Light & Engineering
Vol. 27, No. 2, pp. 105–110, 2019
https://doi.org/10.33383/2017-081

MATRIX TRANSFORMATIONS FOR EFFECTIVE IMPLEMENTATION OF RADIOSITY ALGORITHM USING GRAPHIC PROCESSORS

Alexander S. Shcherbakov [1] and Vladimir A. Frolov [1, 2]

[1] *Lomonosov Moscow State University, Moscow, Russia*
[2] *Keldysch Institute of Applied Mathematics, RAS, Moscow, Russia,*
E-mail: alex.shcherbakov@graphics.cs.msu.ru; vladimir.frolov@graphics.cs.msu.ru

ABSTRACT

A method of form factors matrix transforming, which allows accelerating calculation of secondary illumination by the radiosity method is proposed. An adaptation of this method for graphic processors (graphics processing unit, GPU) is considered. In particular, it is proposed to use DXT textures to store form factors matrix and to reorder columns and lines of the matrix to reduce compression losses. The proposed optimisations increase the speed of radiosity algorithm work to ten times and reduce the GPU occupied memory volume to three times.

Keywords: radiosity, global lighting, GPU

1. INTRODUCTION

Global lighting of 3D scenes is a combination of the light source (LS) primary lighting and of the secondary lighting multiply reflected from the scene surfaces. The calculation of the secondary lighting is maximum complex as the illuminance integral dimension increases with each reflection. Therefore, in real time applications some methods of the approximate calculation are used.

2. A REVIEW OF THE EXISTING METHODS

2.1. Instant Radiosity Method

Instant Radiosity is one of the most popular methods due to its simplicity [1, 2]. To calculate the secondary lighting, secondary LSs are used, which are created by the ray tracing from primary LSs. Thus, the secondary lighting can be considered as the primary from secondary LSs. Reflective Shadow Maps (RSM) algorithm is a development of the Instant Radiosity method for GPU [3]. Instead of the ray tracing in the RSM, to create secondary LSs, the Shadow Maps are used. The main disadvantage of this method is its low accuracy.

2.2. Light Propagation Volumes Method

Light propagation method [4] creates secondary LSs, as well as the Instance Radiosity method, but the calculation of secondary lighting is made by means of the light propagation using a three-dimensional net. The main disadvantage of this method is the high memory consumption and low efficiency when calculating the light propagation through empty spaces.

2.3. Voxel Cone Tracing

Voxel Cone Tracing method [5] makes a “lighting collection” for each pixel by the several cones tracing from a set surface point imitating the Monte-Carlo ray tracing over a hemisphere. This method (just as the previous) uses a voxel net for a simplified geometry presentation, and the cone tracing is similar to the ray marching. The difference is that with the distance increase, the selection is made from more rough MIP-maps of the voxel net, due to which a geometrical cone appro-

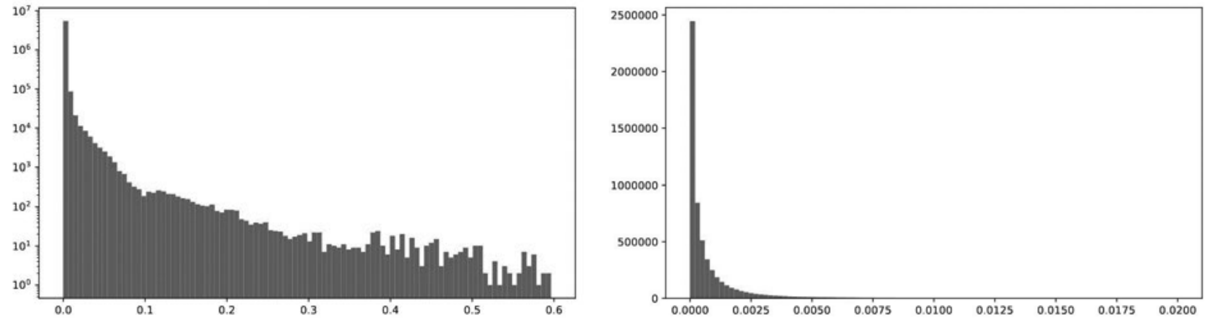


Fig. 1. Distribution of values in a form factor matrix (logarithmic and linear scales)

ximation occurs. A high calculation complexity and a calculation speed dependence on the resolution are the disadvantages of the algorithm.

2.4. Spherical Harmonics Method

Spherical harmonics method [6] is based on the expansion of complex illuminance functions into a sum of values, which are simpler to be calculated. For some surface points (as a rule, for top points) basis expansion coefficients of their lighting functions are calculated. Lighting functions from LSs are also expanded on the basis. As a result, the calculation of lighting in a point with the illuminance function expanded in it is reduced to a dot product of vectors consisting of illuminance function coefficients in this point and of lighting functions from LSs. This method is widely used when visualising open spaces, but it is less accurate than the radiosity method in case of closed rooms.

2.5. Radiosity Method

Radiosity method [7] allows obtaining high-quality images for closed rooms with diffuse surfaces, and in many respects, it is not worse than more modern methods. However, implementation time and required resources strongly depend on the scene. For the scenes containing hundreds of thousands triangles, a direct radiosity application is complicated because of the square complexity and of the memory consumption depending on the primitive numbers. Therefore, in practice, the radiosity algorithm is performed for a simplified scene (containing fewer surfaces) and the calculation result is transferred to the initial scene [8]. It should be noted that along with the spherical harmonics, the radiosity algorithm transfers the main computing complexity to a preliminary calculation stage, and due to this fact, a good accuracy and speed balance is reached in comparison with other methods.

3. KEY TERMS AND DEFINITITIONS

In the classical radiosity algorithm, the form factor matrix F of $n \times n$ size is used, where n is the number of scene sites. To calculate the illuminance m_i after reflection, this matrix is multiplied by n-component vector of initial luminous exitance $m_e^{(0)}$ containing the sites luminous exitance:

$$\overline{m}_i^{(1)} = F \cdot \overline{m}_e^{(0)}.$$

Multiplying the obtained vector by reflectance ρ of the sites, to which light has come, luminous exitance of the sites after reflection is calculated as follows:

$$\overline{m}_e^{(1)} = \overline{m}_i^{(1)} \cdot \rho. \qquad (1)$$

Three-component vectors containing information on each colour channel are vector elements in these formulas, and real numbers from 0 to 1 are form factors matrix elements. The presented calculations can be repeated using vectors $m_e^{(n)}$ instead of the initial site luminous exitance to obtain light, which has come to the scene sites after an arbitrary reflection:

$$\overline{m}_i^{(n)} = F \cdot \overline{m}_e^{(n-1)}. \qquad (2)$$

The full scene lighting after k reflections can be obtained by summation of vectors $m_i^{(n)}$:

$$\overline{I} = \sum_{n=1}^{k} \overline{m}_i^{(n)}. \qquad (3)$$

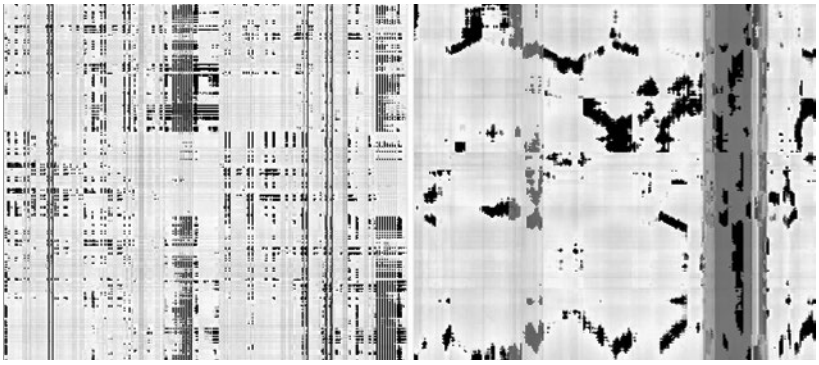


Fig. 2. A fragment of form factor matrix before selection (on the left) and after selection (on the right) of lines and columns. An increased fragment of form factor matrix

4. THE PROPOSED METHOD

4.1. Preliminary Calculation of Several Scene Reflections

The proposed modification is to use of the transformed form factor matrix. In the beginning, a "colour" form factor matrix is introduced:

$$F_{ij}^{C} = F_{ij} \cdot \rho_j ,$$

where ρ_j is colour of j site. This matrix contains information on light transport between scene sites along each channel separately. Thus, its storage requires three times more memory. In this article, the use of the radiosity algorithm for the secondary scene lighting is considered. Therefore, the radiosity calculation begins not from the vector $m_i^{(1)}$ obtaining, but from the luminous exitance calculation after the first reflection $m_e^{(1)}$. One can consider that the vector $m_e^{(1)}$ is already calculated. We can transform the expression (2) using the formula (1) for an arbitrary index.

$$\begin{aligned} m_i^{(n)} &= F \cdot (\rho \cdot m_i^{(n-1)}) = \\ &= F^C \cdot m_i^{(n-1)} = (F^C)^{-1} \cdot m_i^{(1)}. \end{aligned} \quad (4)$$

The summation (3) can be transformed using (4):

$$\overline{I} = \left(\sum_{n=1}^{k} (F^C)^{n-1} \right) \cdot m_i^{(n)}.$$

The matrix polynomial in brackets does not depend on the primary scene lighting and depends only on the scene geometry and on surface materials. Therefore, it can be found at the preliminary calculation stage:

$$S = \left(\sum_{i=1}^{k} (F^C)^{i-1} \right).$$

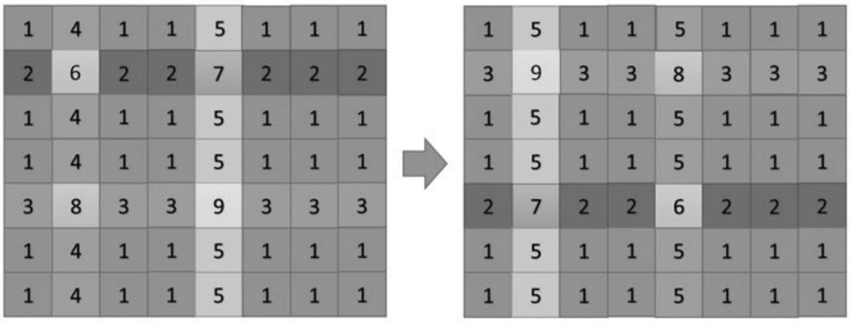
Thus, the calculation of the secondary lighting by the radiosity method after k reflections is reduced to one multiplication of the vector by a matrix:

$$m_i^{(n)} = S \cdot m_i^{(1)}.$$

Using this type of matrix allows accelerating the calculation k times (where k is the number of reflections); however, it requires three times more memory.

To take into account all possible light reflections on a scene, it is necessary to calculate an infinite series

$$S = \left(\sum_{i=0}^{\infty} (F^C)^{i} \right).$$



1	4	1	1	5	1	1	1
2	6	2	2	7	2	2	2
1	4	1	1	5	1	1	1
1	4	1	1	5	1	1	1
3	8	3	3	9	3	3	3
1	4	1	1	5	1	1	1
1	4	1	1	5	1	1	1

1	5	1	1	5	1	1	1
3	9	3	3	8	3	3	3
1	5	1	1	5	1	1	1
1	5	1	1	5	1	1	1
2	7	2	2	6	2	2	2
1	5	1	1	5	1	1	1
1	5	1	1	5	1	1	1

Fig. 3. Layout of lines and columns interchanging used by the authors when selecting

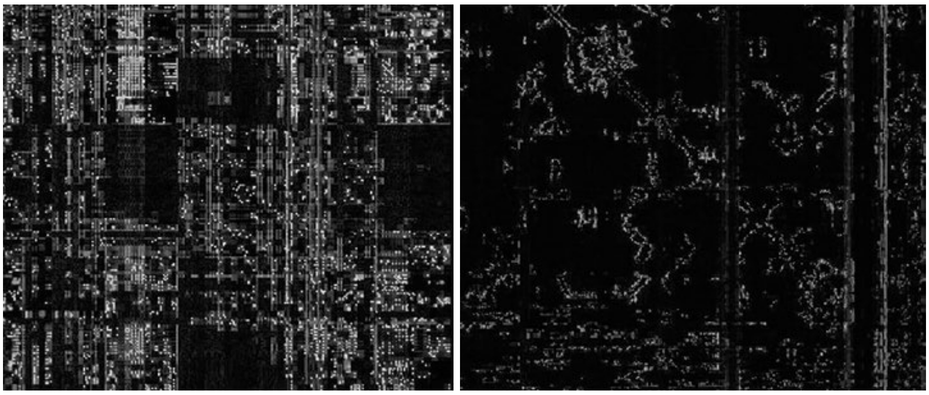


Fig. 4. Visualised difference between the compressed and uncompressed textures (an increased fragment) before (on the left) and after (on the right) lines and columns reordering

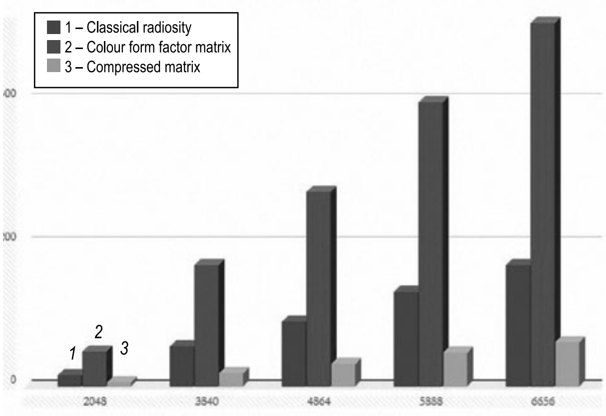


Fig. 5. Comparison of form factor files and matrices size, MB

Limit process. As the initial matrix is positively determined, all numbers in it are less than 1 and, moreover, sum of numbers in one line is less than 1, then the formula for the calculation of the geometrical progression sum is applicable to this series

$$S = F^C \cdot \left(I - F^C\right)^{-1}.$$

However, calculation of a reverse matrix is not always possible because of the solution instability. At the same time, the calculation of a series partial sum does not have such a disadvantage and allows taking into consideration large reflections number at the preliminary calculation stage.

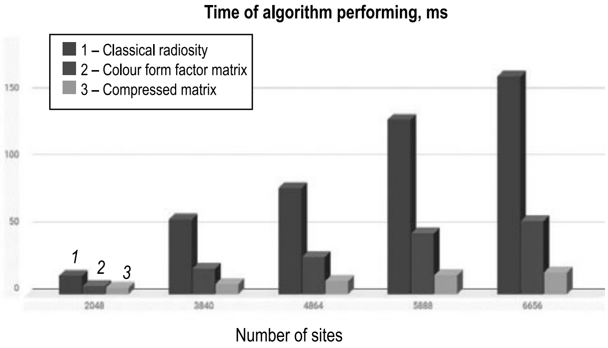


Fig. 6. Comparison of radiosity calculation speeds. (A compression increases the speed since the algorithm is limited by the memory access but not by calculations.)

4.2. DXT Compression

As the proposed algorithm modification requires a greater memory volume, a method of form factor matrix storage in the summary form was developed. As a storage format, DXT texture format supported by many GPUs was selected (a hardware decompression when reading is meant). For this format, all matrix values should be reduced to an interval of 0–255. Fig. 1 shows the value distribution in a form factors matrix. When scaling these numbers in the 0–255 interval, most of the numbers are set to zero. Therefore, large numbers, which make the basic contribution to the radiosity calculation, are stored separately of the rest matrix, and the following transformation is applied to the others:

$$value_i^{'} = \frac{\max\left(\log\left(\frac{value_i}{\max\limits_j value_j}\right) + shift, 0\right)}{shift} \cdot 255.$$

The texture, which is obtained as a result of a such transformation, is shown in Fig. 2, on the left.

However, the DXT compression is followed by losses. In order to minimise them when compressing, matrix lines and columns are sorted to reduce the difference between the neighbour values (Fig. 2, on the right). As the matrix is a relation of the scene sites for couples, lines and columns should change

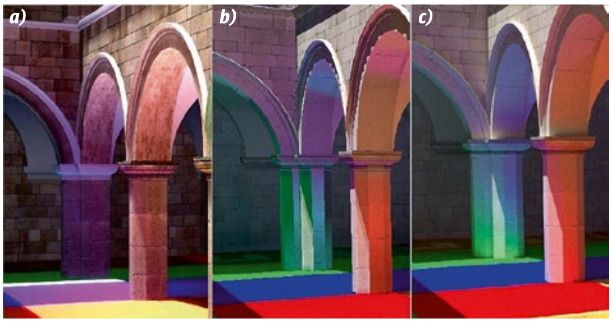


Fig. 7. Images obtained by the Light Propagation Volumes (a) and by the proposed (b) methods, as well as by the path tracing /reference/ (c). The Light Propagation Volumes and the proposed methods work with 40 frames/s frequency / (25 ms period), and the reference is obtained within 5 min.

Fig. 8. An image obtained by the Light Propagation Volumes method (Unreal Engine 4)

positions simultaneously (Fig. 3). As a result, the root-mean-square error, when compressing, decreases to five times (Fig. 4).

4.3. Implementation Details

As modern 3D scenes contain hundreds of thousands of triangles and time of the radiosity algorithm implementation is unacceptable for a such scene site number, to calculate the secondary lighting, a simplified version of the same scene was selected using a simplification method based on the voxel net from the article [9].

Thus, the scene pre-processing is_implemented according to the following schedule:

1. A simplified scene analogue is constructed based on the voxelisation;
2. Form factors are calculated for sites of the simplified scene;
3. A form factor matrix is calculated, which takes into consideration several light reflections;
4. The matrix numbers, which are bigger than the threshold value, are stored in a separate file, and zeroes are put on their places;
5. Values in the matrix are reduced to the 0–255 interval;
6. An interchange of lines and columns is made;
7. The obtained matrix is stored as a DXT texture.

The visualisation occurs according to the following algorithm:

1. A shadow map is created;
2. Lighting of the simplified scene sites made by LSs is calculated;
3. The secondary lighting is calculated using the radiosity method by means of the form factor matrix stored as a texture;
4. The secondary lighting is transferred from the simplified scene to the initial.

4.4. Comparison of the results

The proposed algorithm modifications allow accelerating the radiosity algorithm in total to ten times (Figs. 5–7). The use of the form factor matrix, which is taking into consideration several reflections, increases the matrix file size three times, however, the use of the DXT compression allows reducing the required memory to three times in comparison with the initial form factor matrix

Fig. 9. An image obtained by the proposed method

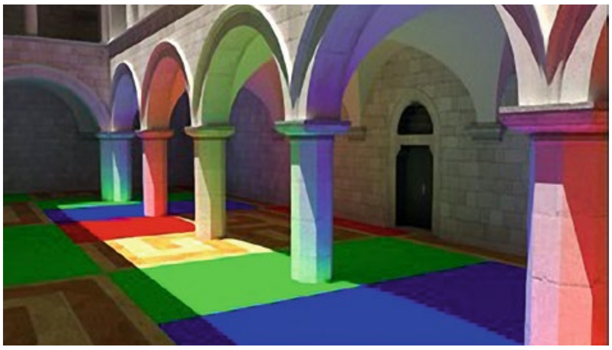


Fig. 10. An image obtained by path tracing (as a reference)

and with the classical radiosity algorithm (Fig. 5). We have carried out a comparison with the images obtained using the Light Propagation Volumes method of the Unreal Engine 4, the classical radiosity, and the ray tracing (reference). The proposed method shows a result comparable by accuracy with the classical radiosity. Based upon the images, it is closer to the reference than the Light Propagation Volumes method with the identical frame frequency (Figs. 7–10).

5. CONCLUSION

Unlike other widespread methods of solving the radiosity equation, the proposed method allows calculating the global lighting during $n^2+O(n)$ arithmetic operations and readings from the memory (where n is the number of scene sites) as it is reduced to the single multiplication of the matrix by the vector.

A similar result could be achieved solving a system of linear algebraic equations by means of LU expansion. However, an effective implementation of LU GPU expansion is nontrivial, and when using third-party libraries (for example, the CUBLAS), there is no possibility to use a compression. The latter, as noted above, is critical for the radiosity algorithms (Figs. 6, 7).

The work is supported by the grant 16–31–60048 mol_a_dk of the Russian Foundation for Basic Research.

REFERENCES

1. Keller A. Instant radiosity. Proc. 24th annual conf. on Computer graphics and interactive techniques (SIGGRAPH’97) // ACM Press/Addison-Wesley Publishing Co., New York, USA, 1997, pp. 49–56.

2. Budak V.P., Zheltov V.S., Kalakutsky T.K. Lokal’ny’e ocenki metoda Monte-Karlo v reshenii uravneniya global’nogo osveshheniya s uchyotom spektral’nogo predstavleniya ob’’ektov [Local evaluations of Monte-Carlo method when solving equation of global illumination taking into account spectral object presentation] // Komp’yuterny’e issledovaniya i modelirovanie, 2012, V. 4, #1, pp. 75–84.

3. Dachsbacher C., Stamminger M. Reflective shadow maps. Proc. 2005 symp. on Interactive 3D graphics and games (I3D ‘05) // ACM, New York, USA, 2005, pp. 203–231.

4. Kaplanyan A., Dachsbacher C. Cascaded light propagation volumes for real time indirect illumination. Proc. 2010 ACM SIGGRAPH symp. on Interactive 3D Graphics and Games (I3D’10) // ACM, New York, USA, 2010, pp. 99–107.

5. Crassin C., Neyret F., Sainz M., Green S., Eiseman E. Interactive indirect illumination using voxel-based cone tracing: an insight. ACM SIGGRAPH 2011 Talks (SIGGRAPH ‘11) // ACM, New York, USA, 2011, Article 20, 1 p.

6. Lisle I.G., Tracy Huang S.-L. Algorithms for spherical harmonic lighting. Proc. 5th int. conf. on Computer graphics and interactive techniques in Australia and Southeast Asia (GRAPHITE ‘07) // ACM, New York, USA, 2007, pp. 235–238.

7. Goral C.M., Torrance K.E., Greenberg D.P., Battaile B. Modeling the interaction of light between diffuse surfaces. Proc. 11th annual conf. on Computer graphics and interactive techniques (SIGGRAPH’84) //SIGGRAPH, 1984, V. 18, #3, pp. 213–222.

8. Martin S., Einarsson P. A Real Time Radiosity Architecture for Video Games, SIGGRAPH, 2010. URL: http://www.geomerics.com/wpcontent/uploads/2014/03/radiosity_architecture.pdf.

9. Shcherbakov A., Frolov V. Avtomaticheskoe uproshhenie geometrii dlya raschyota vtorichnoj osveshhyonnosti metodom izluchatel’nosti [An automatic simplification of geometry for calculation of secondary illuminance by the radiosity method], Proceedings of the 26th International conference on computer graphics and sight, September 19–23, 2016, Nizhny Novgorod, pp. 34–38.

Alexander S. Shcherbakov,
master student of the Faculty of Computational Mathematics and Cybernetics of the Lomonosov Moscow State University

Vladimir V. Frolov,
Ph.D. Graduated from the Faculty of Computational Mathematics and Cybernetics of the Lomonosov Moscow State University. Research associate of the Lomonosov Moscow State University and of the Keldysch Institute of applied mathematics of the RAS