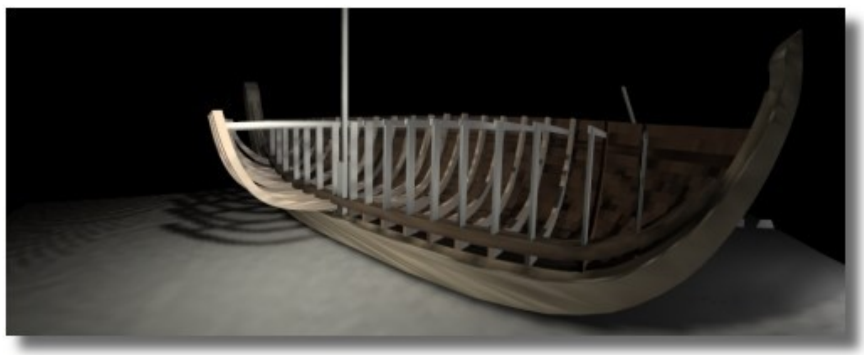
Rendering with Radiosity
by *Xavier Michelon*

A r t i c l e s

Introduction

If you have thrown a glance at the new features available in Blender since the end of the C−Key system, you might have noticed a new tab in the workspace : the **Radiosity** tab. Likewise, you might have noticed the Radiosity menu entry in Moonlight Atelier. Maybe you ignored it. You shouldn't have ! If you had take the time to read the files available on blender web site (especially the 1.8 manual Appendix), you would have discovered how realistic radiosity renderings can be !

In this paper, we will introduce the principles of radiosity, in order to understand its interests and limitations. We will not see here the way to obtain radiosity renderings in Blender. If this is what you're looking for, check the tutorials available from www.blender.nl. You can also find interesting document on NaN ftp site (ftp.blender.nl).

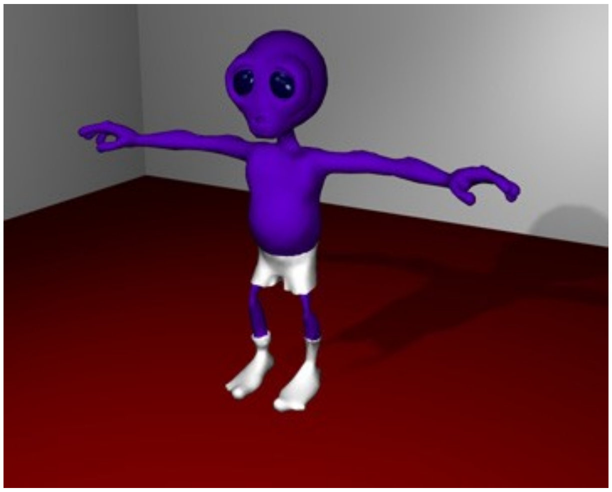


Reconstitution of an antique ship with Blender, rendered with radiosity by Frederic Toussaint

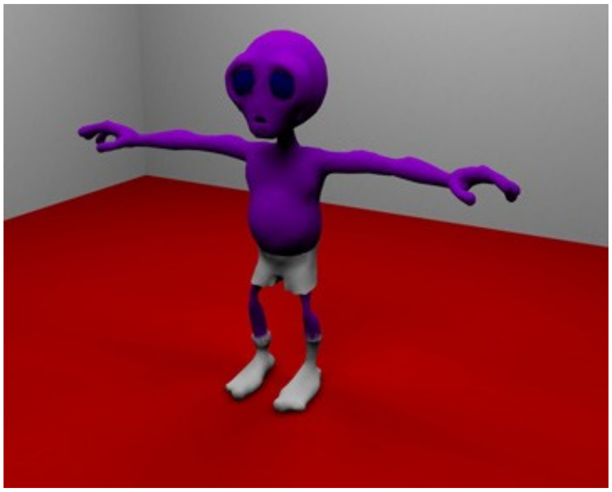
Note : this document is a translation from the french original version. Please excuse my poor english ;)

Comparison

As a starting point, let's have a look at the two following images. The have been rendered from the same Blender scene. The first one was rendered using the classical scanline engine. The second is the result of a radiosity solution calculation. There are many differences. Check by yourself :



scanline rendering



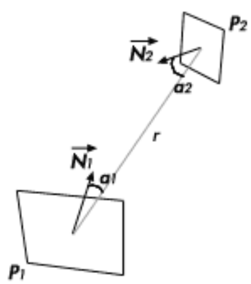
Radiosity rendering

Principles of Radiosity

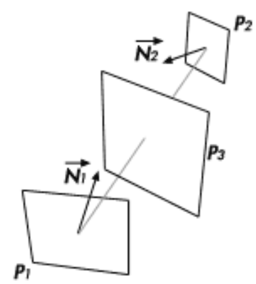
If you have a short scientific background, you probably have already heard of the 'wave−particle duality' principle for light. It means that light behave both as a particle and a wave. Classical rendering algorithms (raytracing,z−buffer(including its scanline variant) use the particle aspect of light. Radiosity is based on on the wave properties of lights

Historically speaking, radiosity algorithms are the adaptation of a model created by physicians for heat transfers. The main principle of radiosity is to simulate the energetic exchanges induced by light, whose color is determinated by a wavelength.

Let's consider a simple 3D scene modelized by a polygonal mesh : an empty room with a cube on the roof. The basic elements for radiosity calculation are patches (plane surface elements). Each face of the cube and the room is patch. Each patch will receive energy (light) from the other patches. It will absorb some energy (according to the patch properties ("the material")), and will reflect the rest of this energy. The amount of energy transmitted from patch A to patch B is a function who depends on the distance between the two patches, on their respective orientations, and on the possible presence of occluding patches.



Calculation of energetic exchanges between two patches



Occlusion phenomenon

Of course, in order to get energy exchanges, there must be light sources in the scene. Some patches are defined as light emittors. In the case of our simple scene, we can define the ceiling as an emittor patche, or we can add a sphere in order to get a light bulb effect.

a

L'équation de radiosité

Until now, we've talked about energy exchanges between patches. In the Global Illumination model, we will in fact use the radiosity, i.e. a light energy emitted per surface unit and per time unit. The symbol B is used for radiosity.

All the energy exchange are ruled by an equation : the radiosity equation. It sums up all we have already seen :

$$B_i = E_i + R_i \sum_j B_j F_{ij}$$

The radiosity emitted by a patch i, noted Bi, equal the auto−emitted energy Ei (only for light source), plus the sum of all the radiosities received from the other patches in the scene. The Ri factor reflects the way a patch absorbs energy (it depends on the material associated with the patch). The Fij factor, called Form Factor, depends on the geometry, and is quite hard to calculate. Fij takes into account the distance between the patches, their orientations, and the presence of possible occluding patches (remember the figure in the previous section).

The radiosity equation is the result of the simplification of a more generic equation (much more heavy too !). The generic equation contains double integrals and other funny mathematical stuff. The hypothesis made to obtain the radiosity equation would make a mathematician scream, but the important thing is too obtain a visually correct result. However, the consequences of those hypothesis are important :

- The radiosity is constant on a patch. As a consequence, each face of the cube in our scene will only have one color. Not really realistic. The solution to this problem consist in subdivising each patch into smaller patches. Each small patch will have its own color. When rendering, a shading algorithm will make the big patch look smooth.
- The light is reflect unvaryingly in all the directions (diffuse surfaces). This explain the dull aspect of radiosity rendering. No object is shiny.

Patch Subdivision

Solving the radiosity equation

Rendering a 3D scene with a radiosity engine means solving the radiosity equation for each patch (remember they have been subdivided). This means solving a N by N linear system, where N is the number of patches. Our computers are not able to solve this in a humanly acceptable time, cause often N is near thousands of hundreds.

Of course, there is a way to obtain faster results : iterative solving methods like Jacobi or Gauss−Seidel. The principle of iterative methods is to proceed step by step, with a partial solution on each step. On each step, you approach more and more the final result. For the end user, this means you progressively see the image appear. The correct solution is never reached. You stop the calculation process wether manually or automatically, when the amount of energy that has not yet been dispatched is under a fixed value. Actually, the iterative methods have a physical meaning : the repartition of the whole light energy in the scene, until it has been absorbed by all the patches.

Advantages

The main advantage of radiosity is the realism of the result. Just throw a glance at the Blender Community Journal if you are not yet convinced.

Another advantage : solving the radiosity equation is viewpoint independant. Once you have computed the solution, you can change the viewpoint without having to recompute the solution. But, even the smallest movement of an object in the scene modify the energy exchanges, and you have to recalculate the solution.

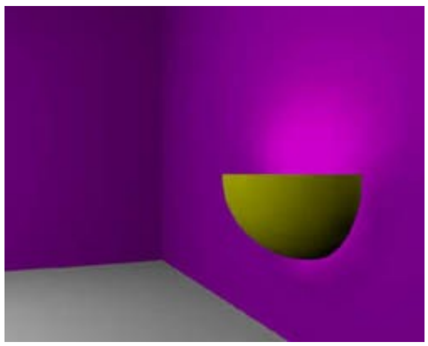
That is why radiosity is perfect for domains where you have to render scenes with static objects, like architecture

Drawbacks

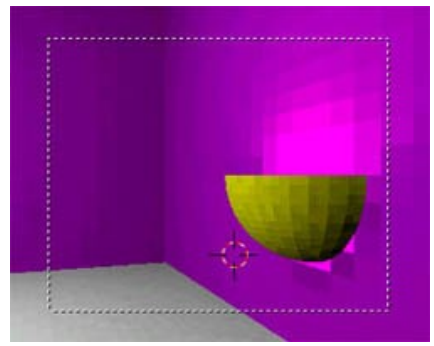
The main drawback with radiosity is the amount of time required for the calculation. This is due to the solving of the equation system, but also to the computation of the form factor.

For animated scenes where only the camera is moving, the time required is acceptable. But in the case of cinematographic scenes where objects move, the radiosity method is not a good solution.

Other drawbacks are linked to the hypothesis made to reduce the radiosity equation : the radiosity and form factor are constant on a patch. Here is a classical example :



Final Image



Solid View

A light source (a sphere) is placed in the bowl and the light 'slobbers' under the ball. If you look at the solid view, the problem can be easily explained. Some patche (uniform color squares) are both inside and outside the bowl. As the radiosity and form factor are supposed to be constant on a patch, the algorithms use a sample point (not necessarily the center of the patch). On the problematic patches, the sample point is taken inside the bowl and the color is applied outside too. Moreover the shading process applied for the final rendering amplify the defects.

As we have already seen, the diffuse light hypothesis is a limitation for the rendering methods. In order to get good results, rendering engines combine radiosity with other rendering method like raytracing which allow specular lights and other important effects. Here is an example provided by André Pascual, and rendered with Moonlight Atelier.

Here, differences between the images are less visible than those in the Comparison chapter, but the radiosity increase the physical correctness of the final image.

Conclusion

You now know the basic principles of radiosity. I hop this article will have given you the will to go further and to experiment radiosity with Blender or Moonlight. You now what you can expect from a radiosity engine, and what you should not.

If you want to read more about Radiosity, here are two really good books dealing with the subject

Computer graphics : principles and practice	J. Foley	Addison−Wesley
This book should be the starting point for everyone interested in computer graphics. A section is dedicated to global illumination.		

Radiosity & Global Illumination	F. Sillion	Morgan Kaufmann
All about radiosity. Very technical.		