Comparison of Lighting Effects for the Visualization of an Object Using Direct3D

Pyae Thida
Computer University (Mandalay)
pyithida@gmail.com

Abstract

This paper seeks to examine the advantages of three dimensional (3D) computer graphics on the personal computers. 3D scene, including the objects named TableDisk, TableLegs and even the landscape with the sky, is designed using a low-level application programming of Microsoft Direct3D and Visual C++. This 3D scene visualization is designed to enable world-class game and interactive three dimensional (3D) graphics application on a computer running Microsoft Windows. Lighting is applied on the 3D objects of the scene to produce the various visualization effects. Nature of the light in actual world is simulated by using the three components of the light: ambient, diffuse and specular. Effects of these lighting components are then calculated using the lighting models and then results are analyzed. Comparison is done with the calculation results via the visualization effects. The main purpose is to promote the software development for high performance 3D applications comparing the lighting effects applied on the objects of a scene.

Keywords: Direct3D, light sources, ambient, diffuse, specular.

1. Introduction

3D computer graphics can exist as a field only due to the existence of actual computer hardware. Today real time rendered 3D is mostly used in computer and video games. The two most commonly used graphics standards for real time rendered 3D that exist today are OpenGL and Direct3D. Direct3D is part of DirectX software development kit which was developed by Microsoft and is available for Microsoft's 32 bit operating systems. An application using Direct3D initializes the interface objects, sets the state of the modules, such as the transformation matrices, light-sources and description of materials, by calling specific functions from Direct3D API, and constructs the execute buffers or display lists which contain geometric information and commands describing transformations and processes which must be performed.

Light is a very important effect in visualization of the computer 3D graphic games. It describes the visual effects of the 3D objects in a scene. Computer generated 3D images will be flat, hard to orient and hard to believe-in without the lights. Lighting effects are applied onto the 3D objects contained in a scene using the lighting components of the nature such as ambient, diffuse and specular.

2. Related Work

When developing applications in Direct3D today, developers can choose between using the fixed function pipeline and the programmable pipeline. Markus Holmaker and Magnus Woxblom proved that the fixed function pipeline is faster than the programmable pipeline in all their tests [6]. In fixed function pipeline, lighting equations are applied by Michal Valient in his project using the vertex and pixel shading techniques [5]. In this paper, the lighting equations are also applied using the vertex shading technique in fixed function pipeline. But there is some little difference in specular component model by which results are computationally and visually compared and analyzed.

3. System Design

There are two main parts to design this lighting system. First is to consider the design of 3D scene objects and next is to apply the lighting model of the nature.

In designing a 3D scene objects, first stage is to construct the required 3D objects on the local system using the algebraic equations. Second is to transform this local system into the world system. And then viewing transformations are carried out to see the 3D scene objects on the screen of the computer. Matrix equations are necessary to know in transforming from one to another system. Finally, input points or vertices of the considered 3D objects are projected onto the screen. And input color and bitmap image as texture for vertex is used in creating the object to be good. When 3D objects are added by lighting effects, the scene is more realistic.

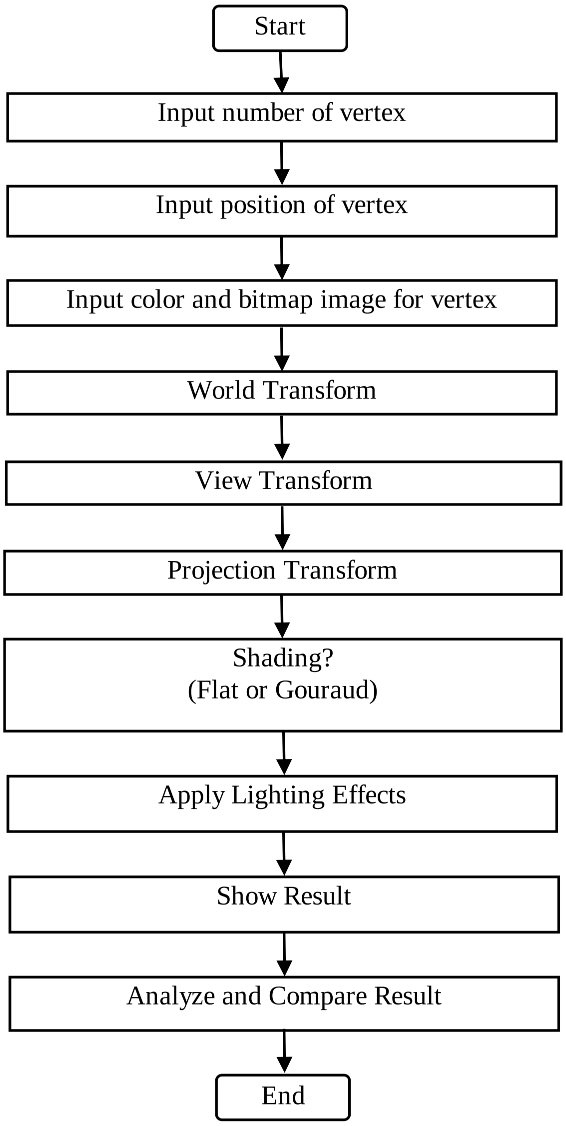


Figure 1. Overview of the System

4. Transformation

Transformation is a function which maps points from one coordinate system to another. Movements of an object in the virtual world and its projections onto the screen are computed using various transformations of coordinates such as translation, rotation and scaling.

4.1. World Transform

The world transformation changes coordinates from model space, where vertices are defined relative to a model's local origin, to world space, where vertices are defined relative to an origin which is common to all the objects in a scene. The world transformation can include any combination of translation, rotation, and scaling.

4.1.1. Translation. Translation transformation moves the points of an object to a new specified position. Translation matrix is constructed as follow:

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}$$

4.1.2. Scaling. It is defined as proportional expansion or contraction of the distances between points of an object. It is performed by multiplying the coordinates by the constant. Scaling matrix is constructed as follow:

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4.1.3. Rotation. The following transformation rotates the point (x, y, z) around the x-axis, producing a new point (x', y', z').

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & \sin\alpha & 0 \\ 0 & -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The following transformation rotates the point around the y-axis.

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1]\begin{bmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 1 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The following transformation rotates the point around the z-axis.

$$[x' \quad y' \quad z' \quad 1] = [x \quad y \quad z \quad 1] \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4.2. View Transform

The view transformation locates the viewer in world space, transforming vertices into camera space. In camera space, the camera, or viewer, is at the origin, looking in the positive z-direction. The view matrix relocates the objects in the world around a camera's position: the origin of camera space and orientation.

$$M_v = \begin{bmatrix} t.x & s.x & n.x & 0 \\ t.y & s.y & n.y & 0 \\ t.z & s.z & n.z & 0 \\ -\bar{C} \circ \bar{t} & -\bar{C} \circ \bar{s} & -\bar{C} \circ \bar{n} & 1 \end{bmatrix}$$

In this matrix, t, s, and n are the right, up, and view-direction vectors and C is the camera's world space position. This matrix contains all the elements needed to translate and rotate vertices from world space to view space.

4.3. Projection Transform

The process of mapping 3D world coordinates into 2D screen coordinates is called a projection. Perspective projection creates an image in which sizes of the projections of objects depend on the objects' distance to the viewer.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/D \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

In this matrix, the distance between the camera required by the projection transformation and the origin of the space defined by the viewing transformation is defined as D.

5. Light Model Equation

The following light model equations are used in this system.

(a) Ambient
(b) Diffuse
(c) Specular

5.1. Ambient Component

The ambient lighting simulates indirect light contribution (for example a light that bounced off a wall and then reached the object) and emission of a light by the surface itself.

$$A = m_a * l_a + m_e \qquad (5.1)$$

where,

A - ambient component of lighting
m_a - ambient coefficient of material
l_a - color intensity of ambient light
m_e - emissive coefficient of material

5.2. Diffuse Component

Diffuse lighting is light that more or less falls directly onto a surface.

$$D = (\bar{N} \circ \bar{L}_{dir}) * m_d * l_d \qquad (5.2)$$

where,

D - diffuse component of lighting
m_d - diffuse coefficient of material
l_d - color intensity of diffuse light
L_{dir} - direction vector from a surface point to light source
N - normal vector of surface point

5.3. Specular Component

Specular lighting is often refered to as specular highlight or, even more commonly, reflection. Specular part of lighting is used to simulate shiny surfaces with the highlights.

$$S = (\bar{N} \circ \bar{H})^{shi} * m_s * l_s \qquad (5.3)$$

where,

S - specular component of lighting
m_s - specular coefficient of material
l_s - color intensity of specular light
H - halfway vector
N - normal vector of surface point
shi - specular reflection power of the material

6. Types of Light Sources and their Effects

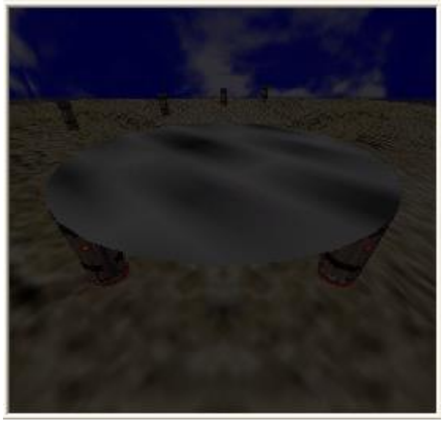
In order to produce the three types of light components, following light sources are used in this system.

(a) Ambient light source

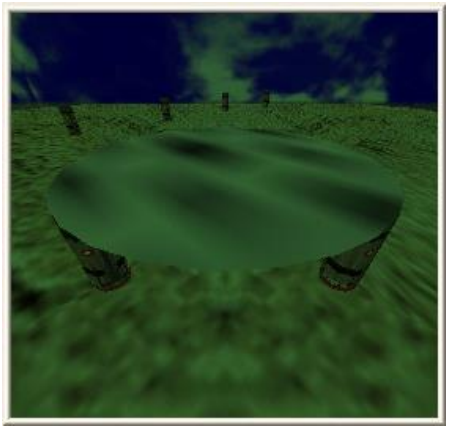
(b) Directional light source
(c) Point light source

6.1. Ambient Light Source Effect

Ambient light source has only the ambient component. There is no concentrated source of light and the illumination is equal from any direction maintaining constant levels throughout the world as shown below.



(a)



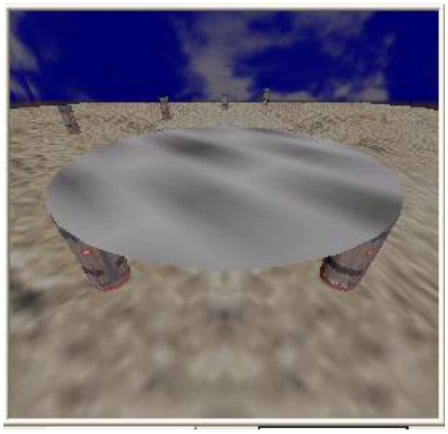
(b)

Figure 2. Ambient Light Source Effects.

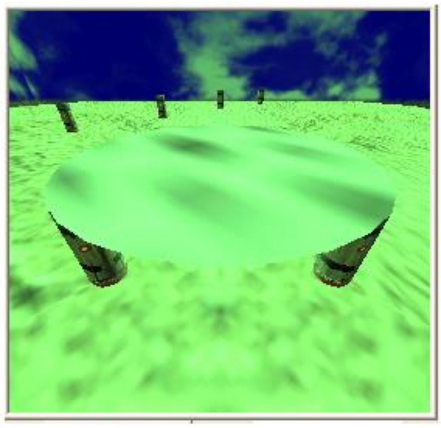
Since the level of ambient light (R=255, G=255, B=255) stays constant, ambient component can be preset for each surface in the scene and thus each object is displayed using its own intrinsic color, i.e. for m_a in Figure.2.(a) R=G=B=51, and in Figure.2.(b) R=B=51, G=102. Of course, such illumination model doesn't provide even minimal amount of realism. All we can see from the comparison are just colored silhouettes of some shapes.

6.2. Directional Light Source Effect

Directional lights produce ambient, diffuse and specular components as shown below.



(a)



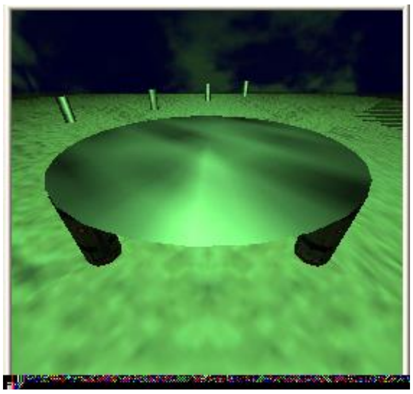
(b)

Figure 3. Directional Light Source Effects.

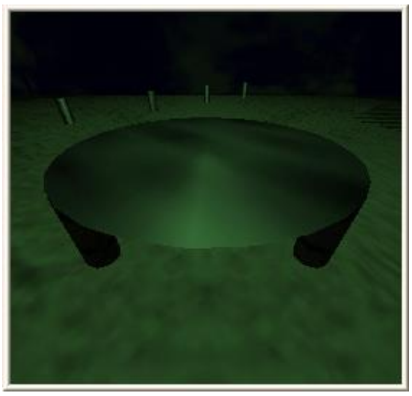
From the comparison between ambient light and directional light effects, it is seen that both the light sources apply and shine the light from everywhere at once. The only difference is that directional light gives the lighting in a specific direction. It is seen from the comparison between Figure.3.(a) and (b) that, if the object's color is not gray or if it has one extreme color (G=102) than the other components (R=B=51), directional light is more effectively shine onto its surface. So, direction lighting effect is more realistic to create a sunlight effect.

6.3. Point Light Source Effect

Point lights emanate the light in all directions from one specific point. They also have ambient, diffuse and specular components. Unlike a directional light, a point light does not go on indefinitely as shown in Figure.4.



(a)



(b)

Figure 4. Point Light Source Effects.

From the comparison between Figure.4.(a) and (b), it is seen that as the light travels away from the source, it gets dispersed, spreading out in every direction equally. Therefore, the farther an object is from a point light, the less illuminated that object will be. The attenuation in these figures are reasonably enough and realistic for creating lamp effect.

7. Conclusion

It can be concluded that, ambient light component is used to complement for effects such as multiple reflections of light from matter surfaces which can be very roughly modeled as ambient light.

Various lighting effects can be obtained by changing the colors of object material and light source with their respective parameter values such as shyness and attenuation.

From the comparison of the effects of light sources, it is seen that directional light source can not only be used for the daylight effect, but also for creating an evening scene. Positive shyness parameter of greater than five units with the less value of ambient components is the good choice for the evening scene. Negative shyness parameter can be used for creating the day light effects. Even though the night scene can be obtained with the directional light effect, it has less effective than the point light effect.

So, point light effects are good choices for creating the night scenes. Attenuation parameter of less than five units and positive shyness parameter of greater than five units are reasonable for the point light effects with the less value of respective ambient components.

8. Further Extensions

The lighting effects for various light sources with their properties can not only be used in creating a 3D game, for which artificial intelligence algorithms must be included, but also be used for analyzing the color pattern and lighting parameters for the future development.

9. References

[1]. David Gosselin "Character Animation With Direct3D Vertex Shaders" On-line document http://developer.amd.com/media/gpu_assets/ShaderX_CharacterAnimation.pdf.

[2]. Jerry Yee, "A Survey Of Graphics Programming Languages", On-line document: http://www.soe.ucsc.edu/classes/cmps203/Fall04/finalreports/ProjectPaper_JerryYee.pdf.

[3]. Louis Bavoil "Real-time Depth Buffer Based Ambient Occlusion" Game Developers Conference, March 2008. http://developer.download.nvidia.com/presentations/2008/GDC/CDC08_Ambient_Occlusion.pdf.

[4]. Louis Bavoil, Miguel Saintz, "Image-space Horizonbased Ambient Occlusion",September2008. http://developer.download.nvidia.com/presentations/2008/SIGGRAPH/HBAO_SIG08.pdf.

[5]. Michal Valient, "Project6 - Lighting per-pixel shading and optimizations for Direct3D", HTML article, November 2002.

[6]. Markus Holmaker & Magnus Woxblom, "Performance evaluation of the fixed function pipeline and the programmable pipeline", School of Engineering, Blekinge Institute of Technology, Sweden, June 2004.