A Fast Silhouette Detection Algorithm for Shadow Volumes in Augmented Reality

Hoshang Kolivand, Mahyar Kolivand, Mohd Shahrizal Sunar, Mohd Azhar M. Arsad

Abstract—Real-time shadow generation in virtual environments and Augmented Reality (AR) was always a hot topic in the last three decades. Lots of calculation for shadow generation among AR needs a fast algorithm to overcome this issue and to be capable of implementing in any real-time rendering. In this paper, a silhouette detection algorithm is presented to generate shadows for AR systems. Δ+algorithm is presented based on extending edges of occluders to recognize which edges are silhouettes in the case of real-time rendering. An accurate comparison between the proposed algorithm and current algorithms in silhouette detection is done to show the reduction calculation by presented algorithm. The algorithm is tested in both virtual environments and AR systems. We think that this algorithm has the potential to be a fundamental algorithm for shadow generation in all complex environments.

Keywords—Silhouette detection, shadow volumes, real-time shadows, rendering, augmented reality.

I. Introduction

SILHOUETTE is the most important part of geometrically based shadow generation techniques. Silhouette is the only part of occluders which contributes in shadow generation. Silhouettes consist of the edges of occluders which belong to two different surfaces in which one of them can be seen from the light source position and the other one cannot. In other words, the normal vector of each surface is towards the position of light source, while the other is not. The first usage of silhouette is referred to Shadow Volumes [6] which is the fundamental research in geometrically based shadow generation techniques. The main issue of the geometrically based shadow generation techniques is silhouette detection which requires a lot of calculations. Nevertheless, improving silhouette detection is in order to enhance the frame per second and quality of shadows especially in AR [9] which will be employed in real-time. In these cases, when the the occluders move or light source position changes, the silhouette will be recalculated in real-time.

Many researchers have spent time to improve shadow volumes based on silhouette detection [1], [7], [4], [10]. Fig. 1 illustrates the concept of silhouette detection. It means that the silhouette of occluders with respect to the light source is important in constructing the shadows. There are two widely used algorithms in silhouetted detection for shadow volumes. Batagelo et al. [3] proposed Hierarchical Face Clusters (HFC)

H. Kolivand, M. S. Sunar, and M. A. Arsad are with the MaGIC-X (Media and Games Innovation Centre of Excellence) UTM- IRDA Digital Media Centre Universiti Teknologi Malaysia 81310 Skudai Johor Malaysia (e-mail: hoshang@utm.my, shahrizal@utm.my, azhar@magicx.my).

M. Kolivand is with the Seri Omega International School Lot 6974, Jalan Beringin 5 Taman Beringin 81400 Senai, Johor, Malaysia (e-mail: shahinshah_kolivand@yahoo.com.)

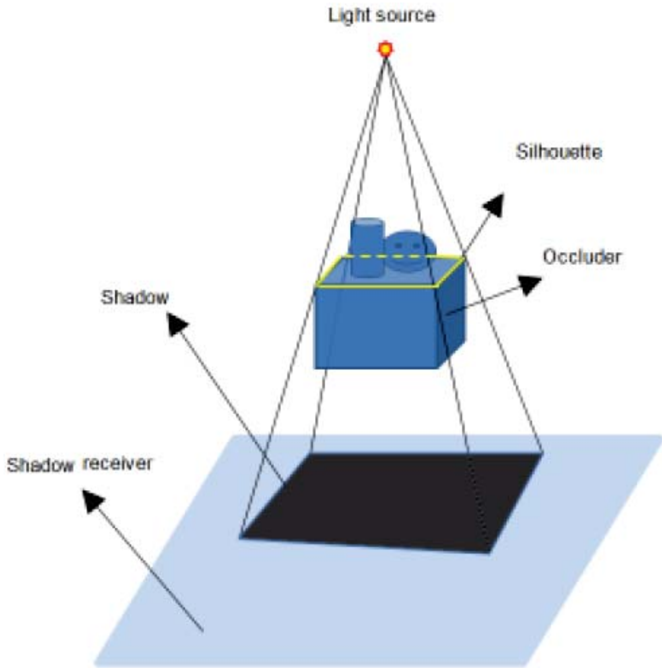

Fig. 1 Silhouette detection

algorithm and Jung et al. [7] presented BSB Tree algorithm. Both of them perform with $O(n^2)$.

Shadow volumes are implemented as a part of outdoor rendering environment to construct the precise shadow on arbitrary objects. Shadow volumes are expensive but the only reason to use these types of shadows is their accuracy without any aliasing. The theory of shadow volumes is illustrated in Fig. 2. The volume between the occluder and shadow receiver is constructed based on the silhouette of the occluders.

The widely used techniques such as BSB Tree and HFC algorithms are difficult to be improved as many years have passes and they are still in use. Thus, the techniques are appreciated. In this paper, we have presented an algorithm called Δ+ Algorithm to speed up the running time for shadow volumes to be employed in real-time rendering like AR.

II. Silhouette Detection

As mentioned early, silhouette of a blocker is the only part of object which contributes in shadow generation. Thus, to create shadows, silhouette of the occluders must be recognized first. This step needs more improvements to enhance the speed of rendering. Calculation of this detection is almost expensive and it needs to be taken into consideration for

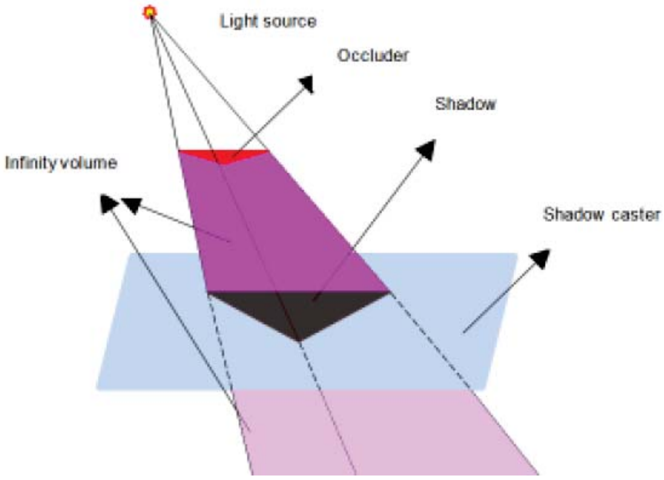

Fig. 2 Theory of shadow volumes

more improvements. In this paper, we have presented an idea to reduce the calculation of this step in shadow generation geometrically and then apply it in AR to be tested in a real-time environment. To begin with, we need to analyze the fundamental concept of silhouette detection in shadow generation techniques.

Given that:
E is the view point which in located in the camera point of view
f_i is i^{th} face of F which is the set of all faces of the occluder
n_{f_i} is normal vector of f_i
v_f is a vertex of f
n is a edge of the occluder in the set of E
e_{f_i} i^{th} face includes e
p_f is a surface consists f

The distance between E and f is:

$$\|E, f_i\| = (E - v_{f_i}).n_{f_i} \tag{1}$$

Now, if $\|E, f_i\|$ is positive, then the face f_i is front face, otherwise, it is a back face. Face recognition algorithm is presented in Algorithm 1.

Algorithm 1 Face Recognition

Step 1. If $\|E, f_i\| > 0$ then
f_i is a front face
Else
f_i is a back face
End if
Step 2. If $\|E, e_{f_1}\| \, \|E, e_{f_2}\| < 0$ then
e is a silhouette
End if

Normalization is suitable for smooth edges which is presented in (2). A simple theory of silhouette detection is provided in Algorithm 2.

$$D_i = \frac{(f_{v_i} - v)n_{f_i}}{\|n_{f_i}\| \, \|f_{v_i} - v\|} \tag{2}$$

Algorithm 2 Silhouette Theory

Step 1. If $D_i > 0$ then
f_i is front face
Step 2. if $D_i < 0$
f_i is back face
Step 3. $D_i = 0$ then
v_i is a silhouette
end if

III. Δ+Algorithm

The regular and basic technique for silhouette detection is by going through all edges and checking one by one whether it is a contributor in shadow generation with respect to the light source or not. The algorithm is not complicated but expensive in terms of rendering. Here, we have presented an algorithm to over come this issue by optimizing the number of edges which need to be checked. The idea behind Δ+Algorithm comes form image based techniques [14], which compare two different view points; but, here we extend the ray from light source view point and compare to the other view point which is camera point of view.

Δ+Algorithm is geometrically-based and it is used to figure out the silhouette in the shorter period of time compared to the current techniques. Δ+Algorithm is presented in the Algorithm 3.

Algorithm 3 Δ+Algorithm

Step 1. Calculate the number of faces and edges of occluder in advance
Step 2. For all edges
Step 3. Extend the edge by ϵ to a new point with respect to the light source position
Step 4. Extend the Ray from light source to ϵ which will be called extended point
Step 5. For all visible faces
Step 6. If the ray and faces have any intersection, the edge is silhouette, otherwise no

To evaluate technically, Δ+ Algorithm needs to be explored. Given that the light source is L and the camera point is C, consider pq is a single edge in E. Extend of p is shown as $p + \Delta p$ where $\Delta p \to 0$. In the next step, the ray (R) which is drown from light source to the vertex of p needs to be extended to $p' = p + \Delta p$ and then extends to infinity. If the extended ray intersects a face, pq is not a silhouette, else it is.

Algorithm 4 Δ+ Algorithm (Details)

Step 1. For each vertex of pq in E
Step 2. $p' = p + \Delta p$
Step 3. Extend R=ray of light source to p'
Step 4. For all visible (f) Faces
Step 5. If Intersection(R, f)=false then
Step 6. pq is a silhouette

In calculation of extend edge, there are 3 summations. To determine the intersection between extended ray and faces,

TABLE I
COMPLEXITY OF ALGORITHM

Line	Worst case
Step 1. For all vertices of pq in E	n
Step 2. $p' = p + \Delta p$	1
Step 3. R=Ray of light source to p'	3 Multiples+3 Sumations
Step 4. R=Ray of light source to p'	$Log_2 n$
Step 5. If intersection (R,f)=false then	1
Step 6. pq is a silhouette	1

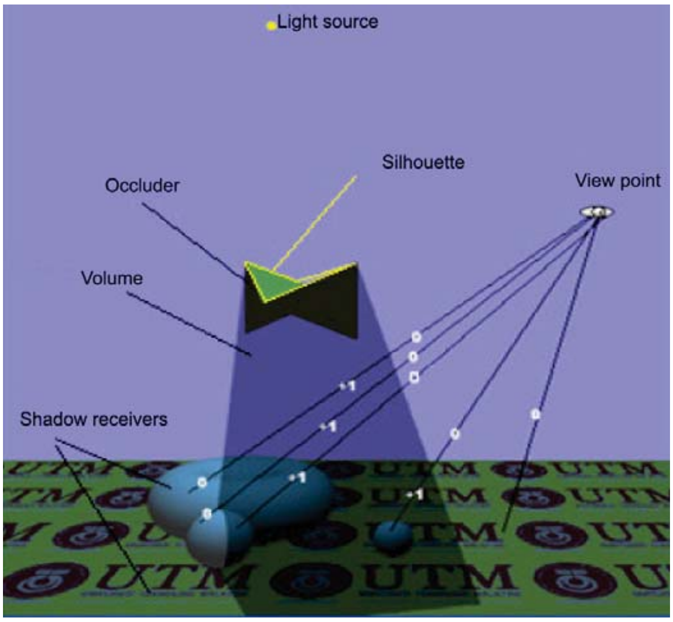

Fig. 3 Illustration of stencil buffer in shadow volumes taking silhouette detection into account

the algorithm puts the ray in the surface equation and checks whether the intersection happens or not. Therefore, given that $x = x0 + \alpha n$ is the ray; to calculate this ray 3 Multiples and 3 Summations are needed. Given that the number of faces in the occluder is F, Table I shows the complexity of Δ+ Algorithm. Then Complexity = $O(nLog_2 n)$

IV. Δ+ ALGORITHM IN SHADOW VOLUMES

Here, we have created shadow volumes using stencil buffer. Silhouette detection is the main part of this algorithm. Δ+ Algorithm recognizes the silhouette as can be seen in Fig. 3.

To use Δ+ Algorithm in shadow volumes, the first step is creating the volume between occluder and shadow receivers. Then, the rays from light source to the vertex are given and extends to the shadow receivers are calculated. The regular process of shadow volume is applied. Meanwhile, when the ray inters into the front face, the stencil buffer is increased, when the ray exits form the back of the face the stencil buffer is decreased. At the end of this process, the amount of stencil buffer is checked. If the stencil buffer is zero the point is in lit else in shadows. The algorithm summarized in Algorithm 5.

If the camera point is out of the shadow region Z-pass algorithm is employed, while, Z-fail algorithm is used when it is inside the shadow region. In fact, steps 5 and 6 will be changed with following steps:

Algorithm 5 Shadow volume generation using Δ+ Algorithm

Step 1. Render the whole scene when light is off
Step 2. Disable depth buffer
Step 3. Enable stencil buffer
Step 4. $\sum_{e=1}^{NE}\sum_{f=1}^{NF} IIf(not(intersection(R_{e_v}, f)), Add\ to\ S, Nil)$
Step 5. $\sum_{i=1}^{Np} IIf(P_i, R_v, \sum_{j=1}^{NP} IIf(ZTest, Stencil++, Nil), Nil)$
Step 6. $\sum_{i=1}^{Np} IIf(Not(P_i), R_v, \sum_{j=1}^{NP} IIf(ZTest, Stencil--, Nil), Nil)$
Step 7. Render scene again with lighting
Step 8. Enable to write in color buffer
Step 9. If (Stencil mod 2=0) then Keep Stencil

NE: Edge number
NF: Face number
NP: Polygon number
S is array of edges
P_i: i^{th} polygon
R_v: a ray from light to v
S: Array to keep Silhouette edges

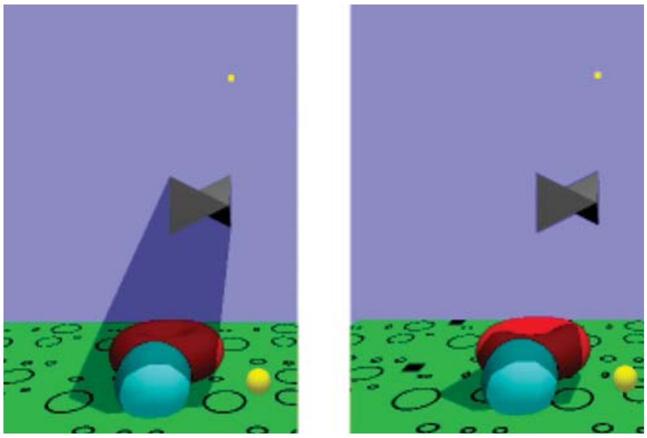

Fig. 4 Shadow volumes using Δ+ Algorithm

Step 5: $\sum_{i=1}^{Np} IIf(Not(P_i), R_v, \sum_{j=1}^{NP} IIf(ZTest, Stencil++, Nil), Nil)$
Step 6: $\sum_{i=1}^{Np} IIf(P_i, R_v, \sum_{j=1}^{NP} IIf(ZTest, Stencil--, Nil), Nil)$

Fig. 4 is the result of shadow volumes using Δ+ Algorithm with 123.25 FPS with one light source. It is 69.93 FPS for BSB Tree algorithm [3] and 65.51 for HFC algorithm [7]. All algorithms are implemented on a 2.5 GHz Intel(R)HD Core(TM) i5-3210 CPU using an ATI Radeon HD 7670M Graphics 4000.

Three different algorithms are employed to demonstrate that the Δ+ Algorithm is faster than BSB Tree algorithm [3] and HFC algorithm [7]. This claim can be also proven by Big-O notation beyond FPS. As shown in other researches, BSB Tree algorithm and HFC algorithm have the same efficiency of $O(n^2)$ while the efficiency of Δ+ Algorithm is $O(nLog_2 n)$. Nevertheless, if the number of vertices of an occluder is 5000, then complexity of BSB Tree algorithm and HFC algorithm will be 25,000,000 while it is 61438 for Δ+ Algorithm. In other words, Δ+ Algorithm is 407 time more effective compared to current fundamental algorithms for silhouette

Fig. 5 View 1: Result of Δ+ algorithm in in AR

detection [5], [13]. It should be mentioned that these two algorithms are currently widely used in silhouette detection due to the fundamental concepts. Table II shows a simple comparison between the complexity of HFC, BSB Tree and Δ+ Algorithm in the case of complexity of the algorithms.

TABLE II
COMPARISON BETWEEN THE COMPLEXITY OF HFC [7], BSB TREE [3] AND Δ+ ALGORITHM

	Algorithms		
Number of Vertices	HFC	BSB Tree	Δ+Algorithm
1000	1000000	1200000	10000
5000	23000000	26000000	60000
20000	3000000000	3840000000	300000
100000	87000000000	1.23E+11	1700000

V. AR SYSTEMS

A virtual object set within a real environment constitutes an AR system [2], [9], [8]. An AR-system incorporates more real objects and a few virtual objects with the real AR taking a dominant role over the virtual. Nowadays, many researchers have focused on AR since most of the computer graphics applications require computer-generated objects to be seamlessly integrated into natural images or videos such as environmental assessments and computer games. Moreover, the appearance of virtual objects should reveal the consistence with effect of the interaction between objects or even sky color in outdoor rendering.

Shadows in AR need sophisticated work. Huge amount of calculation for uploading 3D objects in AR systems demands a fast shadow generation technique which Δ+ Algorithm in shadow volumes covers this requirement. The main issue in realistic virtual objects in virtual environments and AR system is remaining the balance between realism and FPS. Increasing the realistic is not a big issue if the speed of rendering is not taken into account. Noh et al. [12] tried to generate the realistic soft shadows in AR adding soft projection shadows. Madsen et al. [11] considered the sky illumination and volume shadows in AR to enhance the realism.

Figs. 5 and 6 show implementation of the Δ+ algorithm in AR. The shadows are accurate enough and no so heavy in rendering.

Fig. 6 View 2: Result of Δ+ algorithm in AR

The AR system including shadow volumes using Δ+ Algorithm is performed in 74.35 FPS. The 3D objects include 8536 faces. In fact, the results show that the presented algorithm is appropriate to be used in any real-time rendering.

VI. CONCLUSION

Silhouette detection is the heaviest part of geometrically-based algorithms for shadow generation. In this paper, we have presented Δ+ algorithm to over come the heaviest calculation of silhouette detection. A simple idea which came from image based techniques to reduce the number of edges which contribute in shadow generation are taken into account. Δ+ algorithm is working by extending each ray from light source to visible vertices of occluders then using a simple checking can recognize whether the edge is silhouette or not. The algorithm is evaluated by comparing with two other widely used fundamental algorithms. The efficiency of Δ+ algorithm is $O(nLog_2n)$ compared to the others which are $O(n^2)$. Employing Δ+ algorithm on shadow volumes creates a fast enough shadow for real-time rendering. Finally, shadow volumes using Δ+ algorithm are employed on AR to test the algorithm in real-time environments resulting an accurate and fast shadow generation in AR.

ACKNOWLEDGMENT

This research was supported by Vot. Q.J130000.2528.12H18 RUG grant at the Magic-X, Universiti Teknologi Malaysia.

REFERENCES

[1] U. Assarsson and T. Akenine-Moller. A geometry-based soft shadow volume algorithm using graphics hardware. ACM Transactions on Graphics (TOG), 22(3):511-520, 2003.

[2] R Azuma. A survey of augmented reality. Presence: Teleoperators and Virtual Environments, 6(6):355-385, 1997.

[3] H.C. Batagelo and I.C. Junior. Real-time shadow gener- ation using bsp trees and stencil buffers. In SIBGRAPI, 12:93-102, 1999.

[4] Markus Billeter, Erik Sintorn, and Ulf Assarsson. Real time volumetric shadows using polygonal light volumes. Proceedings of the Conference on High Performance Graphics, pages 39-45, 2010.
[5] Johanna P Carvajal-Gonzalez, Juliana Valencia-Aguirre, and German Castellanos-Dominguez. Silhouette classification by using manifold learning for automated threat detection. In Security Technology (ICCST), 2013 47th International Carnahan Conference on, pages 1-5. IEEE, 2013.
[6] F. Crow. Shadow algorithms for computer graphics. Computer Graphics, 11(2):242-247, 1977.
[7] Soon Ki Jung, Soon Il Kwon, and Ku-Jin Kim. Real- time silhouette extraction using hierarchical face clusters. Dept. of Computer Engineering, pages 1-8, 2004.
[8] H. Kolivand and M. S. Sunar. Covering photo-realistic properties of outdoor components with the effects of sky color in mixed reality. Multimedia Tools and Applica- tions, 72(3):2143-2162, 2014.
[9] H. Kolivand and M. S. Sunar. Realistic real-time outdoor rendering in augmented reality. Plos One, 9(9), 2014.
[10] H. Kolivand and M.S. Sunar. An overview on based real-time shadow techniques in virtual environment. TELKOMNIKA, 10(1):171-178, 2012.
[11] Claus B Madsen and Brajesh B Lal. Estimating outdoor illumination conditions based on detection of dynamic shadows. Computer Vision, Imaging and Computer Graphics. Theory and Applications, pages 33-52, 2013. Springer.
[12] Z. Noh and M.S. Sunar. Soft shadow rendering based on real light source estimation in augmented reality. Ad- vances in Multimedia - An International Journal (AMIJ), 1(2):26-36, 2010.
[13] Maarten Slembrouck, Dimitri Van Cauwelaert, Peter Veelaert, and Wilfried Philips. Shape-from-silhouettes algorithm with built-in occlusion detection and removal. In International Conference on Computer Vision Theory and Applications (VISAPP 2015). SCITEPRESS, 2015.
[14] L. Williams. Casting curved shadows on curved surfaces. SIGGRAPH -78, 12(3), 270-274 1978.

Hoshang Kolivand received his MS degree in applied mathematics and computer from Amirkabir University, Iran, in 1999, and his PhD from Media and Games Innovation Centre of Excellence (MaGIC-X) in Universiti Teknologi Malaysia. Previously, he worked as a lecturer in Shahid Beheshti University, Iran. He has published numerous articles in international journals, conference proceedings and technical papers, including chapters in books. He is an active reviewer of many conference and international journals. He has also published many books in object-oriented programming and mathematics. His research interests include computer graphics and Augmented Reality.

Mahyar Kolivand is currently a year 11 student at Seri Omega International School and a research member of MaGIC-X at Universiti Teknologi Malaysia, Skudai. His research focus is on game developing, computer vision, computer graphics, and medicine.

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 degree in computer graphics and virtual environment in 2001 from The University of Hull, UK, and his BSc degree in computer science majoring in computer graphics in 1999 from the Universiti Teknologi Malaysia. He served as an academic member at the Computer Graphics and Multimedia Department, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, from 1999 to 2009. In 2009, he had been vested with the responsibility to lead the department. He has published numerous articles in international and national journals and conference proceedings and has also published numerous technical papers, including articles in magazines. He is an active professional member of ACM SIGGRAPH. He is also a member of Malaysian Society of Mathematics and Science. The current research programs that he leads are driving simulator, Augmented Reality, natural interaction and creative content technology.