# Shadow Mapping or Shadow Volume?

Hoshang Kolivand and Mohd Shahrizal Sunar
UTM ViCubelab, Faculty of Computer Science and Information Systems,Department of Computer Graphics and Multimedia, Universiti Teknologi Malaysia,81310 Skudai Johor, Malaysia
shahinkey@yahoo.com, shah@cs.utm.my

## ABSTRACT

In this paper two techniques of shadow generation are described. Volume shadow is geometric base but shadow mapping is image base. Silhouette detection is a most expensive step to create volume shadow. Two algorithms to recognize silhouette are introduced. Stencil buffer and Z- buffer are two other tools for creating shadow by volume shadow technique. Both algorithms are implemented in virtual environment with moveable light source. Triangular method and the Visible-non visible method are introduced. The recent traditional silhouette detection and implementation techniques used in volume shadow algorithm are improved. With introduce flowchart of both algorithms, the last volume shadow algorithm using stencil buffer is rewritten. A very simple algorithm to create volume shadow is proposed. The last shadow mapping algorithm is rewritten. These techniques are poised to bring realism into commercial games. It may be use in virtual reality applications.

## KEYWORDS

Volume shadow, shadow mapping, silhouette detection, stencil buffer.

## 1 INTRODUCTION

Computer graphics have become one of the most important parts of online and off line games, advertisements, and simulation in a virtual environment. Whether you are watching TV, on the internet, playing games or looking at the billboards on the streets, you can see many computer graphic effects. Computer graphics has frequently been used to produce a simulation of the real world by synthesizing photo realistic images, especially for outdoor scenes, where displaying the sky as the background is indispensable.

There are many algorithms to create shadow. Shadow volume and shadow mapping are classical real-time hard shadow techniques for shadow generation on non-flat surface. Although volume shadow is accurate enough, it is geometrical and needs more calculation. Shadow mapping is easy to implement and it is image base. Volume shadows have great achievements in game makers. Although volumes shadow is considered as established in gaming industry, they have two expensive phases. One of them is update of volume rendering passes and the other one is silhouette detection.

Recognizing the outline of object can increase the speed of algorithm. To find the outline of object, silhouette detection is essential because it can reduce the cost of implementation and it is the main item to improve an algorithm. Silhouette detection is an important phase in all visual effects. It is mentionable that the bases of silhouettes are visual and view point. To generate shadow, silhouette detection plays a crucial role to detect the boundary of occluder.

Shadow mapping is faster than volume shadow and cheap enough in compare with volume shadow but precise algorithm of volume shadow is more important reason to using volume shadow.

The idea of shadow volume was first introduced in 1977 when Frank Crow [1] published his ray-casting based shadow volume algorithm. His method explicitly clips shadow geometry to view the frustum. In 1991 Heidmann published a paper based on volume shadow using stencil buffer which is even today the main shadow volume algorithm [2]. Stencil shadows belong to the group of volumetric shadow algorithms, as the shadowed volume in the scene is explicit in the algorithm. Let's take a look at how the original stencil shadow volume technique works.

The previous algorithms include rays that are traced from infinity towards the eye [3], [4]. Shadow Maps suggested by Fernando et al [5] which are an extension to the traditional shadow mapping technique. In year 2002 Lengyel propose a hybrid algorithm that uses faster Z-pass rendering [6].

Benichou and Elber[7] used a method to compute parallel silhouette edges of polyhedral object based on Gaussian sphere, for the normal vectors which are located on the view direction mapped onto a Gaussian sphere. 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 2010, Matt Olson [9-10], designed a site that lists all related papers. Billeter et al. are the later researchers who worked on volumetric shadow. They introduced a more efficient method for computing single scattering effects in homogeneous participating media [11].

## 2 SILHOUETTE ALGORITHMS

Silhouette detection is a function like $f: R^3 \rightarrow R^2$. Silhouettes have important role to create shadow on shadow receiver. To create shadow, projection of outline of object is enough to generate shadow as a result the cost of projection will be decreased. The most expensive part of shadow volume algorithm is identification silhouette of occluder.

Silhouettes have the most important role to recognize and project shape onto the shadow receiver. To create shadow, projection of silhouette of occluder is enough to generate a shadow of the whole object and as a result the cost of projection will be low. A silhouette edge of polygon is the edges that belong to two neighborhood planes where normal vector of one of them is towards the light and normal vector of the other plane is away from the light. If the volume shadow is desired point by point, it is too difficult to execute the program. It needs a lot of calculation and it takes a substantial CPU time for rendering. To improve this technique we should recognize the contour edges or silhouettes of object and implement the algorithm just for silhouettes. Using silhouette edges of the occluder to generate a volume shadow could optimized the process because the amount of memory is decreased, therefore, rendering will be done faster. The silhouette should be recalculated when position of the light source changes or the occluder moves.

There are many methods to recognize outline of object. Most of them are expensive but our approach is to introduce an algorithm that is not expensive like past algorithms. In stencil shadow algorithm it is needed to divide silhouette face of occluder to triangle

meshes. It means that each edge should shared by just two triangles. To determine which edge is silhouette, it is needed to recognize which aged is shared between a face toward the light source and a face away to the light source.

### 2.1 Triangular Algorithm

In this algorithm it is needed to divide each face of occluder into triangle meshes. It means that each edge should be shared by just two triangles. To determine which edge is silhouette, it is needed to know which edge is shared between faces towards the light source and faces away of the light source.
Here the triangular flowchart is introduced:

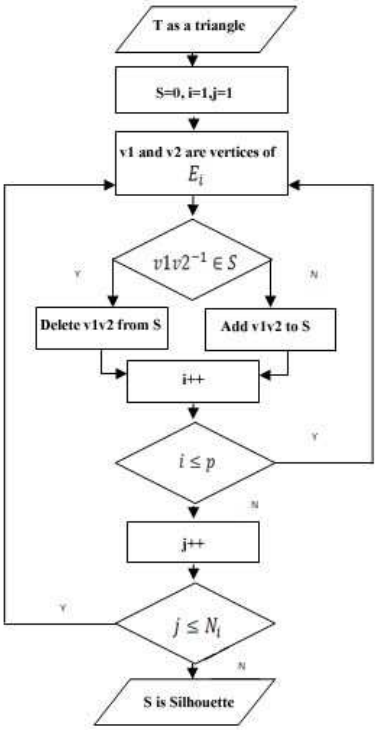


**Fig 1: Triangular flowchart to recognize silhouette**

Where:

$P$ : Number of all polygons

$N_i$: Edge number of i$^{th}$ polygon

$E_i$ : i$^{th}$ Edge

S: An array of vertices

Fig 1 illustrates how to divide one side of a box which is consisting of four triangles. To determine silhouette we need just outline of the box not all of the edges.

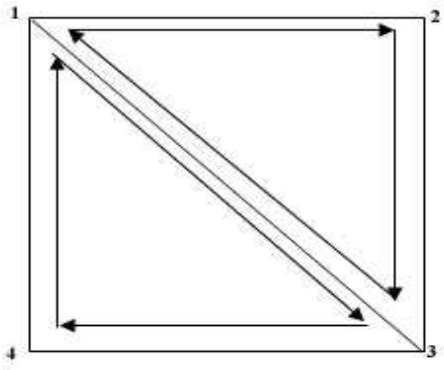


**Fig 2: Triangle dividing for silhouette detection**

### 2.2 Visible-Non Visible Algorithm

In this algorithm, all edges which have just one visible face are silhouette. These edges which have exactly one visible and one invisible face are silhouette, but on the contrary, each edge with two visible faces or two invisible faces is not regarded as a silhouette.

It is important to note that silhouette determination is one of the two most expensive operations in stencil shadow volume implementation. The other is the shadow volume rendering passes to update the stencil buffer. These two steps are the phase that we intend to consider in the nearest future.

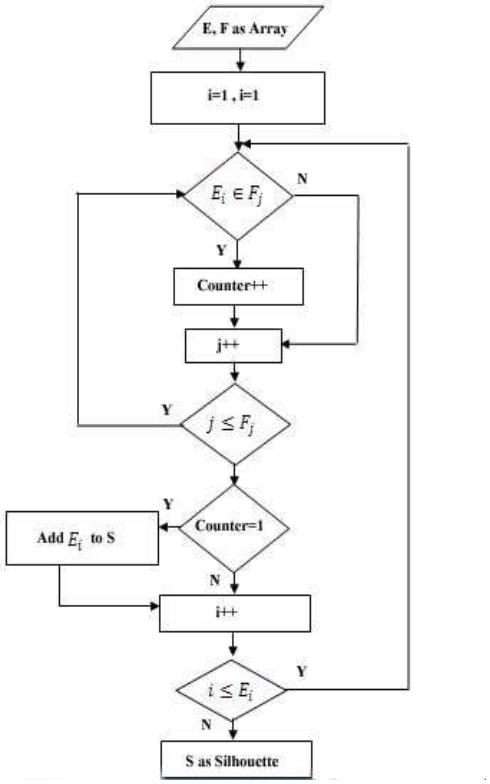


**Fig 3**: Visible-non visible silhouette detection flowchart

Where:

$F_i$: $i^{th}$ Face

$E_i$: $i^{th}$ Edge

S: An array of vertices

## 3 VOLUME SHADOW

Volume shadow is geometric base. Silhouette detection is a most expensive step in volume shadow that described in previous section. Stencil buffer and Z-buffer are two most important tools for creating shadow by volume shadow technique.

### 3.1 Stencil Buffer

Stencil buffer is an extra buffer, in addition to the color buffer and depth buffer found on modern computer graphics hardware. The buffer is per pixel, and works on integer values, usually with a depth of one byte per pixel. The depth buffer and stencil buffer often share the same area in the RAM of the graphics hardware.

In the simplest case, the stencil buffer is used to limit the area of rendering (stenciling). More advanced usage of the stencil buffer makes use of the strong connection between the depth buffer and the stencil buffer in the rendering pipeline.

There is a close relationship between stencil buffer and Z-buffer. Both these buffers are neighborhood and are located in the graphics hardware memory. Z-buffer requires controlling a selected pixel of stencil value that is increased or decreased when it is needed to count the time of entering and leaving the shadow volume respectively. The stencil buffer will be increased when eye's ray enters the volume shadow by passing from front face of volume shadow and it will be decreased when ray leaves the volume shadow by passing of back face of it. It has two phases, first render the front faces; if depth test passes then stencil buffer should be increased. In the second rendering; if depth test passes stencil buffer should be decreased.

For example, stencil values can be automatically increased or decreased for every pixel that fails or passes the depth test.

### 3.2 Z-Buffer Buffer

In computer graphics, z-buffer is the management of image depth coordinates in three-dimensional graphics, usually done in hardware, sometimes in software. It is one solution to the visibility problem, which is the problem of deciding which elements of a

rendered scene are visible, and which are hidden. The painter's algorithm is another common solution which, though less efficient, can also handle non-opaque scene elements. Z-buffering is also known as depth buffering
Z-buffer is most important equipment in computer graphics hardware. It can be used to create real-time shadow for virtual environments. The first researcher who used Z-buffer was Catmull [12], He propose the first algorithm to create shaded images of hicubic surface patches. His algorithm is not entirely private and it is too difficult to implement but requires huge amount of memory.

### 3.3 Volume Shadow Using Stencil Buffer

Now if each object or part of object that is inside of truncated pyramid is in shadow and it should be dark but each object or part of object that is out of truncated pyramid is in lit and it should not be dark To generate volume shadow using stencil buffer and depth buffer together, following steps should be done: After generating volume shadow, we should pass the ray from the eye to each point of object on the shadow receiver. As it mentioned before (in stencil buffer) ,if the ray pass the front face of volume shadow, increase the stencil buffer and when the ray pass the back face of volume shadow, decrease the stencil shadow. Finally, if the number of stencil buffer is not zero it is in shadow. We are going to write this as an algorithm with silhouette detection:

**Step** 1: Render the whole scene just with ambient and diffuse

**Step** 2: Enable stencil buffer, disable depth buffer for writing disable color buffer

**Step** 3: Use silhouette detection (Triangle or Visible-non visible algorithm)

**Step** 4: Create a pyramid from light of source onto the silhouettes and continue to infinity

**Step** 5: Use Z-pass algorithm to recognize each pixel belongs to truncated pyramid

**Step** 6: Draw a ray from point of view in to each pixel in the scene. If the ray enters to volume increase the stencil buffer and if it exits from volume decreases the stencil buffer.

**Step** 7: For each pixel if the stencil buffer is zero, it is in lit else it is in shadow

**Step** 8: Render the whole scene with lighting.

**Step** 9: Enable color buffer for writing

## 4 Shadow Mapping

Shadow mapping is image base and as a result it is faster than the volume shadow. In 1987 Williams introduced a wonder technique that called shadow map [13]. He proposed 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.
Shadow mapping is one of the high usage methods specially to create soft shadow with high quality and high frame per second. In compare with volume shadow it has some advantages and disadvantages.

**Advantages:**

- It does not require any geometrical calculation, since shadow mapping is an image space technique,

working automatically with objects created or altered on the GPU.

- It does not need stencil buffer. Just one single texture is required to hold shadowing information for each light.
- It avoids the high fill requirement of shadow volumes.

**Disadvantages:**

- Bad quality in outline of shadow or aliasing, especially when light source is far from occluder
- For each light, the scene must be rendered once, in order to generate the shadow map for a spotlight, and more times for an omnidirectional point light.

## 5 Performance

In each algorithm, frame per second (FPS) is the most important factor to implement. The triangular method and visible- none visible method are used to recognize silhouette. It is amazing that the visible-non visible method has higher FPS than the triangular method.
To have shadow on arbitrary object such as torus or sphere stencil buffer and Z-pass algorithm are used.

In Figure 4, volume shadow is created using stencil buffer and z-pass algorithm. The following result is obtained in a PC with NVIDIA Graphic Hardware and P IV 2.8GH. When triangular algorithm is used the FPS is 69.93. Using Visible-non visible algorithm produce same result with 65.51 FPS in same PC.
Different between FPS when project is rendered without shadow and when is rendered with volume shadow using Triangular algorithm for silhouette detection is not so much. In additional, in this case FPS is acceptable.

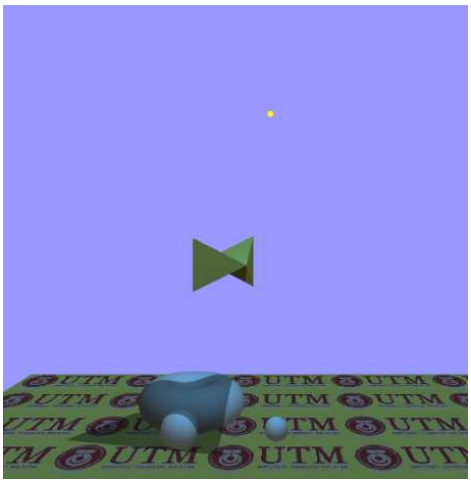


**Fig 4**: Volume Shadow

Figure 5 shows the result of shadow mapping in a PC with NVIDIA Graphic Hardware and P IV 2.8GH as mentioned before. The FPS is

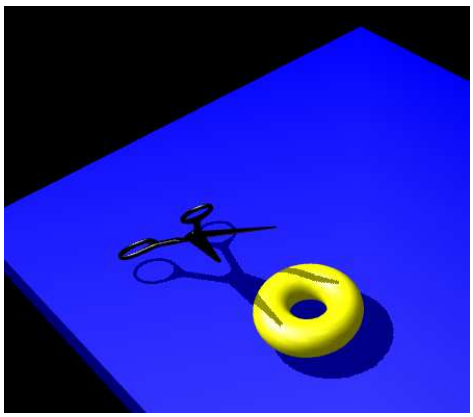

**Fig 5:** Shadow mapping

In Table 1 the FPS for using triangular algorithm and visible-non visible algorithm in volume shadow and FPS for shadow mapping is compared.

**Table 1**: Compare FPS

| *Status* | *FPS* |
|---|---|
| ***Without shadow*** | 71.47 |
| ***Using Triangular Algorithm*** | 69.93 |
| ***Using Visible-Non visible Algorithm*** | 65.51 |
| ***Shadow Mapping*** | 70.03 |

## 6 Conclusion

In this paper two kinds of shadow generation with two kinds of silhouette detection are presented. To create volume shadow, triangular algorithm and visible-non visible algorithm are used. To have volume shadow on an arbitrary object stencil buffer is used. Z pass method is an appropriate tool to check if an object or part of object is located inside the volume shadow or is located in lit. Volume shadow that was difficult to understand and implement is improved.

A comparison between volume shadow using Triangular and Visible-non visible algorithm and shadow mapping is done.

To make easy of the volume shadow, an easy algorithm is proposed and for recognizing silhouette two flowcharts are introduced. 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.

## REFERENCES


1. Crow, F.C.: Shadow Algorithms for Computer Graphics. Comp. Graph., 11, 242--247 (1977)
2. Tim, H.: Real Shadows Real Time. Iris Univ 18, 23--31 (1991)
3. Carmack, J.: Carmack on Shadow Volumes (Personal Communication between Carmack and Kilgard). Referenced:10.4.2002. Available at: http://developer.nvidia.com/view.asp?IO=robust_shadow_volumes
4. Lokovic, T., Veach, E.: Deep Shadow Maps. In:. Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, 385--392 (2000)
5. Fernado, R., Fernadez, S., Bala, K., Greenberh, D.P.: Adaptive Shadow Maps. In: Proceedings of ACM SIGGRAPH ACM Press / ACM SIGGRAPH, Computer. 387--390. (2001),
6. Eric, L.: Mathematics for 3D Game Programming & Computer Graphics. Charles River Media, 437--451 (2002)
7. Benichou, F. Elber, G.: Output Sensitive Extraction of Silhouettes from Polygonal Geometry. In: Proceedings of Pacific Graphics 1999, 60--69 (1999)
8. Zhang, H.: Forward Shadow Mapping,. In: Proceedings of the 9th Eurographics Workshop on Rendering,.131--138 (1998)
9. Olson, M., Zhang, H.: Silhouette Extraction in Hough Space,. Comp. Graph. Forum, 25, 273--282 (2006)
10. Olson, M. : A Survey of Silhouette Papers, Available on: http://www.cs.sfu.ca/~matto/personal/sil_papers.html (2010)
11. Billeter, M., Sintorn, E., Assarsson, U.: Real Time Volumetric Shadows using Polygonal Light Volumes. In High Performance Graphics, 39–45 (2010)
12. Catmull, E.: A subdivision algorithm for computer display of curved surfaces. UTEC-CSc-74--133, Computer Science Dept., Univ. of Utah, Dec (1974).
13. Williams, L.: Casting Curved Shadows on Curved Surfaces, SIGGRAPH ’78, 12, (3), 270--274 (1978)