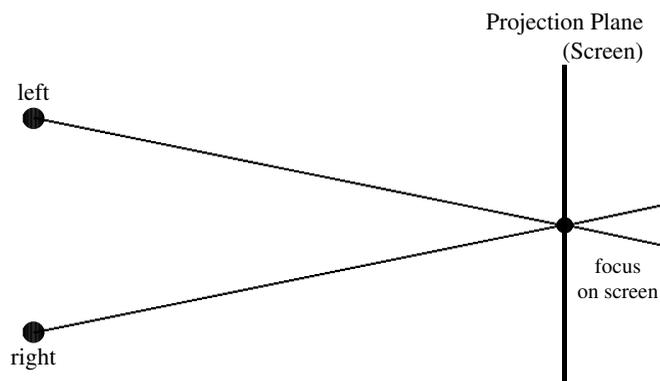
Figure 9-5 shows the case of **negative parallax**, where the projection for the left eye is on the right and the projection for the right eye is on the left. In this case, the model point is in front of the projection plane. Obviously, when the model point is half way between the projection plane and the center of the eyes, the interocular distance is equal to the negative horizontal parallax. The negative horizontal parallax increases to infinity as the model point moves closer to the viewer, and it goes to zero as the model point moves towards the projection plane.



**Figure 9-5 Negative Parallax**

Figure 9-6 shows the case of zero parallax, where objects are at the projection plane. In this case, the projections for the left eye and the right eye converge to the same location at which the model point is located.

In summary, objects associated with positive parallax are seen to be behind the projection plane; those with negative parallax are seen to be in front of the plane and those with zero parallax are at the plane.



**Figure 9-6 Zero Parallax**

To create good stereo pairs, it is crucial to achieve appropriate horizontal parallax. On the other hand, we should avoid or at least strictly limit vertical parallax. In general, we will have difficulties in fusing and holding images together in the presence of vertical parallax. One way vertical parallax will be introduced is that we use different projection planes for the left eye and the right eye. For example, the toe-in method that some people use to generate stereo pairs is simple and quick. However, it also introduces vertical parallax that

could make viewers uncomfortable when viewing the stereo images with glasses. We will discuss this method in more detail in the next chapter.

We can use shear transformation to implement positive and negative horizontal parallax. Actually, shear transformations mechanisms are included in the left and right eye projection matrices.

We can measure parallax in inches or degrees. The parallax distance between homologous points would be in inches and the parallax angle  $\theta$  describing the angle made by the homologous points and the viewer would be in degrees. So if we describe parallax in inches, we refer to the distance and if we describe parallax in degrees, we refer to the angle. The parallax distance  $p$  in inches and the angle  $\theta$  in degrees is related by the formula,

$$p = 2d \times \tan \frac{\theta}{2} \quad (9.1)$$

where  $d$  is the distance from the viewer to the projection plane (stereo plane). Lastly, if the stereo images are projected on a computer screen, we may describe parallax in terms of pixels.

In practice, *when we create stereo images on a computer screen, what parallax value should we use?* There are a few rules we may follow to generate good stereo pairs. Ideally, maximum positive parallax should be equal to the typical human eye separation. In this way we will not force our eyes into divergence, which is what will happen if the positive parallax is greater than our eye separation. If the positive parallax is equal to our eye separation, the object we view will appear to be at infinity as we can tell from Figure 9-4. Maximum negative parallax should be also equal to the eye separation, which occurs when the object being viewed is half way between the screen and our eyes. Actually, negative parallax is more flexible because we can cope with larger changes in negative parallax than with positive. However, measurements of parallax depend on the screen size and viewing distance as shown in Equation (9.1) above. This implies that stereo images that work well on a small screen may not work well when viewed on a large screen. The following calculation gives a guideline for estimating the maximum positive parallax  $p_{max}$  in pixels for a given screen size:

$$p_{max} = \frac{i.o.}{w} \times N \quad (9.2)$$

where  $i.o.$  is the human interocular distance (eye separation), which is about 2.5 inches,  $w$  is the screen width in inches, and  $N$  is the number of horizontal pixels. For example, a small 24-inch wide HD monitor would have a maximum positive parallax of

$$p_{max} = \frac{2.5}{24} \times 1920 = 200 \text{ (pixels)}$$

In other words, the interocular distance covers a range of 200 pixels. If the screen is 30 feet wide but with the same number of horizontal pixels, then the maximum parallax is

$$p_{max} = \frac{2.5}{360} \times 1920 = 13.3 \text{ (pixels)}$$

## 9.4 Calculating Stereo Pairs

### 9.4.1 Perspective Projection

We discussed in Chapter 4 the concept of perspective projection, and in particular perspective projection transformations in Section 4.2.2. This projection is important in creating

stereoscopic images. Figure 9-7 below shows a viewing frustum; any object inside the frustum will be projected into the near-plane.

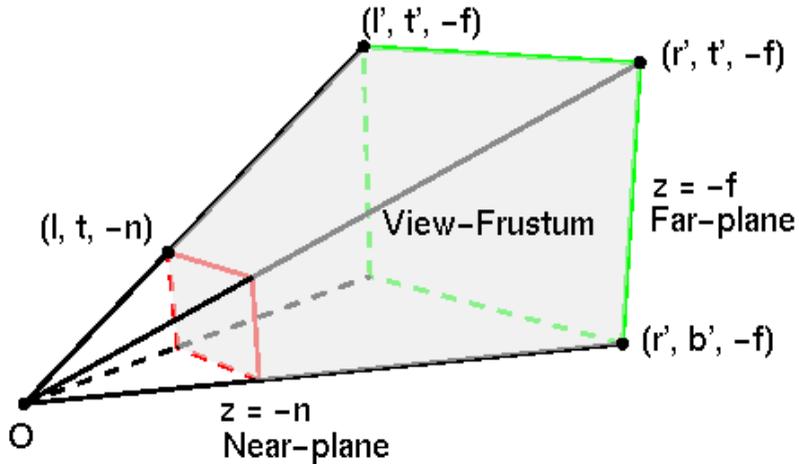


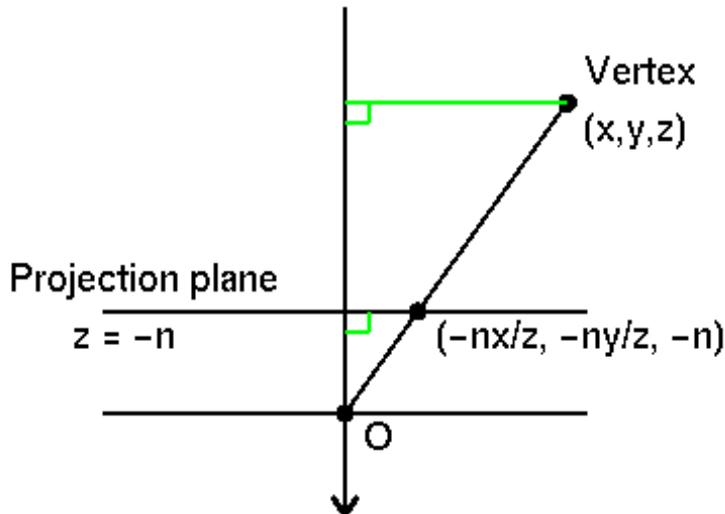
Figure 9-7 Perspective Projection

In Figure 9-7, the eye (viewing point) is at the origin  $O$ , looking towards the negative  $z$  direction; the near-plane is at  $z = -n$  and the far-plane is at  $z = -f$ ; more details,

$$\begin{array}{ll}
 n = & \text{near distance} \\
 l = & \text{left} \\
 b = & \text{bottom} \\
 O = & (0, 0, 0)
 \end{array}
 \qquad
 \begin{array}{ll}
 f = & \text{far distance} \\
 r = & \text{right} \\
 t = & \text{top}
 \end{array}
 \qquad
 (9.3)$$

Any point  $(x, y, z)$  inside the frustum is projected into a point  $(x', y', z')$  on the screen at  $z = -n$ . Obviously  $z'$  is always equal to  $-n$ . The values of  $x'$  and  $y'$  can be easily calculated using similar triangles as shown in Figure 9-8. The calculated values are shown in Equation (9.4) below.

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \frac{-nx}{z} \\ \frac{-ny}{z} \\ -n \end{pmatrix}
 \qquad
 (9.4)$$



**Figure 9-8** Perspective Projection Values

#### 9.4.2 Rotation or Toe-in Method

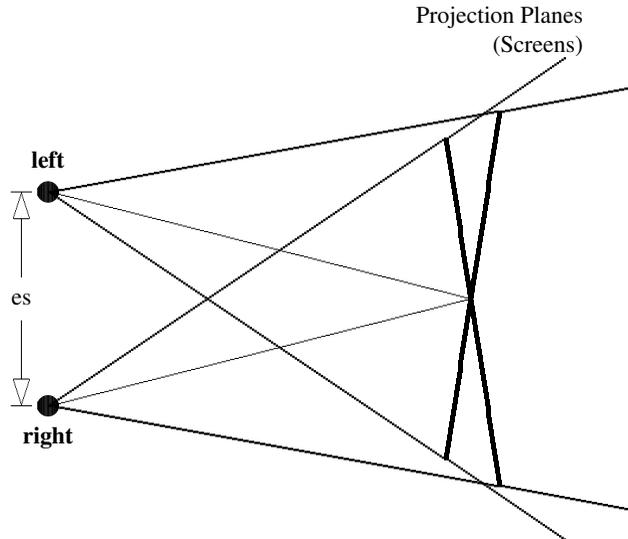
The rotation or Toe-in method is a quick and simple way to create graphics stereo pairs. However, this is also an incorrect method. In this method both the left and right eyes point towards a single focal point as shown in Figure 9-9. Different projection planes are used for the two eyes. If the eye separation is known, the center between the two eyes is specified, and the focal point is defined, then other parameters required to define the viewing volumes can be calculated. For instance, consider that the eyes are in the  $x$ - $y$  plane looking along the negative  $z$  direction and the mid-point of the two eyes is at the origin  $O$ , and suppose

$$\begin{aligned}
 O &= (0, 0, 0) \\
 f &= \text{focal length} \\
 e &= \text{eye separation}
 \end{aligned}
 \tag{9.5}$$

Then we can calculate the eye positions and their distances to the projection plane. For example,

$$\begin{aligned}
 \text{Left eye position} & \quad E_l = \left(-\frac{e}{2}, 0, 0\right) \\
 \text{Distance from left eye to projection plane} & \quad d = \sqrt{f^2 + (e/2)^2}
 \end{aligned}
 \tag{9.6}$$

We can obtain similar equations for the right eye. Once the eye position and its distance to the projection plane (which is the near plane in this case) are known, we can specify the viewing volume and calculate its projected values according to Equation (9.4). OpenGL users usually use the function **gluPerspective** to do the projection.



**Figure 9-9** Toe-in Method ( $es$  = eye separation)

Though the Toe-in method is simple and gives workable stereo pairs, it is not a friendly way to generate stereo pairs. It causes vertical parallax and distorts the stereo plane. The vertical parallax is most noticeable when objects are placed at the outer field of view.

### 9.4.3 Two-Center Projection (Off-axis Projection)

The two-center or off-axis projection is sometimes known as **parallel axis asymmetric frustum perspective projection**. In this case, the view vector for each eye position remains parallel as shown in Figure 9-10. Instead of projecting objects into two different planes, this method projects the objects into a unique single plane and will not cause vertical parallax. Usually, OpenGL users use the **glFrustum** function to describe the perspective projections of this method.

As discussed in Chapter 4, for any given point  $P(x_0, y_0, z_0)$  inside the viewing frustum, it is not difficult to find where it will be projected into the projection plane. For instance, consider the setup of the eye positions shown in Figure 9-11 with index  $l$  denoting left and  $r$  denoting right. Suppose the camera is looking along the negative direction, eye separation (i.o. distance) is  $e$ , the middle of the two eye positions is at  $(0, 0, 0)$  and the projection is at  $z = -d$  as shown in the figure. We can use parametric equations to determine a point  $P = (x, y, z)$  on the line joining the points  $P_0 = (x_0, y_0, z_0)$  given that the left eye position is  $E_l = (e/2, 0, 0)$ :

$$P = P_0 + t(E_l - P_0) \quad 0 \leq t \leq 1 \quad (9.7)$$

Equation (9.7) actually consists of three equations:

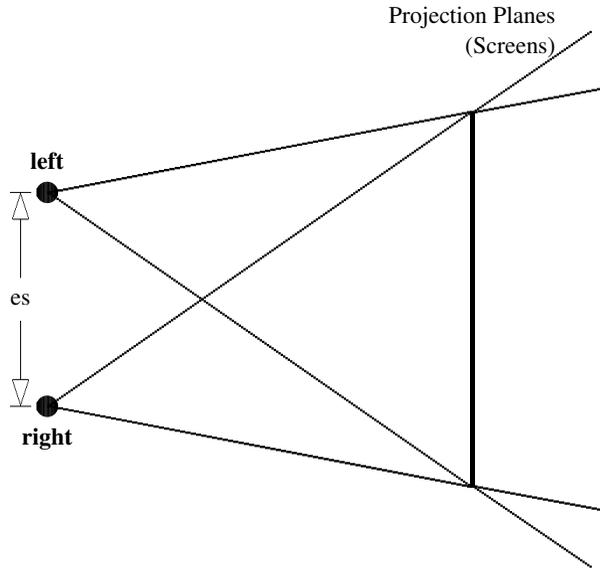
$$\begin{aligned} x &= x_0 + t(e/2 - x_0) \\ y &= y_0 + t(0 - y_0) \\ z &= z_0 + t(0 - z_0) \end{aligned} \quad (9.8)$$

At the projection plane,  $z = -d$ . So we can solve for  $t$ :

$$-d = z_0 + t(0 - z_0) \quad (9.9)$$

which implies

$$t = \frac{z_0 + d}{z_0} \quad (9.10)$$



**Figure 9-10** Two-center Projection ( $es$  = eye separation)

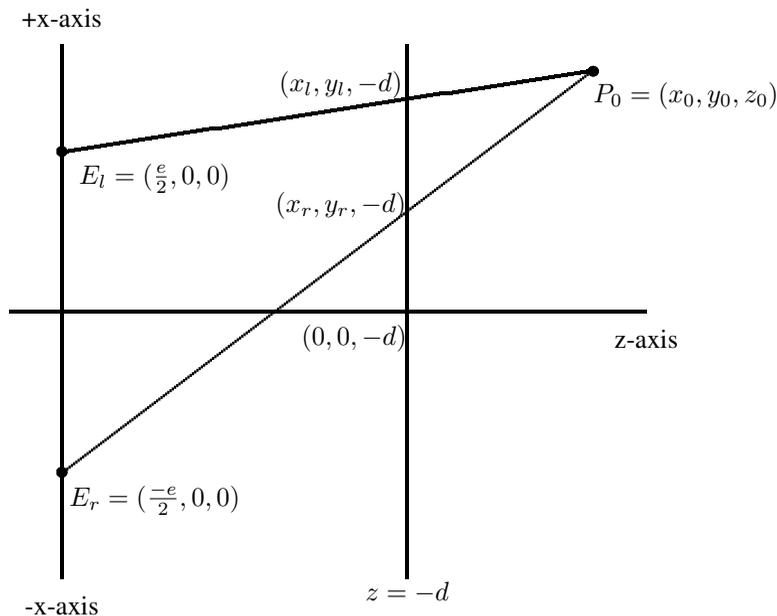
Knowing  $t$ , we obtain the  $x, y$  coordinates at the viewing (projection) plane, where  $x = x_l, y = y_l$ :

$$\begin{aligned} x_l &= -\frac{d \times x_0}{z_0} + \frac{e \times d}{2z_0} + \frac{e}{2} \\ y_l &= -\frac{d \times y_0}{z_0} \end{aligned} \quad (9.11)$$

Similarly, we can solve for the right eye position:

$$\begin{aligned} x_r &= -\frac{d \times x_0}{z_0} + \frac{e \times d}{2z_0} - \frac{e}{2} \\ y_r &= -\frac{d \times y_0}{z_0} \end{aligned} \quad (9.12)$$

Equations of (9.11) and (9.12) demonstrate how the projected points on the view plane can be calculated for the two different eye positions, left and right. However, in reality we do not need to carry out the calculations in our applications. In OpenGL, we can simply implement this shift by translation in the modelview matrix. We can recalculate the  $e$  term for each frame in an animation based on a set of parallax angles. An additional shift (yielding both positive and negative parallax) is created by altering the projection matrix so that the perspective view volume is asymmetrical.



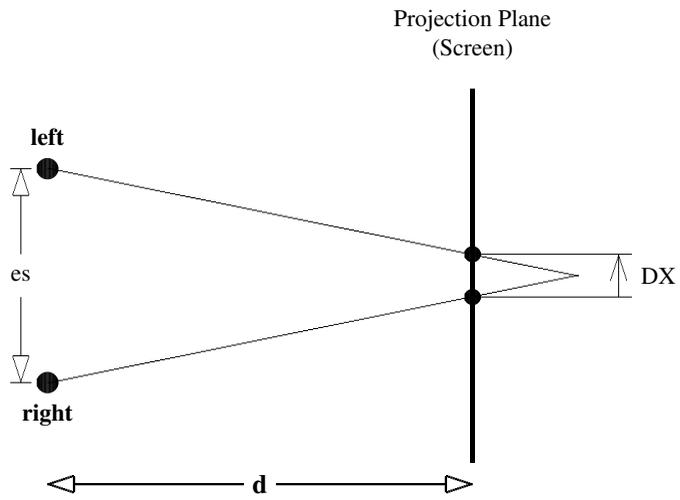
**Figure 9-11** Two-center Projection Calculations

#### 9.4.4 Stereo Effect

As we mentioned before, objects that lie in front of the projection plane will appear to be in front of the computer screen and objects that lie behind the projection plane will appear to be “into” the screen. It is generally easier to view stereo pairs of objects that recede into the screen (positive parallax). To achieve this, we would place the focal point closer to the camera than the objects of interest. A common measure of the **stereo effect** is the parallax angle which is defined as

$$\theta = 2 \tan^{-1} \frac{DX}{2d} \quad (9.12)$$

where  $DX$  is the horizontal separation of a projected point on the screen for the two eye positions and  $d$  is the distance of the eyes from the projection plane as shown in Figure 9-12. For easy fusing by the majority of people, we should make the absolute value of  $\theta$  be less than 1.5 degrees for all points in the scene. Note that  $\theta$  is positive for points behind the scene and negative for points in front of the screen. It is a fairly common practice to restrict the negative value of  $\theta$  to some value closer to zero since negative parallax is more difficult to fuse especially when model objects overlap with the boundary of the projection plane.

**Figure 9-12** Stereo Effect Measurement



Other books by the same author

# Windows Fan, Linux Fan

by *Fore June*

*Windows Fan, Linux Fan* describes a true story about a spiritual battle between a Linux fan and a Windows fan. You can learn from the successful fan to become a successful Internet Service Provider ( ISP ) and create your own wealth. See <http://www.forejune.com/>

Second Edition, 2002.

ISBN: 0-595-26355-0 Price: \$6.86

# An Introduction to Digital Video Data Compression in Java

by *Fore June*

The book describes the the principles of digital video data compression techniques and its implementations in java. Topics covered include RBG-YCbCr conversion, macroblocks, DCT and IDCT, integer arithmetic, quantization, reorder, run-level encoding, entropy encoding, motion estimation, motion compensation and hybrid coding. See

<http://www.forejune.com/>

January 2011

ISBN-10: 1456570870

ISBN-13: 978-1456570873

---

# An Introduction to Video Compression in C/C++

by *Fore June*

The book describes the the principles of digital video data compression techniques and its implementations in C/C++. Topics covered include RBG-YCbCr conversion, macroblocks, DCT and IDCT, integer arithmetic, quantization, reorder, run-level encoding, entropy encoding, motion estimation, motion compensation and hybrid coding.

January 2010

ISBN: 9781451522273