Radiosity

Think of the “world” as a bunch of patches. Some are sources, (and reflect), some just reflect. Each sends light towards all the others.

Consider one color band at a time (some of the computation is shared among bands).

Each surface, \(i\), *radiates* reflected light, \(\mathbf{E}_i\), per unit area.

Each surface, *emits* light \(\mathbf{E}_i\) (if it is not a source, this is 0).

Denote the albedo of surface \(i\) as \(\rho_i\)
Radiosity equation

\[ B_i = E_i + \rho_i \sum_j F_{j \rightarrow i} B_j \frac{A_j}{A_i} \]

The form factor \( F_{j \rightarrow i} \) is the fraction of light leaving \( dA_j \) arriving at \( dA_i \), taking into account orientation and obstructions.

Useful relation

\[ A_i F_{i \rightarrow j} = A_j F_{j \rightarrow i} \]

The equation now becomes

\[ B_i = E_i + \rho_i \sum_j F_{i \rightarrow j} B_j \]

Rearrange to get

\[ B_i - \rho_i \sum_j F_{i \rightarrow j} B_j = E_i \]

In matrix form

\[
\begin{bmatrix}
1 - \rho_1 F_{1 \rightarrow 1} & -\rho_1 F_{1 \rightarrow 2} & \cdots & -\rho_1 F_{1 \rightarrow n} \\
-\rho_2 F_{2 \rightarrow 1} & 1 - \rho_2 F_{2 \rightarrow 2} & \cdots & -\rho_2 F_{2 \rightarrow n} \\
\vdots & \vdots & \ddots & \vdots \\
-\rho_n F_{n \rightarrow 1} & -\rho_n F_{n \rightarrow 2} & \cdots & 1 - \rho_n F_{n \rightarrow n}
\end{bmatrix}
\begin{bmatrix}
B_1 \\
B_2 \\
\vdots \\
B_n
\end{bmatrix} =
\begin{bmatrix}
E_1 \\
E_2 \\
\vdots \\
E_n
\end{bmatrix}
\]

So, in theory, we just compute the Bi’s by solving this (large!) matrix equation.

The fun part: Computing the \( F_{i \rightarrow j} \)

Without obstruction \( dF_{\theta, \phi} = \frac{\cos \theta \cos \phi}{r^2} dA_j \)

Fancy methods exist for of computing and/or approximating storing form factors (e.g. hemisphere and hemicube methods).
Fancy methods exist for computing and/or approximating storing form factors (e.g. hemisphere and hemicube methods).

Hemicube: For a given patch, project other patches onto a cube surrounding it, which is an appropriately scaled version of the projection onto the hemisphere (use cube trick because planar projection is faster).

Hemicube projections can be tagged for distance (like Z-buffer) to deal with occlusion.