Questions about Last Week?
Assignment 4

► You should have started by now
  ● There will not be an extension on the deadline because of department deadlines.

► Updated assignment text on the web page
  ● Clarification of how viewing conditions are specified

Overview of Week 11

► Global Illumination (Part 2)
  ● Radiosity
  ● Acceleration strategies
  ● Two-pass rendering
Ray Tracing: Review

- Ray tracing follows rays from the eye into the scene
  - At every intersection light rays are cast towards the light sources
  - Secondary rays along direction of ideal reflection and refraction
  - Works best for highly specular objects, i.e. where intensity distribution is narrowly distributed around ideal reflection and refraction
  - Uses ambient lighting to model other effects

- Difficulties modeling diffuse reflection and dispersive refraction
- In general, problems exist for all effects where indirect light from several directions contributes to a ray's intensity

Radiosity: Motivation (1)

- Scene geometry (sculpture)
  - No light can be traced from behind the sculpture to the viewer
  - Colored surfaces facing backward, white faces towards the viewer
**Radiosity: Motivation (2)**

- **Ray-Traced Image**
  - Flat appearance
  - No color reflecting to the eye

- **Radiosity Image**
  - Color bleeds through due to diffuse reflection

---

**Radiosity: Overview**

- **To alleviate these shortcomings radiosity determines intensities within the scene by modeling the energy exchange between objects**
  - Eliminates the need for the crude concept of "ambient light"
  - Based on thermal engineering methods for determining the reflection and emission of radiation
  - Accounts for energy received + emitted by a purely diffuse surfaces
  - Based on preservation of light emission and absorption in a static, closed environment
    - No time-varying scenes, i.e. lights have constant brightness and objects don't move
    - Not suited for outdoor scenes
  - View-independent, since the energy exchange does not depend on the position of the viewer.
    - Rendering only requires hidden surface removal
Radiosity: Process

- Divide the scene into surface patches
- Compute the energy emitted and received by each patch taking into account energy exchange between patches
  - Form factor computation to determine "coupling" between patches
  - Solving of simultaneous equations to compute steady state
- Assign light intensity (color) to each patch based on the energy leaving a patch
- Render the scene using flat shading or Gouraud shading

Radiosity: Definition

- Radiosity
  - Rate at which energy leaves a surfaces
  - Measured as energy per unit-time and unit-area, e.g. W/m²
  - We have to consider rates - not absolute quantities - as otherwise the energy level in a scene would increase over time.
Radiosity of a Patch (1)

- **Patch is a surface element in the scene**

- **Radiosity** a patch is the combination of the emitted radiosity and the radiosity received from another patch:

  \[ B_i = E_i + \rho \sum_{j=1}^{n} B_j \cdot F_{j-i} \cdot \frac{A_j}{A_i} \]

  - \( B_i \): Radiosity of patch \( i \) [W/m\(^2\)]
  - \( E_i \): Emitted radiosity [W/m\(^2\)]
  - \( \rho \): Reflectivity of patch \( i \) [dimensionless]
  - \( F_{j-i} \): Form factor between patches \( i \) and \( j \). [dimensionless]
    - Describes how much energy leaving patch \( j \) arrives on patch \( i \).
    - For instance to describe blocking objects.
  - \( A_i \): Area of patch \( i \) [m\(^2\)]

Radiosity of a Patch (2)

- **Geometric Interpretation**
  - \( E_i \) is the radiosity emitted by patch \( i \)
  - \( A, B \) is the total power leaving patch \( j \)
  - \( A, B, F_{j-i} \) is the power from patch \( j \) reaching patch \( i \)
  - \((A, B, F_{j-i} / A_i)\) is the radiosity of patch \( i \) due to patch \( j \)
  - \( \rho_i (A, B, F_{j-i} / A) \) is the radiosity reflected by patch \( i \)
Radiosity between Patches (1)

- Percentage of visible area is the same for both patches

\[ A_i \cdot F_{i-j} = A_j \cdot F_{j-i} \implies F_{i-j} = F_{j-i} \cdot \frac{A_j}{A_i} \]

- Therefore the radiosity equation simplifies to

\[ B_i = E_i + \rho \sum_{j=1}^{n} B_j \cdot F_{i-j} \]

- Solving for the emitted energy gives

\[ B_i - \rho \sum_{j=1}^{n} B_j \cdot F_{i-j} = E_i \]

Radiosity between Patches (2)

- Combining the radiosity expressions for all \( n \) patches results in a system of simultaneous equations:

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

- The solution of this equation describes the steady-state exchange of radiosity within the scene

- Note that form factors \( F_{ij} \) are not always 0, e.g. for concave surfaces. Therefore, the matrix diagonals are not always 1.
Radiosity between Patches (3)

- Solving this system of equations is often impractical
  - The sheer size of the matrix makes inversion very expensive
  - The matrix has $n \times n$ elements.
    - In practice, there are several thousands or tens of thousands of patches!
  - This leads to storage and access problems (page faults etc.)
  - Gaussian methods for matrix inversion are typically $O(n^3)$!!

- Therefore, iterative methods have been developed
  - Start with an approximation of the solution
  - Refine until an error bound is not exceeded
  - Several iterative solution methods are available, e.g. Gauss-Seidel, Jacobi, Southwell or Overrelaxation

Matrix Equations (1)

- Generally the radiosity problem is described by a matrix equation in the form factor matrix $K$, the radiosity vector $B$ and the emitter vector $E$:
  \[ K \cdot B = E \]

- We will iterate to generate a series of approximations to $B$. Each iteration's result will be numbered: $B^{(g)}$
  - Residual is the difference between iterations:
    \[ r^{(g)} = E - K \cdot B^{(g)} \]
  - We will not consider issues related to stability or convergence
Matrix Equations (2)

- Relaxation methods adjust the result \( B^{(g)} \) of iteration \( g \) to force one (!) element of \( r^{(g+1)} \) to zero.
  - This will obviously affect the other values of \( r \), but over many iterations the system will converge to the correct solution.
  - Element \( i \) of \( r \) should be forced to zero with a different value for \( B \):

\[
 r^{(g)}_i = E_i - \sum_{k=1}^{n} K_{i,k} \cdot B^{(g)}_k = E_i - K_{i,i} \cdot B^{(g)}_i - \sum_{k=1}^{n} K_{i,k} \cdot B^{(g)}_k = 0
\]

\[
 B^{(g+1)}_i = \frac{E_i}{K_{i,i}} - \sum_{k=1}^{n} \frac{K_{i,k}}{K_{i,i}} K_{i,k} \cdot B^{(g)}_k
\]

Matrix Equations (3)

- Now the new value of \( B \) must be expressed in terms of its previous value:

\[
 r^{(g)}_i = E_i - \sum_{k=1}^{n} K_{i,k} \cdot B^{(g)}_k = E_i - K_{i,i} \cdot B^{(g)}_i - \sum_{k=1}^{n} K_{i,k} \cdot B^{(g)}_k
\]

\[
 \Rightarrow \frac{r^{(g)}_i}{K_{i,i}} + B^{(g)}_i = \frac{E_i}{K_{i,i}} - \sum_{k=1}^{n} \frac{K_{i,k}}{K_{i,i}} K_{i,k} \cdot B^{(g)}_k
\]

\[
 \Rightarrow B^{(g+1)}_i = \frac{E_i}{K_{i,i}} - \sum_{k=1}^{n} \frac{K_{i,k}}{K_{i,i}} K_{i,k} \cdot B^{(g)}_k = B^{(g)}_i + \frac{r^{(g)}_i}{K_{i,i}}
\]
Solving Radiosity Matrices

The radiosity method describes how energy is exchanged or distributed between patches in the scene.

- Iterative methods solve the problem by refining the energy distribution.
- We distinguish between *shot* energy and *unshot* energy.
- The patch radiosity is the distributed, i.e. shot, energy.
- The difference between the emitted radiosity and the reflected radiosity is unshot energy.
- The residual $r$ is a measure for the unshot energy, i.e. how much more/less energy the patches should be distributing into the scene.

Shot and Unshot Radiosity

For a given iteration $g$:
- Assume $E_i = 0$.
- $B_i$ is the radiosity currently sent into the environment.
- $B_i$ is the total radiosity received from other patches.
- $p_i B_i$ is the radiosity available for distribution into the scene.
- The difference between $B_i$ and $p_i B_i$ is "unshot" radiosity.
  - Positive unshot radiosity: more incident radiosity than emitted.
  - Negative unshot radiosity: less incident radiosity than emitted.
Jacobi Iteration (1)

- This iterative method is the Jacobi Iteration
  - Initial value of all patch radiosities $B_i$ is simply Zero or their emissive radiosity $E_i$
  - All patch radiosities are updated in each iteration; this is fairly costly.
  - Convergence criteria include thresholds for residual or for change in $B_i$, e.g.
    $$\left|B_i^{(g)} - B_i^{(g+1)}\right| < T$$

```
B = E ; // Initialization

for i = 1 to n
    r = E - K * B;
    B_i = B_i + r_i / K_{i,i}

while (max|r[i]| > T)
```

Jacobi Iteration (2)

- Physical interpretation
  - The radiosity of all patches is adjusted to represent the unshot energy

- This method is rarely used for iterative approximation
  - At least initially, only few patches have significant amounts of unshot radiosity. Processing the other patches is wasteful.
  - Sometimes a Jacobi step is the last step in an iteration to distribute the remaining unshot radiosity.
Gauss-Seidel Iteration (1)

- Variation on Jacobi Iteration
  - Jacobi method uses the new values $B_{i(g+1)}$ in the next iteration.
  - Instead Gauss-Seidel uses them immediately for computing the next new $B_i$.
  - This is somewhat more efficient than the Jacobi method.
  - Still all patch radiosities are update in every step.

```plaintext
B = E ;  // Initialization

do
{ for (i = 0 ;  i < n ;  i ++)
  $B_i = B_i - \sum_{k=1}^{n} \frac{B_k \cdot K_{i,k}}{K_{i,i}}$
}
while (\exists |r[i]| > T)
```

Gauss-Seidel Iteration (2)

- Physical Interpretation
  - For each patch all the radiosity from the other patches is collected.
  - This process is applied to all patches in sequence.

\[
B_i = E_i + \sum_{k=1 \atop k \neq i}^{n} K_{i,k} B_k = E_i + \rho_i \sum_{k=1 \atop k \neq i}^{n} F_{i,k} B_k
\]

\[
B_i A_i = E_i A_i + \rho_i \sum_{k=1 \atop k \neq i}^{n} F_{i,k} B_k A_i
\]

\[
F_{i,k} A_i = F_{k,i} A_k
\]
Southwell Iteration (1)

- **Another variation of Jacobi method**
  - Adjusts only the patch radiosity $B_i$ with the largest residual $r_i$.
  - This means that the algorithm focusses better on those patches with the most incorrect radiosity.
    - Jacobi and Gauss-Seidel would have adjusted that radiosity only every $n$-th step!
  - However, to adjust an element repeatedly, the residual vector must be updated in between. This can be done incrementally:

\[
\begin{align*}
B_i^{(g+1)} &= B_i^{(g)} + \frac{r_i^{(g)}}{K_{i,j}} = B_i^{(g)} + \Delta B_i^{(g)} \\
B_i^{(g+1)} &= B_i^{(g)} + \Delta B_i^{(g)} \\
\end{align*}
\]

\[
\begin{align*}
\text{r}^{(g+1)} &= E - K \cdot B^{(g+1)} \\
&= E - K \cdot (B^{(g)} + \Delta B^{(g)}) \\
&= E - K \cdot B^{(g)} - K \cdot \Delta B^{(g)} \\
&= r^{(g)} - K \cdot \Delta B^{(g)} \\
\end{align*}
\]

Southwell Iteration (2)

- **Observation:**
  - Only one element is updated, therefore:

\[
\Delta B_k^{(g)} = 0 \quad \forall \ k \neq i
\]
  - The residual vector can therefore be updated by only considering column $i$ of $K$:

\[
\begin{align*}
r_k^{(g+1)} &= r_k^{(g)} - K_{k,i} \cdot \Delta B_i^{(g)} \\
\implies
\frac{r_k^{(g+1)}}{r_k^{(g)}} &= 1 - \frac{K_{k,i}}{K_{i,j}} \cdot \frac{r_k^{(g)}}{r_k^{(g)}} \\
\end{align*}
\]
Southwell Iteration (3)

- **Initialization:**
  \[ r^{(0)} = E - K \cdot B^{(0)} = E - K \cdot 0 = E \]

  - Converges faster than Jacobi or Gauss-Seidel

  ```
  B = 0; // Initialization
  r = E;
  do
  { select i such that \( r_i = \max(r) \);
    \[ B_i = B_i + \frac{r_i}{K_{i,j}} \]
    \[ t = r_i \]
    \[ \forall k: r_k = r_k - t \cdot \frac{K_{k,j}}{K_{i,j}} \]
  } while (\( \exists |r_i| > T \))
  ```

Southwell Iteration (4)

- **Physical interpretation**
  - The iteration relaxes the element with the largest residual, i.e. the largest unshot radiosity
  - This means, the radiosity of this element is distributed into the environment. (Initially, this element is typically a light source.)

  - Different from Gauss-Seidel in that the radiosity is *shot* into the environment instead of being *gathered*.
Overrelaxation (1)

- Applicable to all iteration methods discussed

- Based on observation that radiosities $B_i$ are adjusted several times before the residual is sufficiently small
  - Instead overrelax the radiosities, i.e. adjust more than indicated by the Jacobi correction.
  - Then the new residuals are no longer zero.

\[
B_i^{(g+1)} = B_i^{(g)} + k \cdot \Delta B_i^{(g)}
\]

\[
\Rightarrow
r_i^{(g+1)} = (1 - k) \cdot r_i^{(g)}
\]

- $k>1$: Overrelaxation
- $k<1$: Underrelaxation (useful for unstable systems)

Overrelaxation (2)

- Physical interpretation
  - Overrelaxation overshoots a patch's energy in the expectation that in future iterations the patch will receive energy from other patches.
Progressive Refinement

- Variation on Soutwell iteration
  - Select the element with the maximum unshot power instead of unshot radiosity, i.e. \( A_i \Delta B_i \)
  - Picks the main contributors of power, instead of the elements with maximal radiosity, i.e. power density.

- Additional refinements in progressive radiosity
  - On-demand computation of form factors reduces storage required for form factor matrix
  - Estimation of ambient lighting, accounting for the unshot radiosity

Estimated Ambient Lighting

- Estimating the unshot radiosity
  - Sum of the residual values is the unshot radiosity: \( \Delta B = \frac{\sum \Delta B_i A_i}{\sum A_i} \)
  - Also compute an average reflectivity: \( \overline{\rho} = \frac{\sum \rho_i A_i}{\sum A_i} \)
  - Releasing the unshot energy into the scene and reflecting it repeatedly off all objects will create a new ambient radiosity:
    \[
    B_A = \Delta B \cdot \left( 1 + \frac{1}{\overline{\rho}} + \frac{1}{\overline{\rho}^2} + \frac{1}{\overline{\rho}^3} + \ldots \right) = \frac{\Delta B}{1 - \overline{\rho}}
    \]
  - The ambient radiosity can be added to all patches
    \( B'_i = B_i + \rho_i B_A \)
  - This result in faster convergence, because all patches receive some energy immediately instead of only after many iterations.
Form Factors

- Remember: $F_{ij}$ is the portion of energy leaving patch $i$ reaching patch $j$

- Since form factors are central to all radiosity methods we will consider several methods to compute them
  - Analytical computation
  - Measure them using physical experiments
  - Approximation

Form Factors: Analytical Computation (1)

- Consider infinitesimally small surface elements $dA_i$ and $dA_j$
  - $H_{ij}$ is 1 if $dA_i$ is visible from $dA_j$ and 0 otherwise
  - $dF_{di-dj}$ is called the differential form factor

$$
 dF_{di-dj} = \frac{\cos \theta_i \cdot \cos \theta_j}{\pi r^2} \cdot H_{ij} \cdot dA_j
$$
Form Factors: Analytical Computation (2)

To determine the differential-to-finite form factor from a surface element $dA_i$ to the entire patch $j$, we integrate over all the surface elements on patch $j$:

$$F_{dA_i-j} = \int_{A_j} \frac{\cos\theta_i \cdot \cos\theta_j}{\pi r^2} \cdot H_{ij} \cdot dA_j$$

The average finite form factor between the patches is the average of all the point form factors on patch $i$:

$$F_{i-j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cdot \cos\theta_j}{\pi r^2} \cdot H_{ij} \cdot dA_j \cdot dA_i$$

- Differential-to-finite form factor can approximate the finite form factor.

Form Factors: Analytical Computation (3)

- Analytical solution of this area integral is often complicated
  - Closed solutions exist for simple geometries
  - More complex geometries may not have a closed form solution

- Conversion of the area integrals to contour integrals simplifies the computation
  - Stoke's theorem relates integrals over an area to integrals over the area's contour
  - Requires that patches are completely visible to each other
Form Factors: Hemisphere Method (1)

The differential-to-finite form factor can be computed using two projection steps:

- Project $A_i$ onto a hemisphere with radius $R=1$, centered at $dA_i$: $A_i'$

- Then project $A_i'$ onto the base of the hemisphere: $A_i''$

- Then, the form factor $F_{d_{i,j}}$ is the ratio of $A_i''$ to the base of the hemisphere.

Form Factors: Hemisphere Method (2)

- Project $A_j$ to $A_j'$
  
  - Project of $dA_i$ to $dA_i'$
  
  \[ dA_j' = dA_j \cdot \cos \theta_j / r^2 \]

  - This is the solid angle $d\omega_j$ subtended by $dA_i'$
  
  \[ d\omega_j = dA_j' \]

  - Now project $dA_j'$ onto the base of the hemisphere
  
  \[ dA_j'' = dA_j' \cdot \cos \theta_i \]

  - Pulling everything together:
  
  \[ dA_j'' = dA_j \cdot \frac{\cos \theta_i \cdot \cos \theta_j}{r^2} \]
Form Factors: Hemisphere Method (3)

- The differential form factor is the portion of the hemisphere base covered by $dA_i$:
  - Assume hemisphere with radius 1, i.e. area $= \pi$

$$F_{d_i-j} = \frac{dA_j''}{\pi} = dA_j \cdot \frac{\cos \theta_i \cdot \cos \theta_j}{\pi r^2}$$

- The differential-finite form factor is computed by integrating of $A_j$

$$F_{d_i-j} = \int_{A_j} \frac{\cos \theta_i \cdot \cos \theta_j}{\pi r^2} \cdot dA_j$$

Form Factors: Solid Angle

- Concept
  - From a given viewpoint, an object covers a certain area of the view
  - This is the solid angle

- Definition
  - 3D analog to a 2D angle measured in radians [rad]
  - The solid angle subtended by an object is the area of its projection onto a unit sphere
  - Measured in steradian [sr].
  - Applies to all types of objects
**Form Factors: Projection Methods**

- Recall that
  \[ d\omega_j = dA_j' \]
- The total solid angle is
  \[ \omega_j = \sum \Delta\omega_j \]
- Precompute a differential form factor \( F_{di,\Delta\omega} \) for a given \( \Delta\omega_j \)
  - The \( \Delta\omega \) do not overlap.
  - Hence, the differential-finite form factor is:
    \[ F_{di-j} \approx \sum F_{di,\Delta\omega_j} \]

**Form Factors: Hemi-Cube (1)**

- Computing form factors and solid angles for a hemisphere is costly
- Use a hemicube instead of the hemisphere
  - Hemicube is centered around the patch
  - Base of the cube parallel to the patch
  - Simpler because of planar faces
- Further simplify solid angle calculation by subdividing the faces of the cube with a grid
  - Each grid cell approximates a solid angle
  - Form-factors are precomputed for each grid cell
  - Typical grid resolutions range from several ten to several hundred
Form Factors: Hemi-Cube (2)

Form Factors: Hemi-Cube (3)

Form factor for cell $P$:

$$\Delta F_p = \frac{\cos \theta_i \cdot \cos \theta_j}{\pi \cdot r_p^2} \cdot \Delta A$$

Top Face:

$$r_p = \sqrt{x_p^2 + y_p^2 + 1}; \quad \cos \theta_i = \cos \theta_j = \frac{1}{r_p}$$

$$\Rightarrow \Delta F_p = \frac{\Delta A}{\pi \left(x_p^2 + y_p^2 + 1\right)^2}$$

Side Faces:

$$r_p = \sqrt{x_p^2 + y_p^2 + 1}; \quad \cos \theta_i = \frac{z_p}{r_p}; \quad \cos \theta_j = \frac{1}{r_p}$$

$$\Rightarrow \Delta F_p = \frac{\Delta A \cdot z_p}{\pi \left(x_p^2 + y_p^2 + 1\right)^2}$$
**Form Factors: Hemi-Cube (4)**

- **Advantages**
  - Use standard polygon rendering techniques to determine the grid cells covered
  - Hardware z-buffering makes this process fast
  - Applicable to different primitive types

- **Basic assumption**: Form factor at the patch center is representative for the entire patch

- **Problems**
  - Proximity: Form factors change noticeably if patches are close
  - Visibility: Form factors vary over the patch in the presence of occluders
  - Aliasing

**Form Factors: Hemi-Cube (5)**

- **Proximity Assumption**
  - The distance between patches is large compared to the patch sizes

- If this assumption is not valid, the form factor is not (even roughly) constant across the patch
  - Form factor is different for the 3 hemicube positions
Form Factors: Hemi-Cube (6)

Visibility Assumption

- If patch \( j \) is visible from the center of patch \( i \), then patch \( j \) is visible from everywhere on patch \( i \).

- This assumption is violated when objects are in between the patches.

Form Factors: Hemi-Cube (7)

Aliasing Assumption

- The sampling frequency defined by the hemicube grid samples all objects in the environment correctly.

In general, this assumption is not valid

- Therefore, sampling artifacts (aliasing) are to be expected as discussed earlier, e.g.
  - Incorrect form factors
  - Staircasing
  - Missed or incompletely sampled objects. This can very noticeable for a large object lit by a distant light source.
Form Factors: Other Sampling Surfaces

- Instead of using the gridded hemicube other surfaces can be used

- Large (infinite) plane parallel to the patch
  - Plane is adaptively subdivided, similar to quadtrees
  - May miss distant objects along the horizon. However, these objects typically do not contribute much to a patch's radiosity.

- Gridded hemisphere
  - Similar to the gridded hemicube, e.g. along latitude and longitude
  - Difficult to scan-convert to a spherical surface

- Gridded hemisphere base
  - Requires fewer cells than a gridded hemicube with comparably accurate form factors
  - Requires efficient method to project objects onto the hemisphere

Form Factors: Adaptive Subdivision

- What if the basic assumptions for computing form factor (proximity, visibility, aliasing) are violated?
  - Can be traced back to large a variation of radiosity over the patch
  - Subdividing the patch into smaller patches can reduce the error

- In order to contain the added complexity, patches are subdivided adaptively
  - Subdivide if difference between radiosity on patch vertices exceeds a given threshold
  - Regular subdivision, e.g. divide the patch into n smaller patches
  - Discontinuity meshing adapts the subdivision to follow features that cause the violation of the assumptions
    - For instance subdivide along an edge creating a shadow.
  - Requires computation of new form factors for the new sub-patches
Rendering Radiosity Solutions

- Radiosities have been computed per patch
  - Translating radiosities into color results in flat shaded images
  - Smooth shaded images (Gouraud shading) can be rendered by computing vertex radiosities as averages of the radiosities of the adjacent faces
  - Details in the textbook!

- Radiosity solutions are view-independent
  - Radiosity method assumes perfectly diffuse materials
  - Therefore, there is no directional preference in how light is reflected
  - Radiosity solution must be computed once for all viewpoints
  - Once computed, walkthroughs can be performed at the speed of regular Gouraud shading!

Radiosity Examples: "Cornell Box"

- Radiosity simulation
  - Photo of the real box
    - Controlled lighting conditions

(C) Cornell University
(C) Cornell University
Radiosity Examples: Factory

In 1988, on a VAX8700:
- 30,000 patches
- Radiosity solution: 5 hrs
- Rendering: 190 hrs

Ray-Tracing and Radiosity

- Ray-Tracing
  - View-dependent and specular reflection only
    - Lighting model can simulate diffuse and ambient lighting

- Radiosity
  - View-independent and diffuse reflection only

- Radiosity and Ray-Tracing can be combined to capture both diffuse and specular effects
  - First pass: view-independent radiosity solution with extra information about specular distribution of light
  - Second pass: view-dependent ray-tracing

- A bit more detail in the textbook
Summary

- Radiosity simulates the power distribution in the scene
  - Simulates exchange of light between patches
  - Exchange is determined by the amount of light received from other patches and the reflectivity of patches.
  - Only applies to purely diffuse environments. View-independent.
  
  - 1. Determine coupling between patches by computing form factors
  - 2. Solve simultaneous equations for the steady state of the system.
  
  - Form factor computation can be accelerated with z-buffer hardware
  - Radiosity solution can be approximated using iterative methods

Homework

- Study radiosity (chapter 16.13)
  - For further study see A. Glassner, Principles of Digital Image Synthesis (Vol. 2), Morgan Kaufman.

- Prepare color theory (chapter 13) and graphics hardware (chapter 4.1-5)
Next Week ...

- Graphics Hardware
- Color Theory

Diagram of electron beam with labels:
- Electron Beam
- Cathode
- Collector Electrode
- Control Grid
- Vertical Deflection System
- Horizontal Deflection System
- Phosphor
- Electrode

Color wheel with labels:
- Cyan
- Blue
- Magenta
- Red
- Yellow
- White
- Green
- Black
- S
- H
- V