



ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

A Survey on Shadowing in OpenGL

Nikhil Bhirud, Rohit Gadiya, Mayuresh Sharma, Ashutosh Yerdoni, Prof. S.V. Chobe

B.E, Dept. of I.T, D.Y.P.I.E.T, Pimpri, Pune, Maharashtra, India

Abstract: This survey paper describes shadowing using OpenGL (Open Graphics Library), which is very important to give realistic effect to the object. Shadowing is useful for multiple reasons: they help understand placement of the objects relative to each other in a 3D scene by providing visual hints, and, they also improve image realism and allow the creation of complex lighting ambiances. This paper gives the overview of few of the shadowing techniques available. It gives the procedures to obtain the shadows of the objects.

KEYWORDS: OpenGL, shadowing, soft shadows, hard shadows, Z-buffer, Stencil buffer.

I. INTRODUCTION

Open Graphics Library (OpenGL) is a cross-platform, cross-language application programming interface (API) for rendering 2D and 3D vector graphics. Hardware-accelerated rendering can be achieved by interacting directly with graphics processing unit with the help of OpenGL API. The graphics and the hardware interface consist of more than 100 graphics functions, with transformation, modeling, light processing, texture mapping, motion blur function object etc. These functions can be used by developers to create 3D models and 3D real time interaction. OpenGL itself is a renderer. It is a low level graphics library and is very useful to directly build the scene in the system's frame buffer with help of OpenGL we render the 3D image on the flat screen. What is rendering? Rendering is the act of taking a geometric description of a three-dimensional object and turning it into an image of that object onscreen.

Shadows are used to provide realistic touch to any image or scene created. The visual phenomenon of a shadow occurs when a shadow caster (object) blocks most of the photons from the light source from reaching a shadow receiver. Shadows gives the realism to image, give the relative geometry between the objects in the image. OpenGL functions are available to produce shadow effects.

In this paper, we first have described the basic terminology used for shadowing. Then the three shadowing techniques- Planner projection method, Z-Buffer and Stencil Buffer are explained. We have concluded the paper with the comments on each of these techniques along with an example which signifies the importance of shadowing in the graphics.

II. RELATED WORK

Space where the light from light source is blocked by an opaque object is termed as Shadow. Shadow gives more realism to objects in the scene Shadows are useful for a variety of reasons: they help understand relative object placement in a 3D scene by providing visual cues. They dramatically improve image realism and allow the creation of complex lighting ambiances.

A common physical shadow consist of following terms:

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

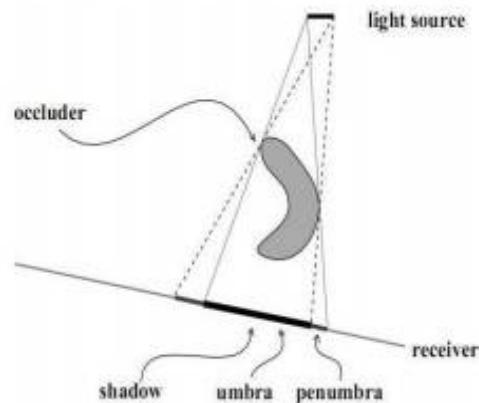


Figure 1.Shadow Terminology

- 1) **Occluder:** Occluder is nothing but an object which shadow is to be obtained. It blocks the light coming from the light source.
- 2) **Light Source:** Light Source emits light, which when blocked completely or partially produces shadow. Light source may be an ideal point light source or volume/area light sources coming from multiple positions.
- 3) **Receiver:** The plain on which the shadow is created is termed as receiver. It may or may not be an even surface.
- 4) **Shadow:** The area on receiver which is blocked by occluder is shadow. A shadow can be Hard Shadow or Soft Shadow.
 - a) **Hard Shadow** is created by an ideal point light source. Hard shadows consist of crisp and sharp edges which make it unrealistic.

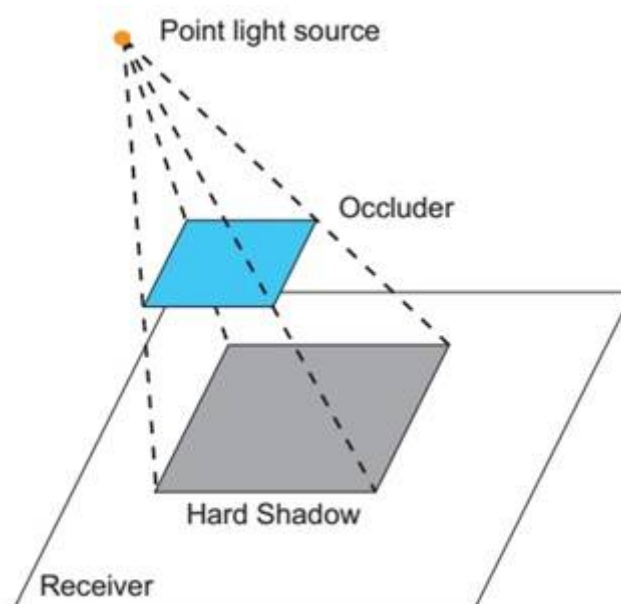


Figure 2.Geometry Of Hard Shadow

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

- b) **Soft Shadow** is created by multiple light sources. Direct and indirect lightning is considered for soft shadowing. Soft shadows give more realism to the image. This is the reason more research is done in enhancing soft shadowing technique.

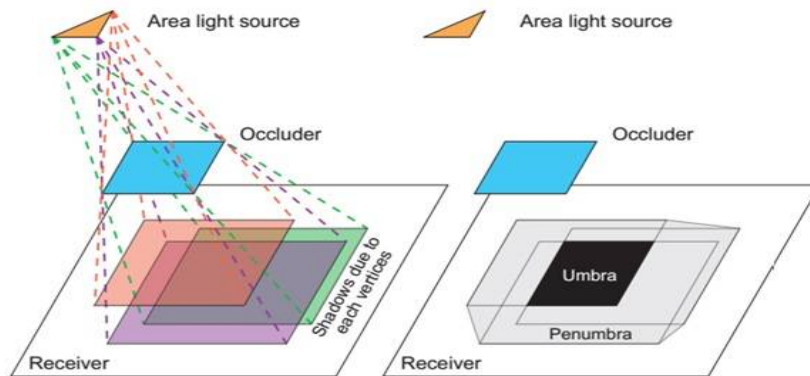


Figure 3. Geometry of Soft Shadow

- 5) **Umbra**: Point on the receiver which completely blocked completely by an occluder is umbra. That point is not visible from any point of light source.
- 6) **Penumbra**: Point on receiver which is blocked by an occluder but can be seen from point of light source is penumbra.

Many techniques are used in process of implementation, where shadow rendering algorithms differentiated like planner projection method, scan-line methods, shadow volume algorithms, shadow mapping, ray-tracing algorithms and Z-buffer, stencil buffers.

1. Planner Projection Method:

The projection shadow algorithms can be quickly computed with a simple idea, but it is limited only to casting shadows onto planar surfaces rather than general surface on earth.

Another method from Gooch et al. (1999) is to approximate the soft-shadows in the scene with a spherical light source. The receiving plane's location is moved up and down in order to average the projections. This shadows created are concentric, which generally looks better and so requires fewer samples. But the shadow rendered in this method is usually larger than the object so that it will induce distortion.

1.1 Implementing Hard Shadows:

In projection shadows algorithm, objects are directly projected onto the receiver directly.

Any shadow receiver plane P can be expressed by the equation: $\mathbf{n} \cdot \mathbf{x} + \mathbf{d} = 0$,

Where n is normal of the plane and x is a point on the plane

Given $\mathbf{n}(\mathbf{a}, \mathbf{b}, \mathbf{c})$ and $\mathbf{x}(\mathbf{P}_x, \mathbf{P}_y, \mathbf{P}_z)$, the Plane P can be expressed as: $\mathbf{aP}_x + \mathbf{bP}_y + \mathbf{cP}_z = 0$.

Let source $\mathbf{s} = (\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w})$, $\mathbf{dot} = \mathbf{ax} + \mathbf{by} + \mathbf{cz} + \mathbf{dw}$,

then projection matrix M is:

$$M = \begin{pmatrix} \mathbf{dot} - \mathbf{ax} & -\mathbf{ay} & -\mathbf{az} & -\mathbf{aw} \\ -\mathbf{bx} & \mathbf{dot} - \mathbf{by} & -\mathbf{bz} & -\mathbf{bw} \\ -\mathbf{cx} & -\mathbf{cy} & \mathbf{dot} - \mathbf{cz} & -\mathbf{cw} \\ -\mathbf{dx} & -\mathbf{dy} & -\mathbf{dz} & \mathbf{dot} - \mathbf{dw} \end{pmatrix}$$

1.2 Implementing Soft Shadows:

For soft shadows, firstly random sample points on the light source needs to be taken. An image is rendered for each of these points. In order to blend these images, enable the GL_BLEND function of OpenGL. The average of these



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

images should be taken which is the image with soft shadows. Users can control resolution of the shadow through Keyboard function of OpenGL to increase or decrease the number of samples. Increasing the number of sample points according to actual requirements can get more realistic shadows.

For Shadow rendering, simply apply this projection matrix M determined by light position and receiver plane to objects that cast shadows and render these projected objects with dark color and no illuminations. Real-time rendering of soft-shadows is still a considerable task on latest GPUs.

2. Z-Buffer:

Z-buffer is also known as depth buffer, it is used to determine whether objects or part of objects lie in the scope of light source and which of the objects will be in shadow. Z-buffer resides in main memory. In every scene there are some actors (or objects) that have same X- coordinates and Y-coordinates, in some part of the object but have different Z-coordinates. These objects create shadow on each other. Z buffer stores depth value and recognize these kinds of objects.

Z-buffer is stored in GPU's memory. The Z-buffer is an enhancement to this mass-memory approach to computer graphics which resolves the visible surfaces in a scene by storing depth (Z) values at each point in the picture. As objects are rendered, their Z values are compared at each point with the stored Z values to determine visibility. Since this determination requires only that a measure exist which orders the surfaces to be displayed, it is not exaggerated statement to say that the Z-buffer algorithm provides a unique and distinct solution to all scenes for which visible surfaces can be computed. The objects to be rendered on to the display are not to be sorted, so Z-buffer algorithm can handle fairly complex scenes. As stated above Z-buffer executes radix sort in X and Y co-ordinates and simple indexing for Z co-ordinate.

3. Stencil Buffer:

Stencil buffer is a bits memory allocated in frame buffer in main memory which is used to restrict rendering within the window. Stencil buffer counts the time of entering and leaving of the shadows, which is helpful in knowing whether the object is in shadow and which parts of the object are lit. Similar to Z-buffer, stencil buffer is also located in the GPU's video memory.

Stencil buffer gives extra control over the fragments of objects which are to be drawn and which should not be drawn.

To setup double-buffered RGB color buffer with stencil (Using glut.):

```
glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE | GLUT_STENCIL);
```

The stencil hardware is relatively faster on modern hardware-accelerated OpenGL implementations.

For example, we can turn on or turn off stencil test with following line of code:

```
glEnable(GL_STENCIL_TEST);  
glDisable(GL_STENCIL_TEST);
```

III. CONCLUSION

In this paper, few of the available shadowing techniques have been studied described in brief. Planar projection algorithm is simple to implement. But it has its own limitations, it is confined to only the scenes with the planar receivers. Z-buffer and stencil buffer keeps the track of depth value of every pixel on the screen. Advantages of Z-buffer is that it can be easily applied to fair amount of complex scenes. Stencil buffer gives you extra control over displaying of objects based on values of stencil buffer.

From the below given images we can say that shadow gives object more realistic effect.



ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

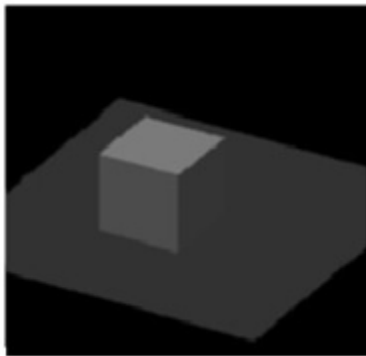


Figure 4. Lighted Cube without shadow

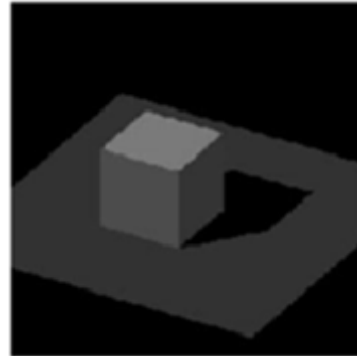


Figure 5. Lighted Cube with shadow

REFERENCES

- [1] Nan Liu, Mingyong Pang, Rendering soft shadows with OpenGL based on planar projected Method. In: Proceedings of IC-NIDC2010
- [2] B. Gooch, P. J. Sloan, et al. Interactive Technical Illustration [A]. In: Proceedings 1999 Symposium on Interactive 3D Graphics [C], 1999, 31-38
- [3] L. Williams, "Casting Curved Shadows on Curved Surfaces", *Computer Graphics (SIGGRAPH '78 Proceedings)*, 1978, 12(3): 270-274.
- [4] Florian Kirsch, Jurgen Dollner, "Real-Time Soft Shadows Using a Single Light Sample".
- [5] Biserica Trpcevska, Vladimir Nasteski, In: "The 7th International Conference for Informatics and Information Technology (CIIT 2010)".
- [6] REAL-TIME SHADOW USING A COMBINATION OF STENCIL AND THE Z-BUFFER, Hoshang Kolivand1*, Mohd Shahrizal Sunar2, Normal Mat Jusoh3, Folorunso Olufemi A.4 1,2,3,4UTM ViCubelab, Department of Computer Graphics and Multimedia, Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, 81310 Skudai Johor Malaysia 1Islamic Azad University, Firozkoh Branch, Iran 1shahinkey@yahoo.com, 2shah@cs.utm.my, 3normal@utm.my, 4femi_folorunso@yahoo.com
- [7] Richard S. Wright, Jr. Benjamin Lipchak Nicholas Haemel, , *OpenGL Superbible : comprehensive tutorial and reference*, 4th Edition, 2007, ISBN-13: 978-0-321-49882-3
- [8] Dave Shreiner Graham Sellers John Kessenich Bill Licea-Kane, Addison.Wesley. *OpenGL Programming Guide. 8th. Edition. Mar. 2013. ISBN. 0321773039*

BIOGRAPHY

Nikhil Bhirud, Rohit Gadiya, Mayuresh Sharma, Ashutosh Yerdoni are pursuing Bachelor of Engineering in Information Technology Department, Dr. D.Y. Patil Institute Of Engineering And Technology, Savitribai Phule Pune University. We are currently working on the project named "Scene Generation with real time rendering using OpenGL", guided by **Prof. S. V. Chobe**.