IEEE/ EG Symposium on Volume and Point-Based Graphics (2008)
H.- C. Hege, D. Laidlaw, R. Pajarola, O. Staadt (Editors)

Isosurface Ambient Occlusion and Soft Shadows with Filterable Occlusion Maps

Eric Penner[1] Ross Mitchell [2]

[1]Department of Computer Science, University of Calgary
[2]Department of Clinical Neurological Sciences and Radiology, University of Calgary

Abstract

Volumetric data sets are often examined by displaying isosurfaces*, surfaces where the data or function takes on a given value. We propose a new method for rendering isosurfaces at interactive rates while supporting dynamic ambient occlusion and/or soft shadows and requiring minimal pre-computation time. By approximating the occlusion in a region as the percentage of occluding voxels in that region, we reduce the ambient occlusion problem to the same problem faced in soft shadow mapping algorithms. In order to quickly extract the number of occluding voxels in an image region, we propose representing distributions using filterable representations such as variance shadow maps or convolution shadow maps. By choosing different sampling patterns from these maps we can dynamically approximate ambient occlusion and/or soft shadows.*

1. Introduction

Cast shadows are known to play a key role in human perception of the 3D world. To qualitatively and quantitatively understand the importance of shadows in our perception of the world, several studies and experiments have been conducted to understand how shadows shape our perception and understanding of a scene. Through these experiments, the importance of shadows has been demonstrated in understanding the position, size and geometry of both the shadow caster and shadow receiver. Hubona *et al.* [HSBW99] discuss the role of shadows in general 3D visualization. Wanger *et al.* [Wan92] study the effect of shadow quality on the perception of object relationships. Kersten *et al.* [KMK94] demonstrate that adjusting the motion of a shadow can dramatically effect the apparent trajectory of a shadow casting object. For the interested reader, a comprehensive discussion of real-time geometry based shadowing algorithms with references to their perceptual importance can be found in [HLHS03].

The most traditional method of computing shadows is with ray tracing using a point light source. Using this method, a ray is traced from each surface point back to the point light that illuminates the surface. If an object obscures the path of the ray, the surface is rendered without the light's contribution. Unfortunately, shadows from simple point sources produce stark discontinuities which aren't present under normal lighting conditions. The depth cues provided also vary depending on the viewing direction. For example, if the camera location aligns with the light location all the shadows become hidden behind the objects and no extra cues are provided. Methods such as shadow maps [Wil78] and shadow volumes [Cro77] can be used to accelerate point-lighting in real-time applications and produce similar results.

More realistic shadows are provided with more complex models such as *ambient occlusion* or *global illumination*. It has been shown that these realistic approaches provide better perception of many shapes than with simple point lighting. Ambient occlusion simulates light arriving equally from all directions or "light on a cloudy day" and is also referred to as *uniform diffuse lighting*. Global illumination simulates lighting from complex area light sources as well as diffuse inter-reflections and caustics. While there have been methods to compute good approximations of these lighting methods for a long time, traditional methods have always had very high computational costs. Usually the calculation involves a monte-carlo simulation of hundreds or even thousands of rays as opposed to the single ray used for a point or directional light source (see figure 1).

In order to capture these effects in interactive applications, many methods based on the theory of light transfer have

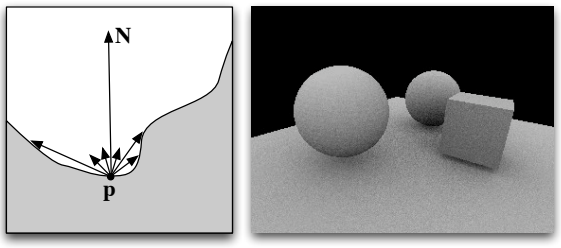

Figure 1: *Left: Illustration of calculating ambient occlusion using monte-carlo ray-tracing. Rays are chosen randomly in the hemisphere above a surface point. The percentage of occluded rays represents ambient occlusion. Right: Example using 36 randomly distributed rays. Notice the noise artifacts due to under-sampling.*

been developed to enable the pre-computation of ambient occlusion and diffuse inter-reflection. Some even allow for arbitrary modification of light and camera parameters. However, the application of these approaches to deformable geometry is much more constrained and thus usually requires a new pre-computation for any deformation.

The goal of our work is to drastically accelerate the calculation of volume ambient occlusion by taking advantage of the image based representation of medical volume data sets. While meshes are stored in a sparse format as a collection of vertices and triangles, volumes are stored in a 3D image grid which is suitable for a host of image processing operations and volume measurements that aren't possible on meshes. Our goal is to very quickly preprocess the volume in such a way that we can quickly extract information about the number of occluding voxels without having to traverse hundreds or thousands of rays. Generally speaking, we wish to replace the geometric definition of ambient occlusion with a statistical approximation that can be computed using image-processing operations.

This paper is structured as follows. In the next section we will discuss the related work with a focus on volume illumination methods. In order to approximate the occlusion for a point on an iso-surface, we query from a very compact representation of the distribution of values in the region above the surface. This is discussed in section 3. The implementation and some special considerations are discussed in section 4 and our results are discussed in section 5 before concluding in section 6 .

2. Related Work

Levoy and Cabral presented some of the first work on light transport for volumetric data sets and accelerated rendering using texture mapping [Lev90] [CCF94]. Max extended the optical models for direct volume rendering to include shadowing, single scattering an multiple scattering effects [Max95]. Max states that lighting from neighboring structures are important in volume rendering. Due to the added computational complexity involved with computing complex lighting effects most medical volume rendering applications only utilize a local illumination model due to its low computational cost. This involves illuminating the volume using one or more point light sources. In a local lighting model the light at each point in the volume is calculated as the sum of diffuse and specular components calculated from a bidirectional reflectance distribution function (BRDF) model. A BRDF model such as the popular Blinn-Phong model [Bli77] provides a means to calculate the amount of locally reflected light based on the directions of the light source, L, the viewer, V, and the surface normal (or gradient), N.

Local illumination methods provide good perceptual cues to the *orientation* of a surface within the volume, due to the diffuse $N \cdot L$ term. Surfaces are bright if lit from directly above, and dark if illuminated from a steep angle. However, as discussed above local illumination methods provide poor cues to the *relationships* between neighboring surfaces. It can be difficult to tell whether a neighboring surface is above or below an adjacent surface. Due to the extra perceptual information and realism they can provide, adding shadows and complex lighting to real-time volume rendering applications has resulted in significant research effort.

Unfortunately, these efforts are complicated by the fact that different volume rendering methods exist and have an impact on how and when light can be propagated. The two primary volume rendering styles are iso-surface and direct volume rendering (DVR), while the two primary volume rendering methods are texture slicing and ray-casting. Simple point light shadows are easily added to iso-surface ray-casting by casting an extra ray from the surface point back to the light. Soft shadows, and shadows within DVR ray-casting have proven much more difficult. A number of more advanced lighting techniques have been developed for slice-based DVR using a technique called half-angle slicing [KPHE02]. Half-angle slicing keeps track of rays from both the light's and eye's point of view and advancing them on the same slicing plane. Unfortunately, slice-based renderers are known to suffer from several rendering artifacts and are very difficult to optimize. Conversely, ray-casting is known to produce very high quality images and is much more easily optimized.

To maintain the quality and performance benefits of ray-casting, considerable research effort has been spent on incorporating efficient lighting algorithms into ray-casting engines. Stewart *et al.* pre-compute diffuse ambient lighting which they call *vicinity shading* in a separate volume which is used as a lookup during iso-surface rendering [Ste03]. They accelerated the computation using a 3D version of Bresenham's line drawing algorithm. Wyman *et al.* and Banks *et al.* furthered this technique by pre-computing or lazily

computing global illumination lighting in a separate volume [WPH06] [BB08]. Unfortunately, pre-computed lighting effects can result in significant aliasing artifacts (see figure 2) unless twice the original resolution is used, resulting in an 8X-100X memory footprint depending on the type of pre-computed lighting [WPH06]. Hadwiger *et al.* [HKSB06] pre-compute *deep shadow maps* that represent a compressed attenuation curve from the light's point of view and can represent area light sources. Unfortunately, these maps must be recalculated and compressed each time the light is moved or transfer function is altered which is undesirable for an interactive application.

To address some of the above limitations, approximations have been proposed which are not strictly physically motivated, but lead to visually convincing and plausible results while still being feasible in real-time. For polygonal models, Bunnell *et al.* [Bun05] represent geometry as a hierarchical tree of discs for which simple form factors can be calculated to approximate occlusion. Shanmugam *et al.* [SA07] and Mittring *et al.* [Mit07] even go so far as to compute occlusion from the depth buffer as a post-process. While these approximate methods can be far from physically correct, they add a surprising amount of realism and accurate depth cues to the scene. In volume rendering, Desgranges *et al.* use a summed area table of the volume's opacity to quickly perform variable width blurring operations to approximate ambient occlusion [DP07]. Unfortunately the summed area table must be recomputed whenever the transfer function or isosurface is changed. Ropinski *et al.* compute approximate ambient occlusion by quantizing all the possible combinations of neighboring voxels such that they can apply the transfer function dynamically [RMSD*08]. This method can approximate ambient occlusion as well as color bleeding but suffers from a lengthy compression process and quantization artifacts during rendering.

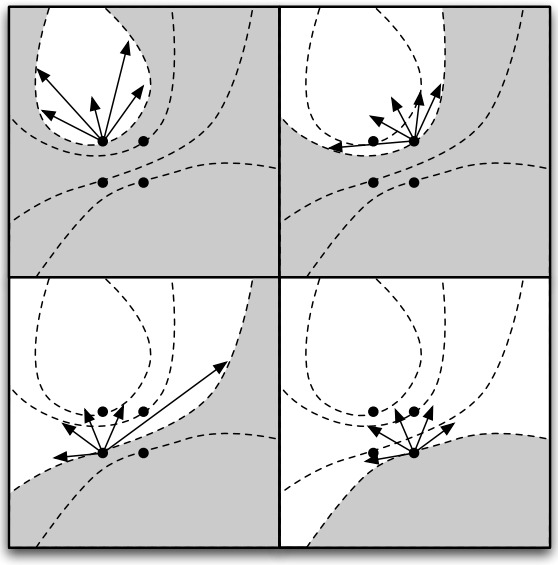

Figure 2: *Illustration of pre-computing volume lighting: Neighboring sampling locations can have highly different occlusion. In order to capture these high frequencies a higher resolution must be used for pre-computed lighting effects.*

3. Algorithm Overview

3.1. Ambient Occlusion

The lighting model used in most applications uses only local reflectance to represent the light leaving a given point p in the direction of the eye. In order to account for complex occlusions from neighboring structures we would like to replace this with a new term that takes into account the irradiance arriving at a surface from all angles. This can be represented as:

$$\int_{\Omega} L(\omega) d\omega \tag{1}$$

where $L(\omega)$ is the radiance arriving from direction ω, and Ω is the set of directions above the surface point (where N is the surface normal and $N \cdot \omega > 0$). Since we are interested in the diffuse reflection of the incoming irradiance, a *cosine-weighted* contribution of incoming radiance is often used:

$$\int_{\Omega} N \cdot L(\omega) d\omega \tag{2}$$

In this case $L(\omega)$ is represented as a vector with magnitude equal to the radiance. This equation can be approximately evaluated using ray casting by discretizing the domain Ω into k sectors of equal solid angle $\Delta\omega$.

$$\sum_{i=0}^{k} N \cdot L(\omega_i) \Delta\omega \tag{3}$$

Since we have chosen to focus on an iso-surface model, computing $L(\omega_i)$ is heavily simplified. If a ray ever enters the iso-surface then it immediately becomes fully occluded. After an occluding iso-surface is found, we can proceed to evaluate the amount of occlusion using an *all-or-nothing* method or a *partial occlusion* method. In the all-or-nothing method $L(\omega_i) = 0$ when ω_i hits an occluding voxel. Otherwise, $L(\omega_i) = 1$. In the partial occlusion method $L(\omega_i)$ uses the unobstructed distance to the first occlusion to determine the amount of occlusion using a basic linear or quadratic fall off function. The partial occlusion method works better in completely occluded spaces or tight spaces where the all-or-nothing method would result in complete occlusion.

3.2. Neighborhood Approximation

Computing the integral mentioned above would require casting many rays. Instead, we would like to make use of the discretized nature of the volume to compute occlusion. The basic assumption we rely on in doing this is that the percentage of occluding voxels surrounding the surface provides a good approximation to the percentage of rays that would be occluded while traversing through the same space. While this is not strictly correct in all cases, it is a commonly used assumption in approximate techniques [SA07] [Mit07] where real-time performance is required.

Under this assumption, we can now define occlusion using the neighborhood around a surface point and define the irradiance at a point p as:

$$\frac{1}{|V|}\int_V L'(v)dv \tag{4}$$

Here, V is a volume above the surface we are evaluating, $L'(x)$ is now simply a binary function of the iso-surface c defined as:

$$L'(x) = \begin{cases} 1 & \text{if } x \leq c \text{ (not occluding)}, \\ 0 & \text{otherwise (occluding)}. \end{cases} \tag{5}$$

By discretizing the volume into k voxels we can express this discretely as:

$$\sum_{i=0}^{k} L'(v_i)\Delta v \tag{6}$$

Whereas the ray-based approximation from equation 3 is dependent on a hemisphere of angles Ω, our approximation in 6 is dependent only on a hemispherical or spherical volume V which lies above a surface point (see figure 3).

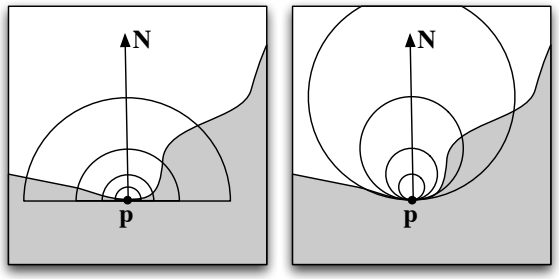

Figure 3: *Occlusion is calculated using the percentage of the a region that is within the iso-surface (grey area). Left: Hemispherical regions. Right: Spherical regions. Different sizes can be combined to better localize occlusions.*

3.3. Filterable Occlusion Maps

While our new definition of occlusion allows us to determine the amount of occlusion using a simple neighborhood around a surface point, computing this directly by testing each voxel in the region would still be prohibitively expensive. We note here that one voxel of a conventional volume data-set can only represent the image value at that one location. If we instead had knowledge of a *distribution* of values in a spherical region surrounding each point, we could do one test against the distribution instead of one test for every voxel. The optimal function for performing this test is the *cumulative distribution function* (CDF). If we think of a region as a distribution of values X, the CDF is defined as:

$$CDF(x) = P(X \leq x) = \sum_{x_i \leq x} P(X = x_i) = \sum_{x_i \leq x} p(x) \tag{7}$$

Given this CDF function we can determine the exact percentage of voxels that are inside/outside any iso-surface in the region. If we look at the properties of the CDF, we can see it is actually equivalent to our new definition of ambient irradiance from equation 6. Unfortunately, while pre-computing CDF functions for each spherical region would allow us to lookup the occlusion with one lookup, the memory required to store all these CDFs uncompressed would be monumental.

Thankfully, by reducing the ambient occlusion problem to storing and evaluating a CDF, we can make use of a lot of research from a slightly different domain. Shadow mapping algorithms [Wil78] are faced with a similar comparison problem when testing an object's depth against a shadow map. In fact a soft shadow technique known as percentage closer filtering [RSC87] performs a brute force depth test against a region in the shadow map which is identical to the test described in equation 6. The main difference is that the shadow map region is in 2D and the volume region is in 3D (see figure 4). A significant amount of recent research has focussed on representing distributions in compact filterable forms.

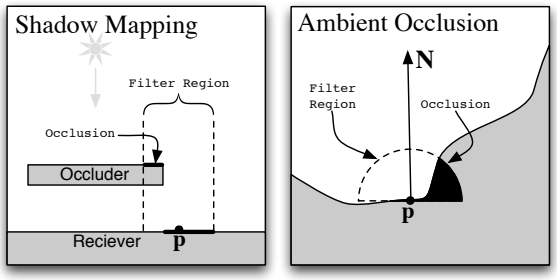

Figure 4: *Comparison of the soft shadow mapping problem to the approximate ambient occlusion problem. Left: In shadow mapping, we want to determine the percentage of non-occluding shadow map texels (greater than or equal to our reciever depth). Right: In our ambient occlusion approximation we wish to determine the percentage of non-occluding voxels (less than the iso-value). Both can be determined using the CDF of the filter region.*

3.4. Variance Occlusion Maps

We chose to use a technique described by Donnelly *et al.* [DL06] to represent a distribution of image values and approximately query the CDF. To very compactly approximate a distribution, they store only the first two moments of the distribution: the image value and the squared image value. The advantage of this representation is that it can approximate the average of several distributions by averaging the moments. This means that the image may simply be blurred to generate a distribution centered at each pixel, where the blurring/filtering kernel represents the image region that will be represented by the distribution. As in [DL06] we can describe these two moments M_1 and M_2 as:

$$M_1 = E(x) \quad = \int_{-\infty}^{\infty} x p(x) dx \tag{8}$$

$$M_2 = E(x^2) = \int_{-\infty}^{\infty} x^2 p(x) dx \tag{9}$$

We can then calculate the mean μ and variance σ^2 of the distribution:

$$\mu = E(x) = M_1 \tag{10}$$

$$\sigma^2 = E(x^2) - E(x)^2 = M_2 - M_1 \tag{11}$$

Since the variance gives us a measure of the width of the distribution, we can place a bound on how much of the distribution can be found a certain distance away from the mean. Chebychev's inequality states this bound precisely as:

$$P(x \leq t) \leq p_{max}(t) \equiv \begin{cases} 1 & \text{if } t \leq \mu\,, \\ \frac{\sigma^2}{\sigma^2 + (t-\mu)^2} & \text{otherwise} \end{cases} \tag{12}$$

Donnelly *et al.* further demonstrate that while this only provides an upper bound on the CDF, it is very accurate in the case of a bi-modal distribution containing only two image values d_1 and d_2. This occurs in shadow maps when there is one occluding object at d_1 and one partially occluded object at d_2. We have found that this is also very often the case in volume data sets where different materials are represented by different iso-values.

In this simple bi-modal case, the inequality becomes an equality and querying equation 12 with d_2 gives the exact percentage of d_2 in the region, which is the same result as the CDF. This represents the percentage of occluding voxels in the region. Of course this becomes less accurate as the distribution takes different forms or the query differs from d_2, but it provides a surprisingly good approximation in many data-sets, especially when only a small carefully chosen region of the volume is considered.

To verify that this approximation is valid, we analyzed a number of data-sets (see figure 5). We found that in medical data-sets, CT values were heavily clustered around different materials such as air, soft tissue, and bone. We found that non-medical data-sets such as the stanford bunny data-set were even more clustered, having almost entirely bi-modal distributions. However, MRI data-sets and distance-field data-sets were much less clustered and thus this technique will be less valid for these types of data.

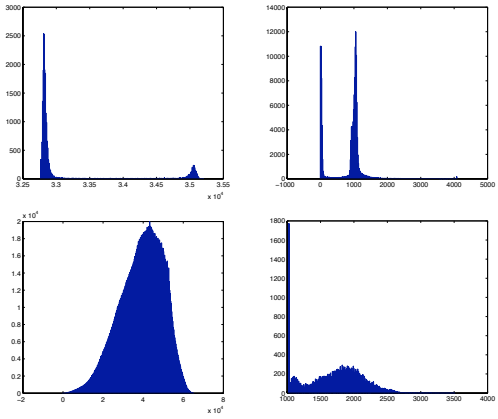

Figure 5: *Histograms of volume data-sets. We found CT data tends to be heavily clustered and works well with our technique. Top-Left: Stanford Bunny CT. Top-Right: Head CT. Bottom-Left: Signed Distance Field. Bottom-Right: Head T1 Weighted MRI*

3.5. Handling Arbitrary Isosurfaces

Unfortunately, even though our data is clustered such that distributions tend to be bimodal, there is no limitation on the isovalue that is chosen by the user. Thus isosurfaces can fall anywhere in between d_1 and d_2. As the isovalue moves towards the mean of the distribution it becomes less accurate. We found this to be a major drawback of this representation but found that it could still be quite useful if the region is chosen carefully (see section 4) .

4. Implementation

4.1. Preprocessing

As a preprocessing step, we create a new volume containing the two moments of the distribution (value and value squared). We then perform separable convolutions with gaussian kernels to generate local distributions centered at each voxel. Storing many of these volumes would take a large amount of memory, so we chose to double the gaussian kernel size at each iteration and down-sample by a factor of two at each iteration. This allows us to store the entire variance ambient map in a mip-mapped two channel floating-point texture. Since we are interested in fairly large filter regions, we also found that we could easily get away with starting at half the resolution of the original volume or even smaller resolutions if only low frequency occlusions are required. At one half resolution the variance ambient map is

roughly 0.6X the size of the original volume including mipmaps if a two channel 32-bit floating point texture is used.

4.2. Blurring Kernel

It is worthwhile to discuss the choice of blurring kernel we used to generate distributions. While we would normally want to use a hemispherical filter region with cosine and distance weighted occluders, this would require a complicated non-seperable blurring process based on the surface normal of each voxel. Since the surface normal can change rapidly, this representation would suffer from aliasing artifacts under linear interpolation of neighboring samples (see figure 6). This becomes especially apparent when lower resolution volumes or mipmaps are used. By using a radially symmetric kernel and offsetting the sampling location from the surface location we can eliminate this aliasing while still allowing high-frequency changes in the surface normal. The one drawback is that the occlusion becomes slightly more concentrated in the direction of the surface normal. Being able to adjust the surface normal during interactive rendering allows us to perform some other interesting effects like averaging the normal and view vector to get view dependent ambient occlusion. Anisotropic spacing can be supported by varying the width of the kernel.

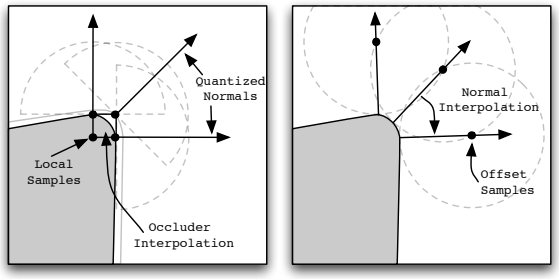

Figure 6: *Illustration of different blurring kernels. While a hemisphere is normally used to search for occluders, it can result in artifacts due to the interpolation of very different occluding regions. By using a spherical kernel and offsetting the sampling location from the surface, the correct normal is used to determine the angle of the occlusion region.*

4.3. Rendering

During interactive rendering, we first search to the isosurface and then perform a binary search to refine the surface location. We then calculate the gradient using central-differencing to perform local lighting. We also use the gradient to calculate offsets for querying the variance ambient map. Since many isosurfaces tend to fall on partial-volumes where one material meets another, we take an extra sample one half of a voxel under the surface in the direction of the gradient (see figure 7) and use this sample as the isovalue to evaluate the amount of occlusion in the variance occlusion map. In the case of ambient occlusion we can evaluate four instances of equation 12 in parallel (one sample from the first 4 mip levels) and then take the maximum occlusion value. In the case of soft shadows, we cast a ray four steps at a time evaluating equation 12 until the ray leaves the volume. To simulate soft shadows from area light sources, the mip level can be increased logarithmically as it travels towards the light.

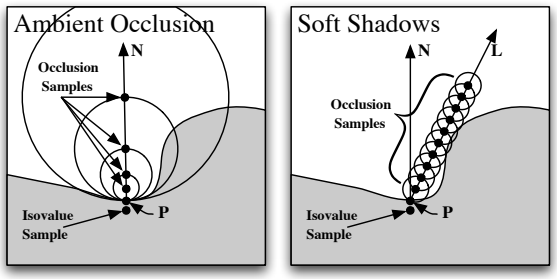

Figure 7: *Illustration of our dynamic ambient occlusion and soft shadow algorithms. The isovalue is chosen one half of a voxel behind the isosurface. Ambient occlusion requires only 3-4 samples of different sizes in the direction of the surface normal. Soft shadows requires one ray cast in the direction of the light. In both cases the occlusion sample with the most occlusion is chosen.*

5. Results

All of the images and timings in this section were generated on a Mac Pro with a single NVIDIA GeForce 8800 GTS. Figure 8 shows 4 offset occlusion samples from separate mip-levels in the occlusion map. Figure 9 shows the combined result with diffuse local lighting. Figure 10 shows soft shadows calculated with one ray combined with ambient occlusion.

5.1. Performance

Preprocessing the volume simply involves blurring and down sampling the volume. We used a 5x5x5 separable gaussian blur to generate variance volumes. This processing can be greatly accelerated on the GPU. Since we are dealing with iso-surfaces, lighting can be calculated as a post process, independent of volume size. The one exception is our soft shadow algorithm which requires an extra ray to be cast until it exits the volume or becomes fully occluded.

6. Conclusion and Discussion

We have presented a novel method for generating approximate isosurface ambient occlusion and soft shadows. Compared with other approaches which may require hours or

Figure 8: *Occlusion sampled from the first 4 levels of the occlusion map. These are combined by taking the maximum occlusion (minimum irradiance) or average occlusion.*

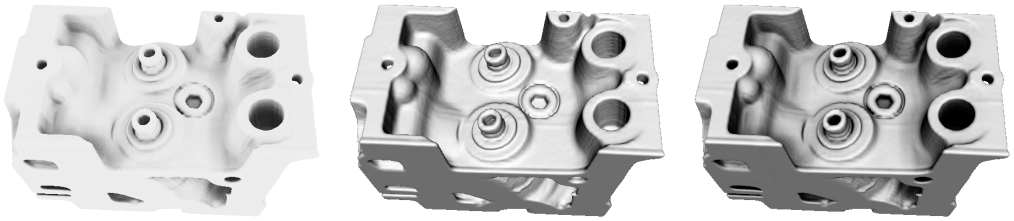

Figure 9: *Left: Combined occlusion from figure 8 using maximum occlusion. Center: Simple diffuse lighting. Right: Diffuse combined with ambient occlusion.*

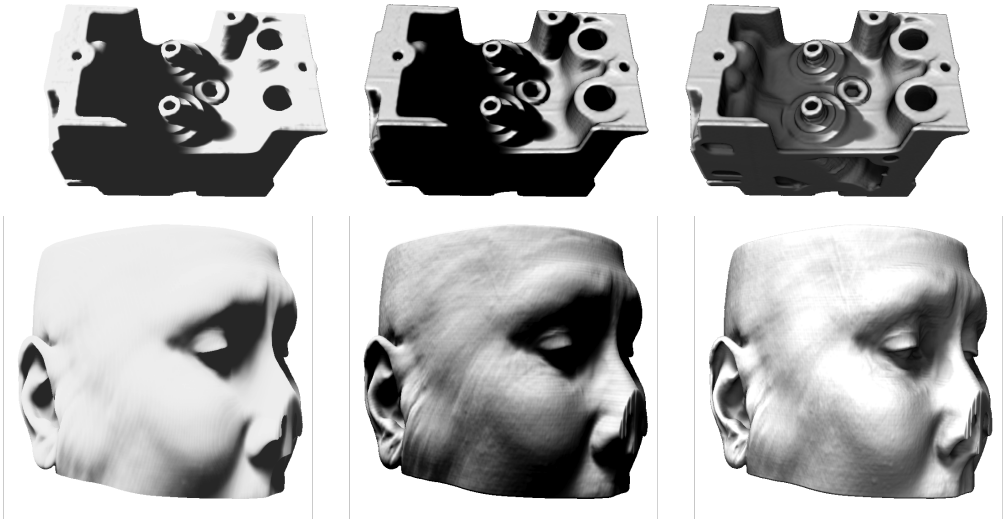

Figure 10: *Left: Approximate soft shadows using only one ray. Center: Diffuse lighting with shadows. Right: Diffuse lighting, ambient occlusion and soft shadows.*

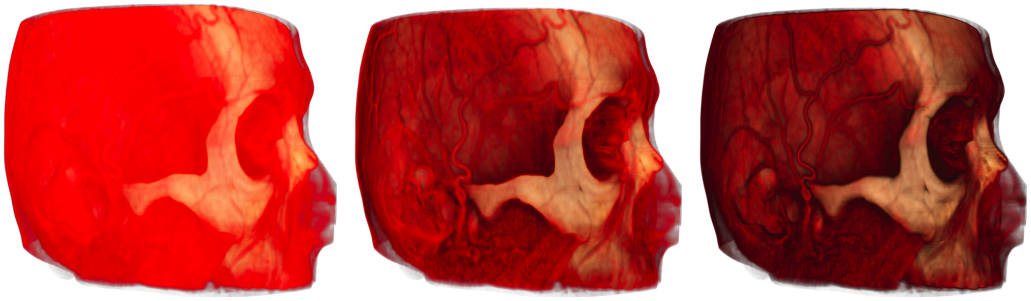

Figure 11: *Left: DVR. Center: DVR with AO. Right: DVR with AO and local phong lighting. If transparency is intended to provide detail-in-context rather than specify transparency to incoming light, then our occlusion method is still useful within DVR. We also found it useful to adjust the shadow contribution based on transparency.*

Preprocessing Time (in milliseconds)		
Dataset	CPU	GPU
Engine (256^3)	170	8.6
Small Head (512^2x256)	678	34.5
Big Head (512^3)	1524	74.1

Table 1: *Preprocessing time on the CPU and GPU*

Rendering Time (in milliseconds)				
Dataset	Phong	AO	HS	SS
Engine (256^3)	7.1	7.4	19.1	25.2
CTHead1 (512^2x256)	7.8	8.2	26.4	30.3
CTHead2 (512^3)	8.0	8.3	32.5	37.0

Table 2: *Rendering time using phong, ambient occlusion(AO), hard shadows(HS) and soft shadows(SS)*

days of pre-computation time and suffer from aliasing and/or quantization artifacts, our approach can load a data set within seconds, has very good performance and doesn't suffer from the aliasing of pre-computed lighting techniques.

We found the most serious limitation of our work to be the statistical approximation provided by the first two moments. We found that adjusting the offset of the spherical distribution produced very interesting effects similar to subsurface scattering, but we were often unable to do this because the variance test would begin to behave incorrectly as the mean of the local distribution approached the isovalue.

In conclusion, we believe that filterable occlusion maps can provide a very quick way to approximate complex lighting effects, but if it is to be used in practice a better CDF approximation will be needed. We are currently investigating Convolution Shadow Maps since they represent the entire CDF and aren't restricted to bi-modal distributions.

References

[BB08] BANKS D. C., BEASON K.: Fast global illumination for visualizing isosurfaces with a 3d illumination grid. *Computing in Science and Engg. 9*, 1 (2008).

[Bli77] BLINN J. F.: Models of light reflection for computer synthesized pictures. In *SIGGRAPH '77* (New York, NY, USA, 1977), ACM, pp. 192–198.

[Bun05] BUNNELL M.: *GPU Gems 2*. Addison-Wesley, 2005, ch. Dynamic Ambient Occlusion and Indirect Lighting.

[CCF94] CABRAL B., CAM N., FORAN J.: Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *VVS '94.* (New York, NY, USA, 1994), ACM, pp. 91–98.

[Cro77] CROW F. C.: Shadow algorithms for computer graphics. *SIGGRAPH Comput. Graph. 11*, 2 (1977).

[DL06] DONNELLY W., LAURITZEN A.: Variance shadow maps. In *I3D '06* (New York, NY, USA, 2006), ACM, pp. 161–165.

[DP07] DESGRANGES P. E. K.: Us patent application 2007/0013696 a1: Fast ambient occlusion for direct volume rendering, 2007.

[HKSB06] HADWIGER M., KRATZ A., SIGG C., BÜHLER K.: Gpu-accelerated deep shadow maps for direct volume rendering. In *GH '06* (New York, NY, USA, 2006), ACM, pp. 49–52.

[HLHS03] HASENFRATZ J.-M., LAPIERRE M., HOLZSCHUCH N., SILLION F.: A survey of real-time soft shadows algorithms. *Computer Graphics Forum 22*, 4 (dec 2003), 753–774.

[HSBW99] HUBONA G., SHIRAH G., BRANDT M., WHEELER P.: The role of object shadows in promoting 3-d visualization, 1999.

[KMK94] KERSTEN D., MAMASSIAN P., KNILL D.: *Moving Cast Shadows and the Perception of Relative Depth*. Tech. Rep. 6, Max Planck Institute for Biological Cybernetics, T§bingen, Germany, jun 1994.

[KPHE02] KNISS J., PREMOZE S., HANSEN C., EBERT D.: Interactive translucent volume rendering and procedural modeling. In *IEEE Vis* (2002), pp. 109–116.

[Lev90] LEVOY M.: Efficient ray tracing of volume data. *ACM Trans. Graph. 9*, 3 (1990), 245–261.

[Max95] MAX N.: Optical models for direct volume rendering. *IEEE Vis. 1*, 2 (1995), 99–108.

[Mit07] MITTRING M.: Finding next gen: Cryengine 2. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses* (New York, NY, USA, 2007), ACM, pp. 97–121.

[RMSD*08] ROPINSKI T., MEYER-SPRADOW J., DIEPENBROCK S., MENSMANN J., HINRICHS K. H.: Interactive volume rendering with dynamic ambient occlusion and color bleeding. *Eurographics 27*, 2 (2008).

[RSC87] REEVES W. T., SALESIN D. H., COOK R. L.: Rendering antialiased shadows with depth maps. In *SIGGRAPH '87* (New York, NY, USA, 1987), ACM.

[SA07] SHANMUGAM P., ARIKAN O.: Hardware accelerated ambient occlusion techniques on gpus. In *I3D* (New York, NY, USA, 2007), ACM, pp. 73–80.

[Ste03] STEWART A. J.: Vicinity shading for enhanced perception of volumetric data. In *IEEE Vis* (10 2003).

[Wan92] WANGER L.: The effect of shadow quality on the perception of spatial relationships in computer generated imagery. In *SI3D '92* (New York, NY, USA, 1992), ACM.

[Wil78] WILLIAMS L.: Casting curved shadows on curved surfaces. *SIGGRAPH 12*, 3 (1978), 270–274.

[WPH06] WYMAN M.-C., PARKER M.-S., HANSEN S. M.-C.: Interactive display of isosurfaces with global illumination. *IEEE Vis 12*, 2 (2006), 186–196.