Hybrid Silhouette Detection for Real-Time Shadow Volume

Hoshang Kolivand [1] and Mohd Shahrizal Sunar[2]

1,2 UTM ViCubelab, Department of Computer Graphics and Multimedia
Faculty of Computer Science and Information System, Universiti Teknologi Malaysia
81310, Johor, Malaysia

***Abstract*— In shadow volume, the most expensive computation is silhouette detection. In this paper, the triangular algorithm (TA) and visible-non-visible (VnV) algorithm that are famous algorithms to detect the outline of occluder are renewed. In this paper, we proposed a hybrid algorithm based on TA and VnV, namely Hybrid Silhouette Detection (HSD) algorithm. HSD is an improved algorithm that can recognize silhouette for generating real-time shadow volume. Our algorithm involves detecting silhouette and decreases the cost of implementation for shadow volume rendering. The last shadow volume algorithm using stencil buffer is rewritten and an algorithm for shadow volume using HSD with respect of culling invisible parts of scene is proposed. An accurate mathematical comparison between TA, VnV and HSD algorithms is undertaken. The obtained results confirm superiority of our proposed algorithm in terms of processing and rendering time. Our algorithm can be used in virtual environment to increase the frame per second and to enhance the realistic of games programming.**

Index Terms*—silhouette detection, shadow volume, visibility culling, real-time rendering

I. INTRODUCTION

Nowadays computer games and effects of computer graphics play most important roles in the living. Shadows have most important contribution to make a virtual scene, realistic. In video games, shadows give the gamers feeling that they imagine they play in realistic world and they can enjoy as much as possible. Although the number of algorithms to create shadow is huge, they are suffering from low Frame per Second (FPS).

Real-time is occurring immediately. The real time word is used for some different features. For example, real-time shadows are that respond to input immediately. They are used for such tasks as navigation, in which the computer must react to a steady flow of new information without interruption.

Real time shadows in real-time applications such as computer games and virtual environment are more important to create realistic scenes so that the users will feel realistic in the scenes. In practice, the rendering performance and the quality of shadows generated are important considerations in the selection of shadow algorithm.

Shadow volume that has been proposed by Crow [1] in 1977 is one of the famous algorithms that still in use to create real-time shadow in computer games. Shadow volume is based on silhouette detection because in this algorithm only part of occluder, which has contribution on shadow generation, is outlining of occluder not all parts.

How to recognize the outline of the occluder can improve the speed of algorithm. To find out outline of object, silhouette detection is so important because it can reduce the cost of implementation and it is the main item to improve an algorithm, which needs silhouette detection. Silhouettes are the especial parts of a 3D object outline which only those parts are important to create shadow at different time of rendering. Silhouette detection plays an important phase to create shadow volume. Silhouette detection is visual and view point base. To generate shadow, especially shadow volume, silhouette detection plays a crucial role to detect the boundary of occluder.

II. PREVIOUS WORKS

Silhouette detection is a function that converts a 3D world to 2D. In 1987 Richards, Koenderink, and Hoffman [2] were the first researchers who proposed how recognize the silhouette of arbitrary objects. Saito and Takahashi, in 1990, used geometric buffer for silhouette detection. G-buffer was based on Z-buffer which produced by adding some geometric data.

Hertzmann[3], proposed an algorithm in irregularity surface with lake of continuity and silhouette based on combine of normal map and depth mapping. The hybrid algorithm changes the face of object in using Z-buffer. Raskar and Cohen [4] introduced a method for silhouette detection by changing the front face and back face of the polygon. In 1999, Gooch et al. [5] used a similar method combined with previous ones to recognize the silhouette.

In 2000, Northup and Markosion [6] proposed an algorithm, although it is difficult to find starting points of each coherence, it is very simple. In the Northup's algorithm, spatial coherence is so impressively.

Pop et al. [7] introduced a method for the on-line silhouette finding using frame coherence. For some consequent different viewpoints, in construct to compute the whole silhouettes each time, they tried to calculate the different changes in the silhouette of a polyhedral model between sequence frames. In the each model they transformed the normal vectors to dual line segments and transformed the viewpoint to a dual plane. The dual line segments intersect with the dual plane. One of the drawbacks of their algorithm is difficulty of finding such dual line segments efficiently; therefore, they improved the problem

of the end points finding for the line segments that are consisted in the double-wedge of two dual planes which are related to two close viewpoints.

Isenberg[8], in 2003, classified all silhouette detection techniques in three groups, image space algorithms which focus on image processing , the second group is object space algorithms which are divided in two groups, free form and polygon mesh and final group is hybrid algorithms. In compare by finding accuracy silhouette and appropriate stylization the second group algorithms are better than the [9, 10, 16].

In 2004, Jung et al.[11] introduced an algorithm that used the spatial coherence and frame coherence to gather.

In our algorithm, the average of operation is lower than the others and as a result, it is faster. In the traditional volume, shadow, which used to triangular silhouette is operated the new technique and the result is high frame per second.

In the following after introduce the silhouette and to famous techniques to recognize, occluder's outline, the number of multiple, addition, and comparison is calculated. Finally an accurate comparison between them has been done.

In 2011, Kolivand and Sunar [12] proposed a geometrical algorithm to recognize the silhouette detection. An accurate comparison between current algorithm and their algorithm have done.

Main Contributions: We have presented a new algorithm for silhouette detection to create shadow volume in complex environments. Our algorithm is not completely geometrically based. A hybrid approach can improve the pure geometrical base algorithm to reduce the cost of rendering.

New Result: Some new aspects of present work are:

1. To propose a novel algorithm for silhouette detection.
2. An improved FPS in generating hard shadow that can be used to accelerate volume shadow and hybrid shadow generation.

We have compared our new algorithm with other famous silhouette detection algorithms. We could increase the FPS from 69.93 to 75.53 using the proposed algorithm to create shadow volume.

Organization: The rest of the current study is organized as follows: In the next section, an overview of related works is introduced. Section 3 is assigned to introduce the concept of silhouette detection. Section 4 provides some material about the conventional silhouette detection techniques. We have proposed our new algorithm in this section 5. In section 6, is allocated for estimate calculation of conventional techniques and our technique in three subsections. A discussion and comparison between other related algorithms and our algorithm to create volume shadow based on number of operators have been done in section 7. A brief conclusion and some advice for future work are suggested in section 8.

III. Silhouette Definition

Silhouette is some parts of occluder outline that can be seen from light source point without respect to the color. Silhouettes have a most important role to recognize and project shape onto shadow receiver. Because to create shadow, projection of occluder silhouette is enough to generate shadow of object and as a result, the cost of projection will be low. A silhouette edge of polygon is some edges of the polygon that are belong to two neighborhood plates which normal vector of one of them is toward the light and normal vector of the other plate is away from the light. To speed up the rendering shadow volume uses the silhouette not point by point of object to create shadow. Creating shadow using point by point, needs a lot of calculation and then it takes a substantial time of CPU to rendering. Using silhouette edges of the occluder to generate a volume shadow will be optimized because in this case the amount of memory is decreased, therefore, render will be done faster. The silhouette should be recalculated when position of the light source changes or occluder moves.

Now with the sign of ||v,f|| can recognize, is the face back face or front face [3].

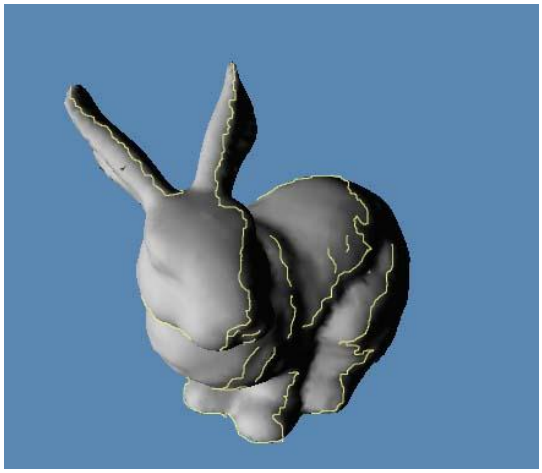

Fig. 1. Silhouette detection

In Fig 1 the yellow line shows the silhouette of the bunny from light source point which is located at the left side of rabbit. To create shadow only the silhouette contribute not whole object.

If dot product (v, f) >0 then

f is a front face

Else

f is a back face

End if

If dot product (v, e_{f_1}) dot product (v, e_{f_2}) <0 then

e is a silhouette

End if

IV. Conventional Silhouette Detection Algorithms

Although, there are many silhouette detection algorithms, they are suffering of low speed. To increase the FPS a comparison between them is done. A new improvement on triangular algorithm is proposed. First, it is necessary to rewrite the famous and useful algorithm, as they will be easy to understand.

4.1 Triangular Algorithm

Triangular algorithm (TA) is a widely use algorithm to recognize the silhouette of an occluder that propose by Harlen and Ilaim [13] in 1999, in this algorithm all surface of occluder must be divided onto triangles mesh. It means that each edge must be shared by just two triangles. To determine which edge is silhouette, it is needed to know which edge is shared between two faces, one toward the light source and one away to the light source.

A pseudo code of triangular algorithm is introduced as follow:

```
S : Array of edges=0;
For (p=1;p<= Number of Faces; p++){
    If (p.visible){
      For (q=1; q<= Number of Side; q++){
        If (v_p v_q^-1 belongs to S){
          Delete vpvq from S
        Else
          Add v_p v_q to S
          }
       }
    }
}
  Cout << S as all silhouette at the current time
```

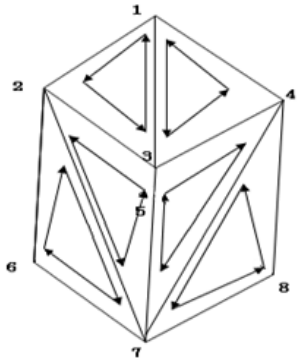

Fig. 2. Triangular mesh

In this algorithm before starting, all faces of occluder must be divided into some triangles and all of the triangles edges must be saved in an array (S). Then, from the first of the array, if inverse of each edge belongs to the S, it is not silhouette and it must be omit form the array. Otherwise, if it is not belong to the array, it is silhouette and it must be saved in the array, until end of array.

4.2 Visible-none Visible Algorithm

In 2004, Jung [10] proposed a very simple algorithm for silhouette detection. Although the algorithm is easy in structure, is not convenient in speed rendering. An overview of this algorithm is belonging to two visible or invisible faces. Every edge that belongs to only one visible face is a silhouette. These edges, which exactly belong to one visible and one invisible face are silhouette, but on the contrary, each edge that belongs in two visible faces or belongs in two invisible faces is not a silhouette.

A pseudo code of Visible-none Visible (VnV) algorithm is introduced as follow:

```
For (e=1; e<= Number of Edges; e++){
    For (f=1; f<= Number of Faces; f++){
      If (f.visible){
        e_ counter =e_counter+1
      }
    }
    If (e_ counter =1) {
      Add e to S
}
Cout<< S as silhouette    in the current time
```

The algorithm is very easy to implement. Suppose that the light source is located in the viewer point of eyes. The edge *ad* in Fig 3 belongs to one visible face (*abcd*) and one invisible face (*adeh*) then it is a silhouette.

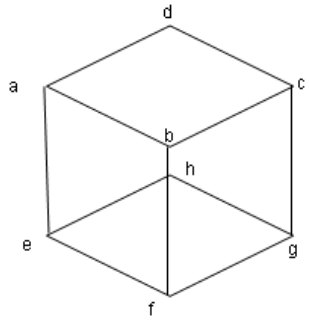

Fig. 3. Silhouette detection for Visible-none Visible algorithm

The edge of *ab* is belonging to two visible faces (*abcd* and *abfe*) then it is not a silhouette and also *hg* belongs to two invisible faces (*dcgh* and *gfeh*) and it is not a silhouette.

V. Hybrid Silhouette Detection Algorithm

In this part, we are going to propose an improvement on triangular algorithm to recognize silhouette detection. In the triangular silhouette detection, each faces must be divided to some triangles. In some cases there are many triangles which have same normal vectors. With respect of light source position, all same normal vectors have same status. For example in a faces with *n* vertices, there is *n-2* triangles and all of the have same normal vectors. If one of these triangles is visible from

light source, all can be seen from light sources. It means none of them is as silhouette.

In our approach, there is no need to convert all faces to triangles. Each face that is flat or each part of occluder that has same normal vectors in one side of occluder it is no need to convert into the triangles. In this case, the number of triangle mesh will be decreased and as a result, the speed of rendering will be increased. We called it Hybrid Silhouette Detection (HSD).

This ratio is based on Chris L. Gorman's idea that he has told ratio of vertices to face is 2 and also the ratio of the number of edges to number of vertices is [14].

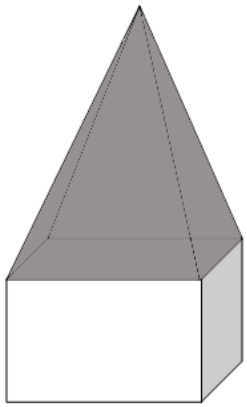

Fig. 4. Hybrid Silhouette Detection

VI. Estimate Calculations

6.1 Estimate Calculation of Triangular Algorithm

The normal face of a plane is needed, to calculate the visibility of each face. It is needed to have a ray of light source into the plane and sign of dot product must be calculated. To determine if an edge is belonging to the array a comparison should be done.

To check all number of faces f_n calculation must be done that f_n is number of faces. To determine a face p is visible or not a ray from eyes point of view is needed. A normal vector of face is needed and by calculate of dot product and sign of result:*3 Adds+ 3 Mults+ 4 Mults+3 Adds+3 Mults +2 Adds+1 Compare +8 Read* are needed respectively.

Assuming average side of each face is m, then to recognize $v_pv_q^{-1}$ is belong to S. e_n *Compare +1Inverse +*e_n *Read* are needed. To delete each edge (v_pv_q) from S, *1 read +2 write* and to add an edge(v_pv_q), *1 write+1 read+2 write* are needed respectively.

The number of faces is f_n. To determine whether a plan p is visible the normal vector of faces should be calculate and it needs *3Adds and 3Mults*. To obtain a ray from light source to the plan, it needs *4Mults and 3Adds* and finally the dot product needs *3Mults and 2Adds*. To determine if the dot product is positive or negative, one compare is require.

The average number side of each plane is considered m. to determine whether *vpvq-1* is belong to S, first needs to calculate *one Inverse* or *1Mults* and follow that search in the array which needs e_n Compares and also all of them should be read. Then estimate calculations of triangular algorithm are as follow:

$$f_n(8A+10M+1C+8R+(m(e_nC+e_n\,R+1M+1R+2W)))$$

$$=n(8A+10M+1C+8R+6nmC+6nmR+1mM+1mR$$

$$+2mW)$$

$$=6n^2mC+(6n^2m+\ mn+8n)R+(m+10)nM+8nA+2mnW$$

6.2 Estimate Calculation of Visible-No visible Algorithm

This algorithm is visible and invisible base. It means the visibility of plans is important. To calculate the visibility of each face, the normal face of plane is needed to and also it is deeded to have a ray of light source into the plane and sign of dot product must be calculate.

Suppose that e_n is number of all edges in face f and f_n is number of all faces in the occluder. Again, to determine a face p is visible or not a ray from eyes point of view is needed. A normal vector of face is needed and by calculate of dot product and sign of result:*3 Adds+ 3 Mults+ 4 Mults+3 Adds+3 Mults +2 Adds+1 Compare +8 Read* are needed respectively. *1 add* for a counter is needed to keep number of all visible faces for each edge. Finally, *1 Compare + 1 Read* to check the number of visible faces is needed. *1 write* requires to add a new silhouette edge to S. Then estimate calculations of visible-none visible algorithm are as follow:

$$e*(f*(10M+8A+1C+8R+1A)+1C+1R+1W)$$

$$=6n*(n*(10M+9A+1C+8R)+1C+1R+1W)$$

$$=60n^2M+54n^2A+(6n^2+6n)C\ +(48n^2+6n)R+6nW$$

6.3 Estimate Calculation of Hybrid Silhouette Detection Algorithm

It is no need to convert all faces to triangles. Each flat face or each part of occluder that has same normal vectors in one side of occluder no need to convert into the triangles. It means the some calculations that related to number of sides in each faces must be decreased.

The first part of calculation is same as traditional triangular algorithm especially to calculate the visibility of faces. The number of faces is f_n. To determine whether a plan p is visible the normal vector of faces must be calculate and it requires *3Adds and 3Mults*. To obtain a ray from light source to the plan, it needs *4Mults and 3Adds* and finally the dot product needs *3Mults and 2Adds*. To determine if the dot product is positive or negative, one compare is require.

The average number side of each plane is considered m. In the Hybrid Silhouette Detection (HSD) this number will be decreased by 1. Thus estimate calculations of HSD algorithm are as follow:

$$f_n(8A+10M+1C+8R+(1(e_nC+e_n\,R+1M+1R+2W)))$$

$=n(8A+10M+1C+8R+6nC+6nR+1M+1R$

$+2W)$

$=(6n^2+n)C+(6n^2+9n)R+11nM+8nA+2nW$

VII. Results and Discussion

Figure 5 and 6 illustrate the comparison between HSD algorithm with the other algorithms with n vertices, f_n faces, e_n edges and m vertices in each faces.

The charts reveals if the occluder is geometrical and number of edge in each face increased the speed of HSD algorithm will be increased dramatically. It means HSD algorithm is more convenient for wide scene with lots of cubes such as buildings. It can be possible to use the algorithm in driving simulator.

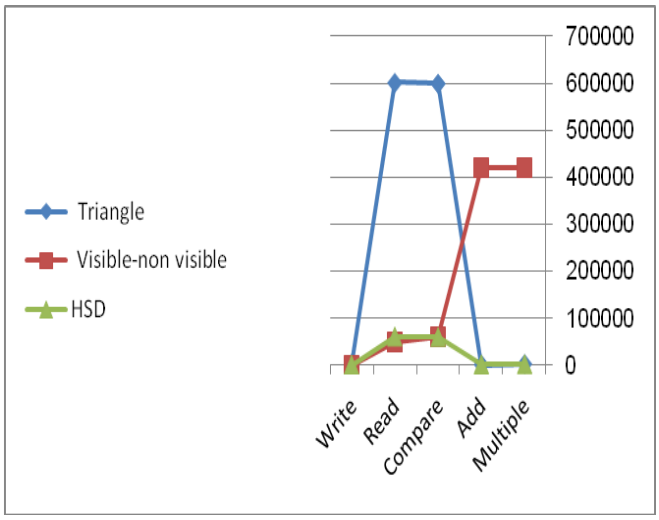

Fig. 5. Complexity with n=10,m=5

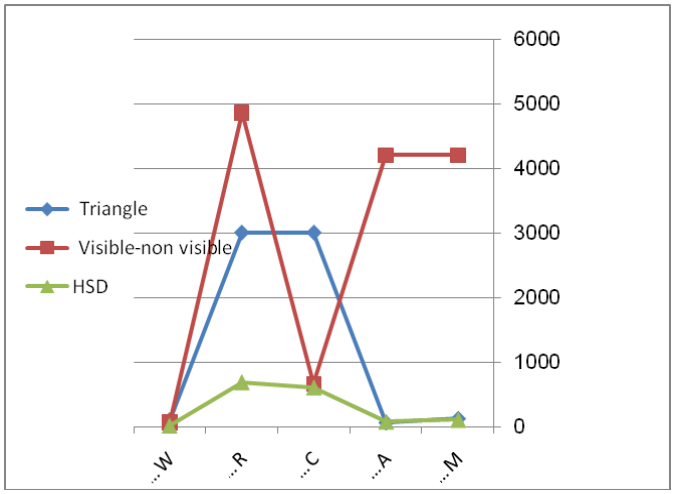

Fig. 6. Complexity with n=100,m=10

Fig 6 illustrates number of write in memory for all algorithm are same. More different is related to number of reading. Number of compare for TA and HSD are same but more different is in number of Adding, Multiplying and reading. VnV and HSD are same in number of writing, adding and multiplying. More different is related to number of comparing and reading

A comparison between Fig 5 and Fig 6 reveals when the number of faces increases the HSD will have better result. For example, by consideration of CPU's cycling for each operators, when the number of faces is more than two million, the sum of all cycling in TA is five times and VnV is six times of HSD. It shows HSD algorithm is convenient for complex virtual environments. It can be possible to use proposed algorithm in commercial games and virtual realities.

VIII. Shadow Volume Synthesis Using Hybrid Silhouette Detection Algorithm

Shadow volume is one of the main algorithm to create shadow in virtual environments that proposed by Crow [1]. To create show volume, the first step is to recognize outline of occluder or in the other word is silhouette detection. In this method after determination of silhouette, draw a ray from light source point to all points of silhouette and continue to infinity. A volume will be appeared between the occluder and shadow receiver. Each object of part of object that is located inside of the volume is in shadow and conversely each object or part of object that is located outside of the volume is in lit.

The stencil buffer is one of the require tools to create shadow volume. Z-pass algorithm is needed to recognize a pixel is located inside of volume or outside.

Culling is another concept that can increase the speed of rendering especially in wide scenes such as a city. In this case, some parts of scene that cannot be seen by point of view should be cut [15, 17]. In our approach, some parts of calculation to find which part should be culled will be saved because in silhouette detection it did. In this case, all faces that have normal vectors toward away of light source should be culled.

We are going to say these all as an algorithm using HSD algorithm for silhouette detection:

I. *Render the all scene without lighting*
II. *Enable stencil buffer*
III. *Enable Cull facing for back faces.*
IV. *Disable depth buffer to write and prevent to write in color buffer.*
V. *Render the all scene again with lighting.*
VI. *Enable to write in color buffer.*
VII. *If (~stencil%2), keep stencil*

Table 1. Number of edges in different data

	Number of edge
Cube	12
Sphere	30
Plane	100
Bunny	2207

The following result is obtained in a PC with NVIDIA Graphic Hardware and P IV 2.8GH. When HSD algorithm is used with bunny data, the FPS is 75.53 in compare to

conventional method that is maximum 67.54 FPS in the same PC and with different quality.

Fig 7 shows HSD can solve some parts of crucial problem of shadow volume. When the number of faces increases, the result of HSD is better that the conventional algorithms. It means the proposed algorithm is suitable for complex scenes.

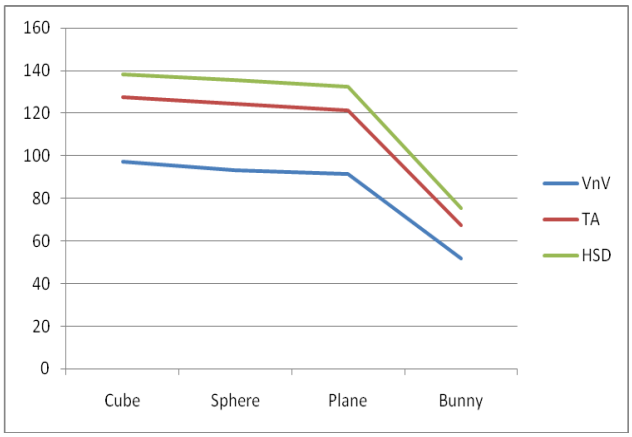

Fig. 7. A comparison between conventional methods (Triangular Algorithm and Visible none Visible algorithm) and Hybrid silhouette detection method to create shadow volume

Fig 8 shows the truncate volume between occluder and shadow receivers. Each object or part of object that is located inside the truncate volume is in shadow.

IX. Conclusion and Future Work

Shadow volume is one of the best techniques to create precise hard shadow in virtual environments. Stencil shadow volume can be used to have shadow on the arbitrary objects. The most expensive part of shadow volume is silhouette detection. Many algorithm tried to reduce the cost of execute. Triangular Silhouette Detection and Visible none- Visible algorithm are the best current ones to recognize the silhouette.

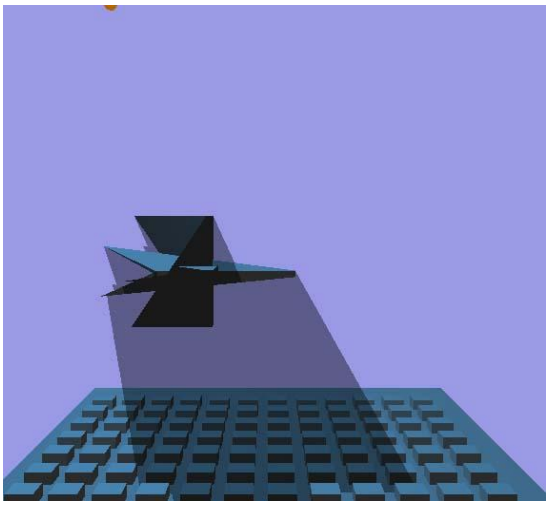

Fig. 8. Stencil shadow volume generation. Generating a truncate shadow volume between occluder and shadow receivers (plane and cubes)

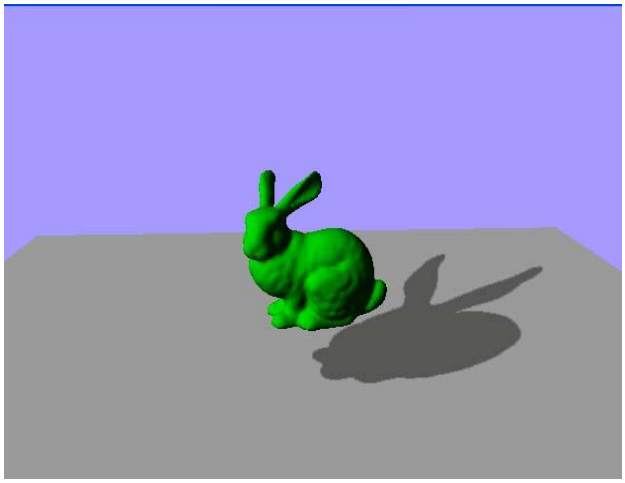

Fig. 9. Shadow volume generation using Hybrid Silhouette Detection with 75.53Freame per Second.

Hybrid Silhouette Detection is a new algorithm to recognize silhouette of object to create real-time shadow volume. In this algorithm, we have tried to remove some parts of triangular algorithm that do not have any contributions in shadow generation. This improvement increases the frame per second in compare with Triangular and Visible none-Visible algorithm. Base of this algorithm is related to mathematics and visualization. The benefit of new algorithm will be revolving where the number of occluder's edge increased. This algorithm is also convenient for complexity scene. To use this algorithm in shadow volume generation, stencil buffer and Z-buffer are so appropriate tools.

Silhouette detection is a crucial problem in shadow volume generation. It is so expensive and needs more improvements. Although our algorithm is faster than the previous ones, it is weak when some parts of occluder are located outside of camera frustum. More improvement on culling can solve this problem.

Acknowledgment

This research was supported by UTMVicubeLab at Department of Computer Graphics and Multimedia, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia. Special thanks to Universiti Teknologi Malaysia (UTM) Vot. J13000.7282.4F085 FRGS for providing financial support of this research.

REFERENCES

[1] F. Crow, Shadow Algorithms for Computer Graphics, Computer Graphics vol. 11,no. 2, pp. 242-247,1977.

[2] W. Richards, J. Koenderink and D. Hoffman, Inferring Three-Dimensional Shapes from Two-Dimensional Silhouettes, Journal of Optical Society of America, 1168–1175, 1987,

[3] A. Hertzmann, Introduction to 3D Non-Photorealistic Rendering: Silhouettes and Outlines,1999.

[4] R. Raskar and M. Cohen, Image precision silhouette edges. In: Spencer SN, editor. Proceedings of the conference on the 1999 symposium on interactive 3D graphics. New York: ACM Press. p. 135-140, 1999.

[5] B. Gooch, P.J. Sloan, A.Gooch, P. Shirley and R. Riesenfeld, Interactive Technical Illustration, In Proceedings of the 1999 Symposium on Interactive 3D Graphics, Atlanta, Georgia, United States, April 26-29, pp. 31-38, 1999.

[6] J. D. Northrup and L. Markosian. Artistic silhouettes, A hybrid approach. In Proceedings of NPAR, 2000.

[7] M. Pop, G. Barequet, A. Duncan, M. T. Goodrich, W. Huang, and S. Kumar, Efficient perspective–accurate silhouette computation and applications, In COMPGEOM: Annual ACMSymposium on Computational Geometry, 60–68, 2001.

[8] T. Isenberg, B. Freudenberg, N. Halper, S. Schlechtweg, and T. Strothotte, A developer's guide to silhouette algorithms for polygonal models, IEEE CG & A, pp. 28-37, July/August 2003.

[9] F. Benichou and G. Elber. Output sensitive extraction of silhouettes from polygonal geometry. In Proc. of Pacific Graphics 1999, pp. 60-69, october 1999.

[10] J. Buchanan and M. Sousa. The edge buffer, A data structure for easy silhouette rendering. In Proc. of NPAR 2000, pp. 39-42, june 2000.

[11] S.K. Jung, S.I. Kwon and K.J. Kim, Real-time Silhouette Extraction Using Hierarchical Face Clusters, 2004

[12] H. Kolivand, M.S. Sunar, New Silhouette Detection Algorithm to Create Real-Time Volume Shadow, Workshop on Digital Media and Digital Content Management(IEEE) , 270-274, 2011

[13] H.C. Btagelo and I.C. Junior, Real-Time Shadow Generation using BSP Trees and Stencil Buffers, XII Brazilian Symposium on Computer Graphics and Image Processing, 93-102, 1999.

[14] A. –Y. Wang, M. Tang and J.-X. Dong, A survey of silhouette detection techniques for nonphotorealistic rendering, Proceedings of Third International Conference on Image and Graphics, Hong Kong, 434-437, 2004.

[15] M.S. Sunar, M.A.M. Azahar, M.K. Mokhtar and D. Daman, Crowd Rendering Optimization for Virtual Heritage System, The International Journal of Virtual Reality, vol. 8, no. 3, pp. 57-62, 2009.

[16] I. Ismail, M.S. Sunar, M.K.M Sidik, C.S. Yusof, A Review of Dynamic Motion Control Considering Physics for Real Time Animation Character, Workshop on Digital Media and Digital Content Management, 86-90, 2011

[17] L. Pujol-Tost, Realism in Virtual Reality applications for Cultural Heritage, The International Journal of Virtual Reality, vol.10, no.3, pp. 41-49, 2011

Authors

Hoshang Kolivand is now pursuing Ph.D in UTM ViCubelab under the supervision of Dr Mohd Shahrizal Bin Sunar. His research interests include Computer Graphics and Virtual Environment. Received the M.S degree in Applied Mathematics and computer from Amirkabir University, Iran in 1999. B.S degree in Computer Science & Mathematic from Islamic Azad University, Iran in 1997. Previously he was a lecturer in Shahid Beheshti University Iran. He has published enormous articles in international journals and conferences as well as national journals, conference proceedings and technical papers including article in a book. Hoshang Kolivand is an active reviewer of some conferences and international journals. He had published many books in object-oriented programming and one in mathematics.

Mohd Shahrizal Sunar obtained his PhD from National University of Malaysia in 2008. His major field of study is real-time and interactive computer graphics and virtual environment. He received his MSc in Computer Graphics and Virtual Environment (2001) from The University of Hull, UK and BSc degree in Computer Science majoring in Computer Graphics (1999) from Universiti Teknologi Malaysia. He served as academic member at Computer Graphics and Multimedia Department, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia since 1999. Since 2009, he had been given responsiblity to lead the department. The current research program that he lead are Driving Simulator, Augmented Reality, Natural Interaction and Creative Content Technology. He had published numerous articles in international as well as national journals, conference proceedings and technical papers including article in magazines. Dr. Shahrizal is an active professional member of ACM SIGGRAPH. He is also a member Malaysian Society of Mathematics and Science.