TELKOMNIKA, Vol.10, No.1, March 2012, pp. 171~178
ISSN: 1693-6930
accredited by DGHE (DIKTI), Decree No: 51/Dikti/Kep/2010

An Overview on Base Real-Time Shadow Techniques in Virtual Environments

Hoshang Kolivand*, Mohd Shahrizal Sunar
ViCubelab, Dept. of Computer Graphics & Multimedia, Fac. of Computer Science & Information Systems
Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia
e-mail: shahinkey@yahoo.com, shah@fsksm.utm.my

Abstrak

Bayangan (shadow) adalah sangat bagus untuk menciptakan suasana realistis pada lingkungan virtual. Bervariasinya teknik shadow mendorong Penulis untuk mempersiapkan tinjauan pada semua teknik shadow dasar. Teknik shadow dapat dikategorikan dua kelas besar, teknik waktu-nyata dan teknik non waktu-nyata. Pada jiplakan sinar teknik non waktu-nyata, pemilihan jiplakan dan radiositas adalah sangat dikenal, dan dijelaskan secara mendalam. Radiositas diterapkan untuk menciptakan bayangan yang sangat realistis pada teknik non waktu-nyata. Karena algoritma radiositas tradisional sulit diimplementasikan, Penulis mengusulkan sesuatu yang sederhana untuk menangani hal ini. Kode pseudo yang diusulkan lebih mudah untuk dipahami dan diimplementasikan. Jiplakan sinar digunakan untuk mencegah tabrakan objek-objek bergerak. Bayangan proyeksi, banyaknya bayangan dan pemetaan bayangan digunakan untuk membuat bayangan waktu-nyata pada lingkungan virtual. Bayangan proyeksi telah digunakan untuk beberapa objek statis dan memiliki bayangan pada permukaan datar. Banyaknya bayangan digunakan untuk membuat bayangan yang akurat dengan garis tajam. Pemetaan shadow yang merupakan teknik dasar dari semua teknik terkini, direkonstruksi. Algoritma rekonstruksi ini memberikan beberapa ide baru untuk mengusulkan algoritma lain berdasarkan pemetaan bayangan.

Kata kunci: *banyaknya bayangan, jiplakan sinar , pemetaan bayangan, pemilihan sinar, radiositas*

Abstract

Shadows are elegant to create a realistic scene in virtual environments. Variety types of shadow techniques encourage us to prepare an overview on all base shadow techniques. Non real-time and real-time techniques are big subdivision of shadow generation. In non real-time techniques ray tracing, ray casting and radiosity are well known and deeply described. Radiosity implemented to create very realistic shadow on non real-time scene. Although traditional radiosity algorithm is difficult to implement, we have proposed a simple one. The proposed pseudo code is easier to understand and implement. Ray tracing used to prevent of collision of movement objects. Projection shadow, shadow volume and shadow mapping are used to create real-time shadow in virtual environments. We have used projection shadow for some objects are static and have shadow on flat surface. Shadow volume used to create accurate shadow with sharp outline. Shadow mapping that is the base of most recently techniques is reconstructed. The reconstruct algorithm gives some new idea to propose another algorithm based on shadow mapping.

Keywords: *radiosity, ray casting, ray tracing, shadow mapping, shadow volume*

1. Introduction

Computer graphics have become one of the most important parts of online and offline games, advertisement and simulation in virtual environment. When you are looking on TV, on the internet, playing games or looking in the each billboard on streets, you can see a lot of computer graphic's effects. Computer graphic has been a dramatically development since about fifteen years back, when users were happy with ability of computer technology without any high quality graphics.

To have a realistic environment, shadow is the most important effect that reveals more information about the distance between objects in the scene. It is the major factor of 3-D graphics for virtual environment but unfortunately is difficult to be implemented in display environment especially in real-time games. In computer games, shadows give the gamers feeling as if them playing in realistic world and provide maximum pleasure. A game with lack of

Received October 6th, 2011; Revised November 23th, 2011; Accepted January 9th, 2012

shadows cannot be attractive especially in this century where gamers' imagination request more realistic environment when they are watching cartoons or playing games.

There are two kinds of shadow, hard shadow and soft shadow. Where the light source is as a point, it produces hard shadow. On the other hand, where the light source is a wide, or there is more than one light point source, soft shadow will be appeared. In the other word, transition is the factor that determines whether the shadow is hard or soft. If the transition occurs, within one pixel, the shadow is classified as hard shadow, but if the transition occurs within more than one pixel and the colour of the shadow blends during the transition, then this shadow is considered as soft shadow.

In 1997 Frank Crow [1] introduces the main idea of shadow volume and published his ray-casting paper base on shadow volume algorithm entitled "Shadow Algorithm's for Computer Graphics". His method explicitly clips shadow geometry to the view frustum. In 1991, Heidmann published a paper base on volume shadow using stencil buffer which is a main idea of shadow volume algorithm [2].

In 1997, Heckbert, P. and M. Herf, proposed the famous algorithm on soft shadow [3]. They explain an algorithm for simulating shadow as a soft shadow for wide light source. In this paper they could generate soft shadow of complexity occluder on arbitrary objects. The wonder thing in their algorithm is that, it can create soft shadow independent of size of light source. But they generate soft shadow by sampling of hard shadow. In this algorithm, it is needed to have more samples. Gooch et al. (1999) is the another researcher who has worked on soft shadow [4].

Another algorithm that Carmack suggested in year 2000 was a bit different from the previous algorithms that include rays that are traced from infinity towards the eye [5]. Shadow Maps suggested by Fernando et al [6], which are an extension to the traditional shadow mapping technique. In year 2002, Lengyel propose a hybrid algorithm that uses faster Z-pass rendering [7].

Matt Olson and Hao Zhang [8], in 2008 work on tangent-space and they focused on tangential distance of objects to be used in polygon mesh silhouette detection. In 2011, Daniel Scherzer et al. published a survey entitled “A Survey of Real-Time Hard Shadow Mapping Methods”. They reviewed most hard shasow techniques based on shadow mapps [9].

Kolivand et al. compared shadow mapping and shadow volume in detailes [10]. They have introduced some advantages and drawback of each technique. In 2009, Liu et al[11] prepared a very good survey of shadow rendering with attention to projection shadows and volume shadows.

2. Shadow Techniques

There are two kinds of technique to generate shadow. Real-time techniques are interactive and so difficult to understand and implement but non real-time techniques are non-interactive (Figure 1). Although non-real-time techniques are not interactive, they have high quality specially to create shadow.

Ray casting, ray tracing and radiosity are non-real-time techniques that are so powerful to create high quality interaction between light sources and objects. Projection shadow, volume shadow and shadow mapping are famous techniques to create real-time shadow.

2.1. Non-Real-Time Techniques

Non-real time is a term used to describe a process that does not happen quickly. For example, non-real-time shadows doesn’t need to response with any input immediately. One of the techniques to create shadow is non-real time techniques. These techniques need a huge amount of calculation and could not be use in a real-time application. Although non- real time techniques are not convenient for real time environment, but they have a high quality. In following three important non-real time techniques will be described.

2.1.1. Ray Casting

Ray casting is a process that uses of ray-surface intersection tests to solve a diversity of problems in computer graphics. It is helpful in recognizing and solving a substantial number of problems that they related in computer graphics. Somebody confused this with the process of ray tracing, but they are different. Ray casting cannot do some of the same functions in ray tracing and it is faster than the ray tracing to move. In 1982, Scott Roth was the first scientist researcher that use ray tracing in computer graphics to describe a method for rendering one kind of modes [12].

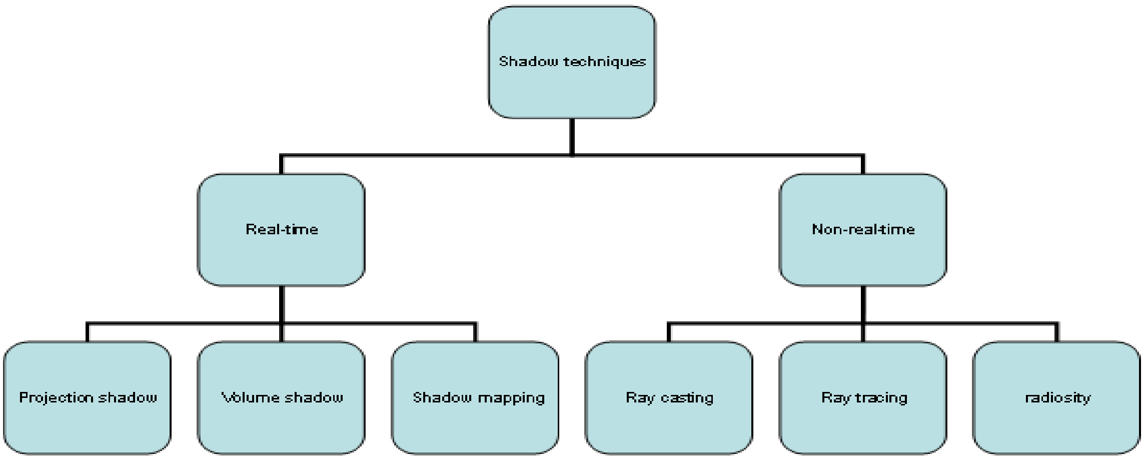

Figure 1. Shadow Techniques

Ray casting is faster version of the ray-tracing algorithm. Both ray casting and ray tracing are image order algorithms to render 3-D scenes to 2-D screens by following rays from point of view through to the light source. The ray casting can be divided in two separate parts. The first part is finding intersection of ray with each pixel of scene. The second part is to recognize color of that using an environment map. The simple algorithm for ray casting is:

```
For each sample
        Construct ray from eye through the plane.
        Find first surface intersect by ray through pixel.
        Compute color sample based on surface radiance.
End for

To implement of this algorithm is so easy:

Image_Ray_Casting(Camera cam, Scene sen, Int with, Int height)
{
    Image img=new Image(with, height);
For(int i=0;i<with;i++){
for(j=0;j<height;j++){
        Ray ray=ConstructRayThroughPixel(cam,i,j);
intersection hit=FindIntersection(ray,sen);
img[i][j]=GetColor(hit);}}
        Return img;
}
```

2.1.2. Ray Tracing

Ray tracing is a method base on rendering that uses for illumination. In the first step it traces each ray of light from the view point back through the image plane into the scene. In the second step, each ray is tested; against whole objects of scene to recognize, if each ray has an intersection with the objects or not. If the ray does not have any intersection it, means pixel is shaded the background color. Ray tracing is a convenient method to have shadows, reflections, multiple seculars and texture mapping in a real simple straightforward manner.

Ray tracing is also a good technique to render 3D graphics with complex light interactions. To generate an image by tracing, the ray of light, draw into the pixels in an image plane and simulating the effects of its collision with virtual objects. By this technique, we can generate a picture with a lot of mirrors, transparent planes and shadows with wonder results. This technique is very powerful to create realism visual scene.

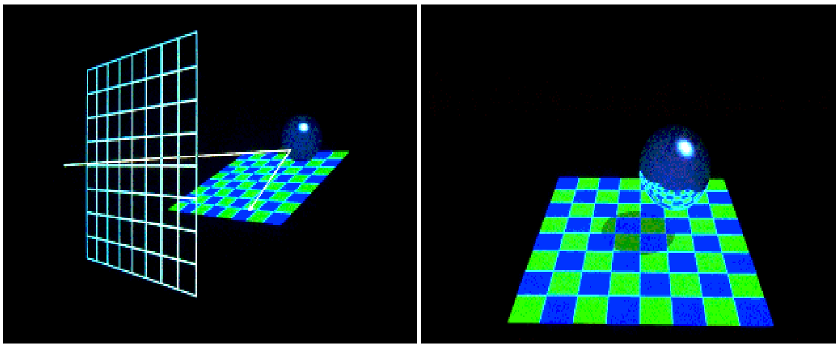

Figure 2. Ray Tracing

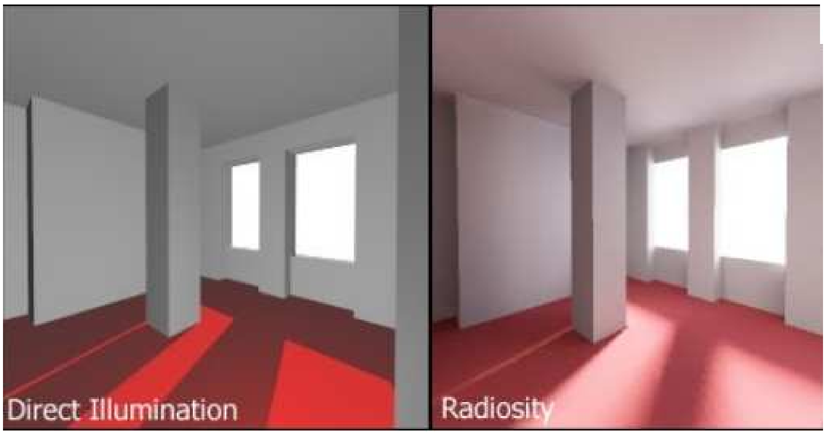

Figure 3. Radiosity

The quality of this technique is usually higher than scan line rendering methods, but is has a greater computational cost. Ray tracing is one of the best suited for applications where the time is not so important, like a movie, some special effects in TV, and more poorly suited for real-time applications such as some computer games where speed is critical. Ray tracing is convenient technique for simulation of a wide diversity of optical effects like reflection and refraction, scattering, and chromatic aberration and it makes shadow really realistic.

2.1.3. Radiosity

Radiosity is another technique to create soft shadow. This kind of algorithm can calculate diffuse inter reflection between surfaces by determining an energy balance in a close environment. These kinds of algorithms are rather expensive and they can be used just for polygonal surfaces. Each surface is divided into some elemental surface patches. Be sure the patches are small enough. Then, the intensity distribution of surface depend on quality of surface and it is approximated a constant value.

2.2. Real Time Techniques

As the name suggests, real-time is occurring immediately[13]. The real time word is used for some different features. For example, real-time shadows are that respond to input immediately. They used for such tasks like navigation, in which the computer must react to a steady flow of new information without interruption. Real time can also refer to events simulated by a computer at the same speed that they would occur in real life.

In graphics animation, for example, a real-time shadow would display objects and their shadow moving across the scene at the same speed that they would actually move[14]. Real time shadows in real-time applications such as computer games and virtual environment are more important to create realistic scenes; so that the user will be feeling realistic in the scenes.

In practice, the rendering performance and the quality of shadows generated are important considerations in the selection of shadow algorithm. Quality of shadows may have to be compromised with high performance but not to the extent of the lowest quality; it must be better than average quality. These criteria's are vital and can only be achieved using certain algorithms [15]. In the next section, some of the real-time algorithms that are widely used in computer graphics will be described.

2.2.1. Projection Shadow

Projection shadows are a one the fast rendering method to create shadow. Although they are fast enough, they can project shadow only on flat surface such as wall or on the ground but not both of them together. In this method we should draw projection of each pixel of occluder on the shadow receiver with Y=0 along ray that started from the light source until the surface.

L is light source; P is a pixel of occluder that S is projection of it on the floor with Cartesian system. According to the basic formula for shadow on the flat surface can provide a matrix as a following:

N: Normal vector of ground
C: an arbitrary point on ground
S=L + α (P-L)
(P-C) N = 0 => $P = L + \frac{E}{NP-D}(P-L)$

$$\text{Shadow matrix}=\begin{pmatrix} LxNx+E & LxNy & LxNz & -ELx-DLx \\ LyNx & LyNy+E & LyNz & -ELy-DLy \\ LzNx & LzNy & LzNz+E & -ELz-DLz \\ Nx & Ny & Nz & -D \end{pmatrix}$$

```
myShadowMatrix(float groundplane[4],float lightpos[4])
{
 GLfloat dot1;
 float  shadowMat[4][4];
  dot = groundplane[X] * lightpos[X] +
    groundplane[Y] * lightpos[Y] +
    groundplane[Z] * lightpos[Z] +
    groundplane[W] * lightpos[W];

  shadowMat[0][0] = dot1 - lightpos[X] * groundplane[X];
  shadowMat[1][0] = 0.f - lightpos[X] * groundplane[Y];
  shadowMat[2][0] = 0.f - lightpos[X] * groundplane[Z];
  shadowMat[3][0] = 0.f - lightpos[X] * groundplane[W];
  shadowMat[X][1] = 0.f - lightpos[Y] * groundplane[X];
  shadowMat[1][1] = dot1 - lightpos[Y] * groundplane[Y];
  shadowMat[2][1] = 0.f - lightpos[Y] * groundplane[Z];
  shadowMat[3][1] = 0.f - lightpos[Y] * groundplane[W];
  shadowMat[X][2] = 0.f - lightpos[Z] * groundplane[X];
  shadowMat[1][2] = 0.f - lightpos[Z] * groundplane[Y];
  shadowMat[2][2] = dot1 - lightpos[Z] * groundplane[Z];
  shadowMat[3][2] = 0.f - lightpos[Z] * groundplane[W];
  shadowMat[X][3] = 0.f - lightpos[W] * groundplane[X];
  shadowMat[1][3] = 0.f - lightpos[W] * groundplane[Y];
  shadowMat[2][3] = 0.f - lightpos[W] * groundplane[Z];
  shadowMat[3][3] = dot1 - lightpos[W] * groundplane[W];
  glMultMatrixf((const GLfloat*)shadowMat);
}
```

2.2.2. Shadow Volume

In 1977, Crow could not implement his algorithm because of lack of enough hardware in that time [1]. Heidmann, in 1991, completed the shadow volume algorithm of Crow and implemented. He could create shadow on arbitrary object [2]. In simple word, it can be illustrate like this:

A line should be draw form point of light source to the each vertex and these rays continue infinite. After recognize of each pair of neighborhood rays draw a polygon from the edge of occluder onto the shadow receiver. These polygons make a infinite truncated pyramid that the part of that which located in the below of occluder is shadow volume that is semi infinite quadrilateral with two finite vertices and two other vertices are located in infinite that showed with green lines in Figure 5.

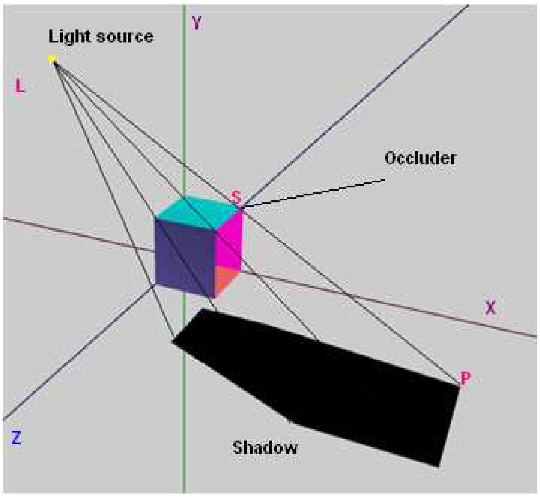

Figure 4. Projection shadow

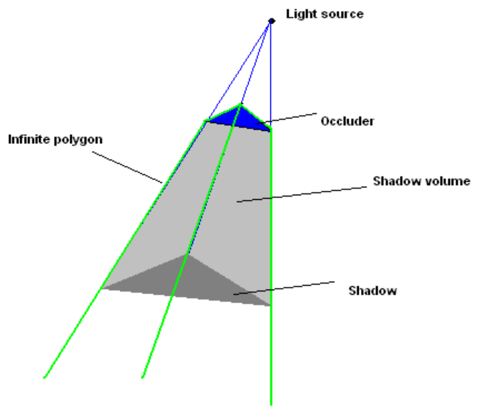

Figure 5. Shadow Volume

2.2.3. Shadow Mapping

In 1978 by Lance Williams introduced a wonder technique that call shadow maps[15]. He propose his method in a paper entitled "Casting curved shadows on curved surfaces" Shadow mapping is a depth map rendered from the point of view of the light source.

There is one simple way to determine whether something/someplace is in shadow or not: If a point can be seen from the point of view but cannot be illuminated from the light source it is in shadow. If a point can be seen from both of the view point and light source, it is not in shadow.

Shadow mapping is one of diversity ways to create shadow with some advantages and disadvantages that we will introduce them in following:

Advantages:

- It does not need any geometry calculation because it is image base technique and most of calculation will be done in GPU.
- On contrast of shadow volume, it does not need high filling.
- It does not need to stencil buffer and just requires a single texture to hold shadowing information for each light.

Disadvantages:

- There is not aliasing, especially when there is a small shadow maps.
- The scene geometry must be render once per light in order to generate the shadow map for a spotlight, and more times for a multi directional point light.

Here we will focus on basic shadow mapping for a single spotlight, but there are plenty of papers about how to extend and improve the technique.

The original of this algorithm has two steps. First render the scene from point of light source in z-buffer. Second step is rendering the scene again from point of view. For each pixel in point of view, if it is farther than z-buffer it is in shadow, else it is in lit. The algorithm in simple way is as follows:

Step 1: Render the whole scene from the point of light source and put it in the z-buffer (depth map)

Step 2: Render the whole scene again but from the point of view

Step 3: For each pixel in view point rendering (Step 1), if it is not visible from light source rendering (Step 2) it is either in shadow it is lit

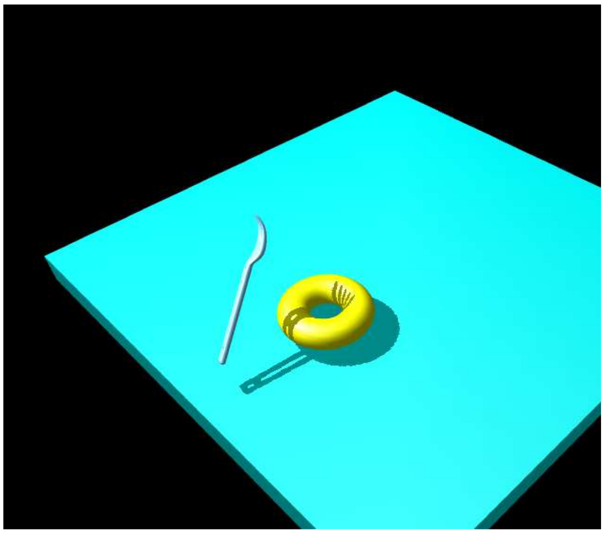

Figure 6. Shadow mapping

Generating shadow with shadow mapping algorithm is easy and fast and the most important usage of shadow mapping is in soft shadow with high quality and high frames per second.

3. Discussion

Variety of shadow techniques encouraged us to prepare an over view on shadow generation. It is difficult to find the best method to create shadow in our project. Some parameters play role to find the convenient method. First, it is most important that the environment that needs shadow is real-time or not. If it is not real-time, radiosity is one of the best techniques to create realistic scenes. However, another parameter is expense on implementation. As mentioned before, radiosity is complicated in implementation.

In case of real-time rendering, the way to choose a base of method still needs some other parameters. If the shadow on flat surfaces is needed, projection shadows are best. High speed rendering of projection shadows helps to have shadow on real-time rendering with very low cost. To have shadow on other objects shadow volume and shadow mapping are suitable. Shadow volume is convenient base for shadow generation when precise shadows are needed. The outline of shadow volume is accurate enough. Although shadow volume is convenient to have shadow on arbitrary objects, high mathematical calculation to recognize the silhouette is big drawback of them method.

Shadow mapping is another technique to have shadow on other objects. Shadow mapping is base of most new techniques to create shadow in real-time rendering. High rendering speed is one of the reasons to choose the method for wide environments real-time rendering.

To create soft shadows, shadow maps are convenient. Using some techniques such as Percentage-Closer Filtering, Deep Shadow Maps, Adaptive Shadow Maps, Perspective Shadow Maps, Variance shadow maps and Cascaded Shadow Maps is most commonly method to create soft shadows. Table 1 shows a briefing comparison between each type of real-time and non real-time techniques.

Table 1. Comparison between base shadow generation techniques.

Type	Technique	Calculation	Quality	Rendering Speed
Non Real-time	Ray Casting	Moderate	Low	Moderate
	Ray Tracing	High	Moderate	Moderate
	Radiosity	Very high	High	Very low
Real-time	Projection Shadow	Low	Moderate	Very high
	Shadow Volume	High	Very high	Low
	Shadow Mapping	Very low	High	High

4. Conclusions

In this paper two kinds of shadow generation in real-time and non real-time aspect have introduced. To create non real-time shadow ray tracing, ray casting are described and a new technique to create radiosity is proposed. To have volume shadow on an arbitrary object stencil buffer is used. Volume shadow that was difficult to understand and implement is improved. A comparison between shadow volume and shadow mapping is done. In comparison, to have high-speed algorithm shadow mapping is suitable but to have precise shadow especially when light small shadow is need, volume shadow is more convenient. Finally, a reconstruction of traditional shadow mapping is proposed. The most important finding of this paper, in addition of reconstruction of all base shadow algorithms, helping user to find the best method to create shadow for real-time and non real-time rendering.

References

[1] Crow F. Shadow Algorithms for Computer Graphics. *Computer Graphics*. 1977; 11(2): 242-247.
[2] Heidmann T. Real Shadows Real Time. IRIS Universe, 1991; 18:28-31.
[3] Heckbert P, Herf M. Simulating Soft Shadows with Graphics Hardware. *Technical Report CMU-CS-,* Carnegie Mellon University. 1997; 97-104.
[4] Gooch B, Sloan PJ, Gooch A, Shirley P, Riesenfeld R. *Interactive Technical Illustration*. Symposium on Interactive 3D Graphics. Atlanta, Georgia, United States. 1999; 31-38.
[5] Carmack, CarmackOnShadowVolumes, (Personal Communication between Carmack, 2002, http://developer.nvidia.com/view.asp?IO=robust_shadow_v
[6] Fernando R, Fernadez S, Bala K, Greenberh D. *Adaptive shadow maps*. In Proceedings of ACM SIGGRAPH 2001; 387–390.
[7] Lengyel E. Mathematics for 3D Game Programming & Computer Graphics, Charles 2002.
[8] Olson M, Zhang H. *Silhouette Extraction in Hough Space*. Eurographics Proceedings 25, 2006 available online.
[9] Scherzer D, Wimmer M, Purgathofer W. A Survey of Real-Time Hard Shadow Mapping Methods. *Computer Graphics forum*, 2011; 30(1): 169–186.
[10] Kolivand H, Sunar MS. Shadow Mapping or Shadow Volume?. *International Journal of New Computer Architectures and their Applications*. 2011; 1(2): 64-70.
[11] Liu N, Pang MY. *A Survey of Shadow Rendering Algorithms: Projection Shadows and Shadow Volumes*. Second International Workshop on Computer Science and Engineering, 2009; 488-492.
[12] Scott R. Ray Casting for Modeling Solids. *Computer Graphics and Image Processing*. 1982;18: 109-144
[13] Basori AH, Tenriawaru A., Mansur A.B.F. Intelligent Avatar on E-Learning using Facial Expression and Haptic. *TELKOMNIKA*. 2011; 9(1): 115-124.
[13] Hermawati FA, Koesdijarto R. A Real-Time License Plate Detection System for Parking Access. *TELKOMNIKA*. 2010; (8)2: 97-106.
[14] Williams L. Casting Curved Shadows on Curved Surfaces. *SIGGRAPH 1978*; 12(3): 270-274.
[15] Olufemi AF, Sunar MS, Ikotun AM. Augmented Reality Prototype for Visualising Large Sensors' Datasets. *TELKOMNIKA.* 2011: 9(1):161-170.