Journal of Computer Science 7 (7): 980-985, 2011
ISSN 1549-3636

Real-Time Volume Shadow using Visible-Non Visible Algorithm

Hoshang Kolivand, Mohd Shahrizal Sunar, Azam Amirshakarami and Zhila Ranjbari
Department of Computer Graphics and Multimedia, UTM ViCubeLab,
Faculty of Computer Science and Information System,
University Technology Malaysia 81310, Johor, Malaysia

Abstract: Problem statement: Shadows are most important effect to make realistic games and visually appealing images, but they are poor in frame per second. Silhouette detection is most important phases to create real-time shadow. **Approach:** Present study describes a real-time shadow generated with volume shadow algorithm using Visible-Non Visible algorithm (VNV) in virtual environment illuminated by a movable light source to improve the frame per second. Silhouette detection that is most expensive part of shadow creation was described. Triangular method and VNV method were compared. An improved algorithm for volume shadow was proposed and volume shadow using both methods in C++ Opengl implemented. **Results:** The VNV algorithm increases the number of Frame Per Second (FPS) and as a result decreases the cost of implementation. **Conclusion:** A verified algorithm for volume shadow makes it easy to understand by everyone how to produce real-time shadow in outdoor and indoor virtual environment. It is possible to use it in current commercial games or other virtual reality systems.

Key words: Real-time shadow, volume shadow, silhouette detection, stencil buffer

INTRODUCTION

Shadows are most important effect to make realistic games and visually appealing images, but they are poor in Frame Per Second (FPS) and difficult to implement in real-time environment (Yaashuwanth and Ramesh 2010). In computer games, shadows can give the best feeling to gamers that they are playing in realistic world. A game with lack of shadow effect cannot be attractive especially in this century that gamers' imagination request more realistic situation.

The idea of volume shadow was proposed when Frank Crow published his ray-casting based shadow volume algorithm. His method explicitly clips shadow geometry to view the frustum (Crow, 1977). Published a paper based on volume shadow using stencil buffer, which is even today the main shadow volume algorithm. Stencil shadows belong to the group of volumetric shadow algorithms, as the shadowed volume in the scene is explicit in the algorithm. He proposed how the original stencil shadow volume technique works.

Another algorithm that Carmack suggested was a bit different from the previous algorithms which includes rays that are traced from infinity towards the eye (Lokovic and Veach, 2000). Shadow Mapping suggested by (Fernando *et al*., 2001) which are an extension to the traditional shadow mapping technique. In year 2002 Lengyel propose a hybrid algorithm that uses faster Z-pass rendering. Soon *et al*. (2004) introduced an algorithm that used the spatial coherence and frame coherence to gather.

Benichou and Elber (1999) 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. Olson and Zhang (2006), worked on tangent-space and they focused on tangential distance of objects to be used in polygon mesh silhouette detection. Olson (2010), designed a site that lists all related papers.

Recently, Ulf Assarsson is working on shadow. He has proposed a lot of algorithms and methods to create real-time shadow (Assarsson and Akenine-Moller, 2004).

MATERIALS AND METHODS

To create volume shadow some techniques like silhouette detection, stencil buffer and Z-pass algorithm are needed. Before to spend on shadow it is needed to introduce require methods.

Corresponding Author: Hoshang Kolivand, Department of Computer Graphics and Multimedia, UTM ViCubeLab, Faculty of Computer Science and Information System, University Technology Malaysia 81310, Johor, Malaysia

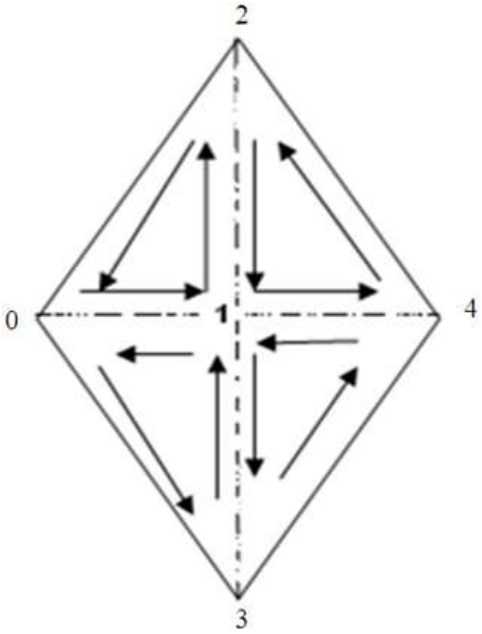

Fig. 1: Triangular mesh

Silhouettes detection: Silhouette is the outline of occluder that is only part of occluder, which contributes in creating shadow. There are some different methods to recognize the silhouette of the occluder. To create real-time shadow using stencil buffer, silhouette detection is needed (Andrea *et al*., 2010).

Triangular Algorithm: In triangular algorithm, each face is divided into some triangles and makes a triangular mesh (Fig. 1). It means that each edge should shared by just two triangles. To determine which edge is silhouette, it is need to know each aged is shared between faces toward the light source and faces away to the light source. One of the ways to this is CPU cycle hungry. Figure 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.

Now we are going to introduce a formula to find a silhouette:

$$S = \sum_{i}^{P} \sum_{j}^{N_i} \text{iif}\,(v_{ij} \in S[u_{ij}], \text{Delete } v_{ij} \text{from S, Add } u_{ij} \text{ to S})$$

Where:

p = Number of polygons
N_i = Number of i^{th} polygon
V_{ij} = First vertex of edge i^{th}
u_{ij} = Second vertex of edge j^{th}
S = A array of vertices

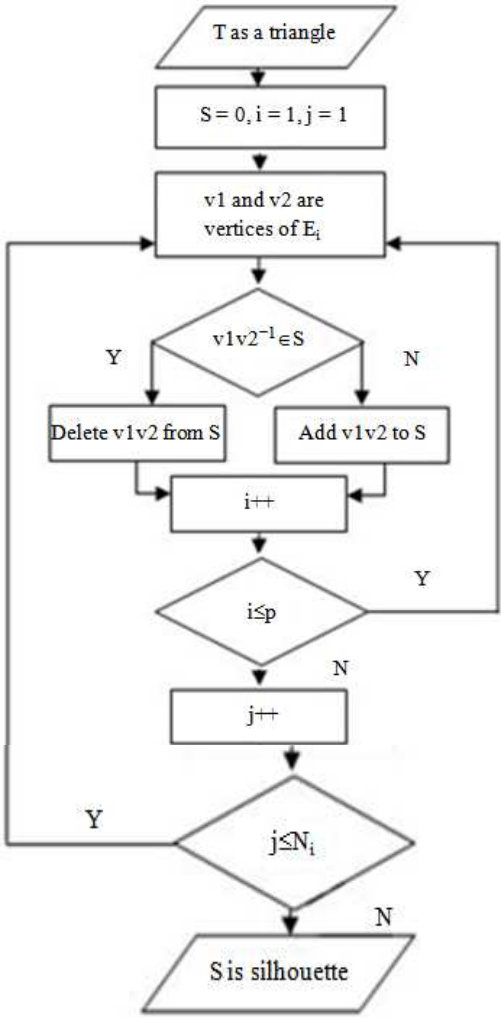

Fig. 2: Triangular flowchart

VNV 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 Fig. 2 and 3.

Silhouette detection is one of the two most expensive operations in implementation of stencil volume shadow. The other is volume shadow rendering passes to update the stencil buffer.

Stencil Buffer: Stencil buffer is a part of memory in 3-D accelerator that can control the rendering of each pixel. In other word the stencil buffer is a number bits in the frame buffer that use for enable and disable drawing of each pixel of object in rendering and also it can be use to mask color buffer.

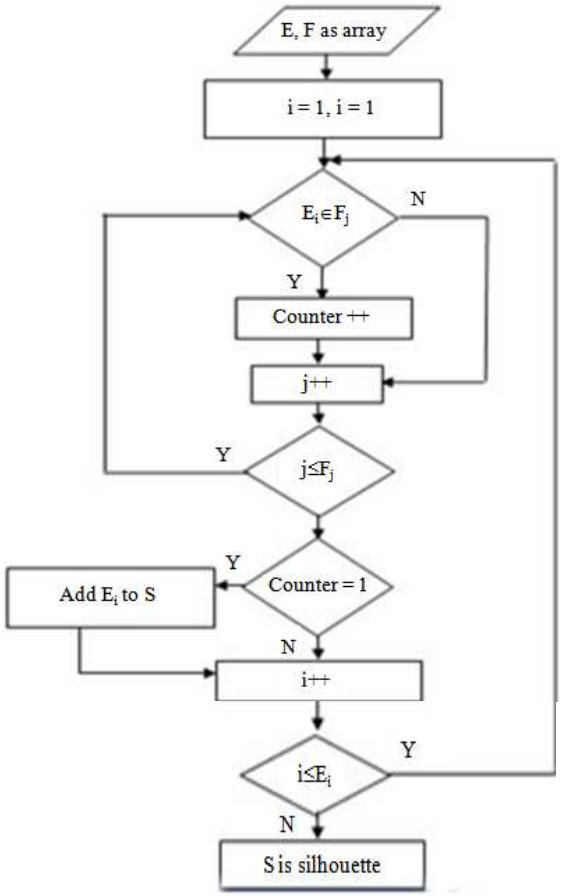

Fig. 3: VNV flowchart

There is a close relationship between stencil buffer and Z-buffer. Because both buffers are, neighborhood and they are located in the graphics hardware memory. Z-buffer needs to control a selected pixel of stencil value that is increased or decreased when we want to count the time of entering and leaving in the shadow volume. Fortunately, the Z-buffer test is after the stencil buffer tests. To get the main idea of Z-pass algorithm e Fig. 4 is provided. The stencil buffer will increase when eye's ray enter the shadow volume by passing from front face of shadow volume and it will decrease when ray leave the shadow volume by passing of back face of that. This has two phases:

1-At first render the front faces.
If (depth test passes)
Stencil Buffer(INCrease)
2-In the second render
If (depth test passes)
Stencil Buffer (DECrease)

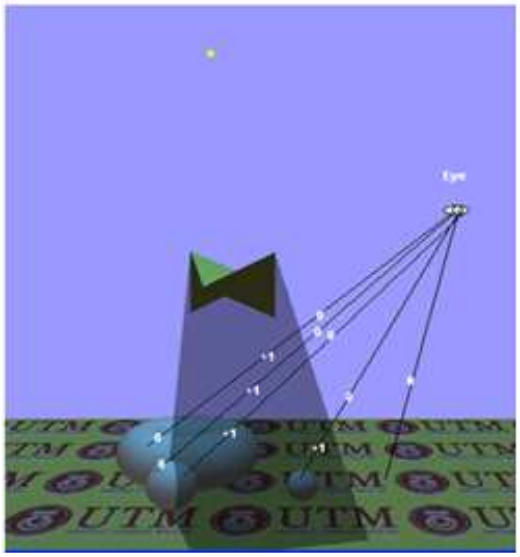

Fig. 4: Z-pass algorithm

Nowadays, Opengl and DiretX support stencil buffer, so we used depth buffer and stencil buffer together. One of the uses of stencil buffer is to limitation of area.

Volume shadow using stencil buffer and VNV algorithm: After produce the volume, 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.

Now we are going to generate volume shadow using stencil buffer and depth buffer together with VNV algorithm. First, in simple word we can say the algorithm in brief:

- Render the whole scene without lighting with this the color buffer will be full for all pints of object in shadow and also Z-buffer will be full with depth value
- After disable Z-buffer and Color buffer to write, render whole scene with lighting should be done
- Subtract of this two depth value provides shadow volume

After generate volume shadow, the ray from eye to the each point of object on the shadow receiver is passed. If the ray passes the front face of volume shadow, the stencil buffer will be increased and when the ray passes the back face of shadow volume, the stencil buffer will be decreased. Finally, if the number of stencil buffer is not zero it is in shadow.

We are going to say this as an algorithm or pseudo code in Opengl:

```
1-Provide the equipment to have stencil buffer
2-Render the all scene without lighting
3-Enable stencil buffer
4 -Disable depth buffer to write and prevent to write in color buffer.
5- VNV algorithm for silhouette detection
6- For Each Plane(P)
    If DotProduct(P,R)>0
        For Each Pixel
            If (Z-test passes)
                Increase StencilBuffer
7- For Each Plane (P)
    If DotProduct(P,R)<0
        For Each Pixel
            If (Z-test passes)
                Decrease StencilBuffer
8-Render the all scene again with lighting
9-Enable to write in color buffer
10-If (the stencil buffer  mod 2=0)
    The pixel is out of shadow
```

Z-pass algorithm is used when eyes are out of shadow. If eyes are located in shadow Z-fail algorithm is used.

Z-pass algorithm is used when eyes are out of shadow. If eyes are in shadow, we should use Z-fail algorithm that is introduced in following:

As a matter of face whole the algorithm is like Z-pass algorithm except the step 6 and 7:

```
6- For Each Plane(P)
    If DotProduct(P,R)<0
        For Each Pixel
            If (Z-test fails)
                Increase StencilBuffer
7- For Each Plane(P)
    If DotProduct(P,R)>0
        For Each Pixel
            If (Z-test passes)
                Decrease StencilBuffer
```

Pseudo code for VNV algorithm is as following:

```
For E=1 to Number Of Edges
    For F=1 to Number Of Faces
        If F.visible =True then
            E_ counter =E_counter+1
        End if
    Next
    If  e_counter=1 then
        Add E to S
    End if
Next
S is silhouettes
```

RESULTS

In each algorithm, FPS is the most important factor to implement. The triangular method and VNV methods are used to recognize silhouette. It is amazing that the VNV method has higher FPS than the triangular method (Yusmawati and Yunus, 2007).

According to Fig. 5 and 6 both algorithms are convenient to implement hard shadow in virtual environment. They can be possible to use for commercial games.

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.

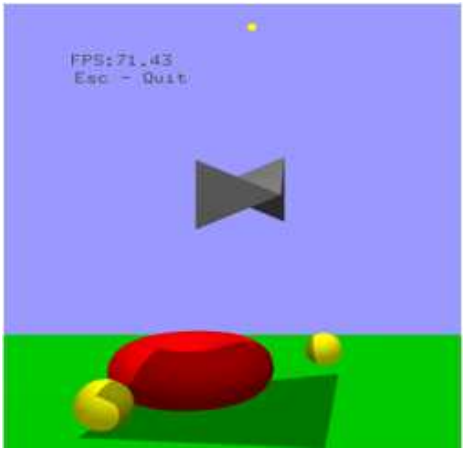

Fig. 5: Volume shadow using triangular algorithm

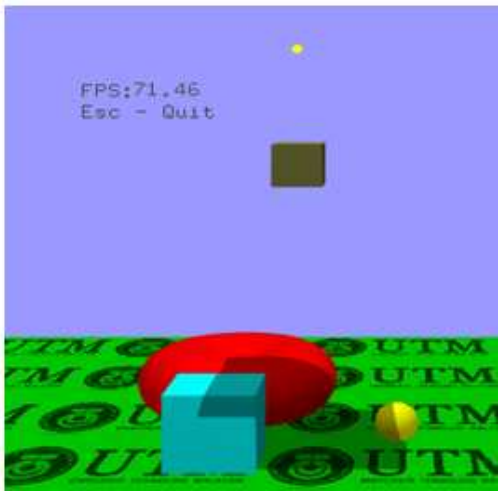

Fig. 6: Volume shadow using VNV algorithm

Table 1: Frame per second

Status	FPS
Without shadow	71.48
Using triangular algorithm	71.43
Using visible-non visible algorithm	71.46

The FPS indicates, volume shadow using VNV algorithm for silhouette detection is better than Triangular detection.

DISCUSSION

To have shadow on arbitrary object such as torus or sphere stencil buffer and Z-pass algorithm are used. In following Table 1 the FPS for using triangular algorithm and VNV algorithm are compared.

The result indicates VNV algorithm is very close to the rendering without shadow. It means, it can be use in current commercial games.

In addition, the following Table 1 illustrates the proposed algorithm for silhouette detection is appropriate for using in real-time shadow in virtual environment.

CONCLUSION

In this study, we have present two method to create real-time shadow. Volume shadow that was difficult to understand and implement we improve it and make it simple to implement. We have proposed the one of simplest ways to implement volume shadow using stencil buffer and depth buffer. Introduce al. To have a real-time shadow on arbitrary objects, volume shadow is convenient method. Volume shadow is geometry computation and it requires supporting of CPU. 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.

To increase the FPS a new method to recognize silhouette is proposed. To create volume shadow, triangular algorithm and VNV algorithm are used.

To make easy of the volume shadow, a pseudo code with VNV algorithm to detect the silhouette and two flowcharts are introduced. In comparison, triangular algorithm has low FPS than the VNV algorithm. To create real-time shadow on arbitrary object volume shadow using stencil buffer and VNV algorithm is convenient. The VNV algorithm is suitable for commercial games.

ACKNOWLEDGMENT

This research was supported by UTMVicubeLab at Department of Computer Graphics and Multimedia, Faculty of Computer Science and Information System, University Technology Malaysia. Special thanks to University Technology Malaysia (UTM) Vot. 00J44 Research University Grant Scheme (RUGS) for providing financial support of this research.

REFERENCES

Andrea, F., G. Andrea and M. Giuseppe, 2010. Rock slopes failure susceptibility analysis: From remote sensing measurements to geographic information system raster modules. Am. J. Environ. Sci., 6: 489-494. DOI: 10.3844/ajessp.2010.489.494

Assarsson, U. and T. Akenine-Moller, 2004. Occlusion culling and z-failfor soft shadow volumealgorithms. Visual Comput. 20: 601-612. DOI: 10.1007/s00371-004-0254-2

Benichou, F. and G. Elber, 1999. Output sensitive extraction of silhouettes from polygonal geometry. Proceedings of the 7th Pacific Conference Computer Graphics and Applications (PCCGA'99), Seoul, pp: 60-69. DOI: 10.1109/PCCGA.1999.803349

Crow, F.C., 1977. Shadow algorithms for computer graphics. Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques, (CGIT'77) ACM New York, USA., pp: 242-247. DOI: 10.1145/965141.563901

Fernando, R., S. Fernandez, K. Bala and D.P. Greenberg, 2001. Adaptive shadow maps. Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (ACCGIT'01), ACM New York, USA., pp: 387-390. DOI: 10.1145/383259.383302

Lokovic, T. and E. Veach, 2000. Deep shadow maps. Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, July 23-28, ACM Press/Addison-Wesley Publishing Co. New York, USA., pp: 385-392. DOI: 10.1145/344779.344958

Olson, M. and Zhang, H., 2006. Silhouette extraction in hough space. Comp. Graph. Forum, 25: 273-282. DOI: 10.1111/j.1467-8659.2006.00946.x

Olson, M., 2010. A survey of silhouette papers. http://www.cs.sfu.ca/~matto/personal/sil_papers.html

Soon, K.J., I.K. Soon and J.K. Ku, 2004. Real-time silhouette extraction using hierarchical face clusters. http://web.kaist.ac.kr/~odinos/HFC_CASE2004.pdf

Yaashuwanth, C. and R. Ramesh, 2010. Web-enabled framework for real-time scheduler simulator: A teaching tool. Proceedings of the 2nd International Conference on Computer Research and Development, May 07-10, Kuala Lumpur, Malaysia, pp: 826-830. DOI: 10.1109/ICCRD.2010.181

Yusmawati, W.Y.W. and W.M.M. Yunus, 2007. Optical properties and kinetic behavior of chlorine in pure water and swimming pool water using surface plasmon resonance technique. Am. J. Applied Sci., 4: 1024-1028. http://www.scipub.org/fulltext/ajas/ajas4121024-1028.pdf

Zhang, H., 1998. Forward shadow mapping. Proceedings of the 9th Eurographics Workshop on Rendering, June 29-July 1, Springer, pp: 131-138. http://www.amazon.com/gp/search?index=books&linkCode=qs&keywords=3211832130