Dynamic Anisotropic Occlusion

Yi Gong, Yubo Zhang, Wei Chen†, Qunsheng Peng

State Key Lab of CAD&CG, Zhejiang University, 310058, Hangzhou, China

Abstract

We present a new real-time shadow rendering approach whose kernel is a hierarchical disk-based approximation to the scene geometry. We show that this approximated representation greatly accelerates the visibility inquiry, facilitating the real-time computation of anisotropic occlusion in GPU. By taking account of anisotropic occlusion, our approach can simulate the shadow cast from not only ambient light, but also point, directional and environment lights. Because no pre-computation is required, it is suitable for dynamically deforming objects.

1. Motivation

Ambient occlusion has been popularly used in the light attenuation simulation caused by the occlusion among scene geometry. It was firstly introduced by Zhukov *et al.* [DS03], and popularized to graphics community by Landis et al. [LGBK02] [JB02] [JMLH01]. The main idea is to compute the extent of ambient attenuation over a point by evaluating the occlusion from nearby scene. The ambient occlusion evolved till now can be expressed as the following function:

$$A(x) = \frac{1}{\pi}\int_{\Omega} V(x,\omega)(\omega\cdot\mathbf{n})d\omega \qquad (1)$$

Here x is the point's location and $\mathbf{n}$ stands for its normal. $V(x,\omega)$ is the visibility function of x along direction ω. The ambient occlusion equals 1 if the point's hemisphere is entirely covered by other objects and zero for none coverage. While ambient occlusion can generates soft shadow due to ambient light, it inherently can not simulate the shadow from non-uniform distributed lights, like point, directional, or non-uniform environment lights.

In 2004, Sattler et al. [LW96] took advantage of occlusion query function provided by graphics hardware to compute the visibility of surface points from directional light source. They also took account of complex illumination from various directions. However, they employed an environmental cubemap and sampled a sequence of points as the light sources. The visibility was then tested from these points' view to determine whether the vertices of the handled model can be illuminated by environment light in these directions. Thus, for slightly complex environment illumination, this approach requires rendering objects several times because a large amount of points on the cubemap have to be sampled, leading to the same number of light sources for visibility testing.

Meanwhile, various other improvement of ambient occlusion have been made. Mendez et al. [JC98] suggested a method for dynamically updating the obscurances information of the moving objects. Kautz et al. [BSS93] created a hemispherical rasterizer which can rapidly compute visibility with a lookup table. They also proposed to compress environment light by spherical harmonics (SH) function. Although this approach is very fit for simulating self-shadow, its efficiency is cut down dramatically if the inter-object shadows are considered. Kontkanen et al. [Sta95] introduced the concept of ambient occlusion field and used it to determine the shadow casting on the receiving objects. Recently, there is a trend to use simplified geometry to participate in high cost calculation, for instance, occlusion evaluation. Bunnell [RT90] presented a dynamic ambient occlusion approach to tackle animated objects. It reformulates the scene geometry as a set of disks, which is similar to our approach. More recently, Hegeman et al. [Wil78] took use of ambient occlusion to approximate the shadowing effect in a vegetation scene. They simulated occlusion from fragmentary trees by approximating a branch of leaves with its bounding plane. Note that, all these aforementioned methods consider only the occlusion to a uniform light to simulate shadow.

On the other hand, most shadow rendering techniques, such as shadow map [MKB*03a] and shadow volume

† Corresponding author, chenwei@cad.zju.edu.cn

[TWL*05] techniques, cannot handle environment light, no matter uniform or non-uniform environment illumination. Precomputed Radiance Transfer (PRT) scheme introduced by Solan *et al.* [SKS02] does take the environment illumination into account, but is typically limited to static models as preprocess is needed. The recent work of PRT [Sil95] applied zonal harmonics (ZH) to enable local deformation, while is unsuitable for dynamic models.

The goal of this paper is to find a simple yet efficient way to fast render shadow of dynamic scenes in complex illumination circumstances. Instead of assigning a single value for one surface location to simulate ambient occlusion, we consider the occlusion from multiple directions by employing a hemi-cube [MKB*03b] to collect the occlusion information from all possible directions. As we execute occlusion estimation and final shading in one pass, memory cost and times of texture access will be reduced. In this pass, we combine the lighting information with the calculated occlusion values of several discrete directions, yielding a comprehensive shading result. To simplify the computation, we approximate the scene geometry with a hierarchical tree whose nodes are a set of disks. In each disk, the center location, normal and approximated area are recorded. One main advantage of the approximated representation is that it greatly simplifies the face-to-face occlusion evaluation. Meanwhile, all kinds of lighting illumination can be compressed and recorded in a set of cubemaps using SH function, facilitating efficient illumination calculation in GPU. Therefore, compared with Sattler's method, our algorithm is much more efficient to process different kinds of lights.

2. Geometry Processes

We transfer the original meshes to a disk-based hierarchical structure–each vertex is replaced by a disk element storing its location, normal, and average area of its neighbor region, as depicted in Figure 1.

Based on the investigation that the influence from far scene is much less than near scene, we divide all disk elements into groups according to their texture coordinates and arrange them into a hierarchical bi-tree structure. The texture coordinates can be generated use various mapping methods, e.g., spherical mapping, cylinder mapping and so on. Perfect texture mapping method is not required because it is used for easy grouping of vertices. This representation facilitates efficient LOD handling, i.e., occlusion from a group of faces far away can be approximated by those of their parent node-a larger disk whose value is the average of those of its child nodes. On the other side, when the occluder is very close to the shadow receiver, the highest resolution is taken.

The bi-tree structure is created with the texture coordinates of vertices once the scene geometry is loaded. During the animation of dynamic models, the bi-tree's nodes has to be dynamically updated to reflect the change of the area,

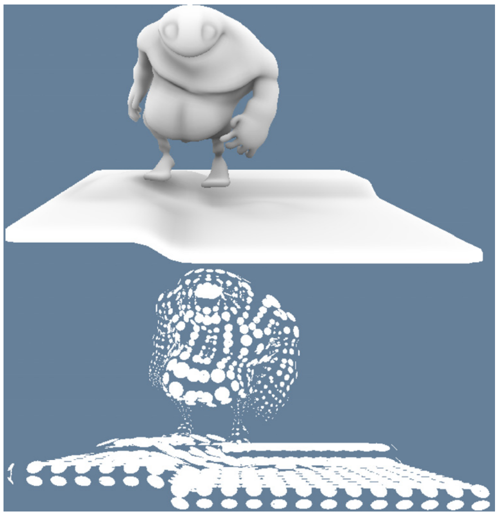

Figure 1: *A model (top) and its disk-based structure (bottom).*

normal and location of disks. To ease GPU's access to the information of these trees, these disk trees are rendered into textures at each frame.

3. Anisotropic Occlusion

Traditional ambient occlusion approach sums up occlusion in all directions equally based on the fact that the ambient light is isotropic. When it comes to environment or directional lights, occlusion over a surface point in different directions will lead to different influence to final outgoing radiance of that point:

$$L_o(x,\omega_o) = \int_\Omega f(x,\omega_o,\omega_i)L_i(x,\omega_i)V(x,\omega_i)(\mathbf{n}\cdot\omega_i)d\omega_i \quad (2)$$

where Ω is the hemisphere domain over x, L_i denotes the incident radiance from direction of ω, f is the BRDF and V is the visibility function. For common environment light and materials, L_i changes in low frequency and BRDF is uniformly distributed. If we further neglect visibility function's sharp change with direction parameter, for the sake of efficiency, we can simplify Equation (2) by discretizing the hemisphere into several directions:

$$L_o(x,\omega_o) \approx \frac{2\pi}{k}\sum_{k=1}^{n} L_i(x,\omega_k)V(x,\omega_k)(\mathbf{n}\cdot\omega_k) \quad (3)$$

It is easy to find out that

$$V(x,\omega_k) = 1 - O(x,\omega_k) \quad (4)$$

where O is the extent of occlusion in given direction.

The remained challenge is how to estimate occlusion over x in several directions. In our implementation, we divide the unit hemisphere space into 16 parts by a uniform grid in xz

plane. Accordingly, the sampling rate increases with the degree of how the direction close to the surface orientation, which happens to match the influence trends of occluder in different directions. In other words, occluder close to the receiver normal will occlude more light than occluder in direction parallel to the receiver surface. After this division, occluder's coverage in each direction can be approximated by the function as follows:

$$O(x, \omega_k) = \frac{(r_o(\omega_k \cdot \mathbf{d}))^2}{2k\pi\mathbf{d}^3} \quad (5)$$

where ω_k denotes certain direction over the hemisphere of point x.

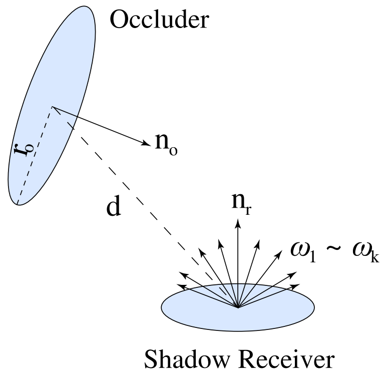

Figure 2: *Occlusion estimation between two disk faces.*

4. Image-space Culling

It is apparent that only the visible surfaces needs to be shaded for each viewpoint. Taking all vertices as receivers to compute the occlusion is redundant. To omit the computation for other invisible surface parts, we employ the early-z-test, occlusion-query and render-to-vertex-array features supported by programmable graphics hardware to perform image-space culling. In the first pass, we render the whole scene without color shading into a renderable buffer. Then we render all vertices as point primitives to check whether they are occluded by other geometry. Finally, the unoccluded vertices are rendered into a vertex array to execute occlusion calculation. For complex and dynamic scenes, this image-space culling scheme can achieve 25% – 30% speedup.

5. Results

We have implemented our algorithms with Cg shading language [HK93]. Performance has been measured on a 2.4 GHz P4 system with 1 GB memory and an NVidia 6800 video card with 256 MB video memory.

We tested three models, including static and deformable ones. Table 1 lists the corresponding configuration and performance. Figure 3, Figure 4 and Figure 5 illustrate several snapshots for these models. For more information, please refer to our submitted videos.

Data	Triangles	Vertices	FPS
Torus	1300	1321	40.64
Shapes	4136	2072	22.31
Big Guy(Animated)	1452	1450	32.87

Table 1: *Performance statistics for various data sets.*

6. Conclusion and Future Works

In this paper, we present a new dynamic anisotropic occlusion simulation algorithm. It can generate shadow by considering the occlusion information from arbitrary direction for each surface location. In contrast to previous techniques, our method is not limited to process simple directional light or point-like light and is suitable to dynamic objects. It achieves high efficiency by fully exploiting the power of GPU, like the newest features of dynamic branches and deferred shading.

Our future work includes bleeding effect process and reducing the frequency of model updating. As our occlusion computation shares some commonplace with the form factor calculation in radiosity methods [MKB*03b], it may be fit for adding bleeding effect. However, there are some unsolved issues, e.g., the depth testing is necessary to check whether the occluder is the closest one to receiver. On the other hand, model updating costs lots in current algorithm, we need to find a way to avoid unnecessary updating. [Sil95], [NVI05], [WTL05], [HV04], [GLL*04].

7. Acknowledgements

This work is partially supported by 973 program of China (No.2002CB312100), NSF of China for Innovative Research Groups (No.60021201), National Natural Science Fund of China (No.60303028), Zhejiang Provincial Natural Science Fund (No.Y105269) and Zhejiang Provincial Natural Science Special Fund for Youth Scientists' Cultivation (No.R603046).

References

[BSS93] Blasi P., Saëc B. L., Schlick C.: A rendering algorithm for discrete volume density objects. *Computer Graphics Forum 12(3)* (1993), 201–210.

[DS03] Dachsbacher C., Stamminger M.: Translucent shadow map. In *Proceedings of Eurographics Symposium on Rendering 2003* (2003), pp. 197–201.

[GLL*04] Goesele M., Lensch H., Lang J., Fuchs C., Seidel H.-P.: Disco: acquisition of translucent objects. *ACM Transactions on Graphics 23(3)* (2004), 835–844.

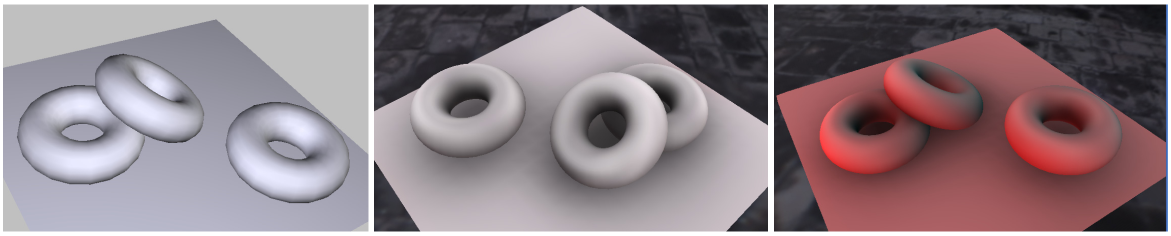

Figure 3: *Results of Torus scenes with or without anisotropic occlusion calculation. Left: without occlusion calculation; Middle: anisotropic occlusion with an environment light; Right: anisotropic occlusion with an environment light and a red point light.*

Figure 4: *Comparison between ambient occlusion (left) and our anisotropic occlusion (right). Note that our algorithm generates more visually plausible shadow with comprehensive illumination environment.*

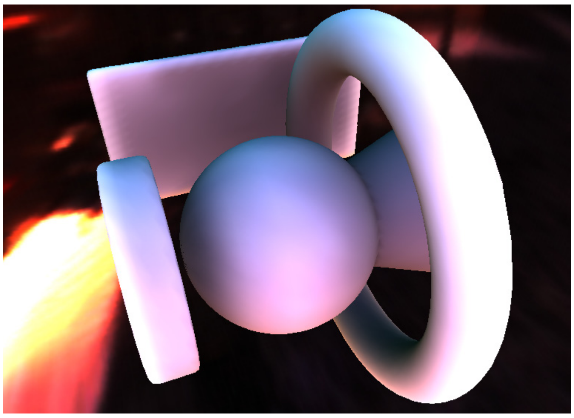

Figure 5: *Models illuminated by an environment light and 4 colored spotlights (red, blue, green and white).*

[HK93] HANRAHAN P., KRUEGER W.: Reflection from layered surfaces due to subsurface scattering. In *Proceedings of ACM SIGGRAPH 1993* (1993), pp. 165–174.

[HV04] HAO X., VARSHNEY A.: Real-time rendering of translucent meshes. *ACM Transactions on Graphics 23(2)* (April 2004), 120–142.

[JB02] JENSEN H. W., BUHLER J.: A rapid hierarchical rendering technique for translucent materials. *ACM Transactions on Graphics 21(3)* (July 2002), 576–581.

[JC98] JENSEN H. W., CHRISTENSEN P.: Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of ACM SIGGRAPH 1998* (1998), pp. 311–320.

[JMLH01] JENSEN H. W., MARSCHNER S., LEVOY M., HANRAHAN P.: A practical model for subsurface light transport. In *Proceedings of ACM SIGGRAPH 2001* (2001), pp. 511–518.

[LGBK02] LENSCH H., GOESELE M., BEKAERT P., KAUTZ J.: Interactive rendering of translucent objects. In *Proceedings of Pacific Graphics 2002* (October 2002), pp. 214–224.

[LW96] LAFORTUNE E. P., WILLEMS Y. D.: Rendering participating media with bidirectional path tracing. In *Proceedings of Eurographics Rendering Workshop 1996* (1996), pp. 91–100.

[MKB*03a] MERTENS T., KAUTZ J., BEKAERT P., REETH F. V., SEIDEL H.-P.: Efficient rendering of local subsurface scattering. In *Pacific Graphics 2003* (2003), pp. 51–58.

[MKB*03b] MERTENS T., KAUTZ J., BEKAERT P., SEIDEL H.-P., REETH F. V.: Interactive rendering of translucent deformable objects. In *Proceedings of Eurographics Rendering Workshop 2003* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 130–140.

[NVI05] NVIDIA: GPU programming guide.

[RT90] RUSHMEIER H. E., TORRANCE K. E.: Extending the radiosity method to include specularly reflecting and translucent materials. *ACM Transactions on Graphics 9(1)* (1990), 1–27.

[Sil95] SILLION F. X.: A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Transactions on Visualization and Computer Graphics 1(3)* (1995), 240–254.

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of ACM SIGGRAPH 2002* (July 2002), pp. 527–536.

[Sta95] STAM J.: Multiple scattering as a diffusion process. In *Proceedings of Eurographics Rendering Workshop 1995* (1995), pp. 41–50.

[TWL*05] TONG X., WANG J., LIN S., GUO B., SHUM H.-Y.: Modeling and rendering of quasi-homogeneous materials. In *Proceedings of ACM SIGGRAPH 2005* (2005), pp. 1054–1061.

[Wil78] WILLIAMS L.: Casting curved shadows on curved surfaces. In *Proceedings of ACM SIGGRAPH 1978* (1978), pp. 270–274.

[WTL05] WANG R., TRAN J., LUEBKE D.: All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Transactions on Graphics 24(3)* (August 2005), 1202–1207.