

APPLYING THE REAL TIME SHADOWING METHOD FOR CREATING SOFT SHADOWS IN OPENGL

UDC ((632.115.2+628.984):004.438)

Dženan Avdić, Dejan Rančić

University of Niš, Faculty of Electronic Engineering, Republic of Serbia

Abstract. *This paper presents a technique for rapidly generating and rendering of soft shadows on the scene. The method has the advantage of real-time creating of shadows which look more realistic than hard shadows and have high quality, also it has not high demands in resources. The main idea is to replace a complex light source with a collection of multiple point sources of light in order to decrease rendering time. The mathematic model, experiments, and comparison with similar techniques are given and described.*

Key words: *light source, OpenGL, penumbra, shadow, soft shadow, umbra*

1. INTRODUCTION

The main aim of computer graphics is to make a depicting of a scene as realistic as possible. When rendering the lighted scenes, next to the light source and illuminated objects, it is necessary to draw out accompanying phenomena, such as shadows. They improve the visual perception of the image because without them the scene would look unnatural. Although shadows should be in the background compared to objects on the scene, they still carry information about the picture. Thus, a shadow can determine the positions and relationships between objects and light sources [1].

When plotting shadows, it is necessary to take care that they look realistic and improve the visual appearance of the scene. However, it sometimes requires a lot of hardware and memory resources and slows down the rendering of a scene. So usually compromises are made because the higher quality of the shadows is a result of complex rendering techniques and vice versa [2].

The appearance of shadows depends on the number and type of light sources presented on the scene. If the type is a point light source, then it causes the creation of the so-called hard shadows. These are shadows consisting of umbra [3]. The reason for this is

Received July 15, 2017

Corresponding author: Dženan Avdić

University of Niš, Faculty of Electronic Engineering, Aleksandra Medvedeva 14, 18000 Niš, Republic of Serbia

E-mail: dzavdic@gmail.com

that the point light source has a negligible surface area. In addition to umbra, there is also a part of the shadow called penumbra [4], and it is visible when the light source has a surface. Shadows consisting of both these components are called soft shadows and are more demanding for plotting and look more natural.

This research is a continuation of papers [4] and [5], where one method for plotting hard shadows was given on the example of primitives, using the common features of these primitive objects. Thus, common features were observed in determining the shadow for the circle, triangle, sphere, and cone, in the presence of one or more point light sources on a flat surface. These sources have the ability to move, in order to demonstrate the speed of calculating and plotting shadows in this way.

If a complex light source is viewed as a collection of multiple point sources of light, we can use the previous method to form a soft shadow. The main idea is to find the characteristic points of the source of lightness and present them as point sources of light, and then determine which one of them creates the umbra, and which one creates the penumbra. An umbra should have a more intense color than a penumbra. The color of the shadow and its components can be determined more closely using the coefficient k , the method of its calculation is given further in the paper.

The paper is organized in the following way. The next section gives a review of the similar research in this field and methods for generating shadows, as well as an overview of their strengths and weaknesses. Below is a description of our method, then a summary of the performed experiments. Finally, we give an evaluation and comparison with other methods, as well as directions for further work.

2. RELATED WORK

There are many techniques for creating shadows in Open GL and the most famous of them are shadow maps, volumes, and projected shadows. All other methods that are sub-indicated by these methods and their overview are given in the paper [6]. In this review, the emphasis will be placed on the papers that describe the soft-shadow algorithms.

The most important methods for generating soft shadows based on the shadow map method are given in the following references [7-12]. All of these methods are based on detecting edges in order to create a penumbra. Some of them need multi-rendering passes which make these algorithms time-consuming. The term 'soft shadow mapping' is the first used in [13] and [14]. Here the main idea was to divide the scene into less complex parts to calculate the radiance, but disadvantages were light leak and projection overlap.

Below is an overview of the latest and most useful papers that describe the creation of soft shadows based on shadow volumes.

In [15] an algorithm for creating soft shadows which uses penumbra wedges is represented. A formation of silhouettes is caused by locating shadow edges. This method has limitations in zooming a soft shadow area and calculating issues when rendering shadowed objects with a huge number of polygons. This fill-rate limitation, which is a feature of all methods based on shadow volume, is removed by algorithms proposed in newer publications [16, 17].

In [18], algorithms that merge shadow volumes and shadow maps are presented in order to create soft shadows. They detect shadow pixels in the shadow map and compute a shadow volume based on these pixels.

Our approach is an extending of the method of projected shadows. Since it does not need multi-rendering passes or extra memory, it does not deal with the shortcomings mentioned above.

3. CREATING OF THE SOFT SHADOWS

In our previous research we dealt with the creation of shadows in the presence of a point light source [4]. Then the research is expanded to include the conditions when the point source of light can move along the scene, in our case by x and y-axis. Since we then established the principles for calculating the hard shadow (the umbra), the main idea in this paper is to use them in the case of a presence of a complex light source and for obtaining the soft shadows. The goal is to create a shadow approximation effectively so that the shadow's realistic look is slightly reduced in favor of increasing the rendering speed. In order for this shadow creation method to be clearer, we recall the expressions from which we then started. We will recall how we formed shadows in previous papers [4, 5] on the basic example of creating a shadow of the circle (Figure 1).

If the light source is near to the Y-axis, the shadow area is enhanced, and vice versa. When the light source shifts away from the object in X-axis, the shadows are also going in the reverse direction to the motion of the light source. Two important points for rendering shadows are the center of the shadow x_s and the size of the radius of the shadow r_1 . The shadow, in this case, is also a circle, and coordinates of its center could be found using the equation of a line which passes through the two points: the center of the light source and the center of the shadowed object.

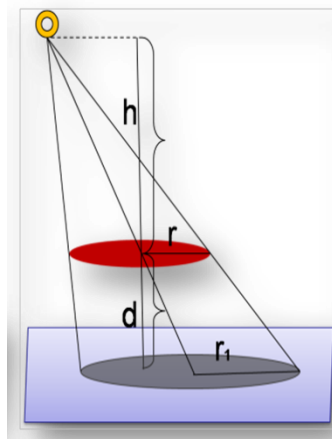


Fig. 1 The shadow of the circle formed by a point light source

The formula for obtaining the center of the shadow is given in Eq.(1). Calculating the length of the radius of the shadow is done using the theorems on the similarity of triangles. Fig. 1 shows the relevant triangles for calculating the increasing factor for the original radius. The equation for calculating the radius of the shadow is given in Eq.(2).

$$x_s = \frac{-d \cdot x_i}{y_i - d} \quad (1)$$

$$r_1 = \frac{r \cdot y_i}{y_i - d} \quad (2)$$

where x_b , y_b , z_b are the coordinates of the position of the light source, r the radius of the circle and the sphere, and r_1 is the radius of the shadow, d is the distance between the object and the surface. Using same logic, we found the formulas for shadowing a triangle, a cone and a sphere.

Therefore, based on these equations, we can determine the shadow formed by any point source of light, regardless of its coordinates, if we have the appropriate parameters of a shadowed object. Therefore, we will consider the complex light source as a collection of point sources of light, where the central one will be used for an approximate and effective creation of the umbra, while others will be used to create the penumbra. This is graphically represented by the diagram in figure 3.

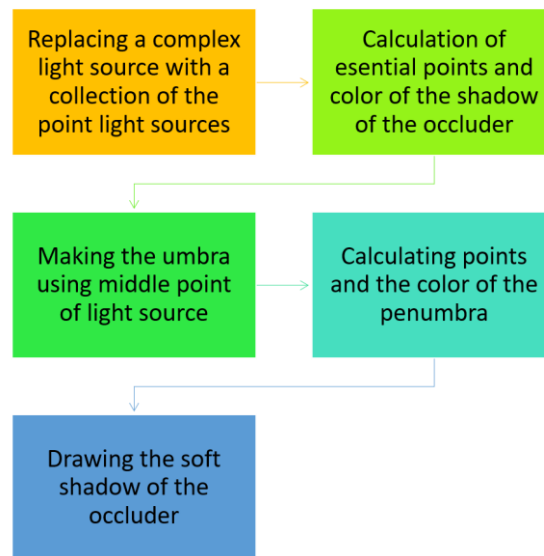


Fig. 2 The steps of the creating soft shadows

In Figure 2, the complex light source considered as a union of point light sources is shown, and forms the shadow of the circle. The shadow created by the central point source of light is considered as the umbra, and two side shadows as the penumbra.

To form these shadows, knowledge about the formation of real-time shadows in the case of presence of point light sources is used, in cases when it is directly above the object and when the source moves left and right of the shadowed object. The union of the created shadows, where the lateral shadows are taken as penumbra, makes a complete shadow.

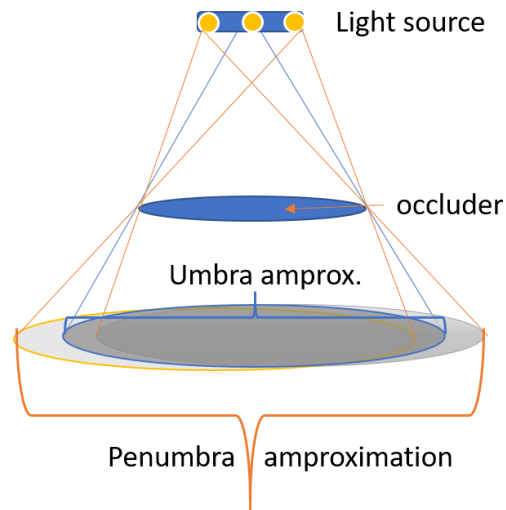


Fig. 3 Forming soft shadows by three point sources of light

In order to achieve a more realistic look of soft shadows, it is necessary to determine the colors of the side shadows, so that they are slightly milder than the central shadow. This is the subject of the next chapter.

4. CALCULATING THE COLOR OF THE UMBRA AND PENUMBRA

In order to make the scene where the shadowed objects are placed as realistically as possible, it is not enough to draw the shadow in a certain shade of gray but is necessary to calculate that shade. It is known that the closer the light source, with respect to the shadowed object, the stronger the shadow. If the source of light is further away, the created shadow is fainter. The question is how to calculate the distance between the object and the light source, and how to include the position of the shadow in this calculation.

The distance between the object and the light source can be calculated as the distance between their centers. The shadow has some shade of gray, and it is known that the shades of gray in RGB model are made if all the components of the color, red and green and blue, are set to the same value. If the value is closer to one, the shade of gray is darker, closer to black RGB (0, 0, 0). If the value of the components is closer to 255, the shade of gray color is closer to white RGB (255, 255, 255).

It is necessary to define the coefficient which would multiply 255, and which would be $0 < k < 1$, to give a realistic shade of the shadow; in accordance with the conditions set above, the shade is darker if the distance of the light source is smaller, and vice versa.

In [4] it is explained that the requirement to form a shadow on the surface is that the light source should be above the observed object, and if it is below, then it forms a reflection. This means that the distance between the light source and the surface (center of the shadow) is always greater than the distance of the object from the surface. Thus, the ratio of these two distances will always be less than 1, if we take the object distance as the

numerator and the distance of the light source as the denominator. In order to meet the requirement to produce a darker shade of shadow if the object and the light source are closer, then the resulting ratio should be deducted from the Eq. (3).

$$k = 1 - \frac{\text{distObj}}{\text{distLightSource}} \quad (3)$$

which would represent the color of dark shadows, hard shadows or umbra. A penumbra would be a little brighter. Bearing in mind that in the real world, the umbra would be a cross-section of three shadows, while the penumbra would be created from one source, then it should be three times milder. This is also experimentally confirmed in the next chapter.

5. EXPERIMENTS AND EVALUATION

In order to confirm the validity of the given formula and to check the speed of drawing the soft shadows, a simple C++ application that uses OpenGL [19] is created. The application has following features:

- changing of the shadowed object (keys K(circle), L(sphere), T(triangle), U(cone));
- changing the position of the camera (keys Y (increasing y coordinate), X (increasing x coordinate), C (increasing z coordinate), A (decreasing y coordinate), S (decreasing x coordinate), D (decreasing z coordinate));
- changing the position of the light source components (keys V (increasing y coordinate), B (decreasing y coordinate), N (increasing x coordinate), M (decreasing x coordinate)).
- choosing the component of the complex light source to move (keys G (left component) and H (right component)).

The results are given with following screenshots (Fig. 4, Fig. 5 and Fig. 6).

We considered two cases. In the first case, the complex light source presented as a collection of the point light sources is located exactly above the occluder (the shadowed object) – position 1, while in the latter case (position 2), the lateral light sources are moved slightly to the left and to the right.

The experiment was made on the example of the sphere shadow creation, since the complex light source is presented as a collection of point sources of light. If the coordinates of the components of a complex light source coincide, then it behaves as a source of light, and then it forms and only the hard shadow (umbra), what can be seen on the scene (Figure 4). By moving the left and right components of the composite light source along the x and y-axes, soft shadows (penumbra) are formed, and they are lighter than the umbra. Figure 5 shows the case where components of the composite light source are shifted by y-axis. Figure 6 shows the case where the shadow components are shifted by x-axis, left and right.

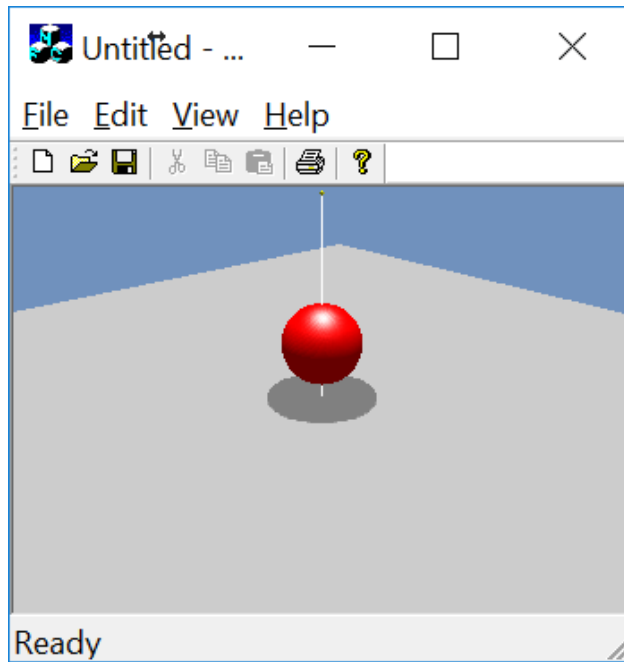


Fig. 4 Forming a hard shadow of the sphere

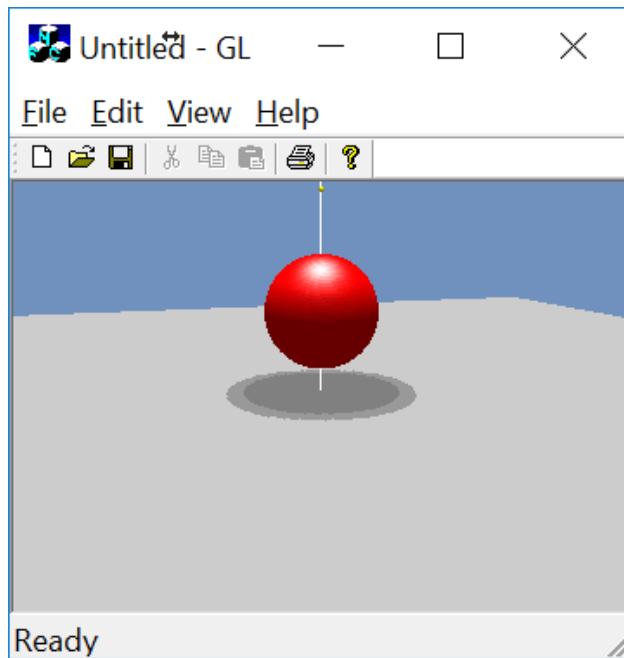


Fig. 5 Making a soft shadow of the sphere – position 1

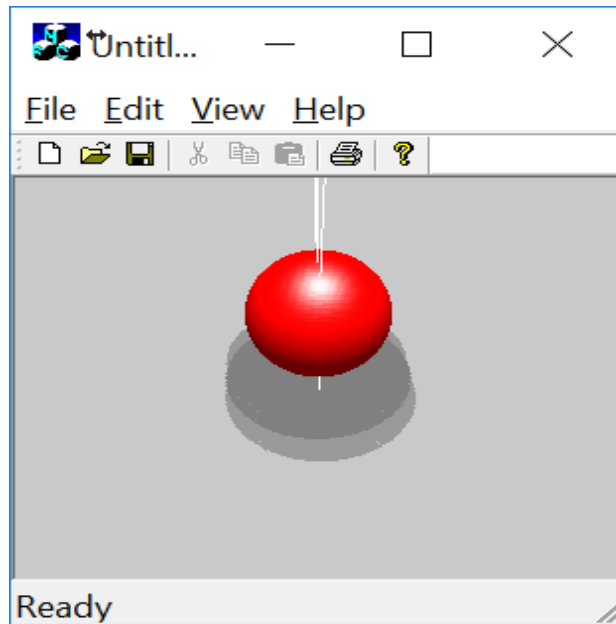


Fig. 6 Making a soft shadow of the sphere – position 2

Here, the formation of shadows is shown only on the case of the sphere, although the application has the ability to change objects in a circle, a triangle and a cone, and also shadows are formed for them in a similar way. In order to avoid repeating the images that show it, a comparison of the time of their rendering has been made (Figure 7).

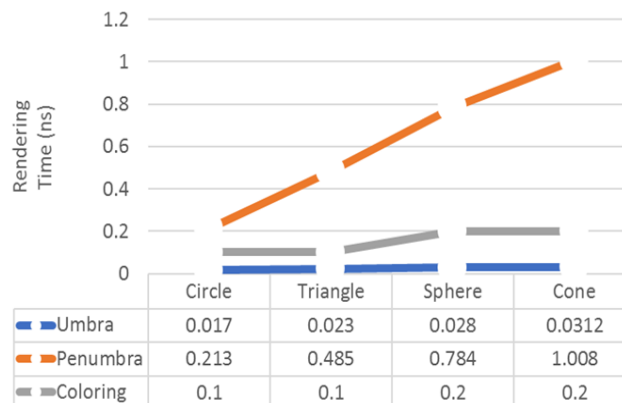


Fig. 7 Rendering time for creating of soft shadows for primitive objects

Based on the graph shown in Figure 7, the following facts can be concluded. It takes more time to form a penumbra, and it increases depending on the number of components

of the complex light source. This means that, depending on the purpose of the application, we can calculate the number of components of a complex light source. Also, with the complexity of the object, the time of rendering its shadow grows.

This comparison was implemented in the environment of VS 2015 and on a PC with Intel CORE i7 2.27 GHz CPU and 8 GB RAM.

5. CONCLUSION AND FUTURE WORK

The paper presents one method for forming shadows that contain a penumbra in addition to the umbra. The method presented uses the previously formed method for real-time formation of shadows, so the shadows are quickly calculated and plotted. Their appearance is at a minimum deviation from reality, and savings in terms of efficiency, when it comes to processors and memory resources are high.

The method has shown that the reality of the shadows will not be visibly disturbed if the complex light source is presented as a collection of several point sources of light, with the corresponding shadow components being umbra and others as the penumbra.

The negative side of this method is that no passages have been developed that would ease the edges of the shadows, and this is what will be the focus of further research on this topic.

REFERENCES

- [1] H. Kolivand, M.S. Sunar, N.M. Jusoh, Olufemi, "Real-Time Shadow Using a Combination of Stencil and the Z-Buffer" *International Journal of Multimedia & Its Applications*, 3, 3(2011), pp. 27-38. [Online]. Available: <http://dx.doi.org/10.5121/ijma.2011.3303>
- [2] T. McReynolds, D. Blythe, "Advanced Graphics Programming Using OpenGL," *Elsevier*, San Francisco, 2005.
- [3] X. Hu, Y. Qi, X.A. Shen, "Real-Time Anti-Aliasing Shadow Algorithm Based on Shadow Maps," *Pattern Recognition CCPR*, Beijing, 2008, pp.1-5. [Online]. Available: <https://dx.doi.org/10.1109/CCPR.2008.27>
- [4] A. Pljaskovic, Dz. Avdic, P. Spalevic, D. Rancic, "Sphere and Cone Composite Realtime Shadows in OpenGL," *Information and Communication Technology Forum*, Sarajevo, 2013, pp.133-137. [Online]. Available: <http://i-scoper.ieice.org/proceedings/ICTF/2013/pdf/COMM6-2.pdf>
- [5] Dz. Avdic, D. Rancic, P. Spalevic, A. Avdic, E. Dolicanin, "Real-Time Shadows In Opengl Caused By The Presence Of Multiple Light Sources," *Tehnički vjesnik*, Vol.24 No.4 accepted for publication. [Online]. Available: <http://dx.doi.org/10.17559/TV-20140324202037>
- [6] N. Liu, M. Pang, "A Survey of Shadow Rendering Algorithms: Projection Shadows and Shadow Volumes," *Second International Workshop on Computer Science and Engineering*, vol. 1, pp.488-492, 2009. [Online]. Available: <https://dx.doi.org/10.1109/WCSE.2009.716>
- [7] W. Lili, Z. Jingchao, S. Zhe, "Real-time approximate soft shadow rendering with bidirectional penumbra map," *In Educational and Information Technology (ICEIT)*, 17-19 Sept. 2010, Vol.1, pp.V1248. [Online]. Available: <http://ieeexplore.ieee.org/document/5607728/>
- [8] E. Chan, F. Durand, "Rendering fake soft shadows with smoothies," *In: Rendering Techniques 2003*, pp.208-218. Available: <http://people.csail.mit.edu/ericchan/bib/pdf/smoothie.pdf>
- [9] L. Wang, S. Zhou, W. Ke, V. Popescu, "GEARS: a general and efficient algorithm for rendering shadows," *Computer graphics forum* Vol. 33, No. 6, pp. 264-275. [Online]. Available: <http://dx.doi.org/10.1111/cgf.12348>
- [10] J. Arvo, M. Hirvikorpi, J. Tyystjärvi, "Approximate soft shadows win an image-space flood-fill algorithm," *Computer graphics forum*, Vol. 23, No. 3, pp. 271-279. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2004.00758.x>

- [11] X. Cai, Y. Jia, X. Wang, S. Hu, R. Martin, "Rendering soft shadows using multilayered shadow fins." *Computer graphics forum*, 2006 Vol. 25, No. 1, pp. 15–28. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2006.00915.x>
- [12] W. De Boer, "Smooth penumbra transitions with shadow maps," *JGraph GPU Game Tools*, 2011. Vol.11, No.2, pp.59-71. [Online]. Available: <http://dx.doi.org/10.1080/2151237X.2006.10129221>
- [13] L. Atty, N. Holzschuch, M. Lapierre, JM. Hasenfratz, C. Hansen, F. Sillion, "Soft shadow maps: efficient sampling of light source visibility," *Computer graphics forum*, 2006, Vol. 25, No. 4, pp. 725–741. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2006.00995.x>
- [14] G. Guennebaud, L. Barthe, M. Paulin, "Real-time soft shadow mapping by Backprojection," In: *Rendering techniques 2006*, pp. 227–234. [Online]. Available: <http://dx.doi.org/10.2312/EGWR/EGSR06/227-234>
- [15] U. Assarsson, T. Akenine-Möller, "A geometry-based soft shadow volume algorithm using graphics hardware." *ACM Transactions on Graphics* 2003 Vol.22, No.3, pp.511–520. [Online]. Available: <http://dx.doi.org/10.1145/882262.882300>
- [16] E. Chan, F. Durand, "An efficient hybrid shadow rendering algorithm," In *Rendering Techniques* 2004, pp.185–195. [Online]. Available: <http://dx.doi.org/10.2312/EGWR/EGSR04/185-195>
- [17] B. Lloyd, J. Wendt, N. Govindaraju, D. Manocha, "Cc shadow volumes," In *Rendering Techniques Proc. of the Eurographics Symposium on Rendering 2004*, pp.197–205. [Online]. Available: <https://dx.doi.org/10.1145/343002.343006>
- [18] M. Mccool, "Shadow volume reconstruction from depth maps," *ACM Transactions on Graphics* Vol.19, No.1, pp.1–26. [Online]. Available: <https://dx.doi.org/10.1145/343002.343006>
- [19] <https://www.opengl.org/about/>