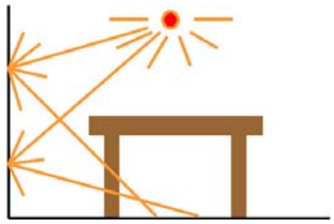
Global Illumination: Radiosity

The radiosity method originates from thermodynamics and models light propagation in a way that preserves the energy equilibrium in a closed system. The procedure describes the physical process of light propagation in a diffusely reflecting environment, i.e. the calculation of the brightness of all surfaces in the scene, taking into account the mutual influence. This is why even surfaces which are not directly illuminated by a light source receive some amount of light. Each illuminated item acts as a secondary light source. For the image generation we first calculate the light propagation in the scene without knowing the camera position, and assuming that the viewer does not influence light propagation. Then the objects can be rendered from different directions without having to recalculate the light propagation every time anew.



The Radiosity Equation

Let the scene be composed of n planar polygons P_i which are denoted as *patches* in radiosity terminology. Simplifying, we assume that each patch has a homogeneous, perfectly diffuse surface. Light sources are also patches, which evenly distribute their emitted light into all directions. P_i's ***Radiosity*** B_i is the total emitted energy, i.e. the sum of self-emitted and reflected energy, measured as power per unit area. This density of light energy is proportional to the perceived brightness. The next simplification that we assume is that every position on a patch has the same radiosity. Under these conditions the equation for the radiosity of a patch is:

$$B_i = E_i + \rho_i \cdot \sum_{j=1}^{n} B_j \cdot F_{ij}$$

Here E_i denotes P_i's self-emission, ρ_i is the diffuse reflection coefficient of the surface (depicts what percentage of the incident light is diffusely reflected, also called *albedo*), n is the number of patches in the scene, B_j are the radiosities of all the other patches and F_{ij} are the so called *form factors*, which determine the ratio of radiosity affecting P_i coming from P_j (which is the same amount of radiosity originating at P_i hitting P_j). F_{ij} are mere geometric quantities and independent of light sources or radiosity values, which means that we can evaluate them before solving for the radiosity values.

The radiosity equations for n patches result in a system of n linear equations of n unknowns B_i, which can be solved numerically using the Gauß-Seidel-Iteration:

$$B_i - \rho_i \sum_{j \neq i} B_j F_{ij} = E_i$$

$$\begin{bmatrix} 1 & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 & \cdots & -\rho_2 F_{2n} \\ \vdots & \vdots & & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

$$B_i^{k+1} = E_i + \rho_i \sum_{j \neq i} B_j^{k} F_{ij}$$

This system of equations has the following properties:

(1) $F_{ii} = 0$ for all i, because a patch cannot illuminate itself,

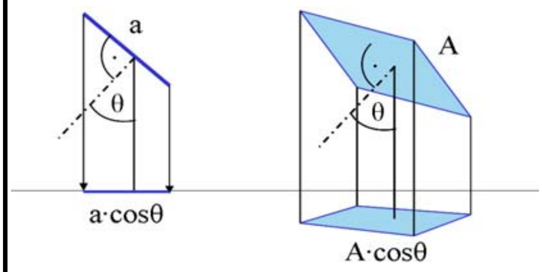
(2) $\Sigma(j=1,n)F_{ij}=1$, because the radiosity-ratios affecting a patch must add up to 100%,

(3) $\rho_i < 1$ for all patches, because it is impossible to reflect more light than is incident.

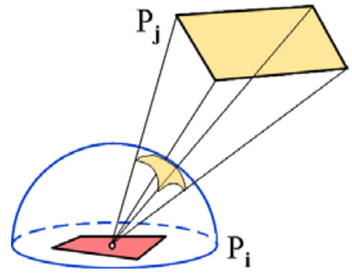
Therefore the matrix is diagonally dominant, thus is numerically well-behaved.
Additionally it holds true that:

(4) E_i equals 0 for most patches, because usually only very few patches are light sources.

Form Factor Calculation

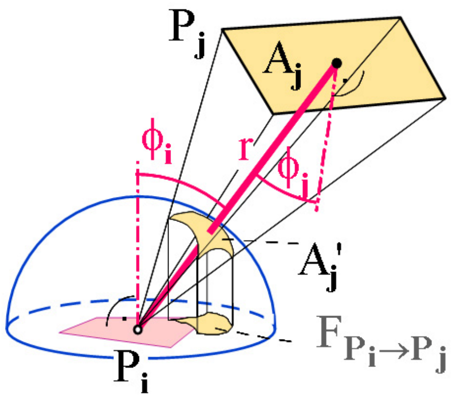


A simple geometric relation helps us to derive an equation for calculating the form factors F_{ij}. The relation states: The area of the normal projection from a surface A onto another surface is reduced according to the cosine of the angle between the two surfaces, i.e. $A \cdot \cos\theta$.



We define the *form factor F_{ij}* as the ratio of radiation energy leaving patch P_i and hitting patch P_j. In other words, what percentage of energy leaving P_i arrives at patch P_j. Trivially, this value also tells us what percentage of the energy arriving on patch P_i originates at patch P_j.

F_{ij} can be calculated as follows: Let us assume that the area of the patches is small compared to their distance r. Let A_j be the area of P_j. Now imagine a hemisphere with radius 1 being placed on patch P_i onto which patch P_j is projected. The resulting area A_j' is approximately $A_j\cos\varphi_j$, where φ_j denotes the angle between the normal of patch P_j and the line between the two patches. Remember that energy hitting patch P_i is proportional to the cosine of this incident angle, which explains why we multiply by $\cos\varphi_i$ (which corresponds to a projection onto the hemisphere's base-plane) to come up with the correct percentage of P_j's influence. Since all form factors F_{ij} have to sum up to 1 (100%), we normalize the result with the hemisphere's base surface area, i.e. with $1^2\pi = \pi$, and finally get the form factor:



$$F_{ij} = \frac{\cos\phi_i \cos\phi_j A_j}{\pi r^2}$$

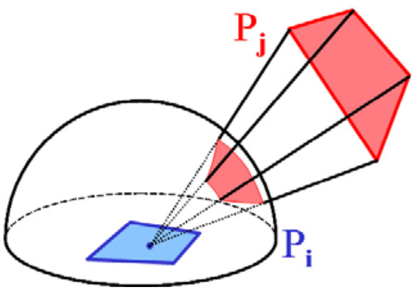
Strictly speaking, the form factor is the sum of all influences of P_j averaged over the area of P_i, so we get:

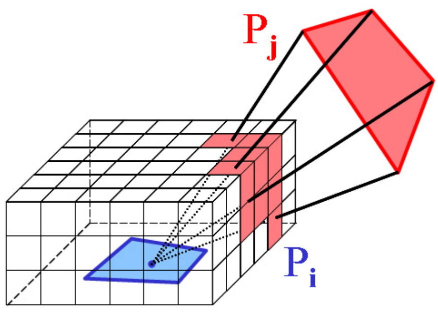
$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\phi_i \cos\phi_j}{\pi r^2} \, dA_j \, dA_i$$

These equations are valid under the assumption that there are no obstacles between the two patches, so that light can travel from P_i to P_j without obstructions. To take this into account, correct form factors have to also include mutual visibility.

Furthermore, the *reciprocity principle* is valid, which relates the dependence of form factors between two patches: $\mathbf{A_i \cdot F_{ij} = A_j \cdot F_{ji}}$.

In most implementations hemicubes are used instead of hemispheres on patches P_i to calculate the form factors. The whole scene is then projected onto the hemicube. We can then make use of the z-buffer technology, i.e. the hemicube's faces are subdivided into a regular grid of pixels, then all other patches are projected onto them with the cube center acting as the center of projection. For each pixel we determine its form factor in advance and then we add that percentage for each patch, which eventually gives us its form factor. Alternatively ray tracing can be used to determine form factors.

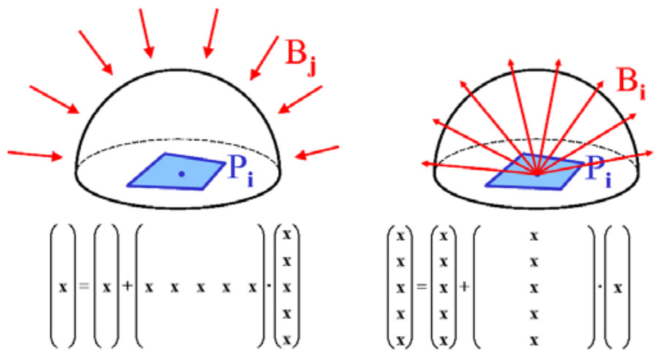




Progressive Refinement

In order to solve the system of equations with the Gauß-Seidel-method, all form factors must be calculated in advance, because we need all entries in the coefficient matrix. For n patches we get nearly n^2 numbers, which not only takes a long time to evaluate but also leads to extreme memory consumption. To solve the systems of linear equations with the already mentioned properties, we can alternatively use the Southwell-method, which can also be interpreted geometrically: Instead of calculating the next iteration B_i^{k+1} for one particular patch P_i in one step by "gathering" (see left figure) the energy of all contributing patches in one step, we choose the brightest patch and distribute its energy onto the other patches („shooting", see right figure). In this way all patches are refined a little bit in every step. The solution process converges much faster because the brightest patch is chosen for energy-distribution for each new iteration step.

$$B_i^{k+1} = E_i + \rho_i \sum_{j \neq i} B_j^k F_{ij}$$



„gathering" „shooting"

Let $B_{(i \text{ from } Bj)}$ be the radiosity-amount of P_i, which is caused by B_j. P_i's influence on B_j is symmetric to P_j's influence on B_i:

$$B_{(i \text{ from } Bj)} = \rho_i B_j F_{ij}, \quad \text{thus} \quad B_{(j \text{ from } Bi)} = \rho_j B_i F_{ji}.$$

From this and the fact that $A_i \cdot F_{ij} = A_j \cdot F_{ji}$ we can conclude that

$$B_{(j \text{ from } Bi)} = \rho_j B_i F_{ij} \, (A_i / A_j),$$

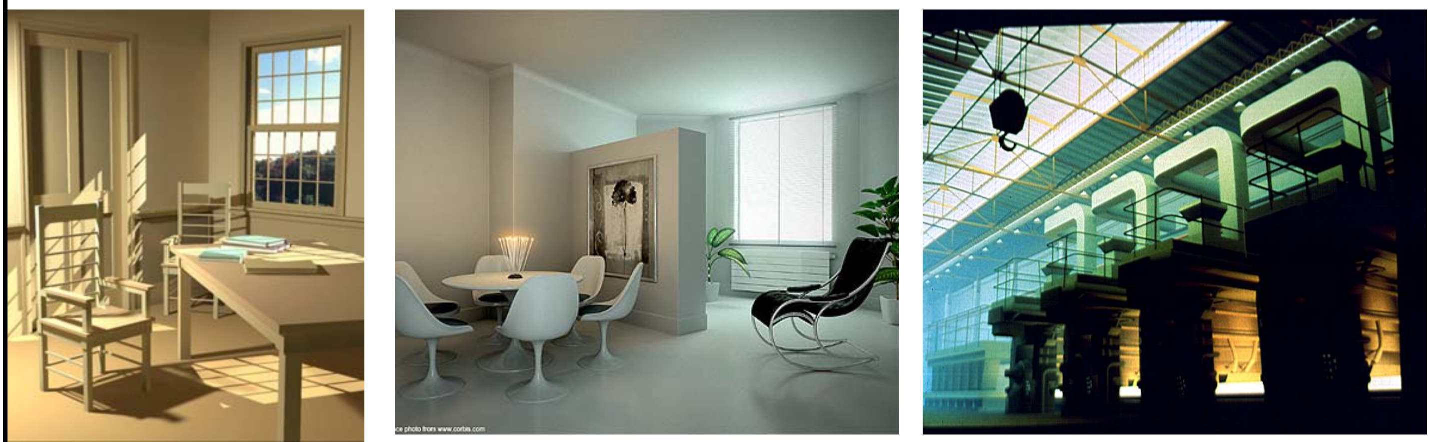
This shows that $B_{(j \text{ from } Bi)}$ can be calculated using the form factors F_{ij}. So for each patch we store the radiosity B_j collected so far (i.e. the best approximation up to that point) and the "not yet distributed radiosity" ΔB_j, which is the basis for selecting the next "brightest" patch. We initialize the B_j and the ΔB_j with E_j: $B_j = \Delta B_j = E_j$ for all j.

Basically an iteration step now looks like this:

```
select patch i with highest Aᵢ*ΔBᵢ
FOR selected patch i {set up hemicube
                      calculate form factors Fᵢⱼ  }

FOR each patch j { Δrad := ρⱼ*ΔBᵢ*Fᵢⱼ*Aᵢ/Aⱼ
                   ΔBⱼ := ΔBⱼ + Δrad
                   Bⱼ := Bⱼ + Δrad    }
ΔBᵢ := 0
```

This method is also called "progressive refinement".

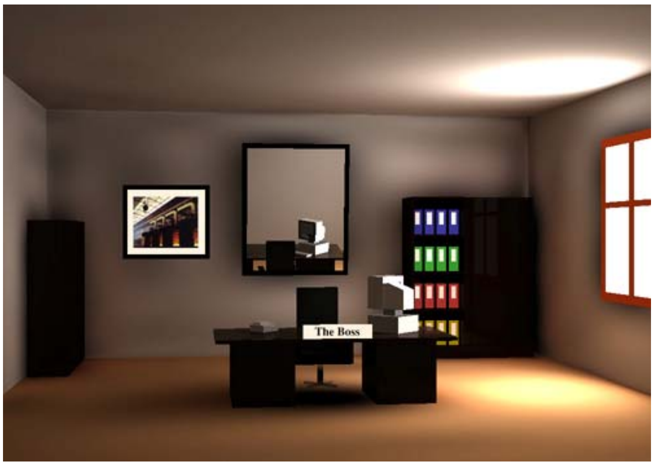


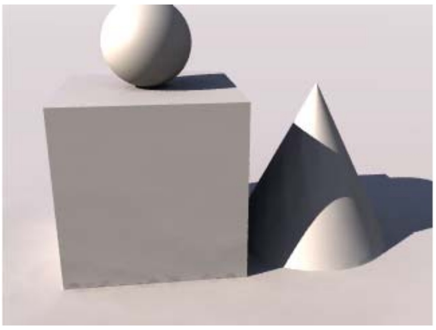
Three examples of scenes rendered with the radiosity technique

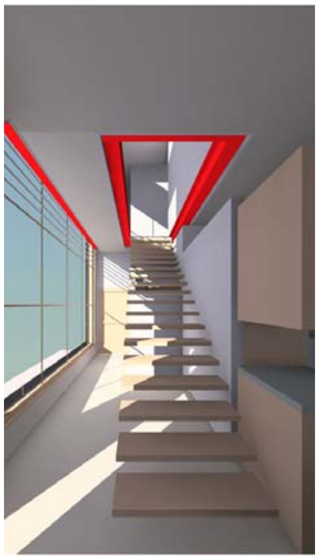
Aspects of Radiosity

Radiosity is a *view-independent* method for calculating the brightness of (diffuse) patches, after which we still need a rendering step. A simple polygon-based Gouraud-shading scanline renderer is often used in this context. The diffuse radiosity values can also be used as “ambient light” input values for a ray tracer to achieve additional effects such as reflections and shadows.

The basic principle of the radiosity method, as presented here, can be extended in many ways. To reduce the number of patches, they can be hierarchically structured, so that patches which are further away don’t need to be treated individually. Furthermore, there are several stochastic approaches, which either try to calculate the form factors or to solve the system of linear equations by employing Monte Carlo methods. The path-tracing method follows rays emanating from light sources, which is similar to ray tracing, but different insofar as the light rays are not traced in reverse direction. At intersection points, the impact of light rays is stored and later these values are interpolated to achieve the object’s appearance.







3 images which were generated by combining radiosity and ray tracing
(left: reflections, center & right: shadows)