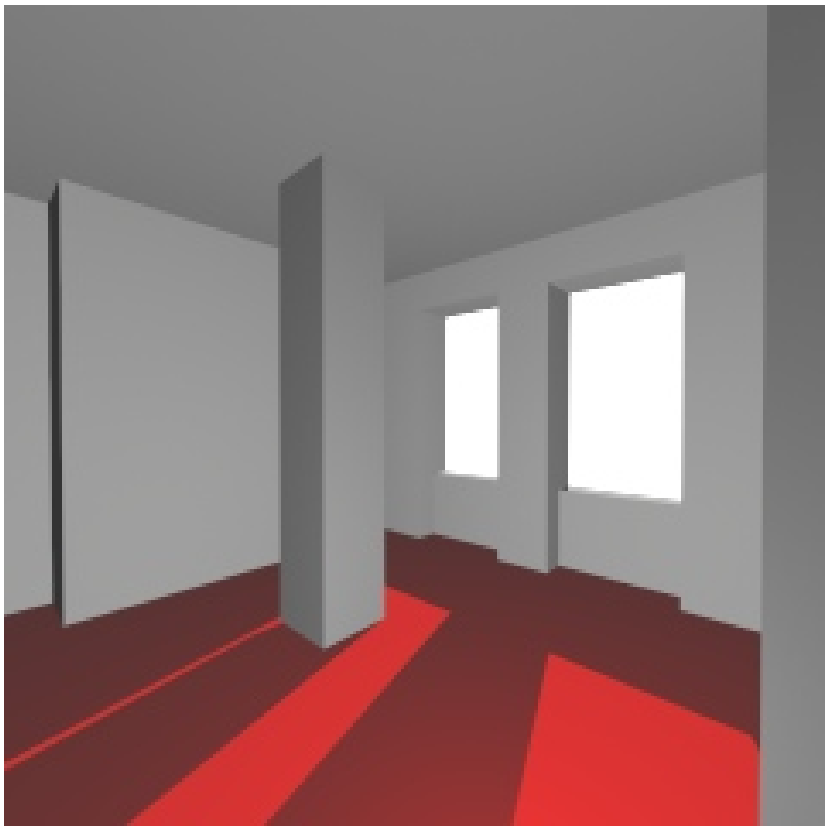
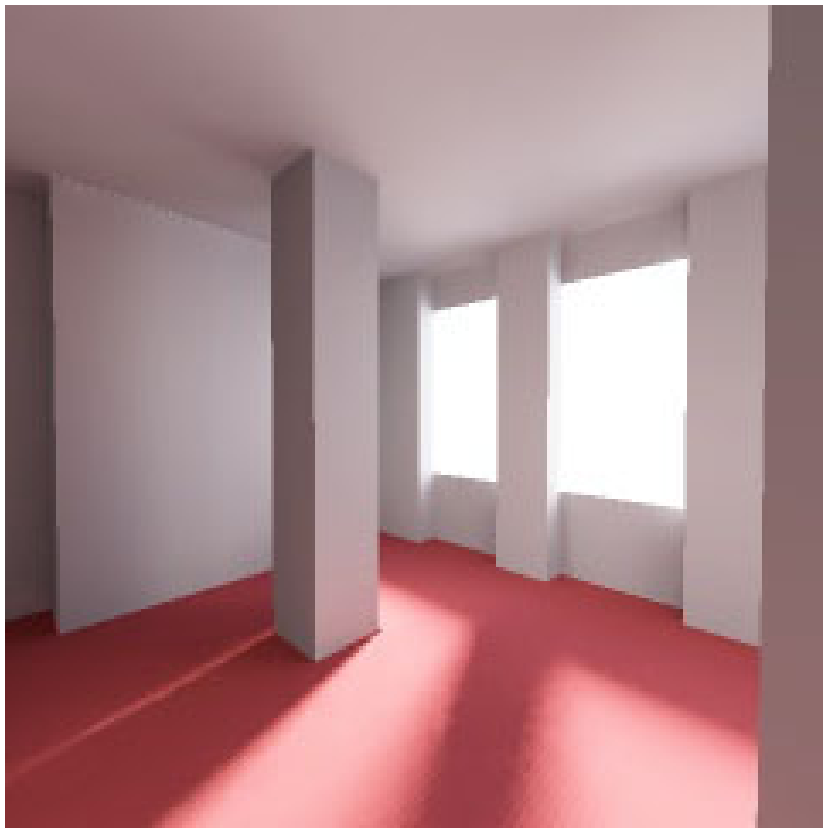
Quadtree application: Realistic Illumination of a Scene

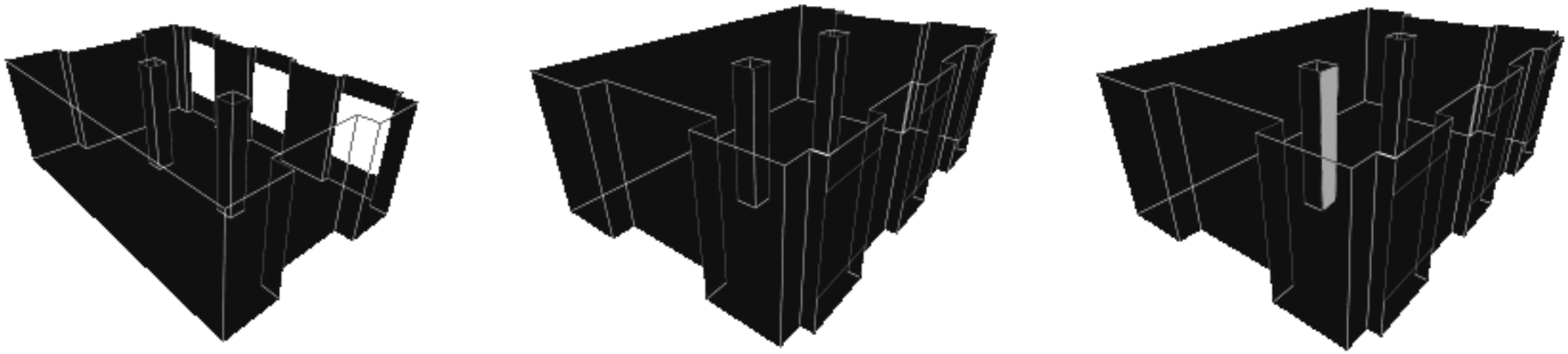
polygons with direct lighting ← → scene with global lighting





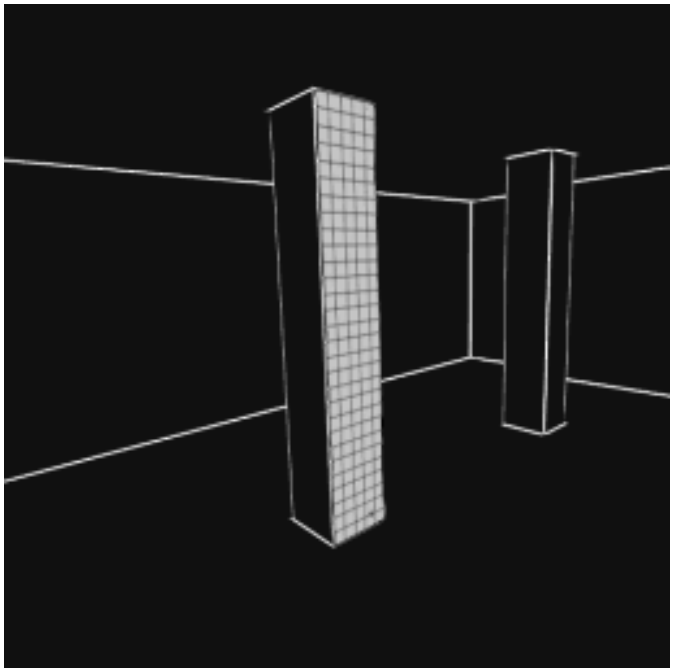
Radiosity

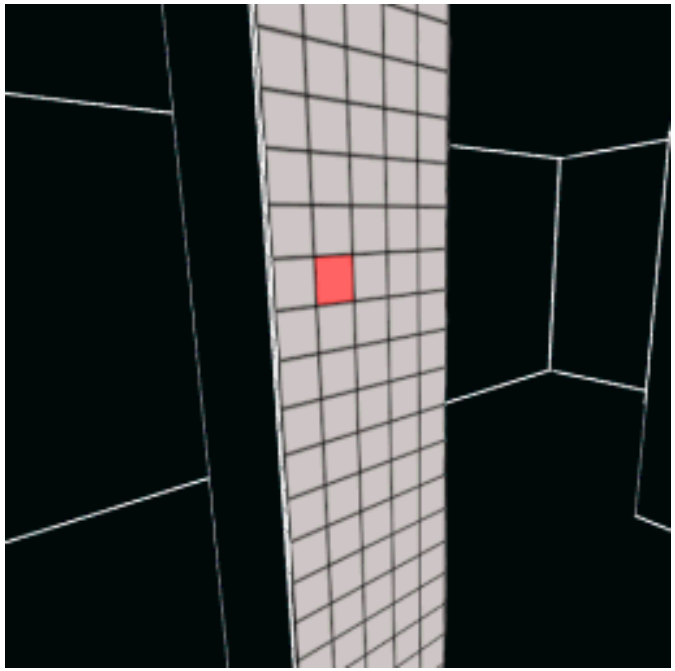
Radiosity: a method for *global illumination* with *diffuse reflections*. It originates from **thermal heat transfer**, i.e. the emission and reflection of heat. Example:



Patches

Radiosity doesn't compute light transport along a discrete number of rays, but instead focuses on **energy transfer between patches**, into which the polygons in the model are subdivided.





Radiosity, brightness and light sources

We define the **radiosity** of a patch as the amount of energy that leaves the patch per unit time per unit area.

Informally, the radiosity of a patch is a measure for its **brightness**.

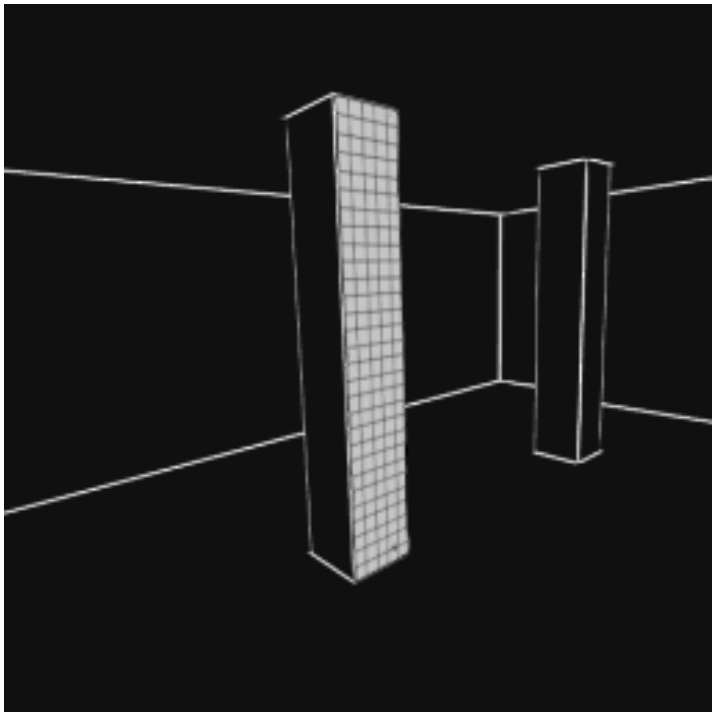
The radiosity of a patch not only depends on the energy it **emits**, but also on the energy that it **reflects**.

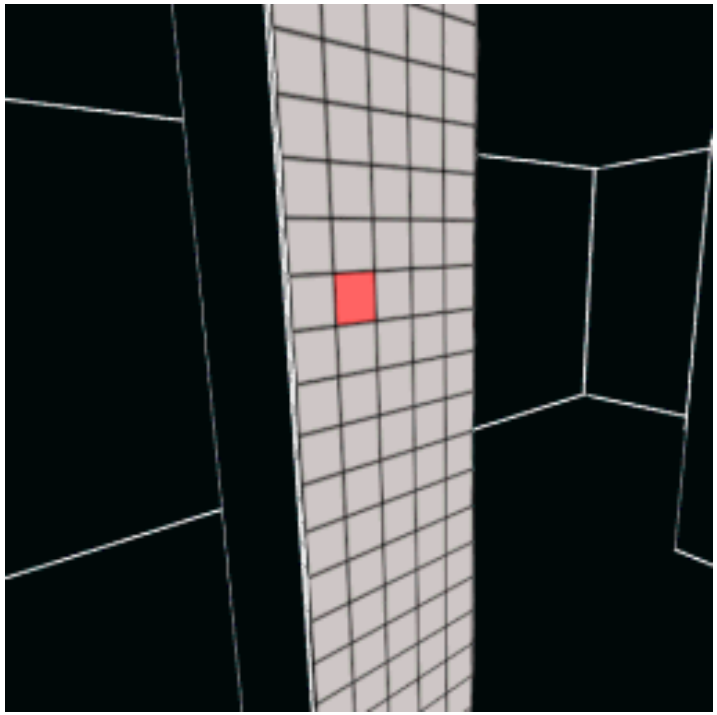
In the radiosity method there are no separate point light sources, i.e. *every patch can be a **light source***.

Patch example

We'll look at the basic idea and formulas (no mathematical details: lots of *physics* involved)

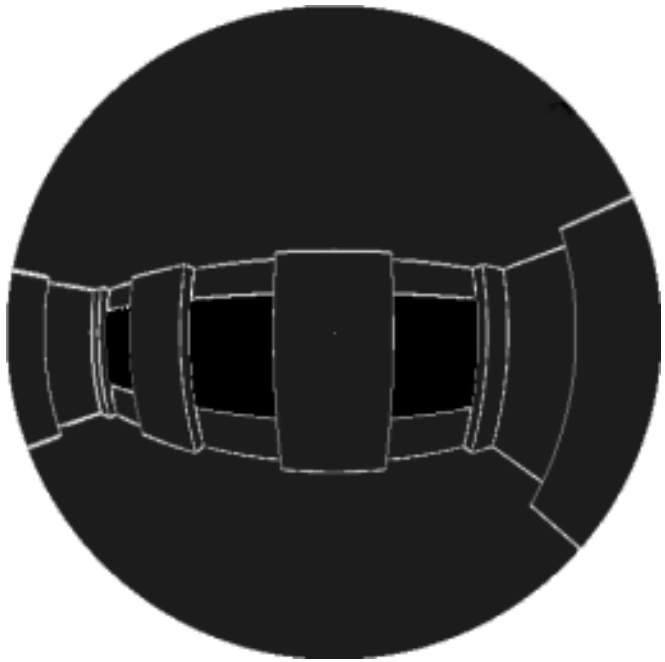
Example illustrating the basic idea of Radiosity:

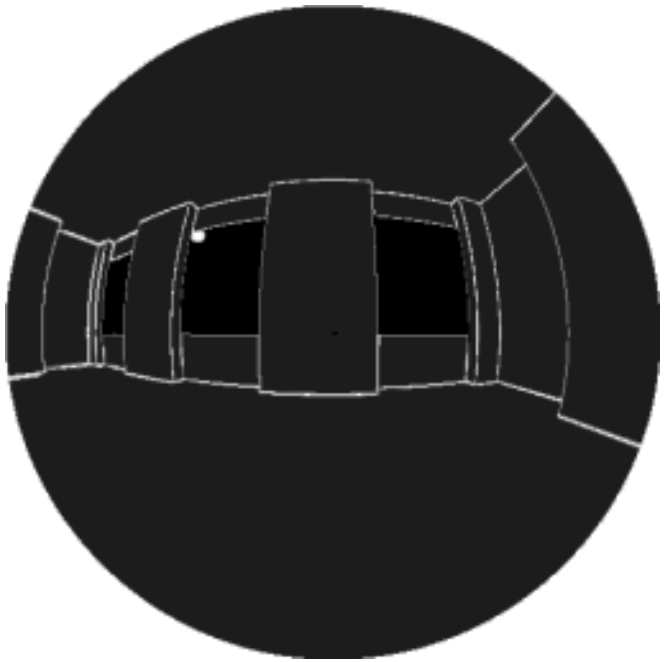




The view from a patch

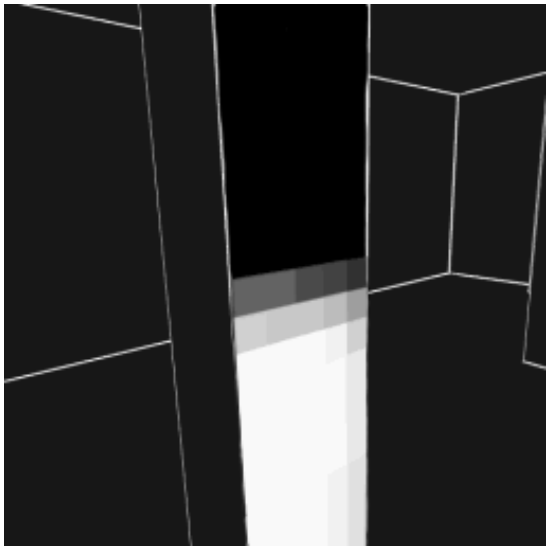
View from a patch ← → View from a lower patch

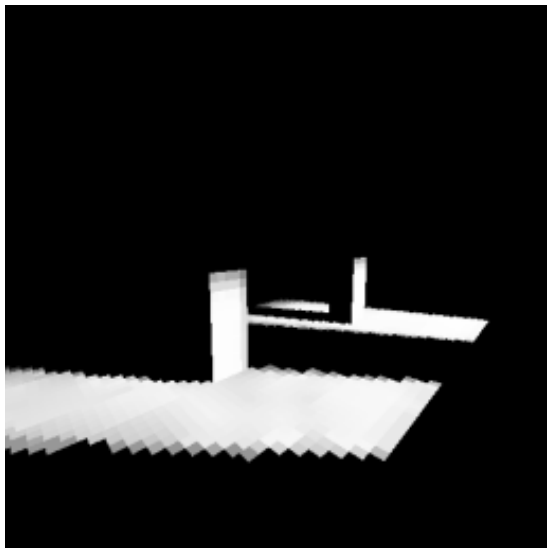




Lighting by patches

Lighting of a pillar in a room, patch by patch. Then the entire room, ... after light emission from the *direct light source* is considered.

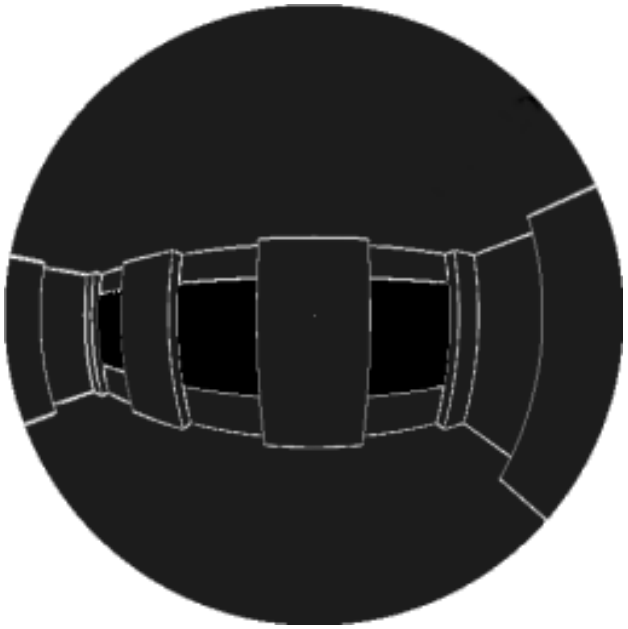


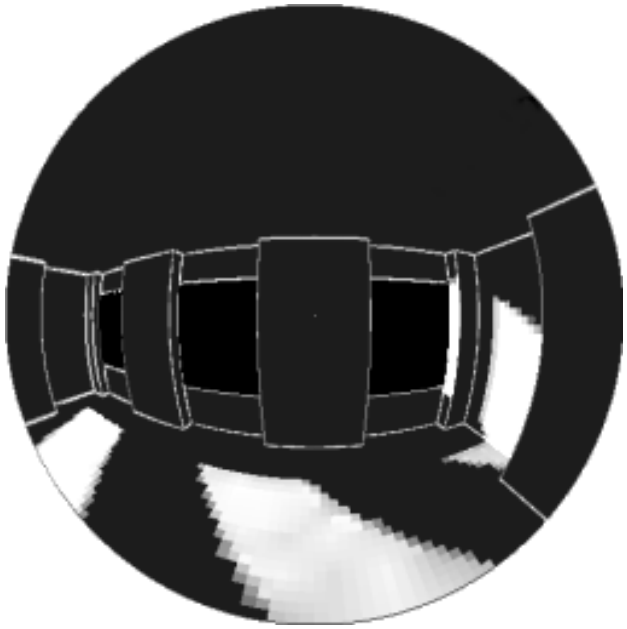


But what about reflections from lit objects?

The view from a patch, before and after

Compare the view from the upper patch on one of the pillars in the room, before -- and after ...



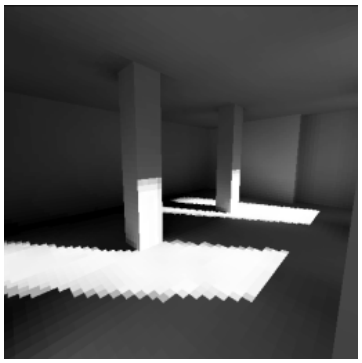
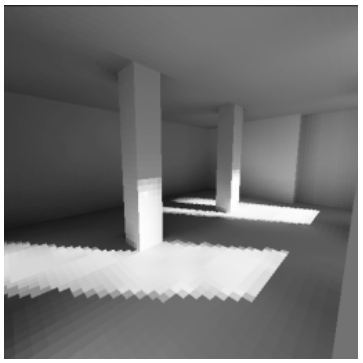
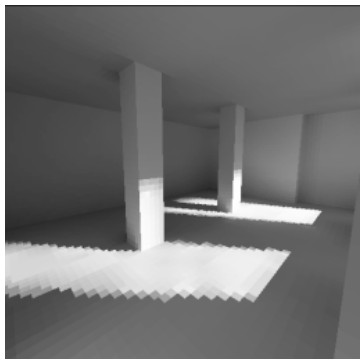
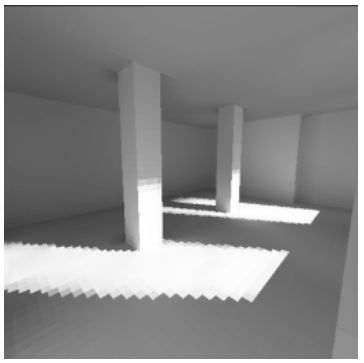


... the light emission from the direct light source is considered.

Iterative process: recompute patches.

There's more light! Let's consider that as well.

And again ... and again ... and again.

2nd pass 3rd pass 4th pass ... 16th pass

Until it converges, or until we are satisfied with the result.

Radiosity Equation

Radiosity Equation:

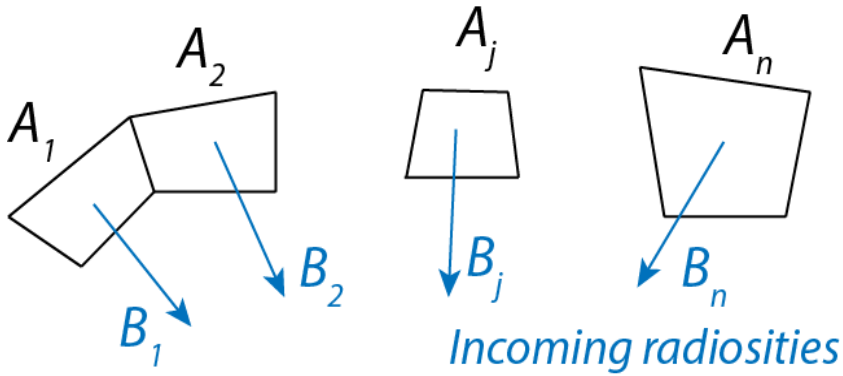
for any patch A_i ,

the radiosity: B_i ,

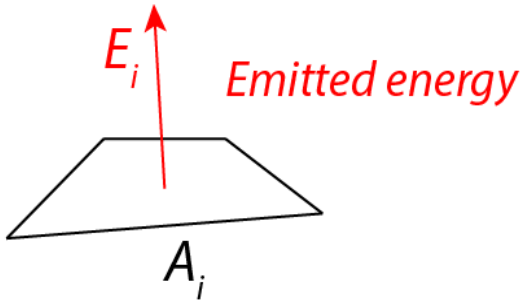
the energy the patch emits: E_i ,

and the patch's reflectivity ρ_i

we can write:



$$B_i = E_i + \rho_i \sum_j B_i F_{ij}$$



where F_{ij} is *dimensionless*

and is called the **form factor**

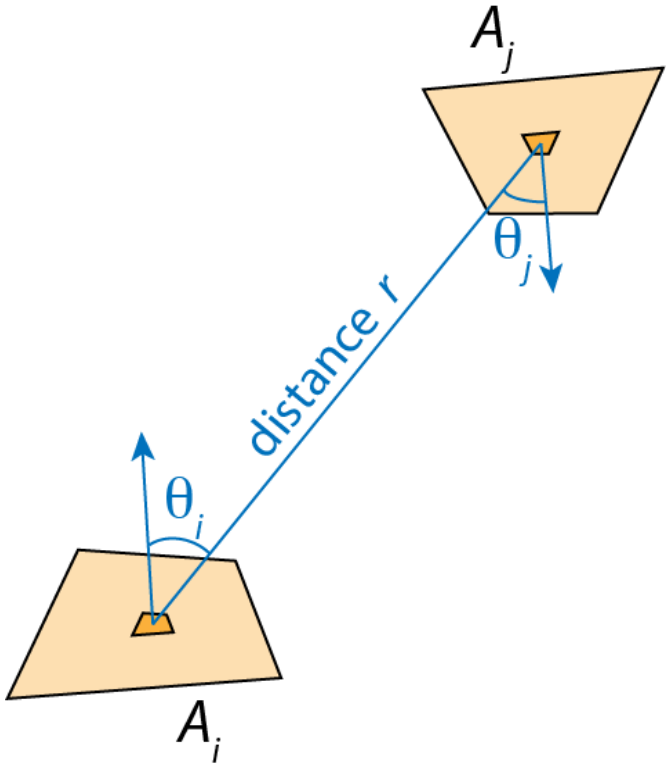
from Ai to Aj.

Form Factors

How much energy leaving A_j reaches A_i depends on:

- their relative *orientation*, modeled by product of cosines
- the distance *r* between them, modeled by the reciprocal of the squared distance times π
- the shapes of patches Ai and Aj (integral over both surfaces)

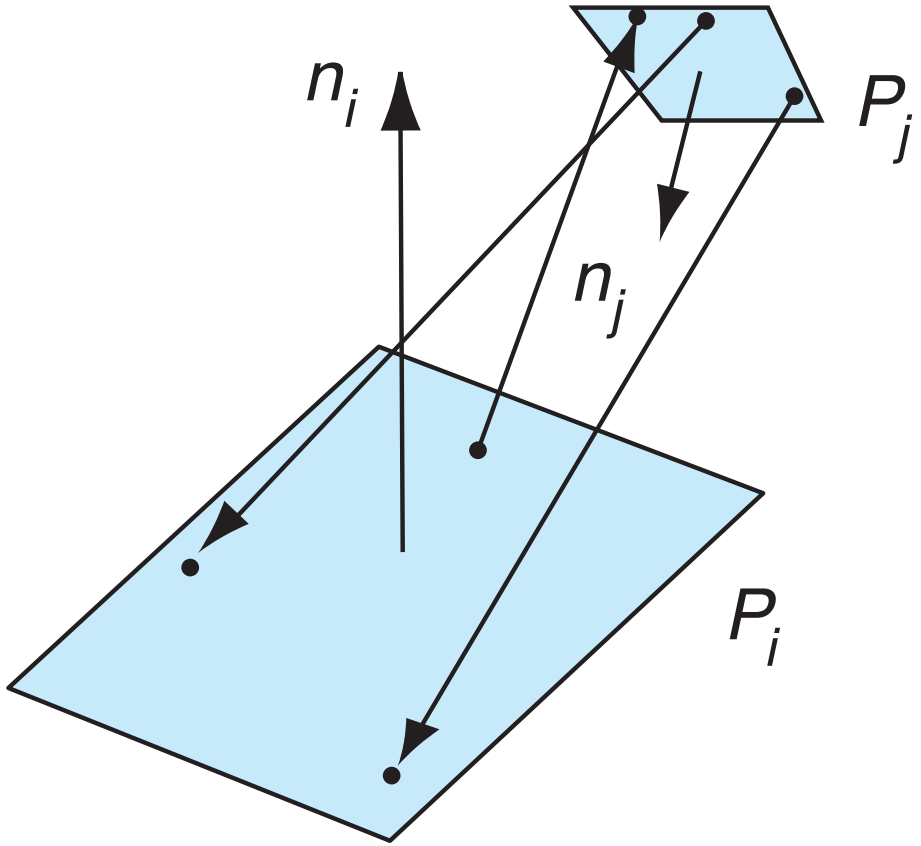
The *form factors* specify the ***fraction of the energy*** leaving one patch that arrives at the other one:



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

Computing Form Factors

Consider two flat patches



Solving the radiosity equations

There is **symmetry** in form factors:

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

$$F_{ji} = \frac{1}{A_j} \int_{A_j} \int_{A_i} \frac{\cos\theta_j \ \cos\theta_i}{\pi r^2} dA_i \, dA_j$$

So we can equate:

$$A_i F_{ij} = A_j F_{ji}$$

And the Radiosity equation can be written as:

$$B_i = E_i + \rho_i \sum_j B_j F_{ji} \frac{A_j}{A_i}$$

Solving the radiosity equations

We have two equivalent equations.

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

These are n equations in n variables. If we know the emissions E_i of all patches, and all form factors F_{ij}, we can solve for the B_i's:

$$B_i - \rho_i \sum_j B_j F_{ij} = E_i$$

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_1 F_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

Solving the radiosity equations

So, we "only" have to...

1. Compute all the form factors.

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

2. Solve the (very large) system of linear equations on the previous slide.

Remember that for each patch we have to take the sum over all patches: quadratic runtime!

Computing form factors analytically

For any pair of patches Ai and Aj we have to compute the form factor:

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

In theory this can be done analytically, but in practice this is too complicated, especially if we have to take care of partial occlusion of patches.

Number of form factors

The number of form factors is **quadratic** in the number of patches. This means that in practice, especially with large models, storing all form factors is impossible.

Therefore, form factors are (re)computed on the fly when they are needed.

Progressive refinement

Now that we know how to compute form factors, we are done halfway.

What is left is the computation of the radiosities.

Theoretically, we could solve the system:

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_1 F_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}.$$

In practice, this is too expensive, and we have to resort to approximation methods.

Progressive refinement

For each patch, we have to compute:

$$B_i = E_i + \rho_i \sum_i B_j F_{ji}.$$

We **approximate** this **iteratively**:

- Initially, set Bi = Ei for every patch.
- For every patch Ai compute the energy reaching it from all other patches and add this (multiplied by ρi) to the radiosity of patch Ai that has been computed so far.
- Repeat the previous step, but only account for the unshot radiosity that was added to each patch Aj in the previous iteration.
- Repeat until the added radiosity per iteration is less than a threshold for each patch.

Meshing

We **subdivide polygons** into patches because in general there is a smooth variation of the radiosity along a polgon. The finer the subdivision, the **more accurate** the results.

On the other hand,

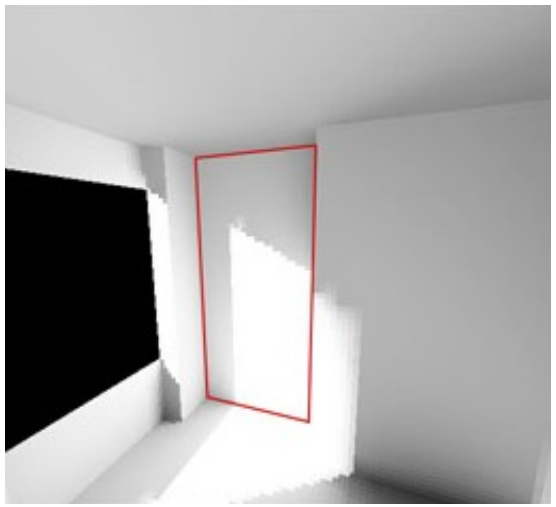
a uniform subdivision into

very small patches leads to

very long computation times.

But: a fine subdivision is

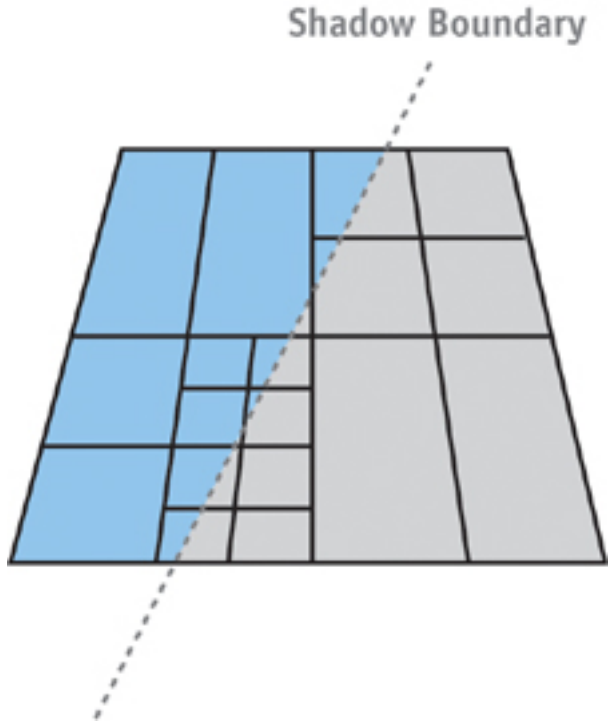
not necessary everywhere!



using Quadtrees for adaptive subdivision

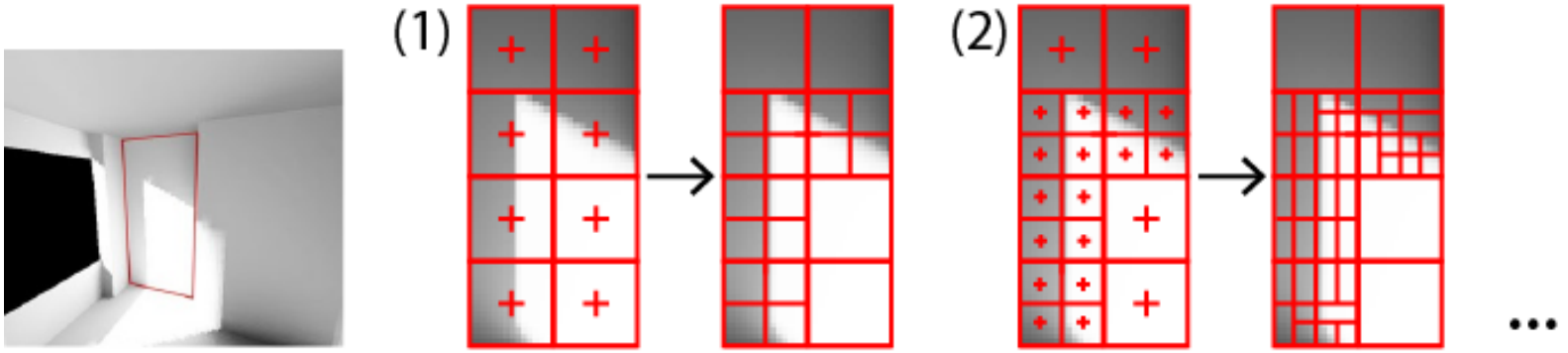
Adaptive subdivision is an iterative process:

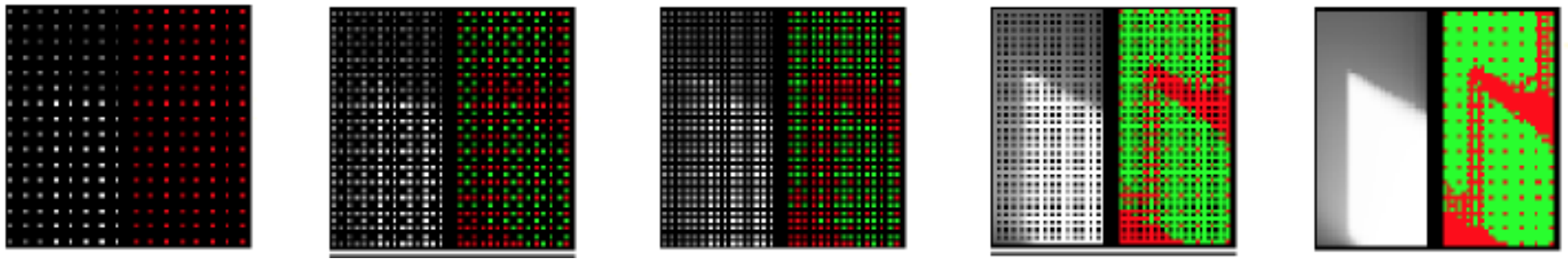
- Make an initial (coarse) subdivision of polygons into patches, and compute the radiosities.
- Check neighboring patches. If their radiosities differ more than a threshold, subdivide the patches into sub-patches
- Repeat until the radiosities of neighboring patches differs less than a threshold, or element sizes have reached a pre-determined minimum.



using Quadtrees for adaptive subdivision

Illustration of adative subdivision:





References

- Hugo Elias: *Radiosity*
  (http://freespace.virgin.net/hugo.elias/radiosity/radiosity.htm)
- E. Angel and D. Shreiner: *Interactive Computer Graphics*, Addison-Wesley
- Greg Coombe and Mark Harris: *Global Illumination Using Progressive Refinement Radiosity*, in "GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation"
  (http://dl.acm.org/citation.cfm?id=1062395)