Obscurances in general environments

Àlex Méndez-Feliu, Mateu Sbert
Institut d'Informàtica i Aplicacions
Universitat de Girona
{amendez,mateu}@ima.udg.es

Abstract

Obscurances is a powerful, simple and robust technique to simulate global illumination of a scene with much lower cost than other global illumination techniques like radiosity or path tracing. On the one hand, this technique has been used to compute the indirect lighting of diffuse scenes in videogames using lightmaps. In this context, it has been even proven useful to recompute in real time the illumination of the polygons around a moving object. On the other hand, it has been used in a ray-tracing environment, allowing to produce highly realistic images that include beautiful effects like the ones given by glossy and specular materials. Despite the introduction of these effects, the computation of the obscurances itself only takes account of the diffuse reflective illumination. In this paper we study how the obscurances should behave when they are computed in an environment with non-diffuse materials, like specular or refractive ones.

Keywords: obscurances, ray-tracing, global illumination

1 Introduction

Obscurances [Zhukov et al. 1998],[Iones et al. 2003] take into account the occlusions around the point or patch of which we want to compute the indirect illumination. To do this, only a local environment is queried and the amount of occlusion determinates the diffuse effects leading to a darker effect (less light intensity) the more occluded is the point or patch. This technique, introduced in videogames as a cheap alternative to radiosity, has been improved with color bleeding effects and allows a real-time update of the illumination when moving objects are in the scene [Méndez-Feliu et al. 2003]. A simplified version of obscurances is used in well known commercial renderers such as *Pixar*'s *Photorrealistic RenderMan* under the name of *ambient occlusion* [Christensen 2003].

In [Méndez-Feliu and Sbert 2004b] and [Méndez-Feliu and Sbert 2004a] the obscurances technique has been used in ray-tracing environments. Although it is not a global illumination technique, it is able to obtain at a low cost high quality realistic looking images that simulate a global illumination solution. Direct illumination and glossy effects are computed in the usual way [Cook et al. 1984], and diffuse effects are computed using the obscurances technique. The overall effect is a nice looking realistic image. In this context, in [Méndez-Feliu and Sbert 2004b] a comparison of the efficiency of different sampling techniques is made. In [Méndez-Feliu and Sbert 2004a], thanks to the decoupling of direct and indirect lighting, the obscurances are used to compute only once the indirect illumination when computing a series of frames with light animation.

Although obscurances have been used in general environments, for the actual computation of the technique, only diffuse reflective materials have been used. In this paper we propose to extend the original algorithm that computes the obscurances with color bleeding in a way that it can cope with other kind of materials and the interactions between them, such as perfect specular surfaces, refractive and translucent objects.

This paper is organized as follows. In the next section we explain the obscurances for diffuse materials, as commonly used. Then we introduce how the obscurances computation should behave if the objects around are perfectly specular (subsection 3.1), perfectly transparent (subsection 3.2), or translucent (subsection 3.3). In section 4, the special case of obscurances of a translucent material and its simple application to plants and trees are discussed. In section 5, a solution for the rest of BRDF's is pointed. Finally, some conclusions and future work are given.

2 Previous work: Obscurances in diffuse environments

2.1 Obscurances with color bleeding

In [Zhukov et al. 1998],[Iones et al. 2003] the obscurances illumination model was defined and in [Méndez-Feliu et al. 2003] color bleeding was added by slightly modifying the original equations. Obscurances take account of secondary diffuse illumination, being totally decoupled from direct illumination. Indirect illumination for point P is defined as:

$$I(P) = \frac{1}{\pi} R(P) I_A \int_{\omega\in\Omega} R(Q)\rho(d(P,\omega))\cos\theta d\omega \qquad (1)$$

where

- $\rho(d(P,\omega))$: function with values between 0 and 1, and giving the magnitude of ambient light incoming from direction ω
- $d(P,\omega)$: distance from P to the first intersected point in direction ω
- θ: angle between direction ω and the normal at P
- I_A: ambient light intensity
- $R(P)$: Reflectivity at P
- $R(Q)$: Reflectivity at point Q seen from P at direction ω or average reflectivity of the scene if no point is seen at a lesser distance than d_{max}
- $1/\pi$ is the normalization factor such that if $\rho() = 1$ over the whole hemisphere Ω then $I(P)$ is $R \times I_A$

Direct illumination is added to (1) to obtain the final illumination at point P. Function $\rho()$ increases with d. Its shape is given in figure 1. A maximum distance for interaction, d_{max} is defined, so that when $d \geq d_{max}$ then $\rho(d) = 1$. This means that we only take into account a d_{max}-neighborhood of P. Obscurance of P is then defined as:

$$W(P) = \frac{1}{\pi} \int_{\omega\in\Omega} R(Q)\rho(d(P,\omega))\cos\theta d\omega \qquad (2)$$

Ambient light in (1) is computed with the formula:

$$I_A = \frac{1}{1 - R_{ave}} \frac{\sum_{i=1}^{n} A_i E_i}{A_T} \quad (3)$$

where

$$R_{ave} = \frac{\sum_{i=1}^{n} A_i R_i}{A_T} \quad (4)$$

and A_i, E_i, R_i are the area, emissivity and reflectivity of patch i, A_T the sum of the areas, and n the number of patches in the scene.

Figure 2 presents an image of a vase made of a white diffuse material. The indirect light is decoupled from direct light and is shown in figure 3. The image of the obscurances values is shown in figure 4. Direct illumination is computed by selecting random points on the light surface and testing visibility. The specular effects are computed by following the corresponding reflected or refracted rays.

All images presented in this paper are made of 800×600 pixels and are computed on a Pentium 4 running at 1.6 Ghz with a memory of 2Gb. For each pixel, 40 obscurance rays are sent.

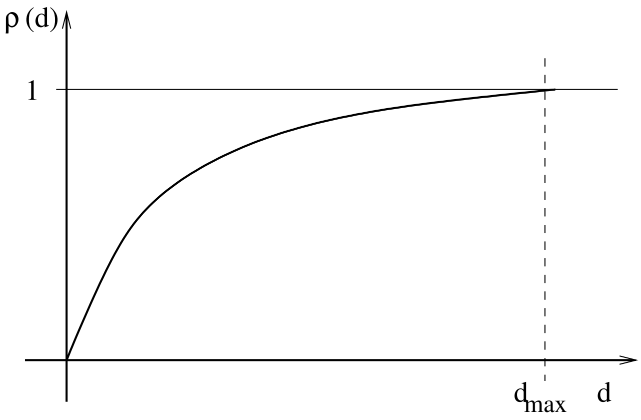

Figure 1: Shape of $\rho(d)$ function.

Figure 2: Model of a vase made of white diffuse lambertian material with indirect illumination computed using obscurances technique. It's a 800×600 px image and it takes 1563 seconds to compute.

Figure 3: Indirect light is computed by multiplying the diffuse color, the ambient intensity and the obscurances. Adding direct light and specular effects to this image, we get image in figure 2

Figure 4: Map of the obscurance values for the diffuse vase image.

3 Generalization to other materials

All these formulae work well for diffuse environments, this is, when the object where the point P is located and the objects around it are all made of diffuse reflective materials. But in general environment, materials can have many different reflective and refractive behaviours. These behaviours are modelled with different BRDF's and BTDF's. Therefore we are going to study how the obscurances computation should be modified according to the basic cases for these BDF's different than diffuse: perfect specular material, transparent material, and translucent material.

We can consider that a point P from a perfect specular object has no "own" color, as it takes the color of the object seen from the reflected direction on the surface. For the same reason, it has no "own" obscurance value. We take the color and obscurance value from the first diffuse object found by the recursive reflections of the viewing ray. When we think of perfect refractive (transparent) materials, we follow the same reasoning, but now considering the refracted ray directions.

If point P is on a translucent object, the indirect light comes from the inside of the object and it should be a function of the *thickness* of the object. In this case, the classic obscurance method does not make sense. We propose a similar method, that we can call *inner obscurances*, and it is presented in section 4.

But even when we consider point P to be on a surface from a lambertian reflective object, the interaction of the obscurance rays with the rest of the non-diffuse materials have to be taken into account. In the next subsections 3.1, 3.2 and 3.3 we study the changes on the obscurances computation when the objects around the diffuse point P have other material properties.

From (1) we have defined Q as the point seen from P in direction ω. If no object is found in direction ω or it is further than d_{max}, we consider $R(Q)$ to be the average reflectivity color (R_{ave}) and $\rho() = 1$. If the object is found at a distance less than d_{max}, and if it has a lambertian material we take its reflective color as $R(Q)$ and we compute ρ according to the shape in figure 1 (we will usually take $\rho(d) = \sqrt{d/d_{max}}$).

But what happens if the object, where Q is located on, is not lambertian?

3.1 What if the object is perfectly specular?

Technically, a pure specular material has no "own" color. The color of a specular object in a certain point depends mainly on the "viewing" direction, taking the color of another object seen at a direction given by the law of reflection. In figure 5 an image of a perfect specular vase is shown.

In this way we can clearly see what value $R(Q)$ should have taken in equation (1) when Q is located on a specular object. As from the definition, $R(Q)$ is the color seen from P in direction ω, we will take the color of the first diffuse object hit by the corresponding reflected direction ω'. Figure 6 shows this situation.

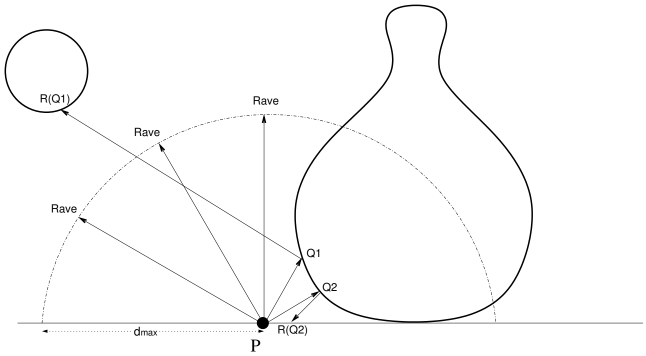

Figure 6: Schema of the reflected obscurance rays. Obscurances of the diffuse point P are computed by sending cosinus distributed rays that query the space around P. If some ray finds a specular surface, it follows the corresponding reflected path.

Figure 5: Vase made of perfect specular material with obscurances computed as in figure 7b. It's a 800×600 px image and it takes 2108 seconds to compute.

We have to discuss also the use of the ρ function. The main obscurances assumption is that the more occluded is a point, the darker it will appear, this is why we apply the ρ function to the *distance* parameter. But when the point is enclosed by specular objects we have to think different, because indirect lighting can come reflected from the rest of the scene and, in consequence, the point considered would not appear so dark as if it were enclosed by diffuse objects.

We have considered three possible solutions. In the first solution, use ρ the same way as with diffuse lambertian materials, i. e., taking into account just the distance from P to Q, and as $R(Q)$, the color of the first hit diffuse material following the corresponding reflected ray. The schema of this solution, the image of obscurances and final image are shown in the first column of figure 7. But we can see that this first solution does not fulfill our expectations of less darkness around the base of the vase. That is because we use the same *distance* parameter as if we were considering a diffuse vase. In a second solution, we can consider the *distance* parameter as the sum of the distances of all the corresponding reflected rays needed to find a diffuse surface. On the one hand we obtain less dark obscurances (and this is what we are looking for). On the other hand, as a collateral effect, we gain computation time, as we can stop sending rays when the sum of the distances overcomes d_{max}. Schema and results are shown in column b of figure 7. In any case, we have to cast additional rays when we find a specular object, and this increases the computational cost. This is why we suggest a third option that comes with no additional cost, though visual results (figure 7, third column) are not so pleasant. In this last case, we use directly $R(Q) = R_{ave}$ and $\rho = 1$.

Comparing visually these three solutions, the one that gives best results is the second one, but if time is an important issue the third one is faster (1693 seconds in the example of figure 7) compared to the two former ones (2179 and 2108 seconds, respectively).

3.2 What if the object is perfectly transparent?

The problem here is similar to the specular case, but with refracted directions instead of reflected ones. Light rays change their directions when traversing through different dielectric materials following the Snell's law of refraction. This is well studied, known and programed in early versions of ray-tracing (see [Cook et al. 1984]). Figure 8 shows an image of a transparent vase with an index of refraction of 1.5 with respect to the air. In figure 9 this situation is shown.

We have similar problems and therefore we propose similar solutions as in previous subsection. This is, an obscurance ray cast from a point of a diffuse object near a transparent object, takes $R(Q)$ as the color of the first hit point of a diffuse material following the refracted (and/or reflected, if it is the case) rays. The *distance* parameter in $\rho(d)$ can be taken into account also in the same way as the three different solutions previously seen in subsection 3.1.

3.3 What if the object is translucent?

A completely different problem is presented here, and depending on the kind of translucency we are considering we can have many different subproblems and complexities. The main feature of a translu-

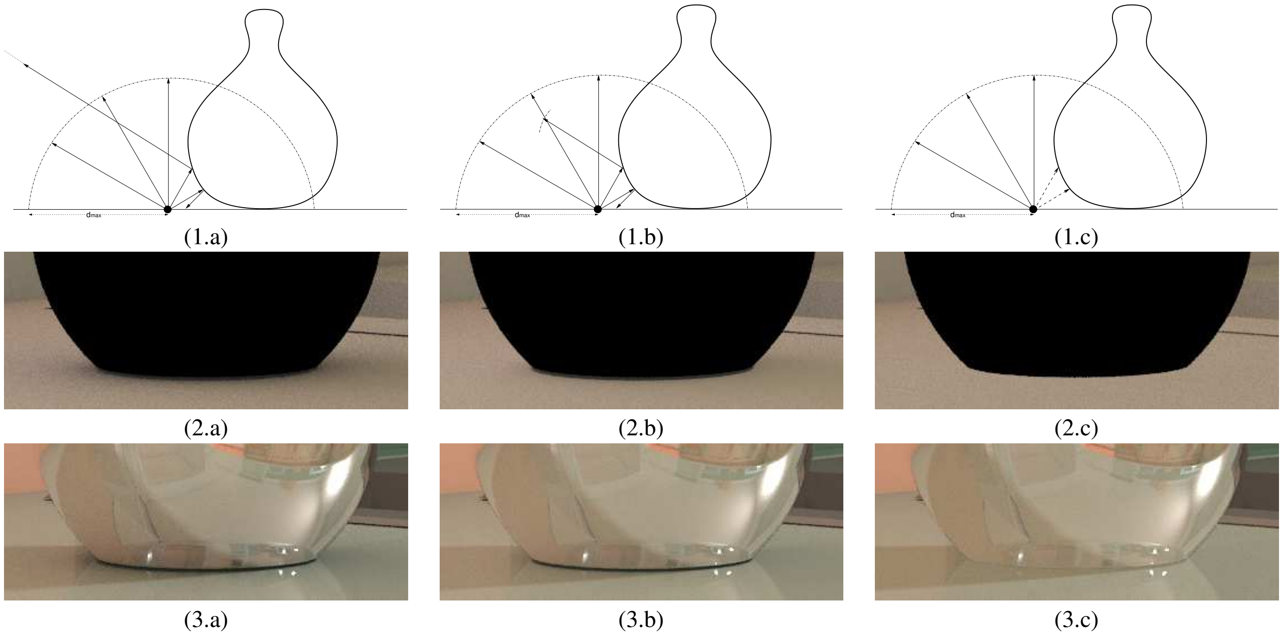

Figure 7: The base of the vase is here detailed with three different computations of the obscurances when the surrounding object is perfectly specular. The first row shows the schema of the way obscurances are computed, the second row presents only the obscurances and the third row presents final images. The complete image ($800 \times 600px$) of the first column takes 2179 seconds to compute. For the image in the b column it takes 2108 secs. The third image takes 1693 secs.

Figure 8: Vase made of transparent material like crystal. It's a 800×600 px image and it takes 2403 seconds to compute.

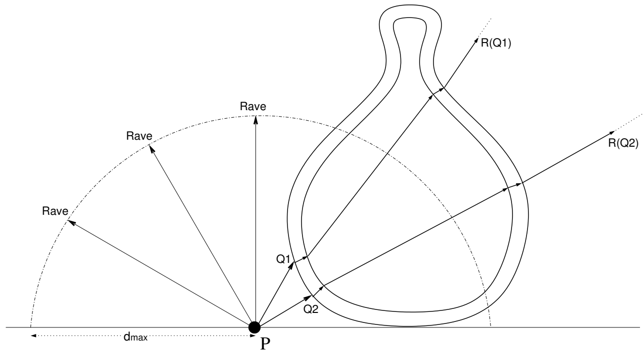

Figure 9: Schema of the refracted obscurance rays. Obscurances of P are computed by sending cosinus distributed rays that query the space around P. If some ray finds a transparent surface, it follows the corresponding refracted path.

cent object is that light gets into it and many things can happen to a photon once inside the object, including scattering, backscattering, attenuation or simply traversing the object (see [Siegel and Howell 1992]). For simplicity we will only consider attenuation and distribution of photons as they get into the object.

Taking all this into account, what should be the color and intensity seen from P at point Q onto a translucent object? This is difficult to answer, specially if we want to adapt it to the obscurances philosophy and preserve the decoupling of direct and indirect lighting, because the light coming from inside the object comes mainly from the direct light pointing to it from the other side. Certainly it does not depend on the reflectivity color $R(Q)$ (remember we are talking of pure translucent objects) and it does not depend on any function of the *distance*. One of the parameters we could think of is an *average inner indirect lighting* for every translucent object, computed using its geometrical properties and its attenuation parameters (possibly an attenuation color and a maximum distance). The computation of these parameters are out of our focus by now and remains as future work.

A provisional solution is to use *black*, R_{ave} or any value between them as $R(Q)$. As no function of the distance is needed, we take $\rho = 1$.

4 Obscurances of a translucent material

As we have said before, we consider here a simplified problem of translucent materials considering only the light absorption of the material, not the diffuse scattering or back scattering. Of course the original concept of the obscurances does not apply but we can use the same concept of space querying (inside the object, in this case) and maximum distance to compute a similar concept, that we can dub *inner obscurances*, and are shown in figure 11. A similar idea but with a different solution is the *vicinity shading* used by Stewart [Stewart 2003].

Figure 10: Vase made of translucent material like wax. It's a 800 × 600 px image and it takes 1433 seconds to compute.

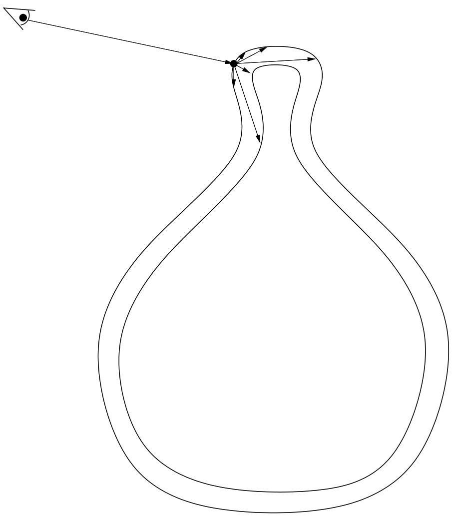

Figure 11: Schema of the computation of the *inner obscurance value*. From the inner part of the objects its *thickness* is queried to obtain the inner obscurances value.

We can define the *inner obscurance* as:

$$I(P) = \frac{1}{\pi} a(P) R_{ave} I_A \int_{\omega \in \Omega} \gamma(d(P,\omega)) \cos\theta \, d\omega \quad (5)$$

where

- $\gamma(d(P,\omega))$: function with values between 0 and 1, and giving the magnitude of ambient light incoming from direction ω. It is a special function useful for translucent objects and is equivalent to $1-\rho$.
- $d(P,\omega)$: distance from P to the boundary in direction ω, from the inside of the object.
- θ: angle between direction ω and the normal at P
- I_A: ambient light intensity. It is multiplied by R_{ave} because no color bleeding is used inside the object.
- $a(P)$: Absorption coefficient.
- $1/\pi$ is the normalization factor such that if $\gamma() = 1$ over the whole hemisphere Ω then $I(P)$ is $a \times R_{ave} \times I_A$

Here a different maximum distance d_{max} is defined for each different translucent material, giving the idea of the distance at which the light energy is completely absorbed by the material. The function γ can be defined as $\gamma(d) = 1 - \sqrt{d/d_{max}}$.

Figure 10 shows our vase as made of a perfect translucent material. Though it is directly illuminated, this kind of material does not reflect the direct light in any way, as it is suposed to fully *get into* the object and scatter. We only see our aproximation of the indirect light getting out of the object as if direct light had come to it from all directions. The slimmer parts of the object appear more intensely illuminated than the thicker parts.

One specific case of translucency can be found in complex objects like trees and plants. Normally the leaves are so slim that they are represented with a single polygon with texture (or sometimes we represent many leaves in a polygon, forming *clusters*). We modified our algorithm to deal with simple objects made with single textured polygons, and applied it to trees and plants. In this case we don't take into account the inner and outer spaces of an object, and in consequence the *inner obscurances* do not make sense. We simply compute the obscurances of the backface of a leave and multiply them by the translucency factor. We finally add the back and front obscurances, and the final result is a slightly more clear tree. Figure 12 shows our result with a lemon tree.

Figure 12: This method can be used with good results with models of plants and trees. It's a 800 × 600 px image and it takes 894 seconds to compute. We have to note that it is a simpler model than the kitchen one.

(a) (b)

Figure 13: Image of the vase with a combination of basic BRDF's and BTDF's. Left: using a combination of the different strategies to compute obscurances (7723 seconds). Right: a global illumination path tracing image of the same vase (19431 seconds).

5 What about the other BRDF's and BTDF's?

So far, we have only explained the basic cases. But we can consider any material as partly diffuse, partly specular, partly transparent and partly translucent. Though reality is much more complex than these simple cases or even a linear combination of these, a wide range of materials can be modelled with this method.

We consider in our examples every material as made of a percentage of each four basic materials. Thus, a maximum of four different obscurances computations per hit have to be done using the appropiate algorithm for each case. The problem comes with the specular and transparent cases, that require sending additional rays to follow the paths of the viewing rays, and possibly (if a material has both specular and transparent properties) casting two paths per hit, leading to a quadratic increase of the number of rays sent. To reduce this problem we can choose only one of the two possible ways at each impact hit, using importance sampling. Also, a russian roulette technique can be used to stop recursion.

Figure 13a shows an image of a vase made of a material that combines all BRDF's and BTDF's seen in this paper. Image in figure 13b is a path-tracing (global illumination) version of the same model. Note it presents much more noise.

6 Conclusions and future work

We have introduced in this paper how the obscurances technique can be extended to deal with general environments. We have first presented the original obscurances idea, aimed to simulate the indirect global illumination for diffuse environments. Then the different algorithms and techniques for different basic BRDF's and BTDF's have been introduced. Finally one possible way to combine these basic material properties has been shown.

As future work we plan to extend the obscurances with more general material properties, such as glossy, anisotropic, fluorescent or phosphorescent materials. Obscurances can be applied also to more specific translucent materials as skin, hair, or to participative media as clouds, fog and submarine rendering. Its application to leaves, trees and plants can also be studied more deeply.

Acknowledgements

This project has been funded in part with Gametools project from the European VIth Framework and with grant number TIN2004-07451-C03-01 from the Spanish Government.

References

CHRISTENSEN, P. H., 2003. Global illumination and all that. SIGGRAPH 2003 course notes #9, RenderMan: Theory and Practice, July.

COOK, R. L., PORTER, T., AND CARPENTER, L. 1984. Distributed ray tracing. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, ACM Press, 137–145.

IONES, A., KRUPKIN, A., SBERT, M., AND ZHUKOV, S. 2003. Fast, realistic lighting for video games. *IEEE Computer Graphics & Applications 23*, 3 (may-june), 54–64.

MÉNDEZ-FELIU, A., AND SBERT, M. 2004. Combining light animation with obscurances for glossy environments. *Computer Animation and Virtual Worlds 15*, 3-4 (july), 463–470.

MÉNDEZ-FELIU, A., AND SBERT, M. 2004. Comparing hemisphere sampling techniques for obscurance computation. In *3IA '04: Proceedings of the International Conference on Computer Graphics and Artificial Intelligence*.

MÉNDEZ-FELIU, A., SBERT, M., AND CAT, J. 2003. Real-time obscurances with color bleeding. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics*, ACM Press, 171–176.

SIEGEL, R., AND HOWELL, J. R. 1992. *Thermal Radiation Heat Transfer, 3rd Edition*. Hemisphere Publishing Corporation.

STEWART, A. J. 2003. Vicinity shading for enhanced perception of volumetric data. In *IEEE Visualization*.

ZHUKOV, S., IONES, A., AND KRONIN, G. 1998. An ambient light illumination model. In *Rendering Techniques*, Springer, G. Drettakis and N. L. Max, Eds., 45–56.