Area Lights Voxelization for Light Propagation Volumes

Cristian Lambru[a], Florica Moldoveanu[b], Anca Morar[c] and Victor Asavei[d]
Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, 060042, Bucharest, Romania

Keywords: Computer Graphics, Area Lights, Voxelization, Light Propagation Volumes.

Abstract: Simulation of the area light sources direct illumination is a topic of interest in the field of Computer Graphics. In the real world, all light sources have a surface from which light is emitted. Thus, for a physically correct simulation of the light transport in graphical applications, area light sources are required. In addition, there are complex lighting effects that can only be simulated with such light sources. In this paper, we present an improvement of the direct lighting simulation for area light sources within the real-time global illumination technique called light propagation volumes. Our method is based on a voxelization of the area light source geometry in a voxel volume of the same resolution as the light propagation volume used in the global illumination technique. With a sample for every voxel that intersects a triangle, for every triangle of the mesh, we obtain an optimal distribution of the samples needed to approximate the direct illumination of the area light source for the light propagation volumes technique.

1 INTRODUCTION

An area light source is defined by a geometry on the surface of which radiance is emitted. Such a light source has applicability in Computer Graphics because it can reproduce effects such as the light of a TV display. The light emitted by such a source can reach the observer directly, but it can also reach other surfaces in the scene and by reflection, only later reach the observer. In particular, the simulation of the first bounce of light is a topic of interest in Computer Graphics. Analogous as for analytic light sources, such as directional or point lights, we consider the simulation of this first bounce of light to be direct illumination.

By comparison, the analytic light sources are a useful mathematical concept in practice. These light sources have no surface or volume and are usually represented by a position in space, around which they scatter radiance according to a chosen behavior. Their simulation is easily performed analytically and offers fast solutions for real-time applications. They can also produce plausible visual results. However, this type of light does not exist in real world. All the lights in nature have a surface and in addition, effects such as the one mentioned above cannot be simulated with analytic lights. Therefore, the simulation of direct lighting of area light sources has been researched for a long time.

In this paper, we present a method to improve the simulation of area light sources direct illumination within the real-time global illumination technique called light propagation volumes (Kaplanyan and Dachsbacher, 2010). Its authors specified that this technique can be adapted to simulate the direct illumination of area light sources by sampling the surface of these sources to obtain an initial set of points that are used as input for the technique. More details about the light propagation volumes technique are provided in Section 2.2. Our method uses a voxelization of the area light source mesh representation into a voxel volume of the same resolution as the 3D grid used in the global illumination technique. Taking a sample for each voxel that intersects a triangle, for every triangle of the mesh, creates an optimal distribution for the samples needed to approximate the direct illumination of area light source for the light propagation volumes technique.

The paper is structured as follows. Section 2 presents the related works and the necessary background for our method, followed in Section 3 by a detailed description of it. Section 4 presents the quantitative and qualitative results obtained by our method. Section 5 presents some conclusions about our work.

[a] https://orcid.org/0000-0002-3168-9969
[b] https://orcid.org/0000-0002-8357-5840
[c] https://orcid.org/0000-0002-4773-6862
[d] https://orcid.org/0000-0002-4776-2542

Lambru, C., Moldoveanu, F., Morar, A. and Asavei, V.
Area Lights Voxelization for Light Propagation Volumes.
DOI: 10.5220/0010841600003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 1: GRAPP*, pages 212-219
ISBN: 978-989-758-555-5; ISSN: 2184-4321

2 RELATED WORK

2.1 Area Lights

The real-time techniques approached by the game industry are based on analytical solutions that provide plausible results. One of them uses a set of representative points to approximate the surface of the light source. These points are numerically integrated to obtain an approximation of the lighting of the entire surface. The approach requires closed-form formulas specific to the area light source shape. Several solutions have been proposed for different shapes, such as lines (Picott, 1992), spheres (Snyder, 1996) and rectangles (which also applies to disks) (Drobot, 2014). For generic planar polygonal light sources, Arvo (Arvo, 1995) proposed a solution that was later improved by Lecocq et al. (Lecocq et al., 2017) to provide real-time performance. Recently, Heitz et al. (Heitz et al., 2016) introduced a new class of techniques that use linearly transformed spherical distributions to offer great flexibility over the shape of the polygon that describes the light source.

The direct illumination of an area light source behaves similarly to the indirect illumination of the surface when the light comes from another light source. So that, in general, real-time global illumination techniques can easily simulate direct illumination of area light sources, as shown by Lambru et al. (Lambru et al., 2021). Nichols et al. (Nichols et al., 2010) proposed such an approach that uses the information on the screen. However, this approach is limited by the information on the screen and have problems with temporal coherence as the camera moves. The light propagation volumes technique (Kaplanyan and Dachsbacher, 2010), which our method improves, offers good visual results. This technique is presented in more detail in the next subsection. The techniques that use the scene geometry in the form of voxel representation can also simulate the direct illumination of area light sources, as shown by Villegas and Ramírez (Villegas and Ramírez, 2016).

2.2 Light Propagation Volumes

This technique was proposed by Kaplanyan and Dachsbacher (Kaplanyan and Dachsbacher, 2010) to produce real-time indirect illumination and uses a 3D grid, known as light propagation volume (LPV). Every LPV cell stores the radiance that leaves the center of the cell in all directions around it. To store such a distribution around a point, spherical harmonics are used. In order to compute the indirect illumination for every pixel on the screen, obtained by rendering the scene from the position of the observer, the information from the LPV cell whose center is closest to the 3D space position of the pixel is queried. In this way, the radiance that reaches the pixel from the center of the LPV cell is obtained. However, for this purpose it is required to compute in each cell the radiance that reaches its center from all areas of the scene. This process is done in two steps. First, an initial radiance is introduced in the cells whose center is closest to the positions where the light is reflected by the surfaces of the scene. In the second step, this radiance information is propagated to all LPV cells. The second step propagates the radiance information through an iterative process, that at every iteration, propagates the radiance information on each cell to all its neighbors. This can also be a gathering process, in which at each iteration, each cell accumulates the radiance information from its neighbors.

Kaplanyan and Dachsbacher (Kaplanyan and Dachsbacher, 2010) specified that the initial radiance can be obtained from any source. In order to simulate the area light sources direct illumination, they proposed sampling the area lights surface in virtual point lights (VPLs), which would be subsequently introduced into the LPV. Di Koa et al. (Di Koa et al., 2017) improved this approach and used a Poisson sampling to acquire the VPLs. Di Koa and Johan (Di Koa and Johan, 2016) proposed a technique to obtain a second bounce of light for area light sources, by using the LPV for direct illumination and the information on the screen for indirect illumination. However, this approach has problems due to the fact that it is limited to the information available on the screen for the second bounce of light.

2.3 Mesh Voxelization

A voxel represents a unit of space of cubic shape and is oriented parallel to the coordinates axes. The representation of spatial information as a grid of voxels has applicability in generic volumetric rendering.

The conversion of polygonal meshes into a voxel volume is known as voxelization. Several real-time techniques (Fang and Chen, 2000; Dong et al., 2004) have taken advantage of the hardware acceleration of GPUs. Eisemann and Décoret (Eisemann and Décoret, 2008) proposed an approach that uses the GPU rasterizer to voxelize the mesh representation of the scene in a single rendering pass, but can only produce binary information per voxel. Schwarz and Seidel (Schwarz and Seidel, 2010) also proposed a single pass solution, but without limitations on the information stored in a voxel.

3 PROPOSED METHOD

Our method is an improvement of the light propagation volumes technique (Kaplanyan and Dachsbacher, 2010). Its authors proposed to simulate the direct illumination of area light sources by sampling the surface of these sources and creating a VPL for every sample. These VPLs are subsequently introduced into the LPV as initial radiance. However, they failed to provide a sampling strategy and a distribution of the samples. They stated that they simulated area light sources direct illumination with a dense sampling of VPLs without providing an analysis and a quantitative evaluation of this adaptation.

The purpose of this set of samples proposed by the authors is to approximate the area light source direct illumination inside the LPV. A sample set that does not cover every LPV cell that intersects the area light source surface does not introduce the radiance in all the cells necessary to approximate the direct illumination of this source. This scenario is visible in Panel (a) of Figure 1, where it can be seen that the visual result is not plausible because the scene is dimly lit compared to the surface of the area light source. By contrast, in Panels (b) and (c) of Figure 1, the visual results obtained with two sample sets that cover all intersecting LPV cells can be observed.

This initially injected illumination is later propagated throughout the LPV so that it can be quickly accessed at any point in the scene. The radiance of a VPL is injected into the LPV cell whose center is closest to the VPL position. Thus, the VPL position is not stored inside the LPV and it is considered that the radiance emission is made from the center of this LPV cell. In the situation where the intersection area between the area light source surface and a LPV cell has homogeneous information on its entire surface, such as the same radiant flux and emission direction, a single VPL is sufficient to approximate the direct illumination of the area light source in the intersected LPV cell. Such a scenario is visible in Panel (c) of Figure 1, where it can be observed that the same visual result is obtained as in Panel (b) of Figure 1, for which a larger sample set was used. All VPLs in the same intersection area with homogeneous information on its surface contain the same information except for the position, which is not stored inside the LPV cell. For a mesh representation of the area light source, this homogeneity on the surface of the intersection area is obtained when the homogeneity is at the level of an entire triangle. We continue this section by presenting this simplified scenario with homogeneous information and subsequently we present our proposal for the situation in which the information is not homogeneous at the level of an entire triangle. For both scenarios, our proposal uses only one sample per intersection area.

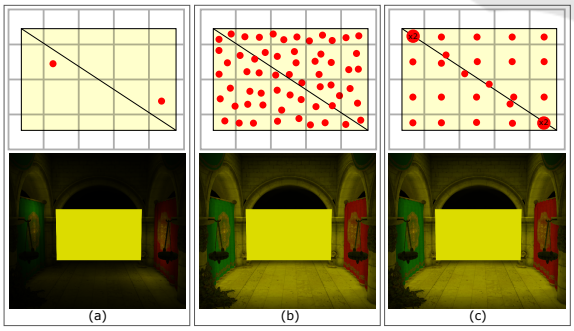

Figure 1: Three panels that show different scenarios of sampling the triangles of the area light source mesh. The yellow rectangle represents a planar area light source composed of two triangles. The top figure in each panel shows the distribution of samples that approximate the area light source direct illumination. The samples are marked by red dots, inside the triangles of the mesh and the LPV grid. Panel (a): an under-sampling scenario is presented. Panel (b): an over-sampling scenario. Panel (c): The VPL distribution obtained by our voxelization sampling method. The large red dots mark two overlapping samples.

The use of a single sample per intersection area offers multiple performance advantages. It should be mentioned that the sample set cannot be processed entirely offline and the radiance of the VPLs in it must be injected into the LPV at each frame based on the transformations of the area light source and the position of the LPV. Thus, a large set of VPLs has a high cost because it requires processing at each frame. For this reason, dense sampling is not always the optimal solution for the simulation of area light source direct illumination.

3.1 Sampling by Voxelization

We further refer to the term intersection area for the entire portion of a triangle that is found inside a LPV cell. To obtain all the intersection areas of an area light source mesh, we use a process similar to the voxelization approach proposed by Schwarz and Seidel (Schwarz and Seidel, 2010). This technique uses the GPU rasterizer to voxelize the geometry in a single rendering pass. The pipeline of our approach is shown in Figure 2 and is described below.

The voxelization process starts by rendering the area light source mesh in a viewport having the resolution of the LPV faces. In the geometry shader, the vertices of the triangle are projected on the plane of one of the LPV faces. In the fragment shader, the clipspace coordinates (x, y, z) are normalized to the resolution of the LPV $(side^3)$, obtaining the position

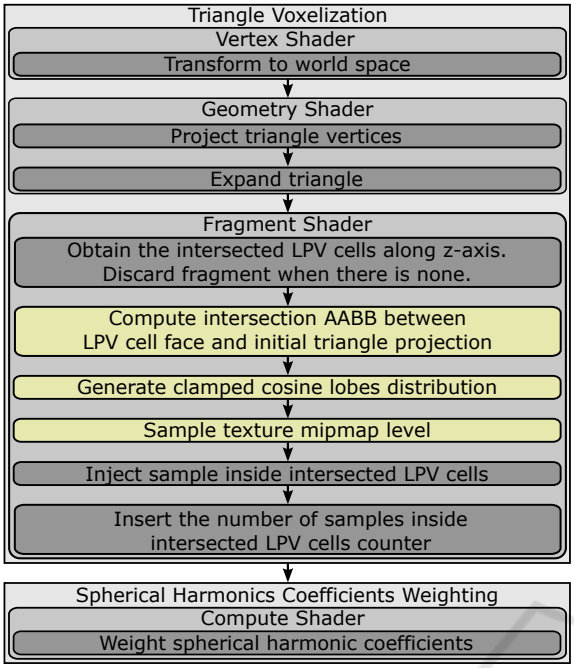

Figure 2: The pipeline of our sampling method that uses the voxelization of the area light source mesh. Operations marked in yellow represent the necessary additional steps when the information is not homogeneous on the entire surface of the mesh triangles.

of the cell (A_x, A_y, A_z) in the LPV grid.

A triangle can intersect several LPV cells on the z-axis, cells for which no fragments are produced by the voxelization approach we use. Thus, all LPV cells that have the same coordinates (A_x, A_y) must be checked whether they intersect with the triangle. For this process, we used the test proposed by Akenine-Möller (Akenine-Möller, 2001). To optimize this approach, we project the triangle on the LPV face for which the triangle projection has the largest surface. This approach ensures that a maximum of 2 cells intersect the triangle projection on the z-axis. To obtain the 2 cells, we check only the following 3 cells: $\{(A'_x, A'_y, A'_{z-1}), (A'_x, A'_y, A'_z), (A'_x, A'_y, A'_{z+1})\}$, where (A'_x, A'_y) represents the coordinates on the face chosen for the projection of the triangle and A'_z is the coordinate in depth along the axis perpendicular to this face. Such a scenario is presented in Figure 3.

The GPU rasterizer does not generate the fragments whose centers are outside the rasterized triangle, even when the triangle partially intersects them. Thus, a VPL for the intersection area cannot be generated when the fragment shader process has not started. To generate these fragments, we extend the rasterized triangle outwards with the approach proposed by Hasselgren et al. (Hasselgren et al., 2005). This approach sends to rasterization a larger triangle which covers the centers of all partially intersected fragments. However, the larger triangle can also cover the center of other fragments that are completely outside the triangle. In order not to introduce samples into LPV cells that should not receive radiance, we compute the cells of the LPV, along the z-axis, which intersect the initial triangle, before the expansion process.

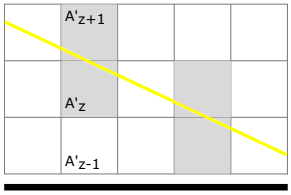

Figure 3: Section perpendicular to the plane of the LPV face on which the triangle is projected. The section of the triangle is marked in yellow. The section of the plane on which the triangle is projected is marked in black. The gray cell pairs are intersected by the section of the triangle along the axis perpendicular to the face on which it is projected.

For every intersection area, a VPL is created with the position of the LPV cell center in which the area is located and with the direction of radiance emission and the radiant flux of the triangle. This situation is valid when the information is homogeneous over the entire surface of the triangle. Because the intersection areas are computed for every triangle, this approach covers the situation in which the triangles have different information from others.

The radiance of each VPL is injected into the spherical harmonic inside the LPV cell similarly to that of the VPLs for indirect illumination in the light propagation volumes technique. We use a clamped cosine lobe in terms of zonal spherical harmonics along the negative z-axis (Ramamoorthi and Hanrahan, 2001) rotated in the direction of radiance emission (Sloan, 2008). To obtain the radiance distribution for all VPLs in a LPV cell, we use:

$$D' = \frac{1}{N} \sum_{i=0}^{N} D_i \cdot C_i \qquad (1)$$

where D' represents the spherical harmonic coefficients that approximate the radiance distribution in the LPV cell, N is the number of VPLs inside the cell and D_i represents the coefficients of the clamped cosine lobe in terms of zonal spherical harmonics that approximate the radiance distribution of the i-th VPL. We compute the coefficients D' of 3 spherical harmonics inside each cell, one for every color channel $\{R, G, B\}$. Thus, C_i represents the intensity of the color channel corresponding to the coefficients D' for the i-th VPL. We use 2-band spherical harmonics, so a total of 4 coefficients are used.

We count the number of VPLs in each LPV cell

in parallel with the computation of the spherical harmonic coefficients corresponding to the cell. During the voxelization process, we store this VPL counter for each cell and we update the coefficients without weighting. After this process, we weight all the coefficients in a compute shader.

The GPU can start several fragment shader processes in parallel, so that several VPLs can be injected in a LPV cell at the same time. To avoid concurrency problems, we use atomic operations to update the spherical harmonic coefficients and the counter inside the LPV cells.

3.2 Radiance Distribution

To store a specific radiance distribution on the surface of a mesh triangle, we use an emission direction for every triangle vertex. The emission direction at a point inside the triangle is computed through interpolation.

In this situation, where the emission direction is not the same on the entire surface of the intersection area, it is necessary to create a spherical harmonic that approximates this radiance distribution. Several methods (Wang and Ramamoorthi, 2018; Belcour et al., 2018) have been proposed to compute such a spherical harmonic for any number of bands. However, due to the fact that we use 2-band spherical harmonics, such an approach would be unnecessary. Our approach to approximate the radiance distribution uses a set of clamped cosine lobes to approximates all emission directions on the intersection area surface.

To obtain the distribution of the clamped cosine lobes, we compute the projection of the intersection area on the LPV face on which the triangle is projected. For simplicity, we compute only the axis-aligned bounding box (AABB) of the intersection area between the face of the LPV cell and the projection of the initial triangle, before expansion. For this process, we compute the AABB of the polygon formed by the projected vertices of the triangle that are inside the LPV cell face, the corners of the LPV cell face that are inside the triangle projection and the intersection points between the edges of the LPV cell face and the edges of the triangle. These points can be seen in Figure 4, marked with red dots.

After obtaining this AABB of the intersection area projection, the emission direction in each corner of the AABB is computed. The clamped cosine lobes are distributed between the emission directions from the corners, vertically and horizontally at angular distances of the same degree as that of the cosine lobe. Each generated cosine lobe is injected into the LPV cell with a weight of 1. The only observation is that

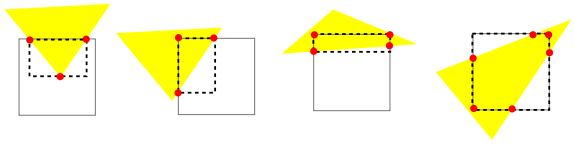

Figure 4: Several examples of AABBs, marked with dotted line, computed for the intersection of the triangle projection, marked in yellow, with a LPV cell face. An AABB is obtained for the polygon determined by the points marked with red dots.

this distribution is created locally and only the final approximation with the weight as the number of lobes required for the approximation is injected inside all LPV cells intersected along the z-axis.

To obtain the emission direction in a corner of the AABB, we compute the barycentric coordinates of the respective corner inside the initial triangle. With these barycentric coordinates, the direction from the corner of the AABB is computed based on the direction information from the vertices of the triangle.

This approach can produce minor visual errors because it generates a single radiance distribution for the AABB of the intersection area projection. This intersection area, as presented in Section 3.1, can cover two LPV cells. For a correct computation of the radiance distribution, the intersection area for every intersected LPV cell must be processed.

3.3 Textured Area Lights

To store the radiant flux information on the surface of a triangle mesh, we use the texture mapping technique. We use texture coordinates at the vertices of the triangle and interpolate them to obtain the texture coordinate at any point inside the triangle. This coordinate is used to sample the radiant flux map.

To approximate the entire radiant flux of the intersection area, we use, similarly as in Section 3.2, the AABB of this area projection on the LPV face on which the triangle is projected. Based on the barycentric coordinates of the AABB center, the texture coordinate from this position is computed. To quickly compute the entire radiant flux information of the intersection area, the mipmaping technique is applied to the radiant flux map. We compute the mipmap level required to approximate the radiant flux similar to the specifications of the OpenGL standard (Segal and Akeley, 1999), based on the texture coordinates in the corners of the AABB.

This approach, similar to Section 3.2, can produce minor visual errors due to the fact that a single radiant flux sample is computed for all LPV cells intersected along the z-axis. In addition, due to the approximation made by the mipmapping technique, radiant flux

information can be introduced from areas outside the intersection area.

4 RESULTS

4.1 Dense Sampling

Because the authors of the light propagation volumes technique did not provide a strategy to sample the surface of the area light sources, we had to design our own strategy to obtain a dense sampling. We compare the results obtained by this strategy with our optimal method that uses the voxelization process of the area light source mesh.

To obtain a dense sampling, we decided to sample each triangle of the area light source mesh with a number of samples evenly distributed inside the triangle. The distribution of the samples inside a triangle was made similarly to Turk (Turk, 1990):

$$P = (1-\sqrt{r_1})\cdot A + \sqrt{r_1}\cdot(1-r_2)\cdot B + \sqrt{r_1}\cdot r_2 \cdot C \quad (2)$$

where P is the sample position, (A,B,C) are the vertices positions of the triangle and $\{r_1, r_2\}$ are two random numbers in the interval $[0,1]$.

4.2 Evaluation

We implemented both the dense sampling strategy described in Section 4.1 and our method described in Section 3 in the C++ programming language with OpenGL 4.6 graphics API. Performance evaluation was performed on a machine with Nvidia GTX 1660 GPU.

We evaluated 5 different area light source meshes with a LPV of 32^3 and 64^3 resolutions. For the dense sampling strategy, we used 3 sets of samples with 1, 100 and 1 000 samples/triangle. The results of the quantitative evaluation are presented in Table 1 and the visual results, only for 64^3 LPV resolution, are presented in Figures 5 and 6.

We also made a quantitative comparison between the visual results from Figures 5 and 6. We computed the mean squared error (MSE) between the pixels of each image obtained with a dense sampling and the ones of the image obtained with our method for every area light source mesh. This MSE is found under each image obtained with the dense sampling in Figures 5 and 6.

It is visible in Table 1 than a LPV resolution of 64^3 has better performance that a 32^3 resolution for almost all test data. This is because atomic operations were used to inject the radiance of each sample into the LPV, as presented in Section 3.1. Thus, for a high LPV resolution, few samples are introduced into the same cell. From our tests, this operation is the most expensive for the entire sampling process, far beyond the actual samples generation.

The performance of the strategy that obtains a dense sampling is poor for a large number of triangles, for obvious reasons. The best performance is obtained for 1 sample/triangle, but as can be seen in Figure 5, the visual results are not always the best and can produce a dimly lit scene. The visual results become plausible and approach the optimal result when a large number of samples is used. Our method obtains this result for area light source meshes that have homogeneous information on the surface of all triangles. It detects the LPV cells intersected by each triangle and produces a single sample for every one. The use of a dense sampling does not always guarantee that a sample is produced for an intersection area.

The reasons why our method obtains optimal visual results are explained in Section 3, but a practi-

Table 1: Performance evaluation for both the strategy that obtains a dense sampling and our method that uses the voxelization of the area light source mesh. The number of triangles for every evaluated mesh is as follows: Quad - 2 triangles, Square Pyramid - 6 triangles, Tube - 12 triangles, Sphere - 720 triangles and Stanford Lucy - 33 446 triangles.

LPV res.	Area light source mesh	Sampling strategy			
		Dense sampling			Our method
		1 smp/tri	100 smp/tri	1 000 smp/tri	
32^3	Quad	0.007 ms	0.007 ms	0.015 ms	0.024 ms
	Square Pyramid	0.007 ms	0.009 ms	0.034 ms	0.024 ms
	Tube	0.007 ms	0.012 ms	0.053 ms	0.025 ms
	Sphere	0.009 ms	0.235 ms	2.475 ms	0.031 ms
	Stanford Lucy	0.132 ms	9.850 ms	98.500 ms	0.307 ms
64^3	Quad	0.007 ms	0.007 ms	0.015 ms	0.020 ms
	Square Pyramid	0.007 ms	0.009 ms	0.024 ms	0.020 ms
	Tube	0.007 ms	0.016 ms	0.047 ms	0.020 ms
	Sphere	0.008 ms	0.140 ms	1.581 ms	0.032 ms
	Stanford Lucy	0.123 ms	9.250 ms	91.760 ms	0.304 ms

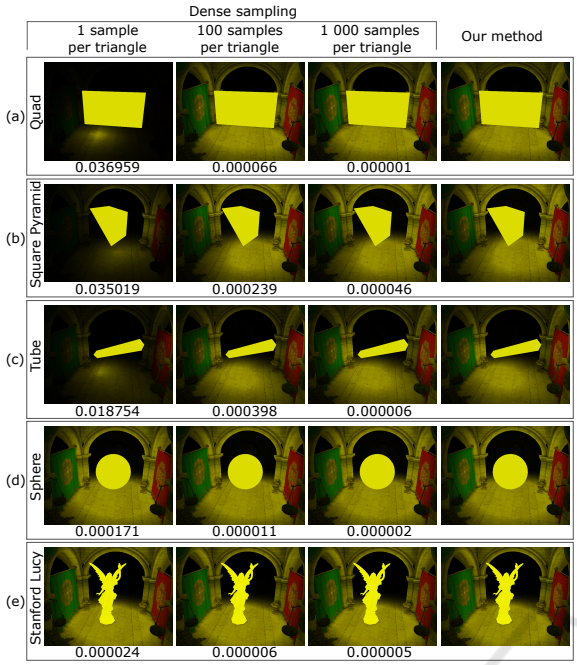

Figure 5: The visual results obtained with both the strategy that obtains a dense sampling and our method that uses the voxelization of the area light source mesh. For the first method, the visual results obtained for the use of 3 sets of samples with 1, 100 and 1 000 samples/triangle are presented. The number below each image represents the MSE between the pixels of the image and the ones of the image obtained with our method, visible on the right side of each panel.

cal example is also provided in Figure 5. This figure shows the MSE between the pixels of each image obtained with a dense sampling and those of the image obtained with our method. It can be seen that the MSE is small when a dense sampling with 100 samples/triangle is used and approaches 0 when the number of samples/triangle is 1 000. Thus, it can be observed that our method offers visual results almost identical to these provided by the use of a dense sampling with a large sample set.

Our method provides optimal visual results in general at performances superior to a dense sampling. This can be observed from the quantitative evaluations visible in Table 1 and Figure 5, where in Panel (b) of Figure 5 the optimal visual result is obtained for 1 000 samples/triangle in 0.034 ms compared to our method that obtains an almost identical result in 0.024 ms for a LPV resolution of 32^3. A similar situation is in Panel (c), with 1 000 samples/triangle with a performance 2 times poorer than that of our method, in Panel (d) with 100 samples/triangle at a performance 4.3 times poorer and in Panel (e) with 100 samples/triangle at a performance 30.4 times poorer. The only area light source mesh for which the use of a dense sampling has superior performance to our method is the one in Panel (a) of Figure 5, where for 1 000 samples/triangle, dense sampling obtains the optimal result at a performance of 0.6 compared to our method.

For the situation where the information is not homogeneous on the entire surface of the mesh triangles, our method produces visual results similar to those obtained by a dense sampling. This is visible in Panel (a) of Figure 6. Our approach can produce minor visual errors, as explained in Sections 3.2 and 3.3. However, it can be seen that the MSE between the pixels of the image obtained with a dense sampling and those of the image produced by our method is small when the number of samples is large for the dense sampling. Therefore, the result obtained by our method is close to the optimal one, obtained with the dense sampling when a large number of samples is used.

Our method is an improvement of the light propagation volumes technique. It offers an optimal distribution of VPLs that approximates the direct illumination of area light sources, but after this step, the light propagation volumes technique does not support changes. Therefore, the direct lighting simulation has the same level of flexibility and quality of the visual results as this technique allows. Our method can produce, at different quality levels, several lighting effects. The effect with the highest quality of the visual results is the direct diffuse illumination, visible in Panel (a) of Figure 6. The direct specular illumination is not of good quality due to the rather low resolution of the LPV and the use of 2-band spherical harmonics, as shown by Lambru et al. (Lambru et al., 2021). Both limitations are motivated by performance. The visual results are visible in Panel (b) of Figure 6.

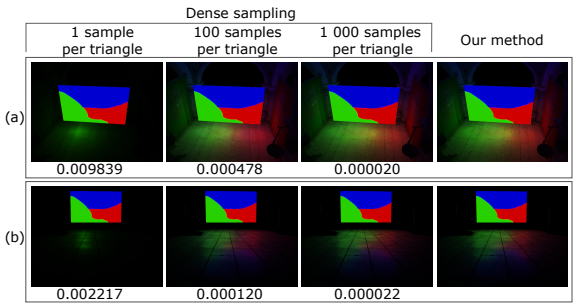

Figure 6: The visual results obtained with both the strategy that obtains a dense sampling and our method that uses the voxelization of the area light source mesh. The number below each image represents the MSE between the pixels of the image and the ones of the image obtained with our method, visible on the right side of each panel. Panel (a): only the direct diffuse illumination is presented. Panel (b): only the direct specular illumination.

5 CONCLUSIONS

In this paper, we have presented an improvement of the light propagation volumes technique for the particular case of simulating area light sources direct illumination. This technique requires the approximation of these sources direct illumination in the form of a set of VPLs, that are later used as input in the technique. The method proposed by us in this article provides an optimal distribution of VPLs on the area light source surface, which offers an optimal approximation for the light propagation volumes technique.

A further improvement can be represented by the better approximation of inhomogeneous information on the surface of intersection areas. The explanation of the visual errors obtained with our approximation was provided in Sections 3.2 and 3.3, together with a theoretical solution to obtain a better approximation.

Based on the light propagation volumes technique, our method offers plausible visual results for both diffuse and specular direct illumination. However, there are other illumination effects that could be produced, such as shadows, volumetric lighting and the simulation of supplemental light bounces for area light sources. These illumination effects are important for area light sources and their simulation may be a topic of interest in the future.

REFERENCES

Akenine-Möller, T. (2001). Fast 3d triangle-box overlap testing. *J. Graph. Tools*, 6(1):29–33.

Arvo, J. (1995). Applications of irradiance tensors to the simulation of non-lambertian phenomena. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, page 335–342, New York, NY, USA. Association for Computing Machinery.

Belcour, L., Xie, G., Hery, C., Meyer, M., Jarosz, W., and Nowrouzezahrai, D. (2018). Integrating clipped spherical harmonics expansions. *ACM Trans. Graph.*, 37(2).

Di Koa, M. and Johan, H. (2016). Efficient screenspace rendering for area lights. In *Proceedings of the 33rd Computer Graphics International*, CGI '16, page 29–32, New York, NY, USA. Association for Computing Machinery.

Di Koa, M., Johan, H., and Sourin, A. (2017). Voxel-based interactive rendering of translucent materials under area lights using sparse samples. In *2017 International Conference on Cyberworlds (CW)*, pages 56–63.

Dong, Z., Chen, W., Bao, H., Zhang, H., and Peng, Q. (2004). Real-time voxelization for complex polygonal models. In *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.*, pages 43–50.

Drobot, M. (2014). Physically based area lights. In Engel, W., editor, *GPU Pro 5*, pages 67–100. CRC Press.

Eisemann, E. and Décoret, X. (2008). Single-pass gpu solid voxelization for real-time applications. In *Proceedings of Graphics Interface 2008*, GI '08, page 73–80, CAN. Canadian Information Processing Society.

Fang, S. and Chen, H. (2000). Hardware accelerated voxelization. *Computers & Graphics*, 24(3):433–442.

Hasselgren, J., Akenine-Möller, T., and Ohlsson, L. (2005). *Conservative Rasterization*, pages 677–690. Addison-Wesley.

Heitz, E., Dupuy, J., Hill, S., and Neubelt, D. (2016). Real-time polygonal-light shading with linearly transformed cosines. *ACM Trans. Graph.*, 35(4).

Kaplanyan, A. and Dachsbacher, C. (2010). Cascaded light propagation volumes for real-time indirect illumination. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '10, page 99–107, New York, NY, USA. Association for Computing Machinery.

Lambru, C., Morar, A., Moldoveanu, F., Asavei, V., and Moldoveanu, A. (2021). Comparative analysis of real-time global illumination techniques in current game engines. *IEEE Access*, 9:125158–125183.

Lecocq, P., Dufay, A., Sourimant, G., and Marvie, J.-E. (2017). Analytic approximations for real-time area light shading. *IEEE transactions on visualization and computer graphics*, PP.

Nichols, G., Penmatsa, R., and Wyman, C. (2010). Interactive, multiresolution image-space rendering for dynamic area lighting. *Computer Graphics Forum*, 29(4):1279–1288.

Picott, K. P. (1992). Extensions of the linear and area lighting models. *IEEE Computer Graphics and Applications*, 12(02):31–38.

Ramamoorthi, R. and Hanrahan, P. (2001). On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *J. Opt. Soc. Am. A*, 18(10):2448–2459.

Schwarz, M. and Seidel, H.-P. (2010). Fast parallel surface and solid voxelization on gpus. *ACM Trans. Graph.*, 29(6).

Segal, M. and Akeley, K. (1999). The opengl graphics system: A specification (version 1.1).

Sloan, P.-P. (2008). Stupid spherical harmonics (sh) tricks. Game developers conference.

Snyder, J. (1996). Area light sources for real-time graphics. Technical Report MSR-TR-96-11.

Turk, G. (1990). Generating random points in triangles. *Graphics gems*, pages 24–28.

Villegas, J. and Ramírez, E. (2016). Deferred voxel shading for real-time global illumination. In *2016 XLII Latin American Computing Conference (CLEI)*, pages 1–11.

Wang, J. and Ramamoorthi, R. (2018). Analytic spherical harmonic coefficients for polygonal area lights. *ACM Trans. Graph.*, 37(4).