Obscurances for ray-tracing

À. Méndez,[1] M. Sbert[1] and L. Neumann[2]

[1] amendez@ima.udg.es, mateu@ima.udg.es, Institut d'Informàtica i Aplicacions, Universitat de Girona
[2] lneumann@ima.udg.es, ICREA

Abstract

We present a powerful method to create realistic-looking pictures of scenes with objects that have diffuse and non-diffuse properties. The method recreates the obscurances technique, introduced some years ago, with a new approach based on ray-tracing. The first version of the obscurances technique was used only for diffuse environments. It is already working successfully in some widely known video games because it is able to avoid the high cost of radiosity techniques in the interactive and real-time game environments. Soft shadows, nice colour reflection effects, visually pleasant rendering of corners and other partly occluded surfaces of the scene are reproduced at a small fraction of the cost of radiosity. We extend here obscurances to handle non-diffuse environments via ray-tracing. Instead of computing the obscurance of every patch in the scene, we will compute view-dependent obscurances. The direct illumination is separately handled and diffuse color albedo functions are used for obscurance computation. Ray-traced obscurances can be useful in animation, both in the editing phase and/or in final images, and in ray-traced games, when the use of future graphics cards will decrease dramatically the cost of tracing a ray.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and realism, Raytracing

1. Introduction

Ambient term (see [3],[11]) has been used for a long time as a cheap way to complete local illumination techniques. It was meant to give a faint approximation for all interreflections of second and higher order. In [1] this simple technique was enhanced in the Extended Ambient Term approach to obtain all possible information without occlusion computation, using backface and quasi-shadow effects. Obscurances [15, 6] can be classified as a third step forward Occlusions are now taken into account, but only a local environment is queried. The amount of occlusion will determinate the diffuse effects. 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 [10].

Global illumination effects are very costly to simulate. Sophisticated techniques like bidirectional path tracing [8],[13], Metropolis lighting [14], photon map [7] have allowed to reduce significantly the cost of obtaining a noiseless image but they are still computationally very expensive.

The technique presented here, although it is a not a global illumination technique, 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, and diffuse effects are computed using the obscurances techniques. The overall effect is a nice looking realistic image.

This paper is organized in the following way. Previous work is explained in the next section. In section 3 we introduce the new algorithm and present the results, and finally in the last section we present our conclusions and give some directions for future work.

2. Previous Work

2.1. Obscurances

In [15, 6] the obscurances illumination model was defined. Obscurances take account of secondary (diffuse) illumination, being totally decoupled from direct illumination. Indirect illumination for point P is defined as:

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

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 of 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
- $1/\pi$ 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 the point. 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}-neighbourhood of P. Obscurance of P is then defined as:

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

Clearly $0 \leq W(P) \leq 1$. Obscurance for a patch is defined as the average of obscurances for all points in it. An obscurance value of 1 means that the patch is totally open (not occluded with neighbour polygons), while a value of 0 means that it is totally closed (or occluded with neighbour polygons). A simplified version of this technique, with constant function $\rho = 0$, when $d < d_{max}$, is used under the name of ambient occlusion in several commercial renderers, such as Photorealistic Renderman [9, 2].

Ambient light in 1 is computed with the formula:

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

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. The ambient term considered here corresponds to the secondary illumination only, as direct illumination is computed apart.

2.2. Real-time Obscurances with color bleeding

Computation of obscurances is very fast as it has only to consider a neigboorhood of a point (or a patch). In [10] a Monte Carlo implementation with real-time update of moving objects was presented. Also in [10] color bleeding was incorporated. To do it we have to modify the ambient light computation by taking out the average reflectivity factor in (3):

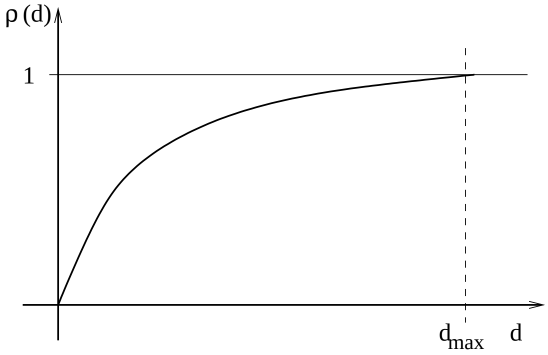


Figure 1: *Shape of* $\rho(d)$ *function.*

$$I_A = \frac{1}{1 - R_{ave}} \frac{\sum_{i=1}^{n} A_i E_i}{A_T} \quad (5)$$

and modify accordingly the obscurances (2):

$$W(P) = \frac{1}{\pi} \int_{\omega \in \Omega} R(Q) \rho(d(P,\omega)) \cos\theta d\omega \quad (6)$$

where $R(Q)$ is the reflectivity of the point Q seen from P in direction ω. When no surface is hit at a distance less than d_{max} in direction ω the obscurance takes the value of R_{ave}.

Obscurances with color bleeding have been implemented within RenderPark software[5] and within Crystal Space 3D engine [4].

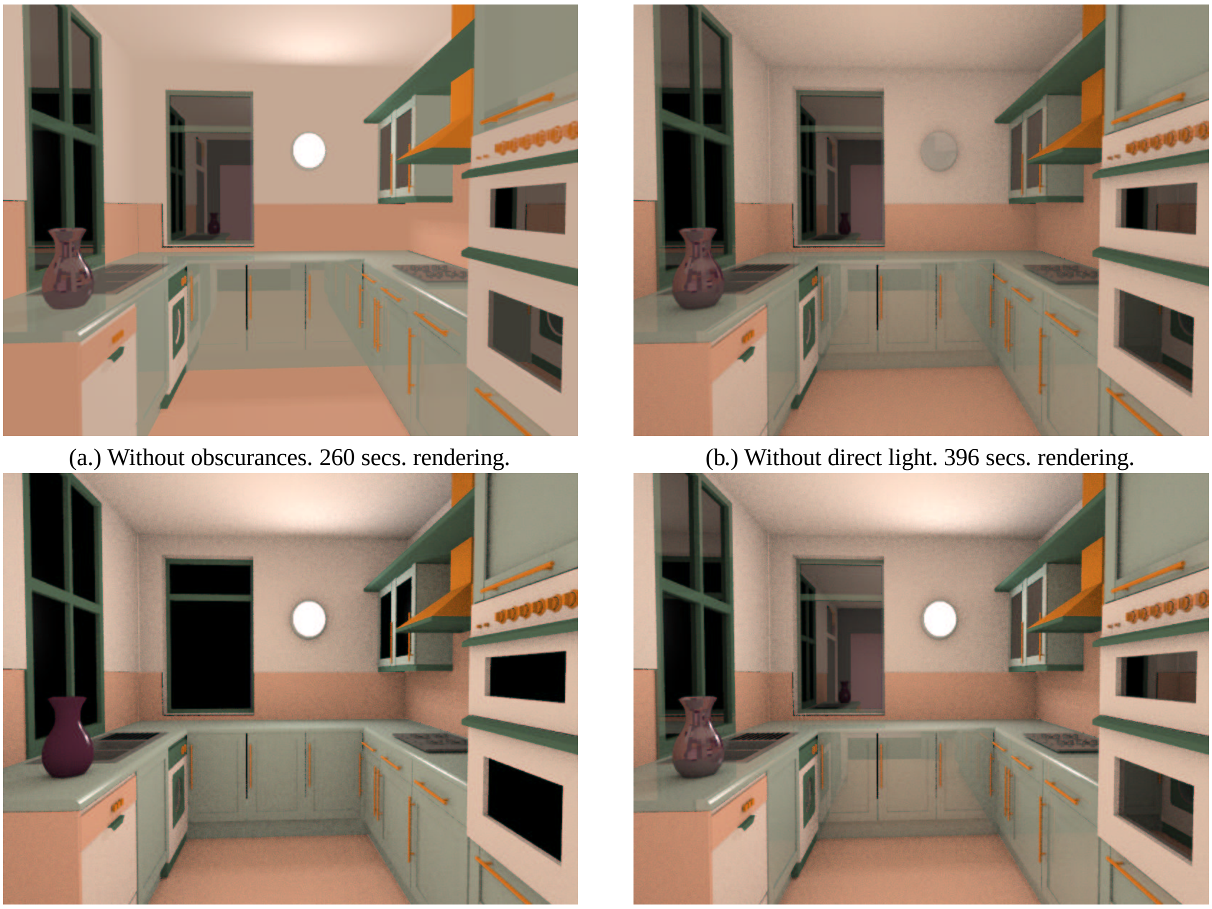
3. Extension to glossy environments: Ray-traced obscurances

The extension of the technique to deal with general environments is done in the context of ray-tracing.

3.1. Algorithm

A ray is traced in the usual way from the view point through a pixel, hitting a point P of some surface in the scene. Then from P, N rays are stochastically traced with a $\cos\theta$ distribution to compute the obscurance of point P, this is, we solve by Monte Carlo the integral in (6). When the hit point corresponds to a glossy or specular surface a path is followed according to usual path-tracing. Direct illumination rays are also traced to the light from the hit points. The final illumination is then given by adding the direct lighting, specular effect and diffuse effect represented by the obscurances multiplied by ambient light (5).

In Fig.2 the separated effects treated by our algorithm are shown. In Fig.2a the obscurances are missing, in Fig.2b direct illumination is missing, Fig.2c lacks the specular effects and in Fig.2d all effects are included.



(a.) Without obscurances. 260 secs. rendering.

(b.) Without direct light. 396 secs. rendering.

(c.) Without mirroring surfaces. 355 secs. rendering.

(d.) Final image with every feature included. 527 secs. rendering.

Figure 2: *These four images demonstrate the contribution that each of the features of our rendering algorithm give to the final image. All the images of the kitchen scene shown here have a resolution of 800x600 pixels.*

3.2. Implementation and results

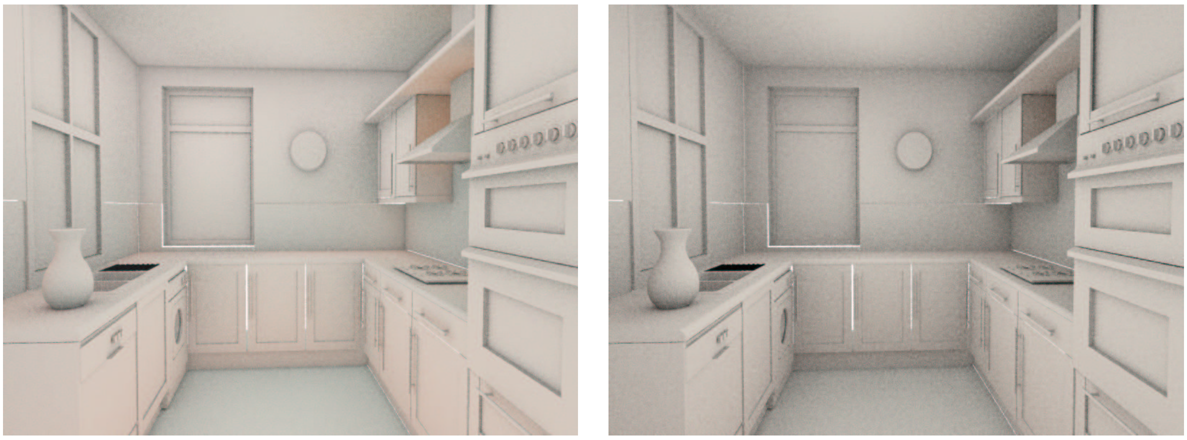
The ray-traced obscurances have been implemented in the SIR system [12]. Several parameters influence the behaviour of the obscurances. The first one is the $\rho()$ function selected. In Fig.3 we compare two of these possible functions. We see in Fig.3b that the negative exponential function emphasizes the darkening of the corners, while with the square root function, Fig.3a, we obtain a nicer color bleeding effect. The second parameter is the maximum distance selected, d_{max}. A small distance will allow for a faster computation, while a longer one will take into account more darkening (and color bleeding) effects. For the purpose of our comparisons we selected the negative exponential function and $d_{max} = 1$.

All images have been rendered on a Pentium 4 1.6 Ghz with 1 Gb RAM. We have used in the obscurances computation 8 rays per pixel and 5 rays per hit point, that makes a total of 40 rays per pixel. The kitchen scene is composed of 28937 triangles featuring 203 objects.

4. Conclusions and future work

We have presented in this paper a fast technique for realistic looking ray-traced images. It is partly non physically realistic, as the diffuse part of the illumination is computed with an improvement over the ambient illumination, the obscurances technique. The use of non-diffuse mean albedo has not yet been implemented and we keep it as future work, instead we have used diffuse-only albedo. Addition of other global illumination effects such as refraction is straightforward. We also expect in the near future to solve the problem posed by strong secondary reflectors. Application to animation in dynamic environments will be also considered. As obscurances are decoupled from direct illumination, a sequence of frames corresponding to a moving light source, i.e. a torch, can be efficiently computed.

The technique presented here can be used as a fast editing tool or as the final image. We expect that advances in graph-



(a.) Square root function (b.) Negative exponential function

Figure 3: *These are two of the many possible functions to compute the obscurances. The first one gives more color bleeding and the second one gives a darker obscurance factor. Image resolution is 800 x 600 pixels.*

ics hardware will make it possible to run in real-time in a not so distant future.

Acknowledgements

This project has been funded in part with grants number TIC2001-2416-C03 from the Spanish government and 2001/SGR/00296 and ACI2002-52 from the Catalan government.

The kitchen model is courtesy of LightWork Design Ltd. Thanks to Coloroid Ltd. and to professor Antal Nemcsics for his kind permission to use the Color Harmony Designer software.

References

1. Castro, F., Neumann, L. and Sbert, M., "Extended Ambient Term", Journal of Graphics Tools, 5(4), pp. 1-7, 2000. 1
2. Per H. Christensen, "Global illumination and all that", SIGGRAPH 2003 course notes #9 (RenderMan: Theory and Practice), pages 31-72, ACM, July 2003. 2
3. Glassner, A. S., Principles of Digital Image Synthesis, Morgan Kaufmann, San Francisco, CA, 1995. 1
4. http://crystal.sourceforge.net/ 2
5. http://wwww.cs.kuleuven.ac.be/cwis/research/graphics /RENDERPARK/ 2
6. Andrey Iones, Anton Krupkin, Mateu Sbert and Serguei Zhukov, "Fast realistic lighting for videogames". IEEE Computer Graphics & Applications, may-june 2003. 1, 1
7. Jensen, H., Christensen, P.: "Photon Maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects", Computers and Graphics 19(2), pp, 215-224, 1995 1
8. E. P. Lafortune and Y. D. Willems, "Bi-directional Path Tracing", Compugraphics '93 Conference Proceedings, pp.145–153, Alvor, Portugal 1
9. Hayden Landis, "Production-ready global illumination", SIGGRAPH 2002 course notes #16 (RenderMan in Production), pages 87-102, ACM, July 2002. 2
10. Méndez, À., Sbert, M., Catà, J., "Real-time obscurances", SCCG'03, Budmerice, Slovak Republic. 1, 2, 2
11. Neumann, L. and Neumann,A., "Radiosity and hybrid methods", JACM Transaction on Graphics, 14(3), pp. 233-265, 2000. 1
12. I. Martín, F. Pérez, and X. Pueyo, "The SIR rendering architecture", Computers&Graphics, 22(5):601-609, 1998. 3
13. E. Veach and L. J. Guibas, "Bidirectional estimators for light transport", SIGGRAPH 95 Conference Proceedings, pp.419–428,1995 1
14. E. Veach and L. J. Guibas, "Metropolis Light Transport", SIGGRAPH 97 Conference Proceedings, pp.65–76,1997 1
15. Zhukov, S., Iones, A., Kronin, G.: "An Ambient Light Illumination Model", Rendering Techniques'98, Springer-Wien, NewYork (Proc. of Eurographics Rendering Workshop98 - Vienna, Austria), pp. 45-55 1, 1