Real-time Obscurances with Color Bleeding
(the three cubes demo)

Àlex Méndez , Mateu Sbert and Jordi Català

amendez@ima.udg.es, *mateu@ima.udg.es*, *jcata@ima.udg.es*, IIiA, Universitat de Girona

Abstract

Here we present an animation that demonstrates the power of the obscurances technique and how it can be used in a video-game context when moving objects are in the scene.
Obscurances is a powerful technique that simulates diffuse illumination, i.e. radiosity, with a much lower cost. Its advantage is based in that it considers only neighbour interactions instead of considering global ones, and in being decoupled from direct illumination computation. The implementation of this technique is made on a 3D game engine that allows real-time recomputation of obscurances for moving objects. Extensions to deal with color bleeding can also be appreciated.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism, Radiosity, Animation, Shading.

1   Introduction

Radiosity techniques [2, 3, 6] are commonly used to simulate diffuse illumination in a closed 3D environment. These techniques, although very powerful and increasingly faster [1], do not fulfill yet the requirements for a fast and efficient scene editing and/or real time update. The interaction of each surface in the scene with each other surface has to be (at least potentially) considered. Obscurances [7, 8] were designed to offer an alternative to radiosity, while keeping its good visual properties that makes a scene appear realistic. In addition they can be applied without any change to open 3D environments. Obscurances have been already used in 3D computer games and animations (intros and cut scenes) for Heretic-II, Civilization-III, etc.

In this movie we show an animation captured from an implementation on a 3D engine, Crystal Space, allowing real-time update of obscurances for moving objects. We also discuss a further improvement to obscurances, color bleeding between surfaces that adds realism to the generated illumination.

2   Previous Work

In [7,8] 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:

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

where

- $\rho(d(P,\omega))$: function that determines the magnitude of ambient light incoming from direction ω and taking values between 0 and 1
- θ angle between direction ω and the normal at P
- I_A: ambient light intensity
- $R\ (P)$: Reflectivity at P
- $\frac{1}{\pi}$ is the normalization factor such that if $\rho()=1$ over the whole hemisphere Ω then $I(P)$ is $R*I_A$

Direct illumination will then be added to (1) to obtain the final illumination of the point.

Function $\rho()$ increases with d. Its shape is:
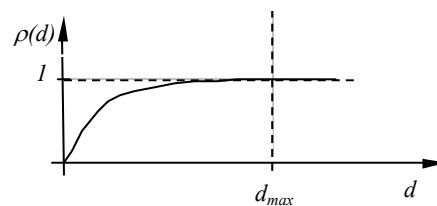
$\rho(d)$ will be equal to 1 for $d \geq d_{max}$. This means we are not taking into account intersections farther than d_{max}. We only take the "neighbour" ones.

Obscurance of point P is then defined as:

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

Clearly, $0 \leq W(P) \leq 1$. Obscurance for a patch (in radiosity terms, a polygon from a mesh in which surfaces are subdivided) is the average of the obscurances for all points in the patch. 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).

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

$$I_A = \frac{R_{ave}}{(1 - R_{ave})} * \frac{\sum_{i=1}^{n} A_i * E_i}{A_{total}} \tag{3}$$

where

$$R_{ave} = \frac{\sum_{i=1}^{n} A_i * R_i}{A_{total}}$$

and A_i , E_i and R_i are the area, emissivity and reflectivity of patch i, A_{total} 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.

As the obscurance computation only takes into account the neighbourhood of a patch (understanding by it the patches near than d_{max}) the computation can be done very fast and efficiently. For further details see [7,8].

3   Color bleeding

The obscurances as presented in previous section lack one of the qualities of radiosity computed scenes, color bleeding. This quality allows for much greater realism. We present here a straightforward way to account for that, with no added computational cost[1].

We first modify the ambient light computation (3) in the following way:

$$I_A = \frac{1}{(1 - R_{ave})} * \frac{\sum_{i=1}^{n} A_i * E_i}{A_{total}} \tag{4}$$

Then, obscurances (2) are modified accordingly to include a reflectivity term:

$$W(P) = \frac{1}{\pi} \times \iint_{\omega \in \Omega} R(Q) \rho(d(P, \omega)) \cos \theta \, d\omega \tag{5}$$

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}.
The improved obscurances model has been implemented on the RenderPark [4] radiosity software package, and is computed using the same Monte Carlo (and Quasi Monte Carlo) ray casting technique that the Random Walk and Hierarchical Monte Carlo techniques use. This is, we cast $\cos\theta$ distributed

---

[1] Color bleeding was first discussed in the unpublished report [9].

rays from a patch.   The obscurance for patch i will be the average of the values gathered by the rays cast from i:

$$W(i) = \frac{1}{N_i} \sum_{j=1}^{N_i} \rho_j R_{int} \tag{6}$$

Where N_i are the rays cast from patch i, R_{int} is the reflectance from intersected patch (if no patch is intersected then we take R_{ave}) and ρ_j is the value of the function $\rho()$ for line j.
The number of rays and patches for all the scenes shown in the images are similar, and the cost for obscurances computation is less than one third of the cost for secondary illumination computation in the two radiosity algorithms used, shooting Random Walk and Hierarchical Monte Carlo. As stated before, the computation takes advantage of the fact that only a d_{max}-neighboorhood of the patch has to be examined for intersection. The total number of available rays is distributed in the obscurance computation according to the area distribution.
In Fig.1 we show the Cornell box scene computed with the obscurances without color bleeding, while in Fig.2 we have used the improved algorithm. Color bleeding is clearly visible, adding a lot of realism to the image. We can compare these images with the ones obtained with radiosity algorithms (Fig.3&4). We see that the improved obscurances represent a step forward towards simulating a radiosity image, with no added cost.

Fig 1. Obscurances without color bleeding

Fig.2. Obscurances with color bleeding

Fig.3. A Monte Carlo radiosity solution (shooting Random Walk)

Fig.4. A Hierarchical Monte Carlo radiosity solution

4   Real-time

Although the computation of the initial obscurances is not real time, we can update them in real time for a small number of polygons. We do it as follows:

We store initially in a list the patches influenced by the dynamic objects, i.e., the patches that have been reached by rays from a patch in a dynamic object in the initial obscurances calculation. When the object moves to a new position the obscurances of this list have to be recalculated. Obviously, also the obscurances of the patches of the moving object have to be recalculated, and this causes the update of the list of influenced patches. Lastly, the obscurances of these patches have to be recalculated, ending the whole updating process. The algorithm thus starts by creating the list of influenced patches:

For each patch

    Compute initial obscurances

    If ray from patch hits moving object

      Store in list of influenced patches

    Endif

    If patch in moving object

      Store hit patches

    Endif

Enfor

Once computed, and when an object moves to a new position, the obscurances are recomputed:

For each patch in the moving object

    Recompute obscurances

    Store hit patch in list of influenced patches

    Store hit patch in new list of influenced patches

Endfor

For each patch in the list of influenced patches

    Recompute obscurances

    If ray from patch hits moving object

      Store in new list of influenced patches

    Endif

Endfor

List of influenced patches:=new list of influenced patches

5   Implementation and results

Obscurances have been implemented within the 3D engine Crystal Space [5]. In Fig5-8 we show some screen snapshots of scenes Room and Infinite. Fig.5&7 correspond to direct illumination plus ambient light, while in Fig.6&8 we have direct illumination plus obscurances. In Fig.6 the effect of color bleeding caused by the red and yellow cubes contributes clearly to increase realism in the scene. See also the increased realism in   Fig.8   respective   to   Fig.7.   In http://ima.udg.es/iiia/GGG/Gallery/users/~jcata/realtime-obscurances.avi another example, a bit less complex than "3 cubes", of real time animation of scene Room can be found. Table 1 shows the characteristics of both scenes. Times correspond to running the program on an 900 Mhz AMD[2]. With our non-optimized implementation we get rates of 7 fps in the scene Room with 4016 patches while recomputing in real time the obscurances of the moving object (around 150 patches).

---

[2] Note that the rate that appears at the video is lower due to the CPU requirements of the video capture program.

Scene	Poly.	Patches	Rays	TimeC	FPS	FPS/R
Room	73	1423				
			60	1.7	56	17
			80	2.2	56	14
			100	2.8	56	11.5
		4016				
			60	4.6	56	10.5
			80	5.9	56	8
			100	7.3	56	6.5
Infinite	180	5508				
			60	6.4	40	-
			80	8.3	40	-
			100	10.3	40	-
		13824				
			60	19.8	40	-
			80	20.7	40	-
			100	26.1	40	-

Table 1. Characteristics for scenes Room and Infinite, for two different meshing in patches. Second column gives the number of polygons, third one the patches, fourth one the number of rays per patch to compute obscurances, fifth one its computation time in seconds, sixth one the fps, and seventh one the fps when recomputing obscurances.
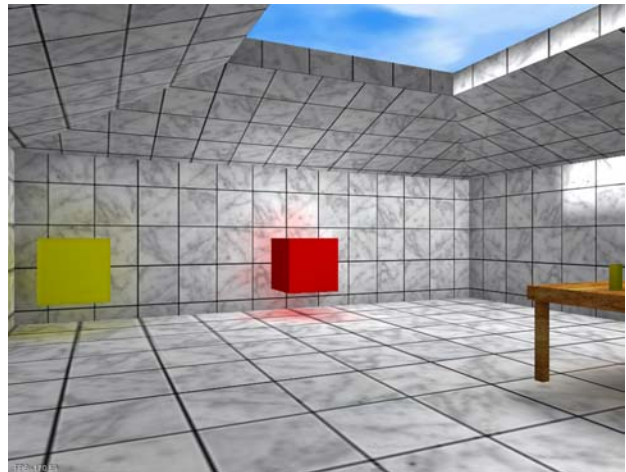

Fig.5 Screen snapshot of the Room scene without obscurances

Fig.6 Screen snapshot of the Room scene with obscurances

6   Conclusions and Future work

The "3 cubes" animation that we present demonstrates the obscurances illumination technique to deal with color bleeding in a real-time version embedded within the Crystal Space 3D game engine, showing the enormous potential of this technique to simulate radiosity-like realistic-looking illumination in virtual and dynamic environments.
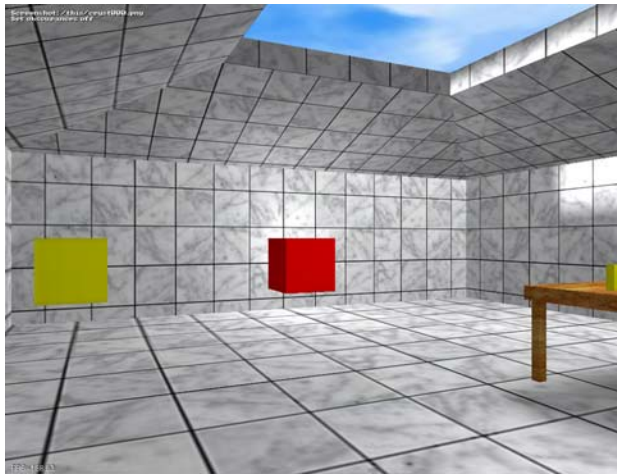

Fig.7 Screen snapshot of the Infinite scene without obscurances

Fig.8 Screen snapshot of the Infinite scene with obscurances

[10]   Méndez, À, Sbert, M., Catà, J., *Real-Time Obscurances with Color Bleeding*, Proceedings of the Spring Conference in Computer Graphics 2003, Budmerice, Slovak Republic.

Acknowledgements

References

[1]   Bekaert, P., Neumann, L, Neumann, A., Sbert., M., Willems, Y.: "Hierarchical Monte Carlo Radiosity", Springer-Wien, NewYork (Proc. of Eurographics Rendering Workshop'98 - Vienna, Austria), pp. 259-268

[2]   Goral C., et al.: "Modeling the interaction of light between diffuse sources", ACM SIGGRAPH'84 Conf. Proc

[3]   Hanrahan P., Salzman D., Aupperle L.: "A Rapid Hierarchical Radiosity Algorithm." Computer Graphics'91 (SIGGRAPH'91 Conference Proceedings) Vol.25(4), July 1991, pp.197-206

[4]   http://www.cs.kuleuven.ac.be/cwis/research/graphics/RENDERPARK/

[5]   http://crystal.sourceforge.net/

[6]   Sillion F., Puech C.: "Radiosity and Global Illumination." Morgan Kaufmann, SF, US, 1994

[7]   Zhukov, S., Iones, A., Kronin, G.: *"An Ambient Light Illumination Model"*, *Rendering Techniques'98*, Springer-Wien, NewYork (Proc. of Eurographics Rendering Workshop'98 - Vienna, Austria), pp. 45-55

[8]   Iones, A., Krupkin, A., Sbert, M, Zhukov, S. *Fast realistic lighting for video games*, IEEE Computer Graphics&Applications, may-june 2003

[9]   M.Garcia, M.Sbert. A.Iones, A.Krupkin, "Improved obscurances with color bleeding", Research Report IIiA 01-12-RR, 2001, University of Girona, Girona