
CS482: Radiosity

Sung-Eui Yoon
(윤성의)

Course URL:
<http://sglab.kaist.ac.kr/~sungeui/ICG>

Class Objective (Ch. 11)

- **Understand radiosity**
 - Radiosity equation
 - Solving the equation

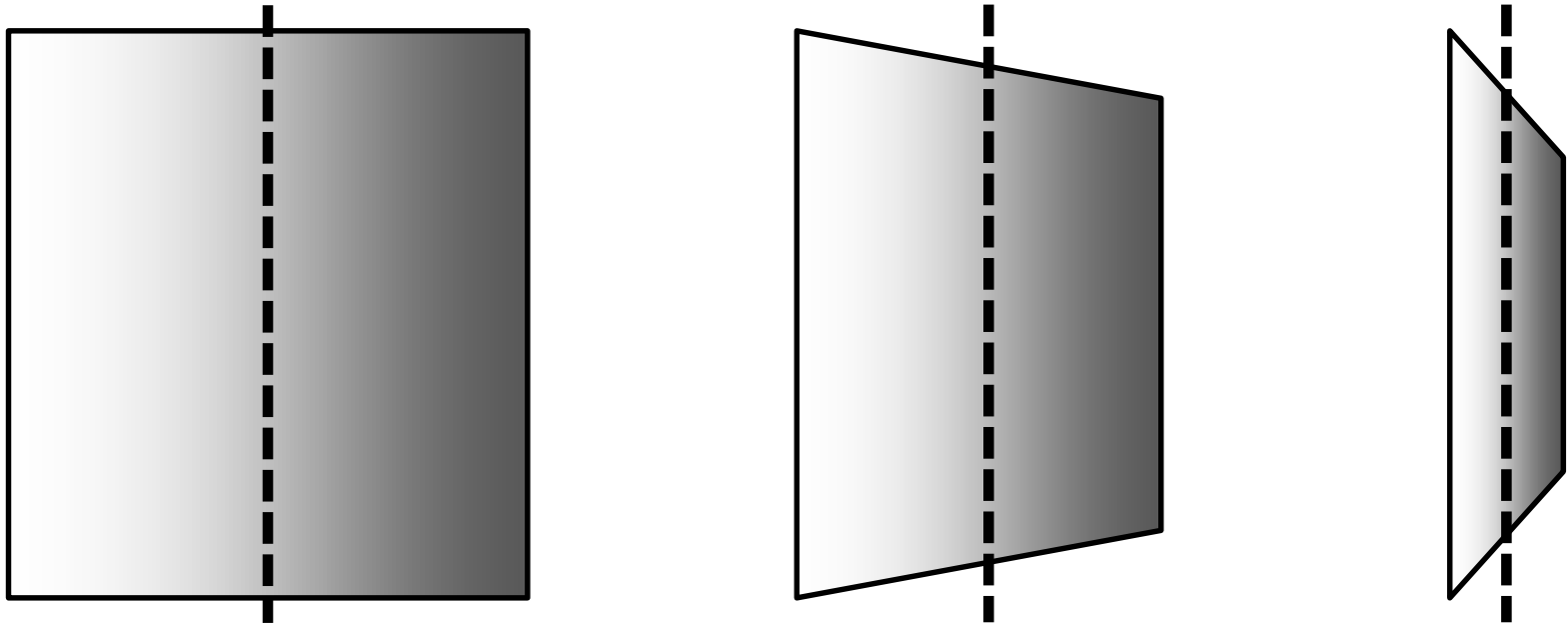
History

- **Problems with classic ray tracing**
 - Not realistic
 - View-dependent
- **Radiosity (1984)**
 - Global illumination in diffuse scenes
- **Monte Carlo ray tracing (1986)**
 - Global illumination for any environment

Radiosity

- **Physically based method for diffuse environments**
 - **Support diffuse interactions, color bleeding, indirect lighting and penumbra**
 - **Account for very high percentage of total energy transfer**
 - **Finite element method**

Key Idea #1: Diffuse Only

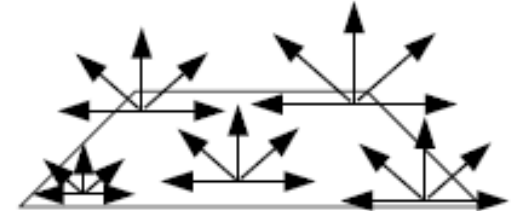


- **Radiance independent of direction**
 - Surface looks the same from any viewpoint
 - No specular reflection

Diffuse Surfaces

- **Diffuse emitter**

- $L(x \rightarrow \Theta) = \text{constant over } \Theta$



- **Diffuse reflector**

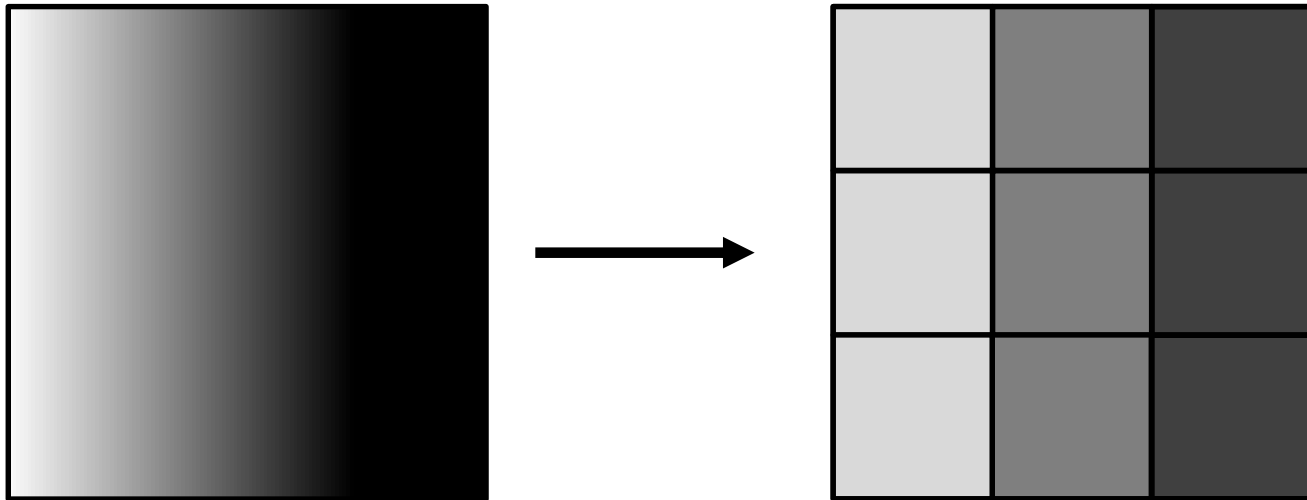
- **Constant reflectivity**



From kavita's slides

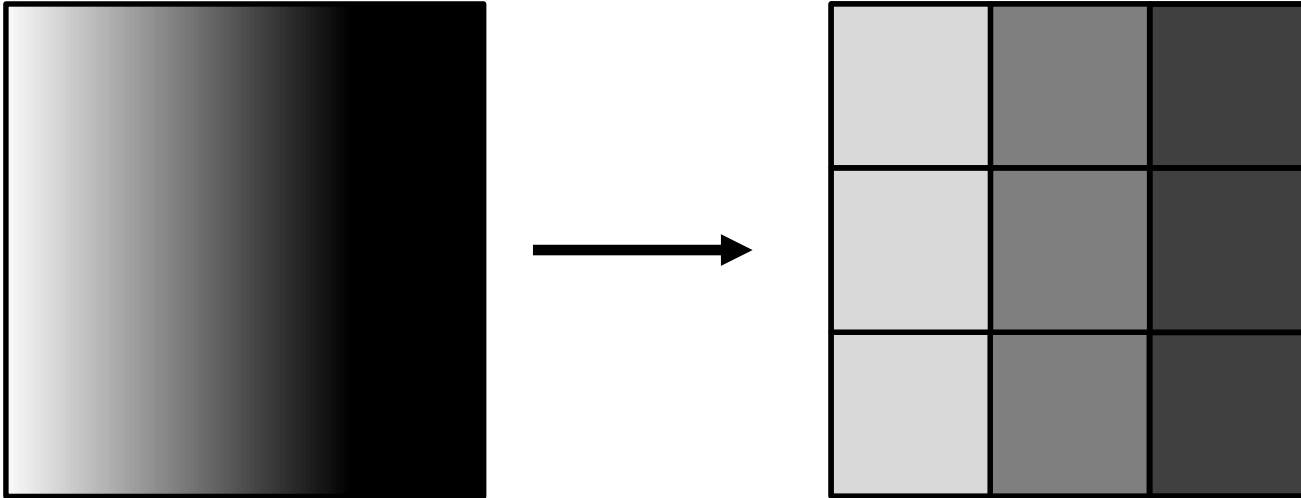
Key Idea #2: Constant Polygons

- **Radiosity is an approximation**
 - **Due to discretization of scene into patches**



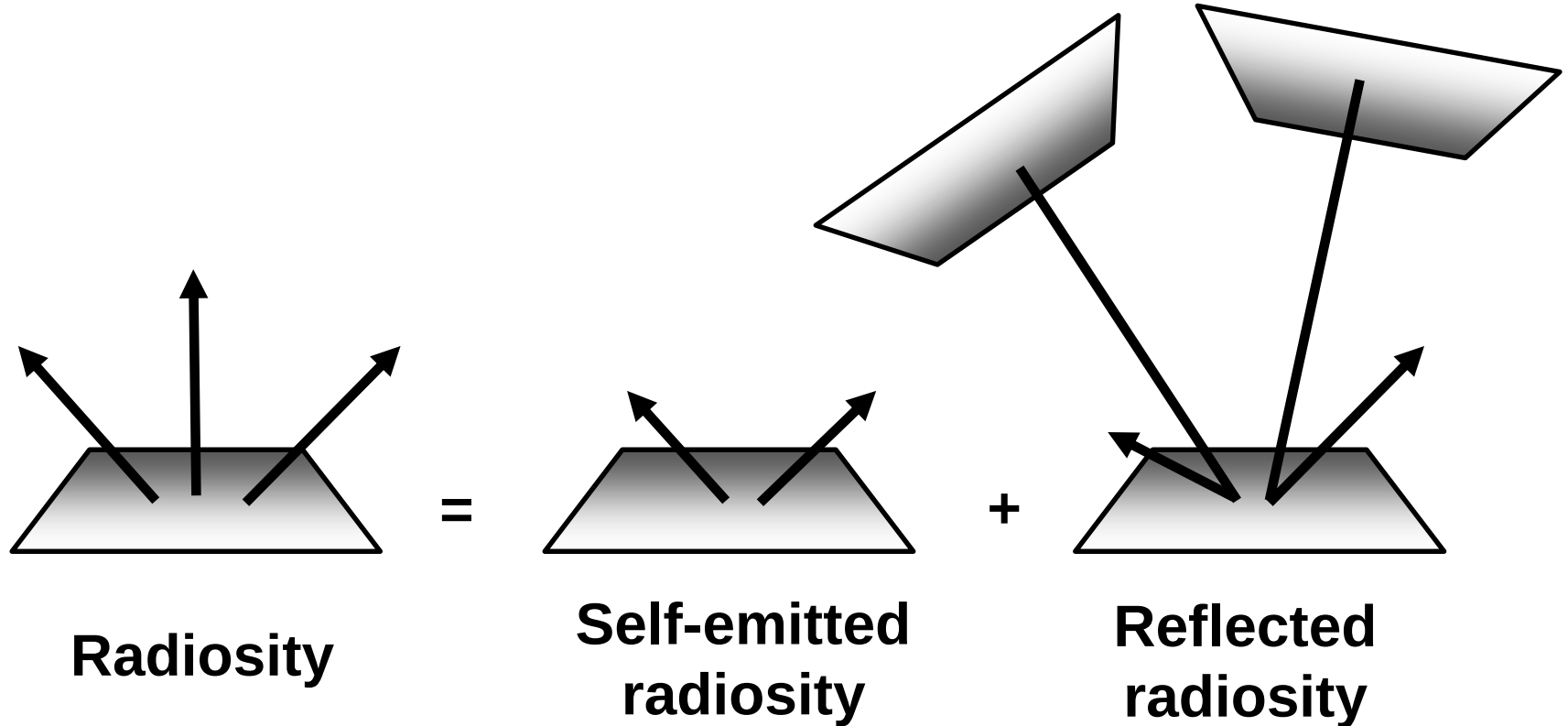
- **Subdivide scene into small polygons**

Constant Radiance Approximation



- **Radiance is constant over a surface element**
 - $L(x) = \text{constant over } x$

Radiosity Equation



$$Radiosity_i = Radiosity_{self,i} + \sum_{j=1}^N a_{j \rightarrow i} Radiosity_j$$

Radiosity Equations

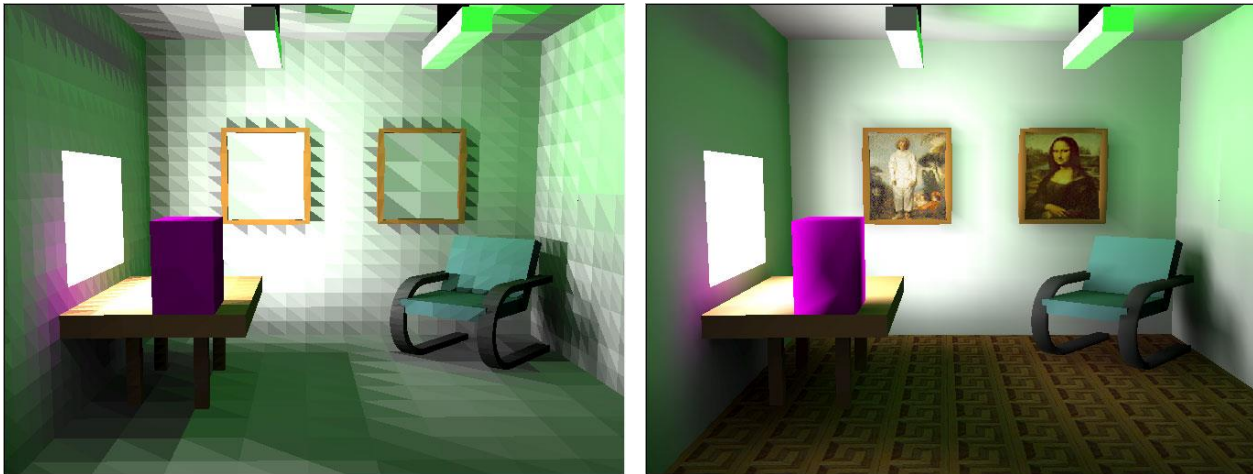
- Radiosity for each polygon i

$$\begin{aligned} \text{Radiosity}_1 &= \text{Radiosity}_{self,1} + \sum_{j=1}^N a_{j \rightarrow 1} \text{Radiosity}_j \\ &\vdots \\ \text{Radiosity}_i &= \text{Radiosity}_{self,i} + \sum_{j=1}^N a_{j \rightarrow i} \text{Radiosity}_j \\ &\vdots \\ \text{Radiosity}_N &= \text{Radiosity}_{self,N} + \sum_{j=1}^N a_{j \rightarrow N} \text{Radiosity}_j \end{aligned}$$

- N equations and N unknown variables

Radiosity Algorithm

- **Subdivide the scene in small polygons**
- **Compute a constant illumination value for each polygon**
- **Choose a viewpoint and display the visible polygon**
 - **Keep doing this process**



From Donald Fong's slides

Radiosity Result

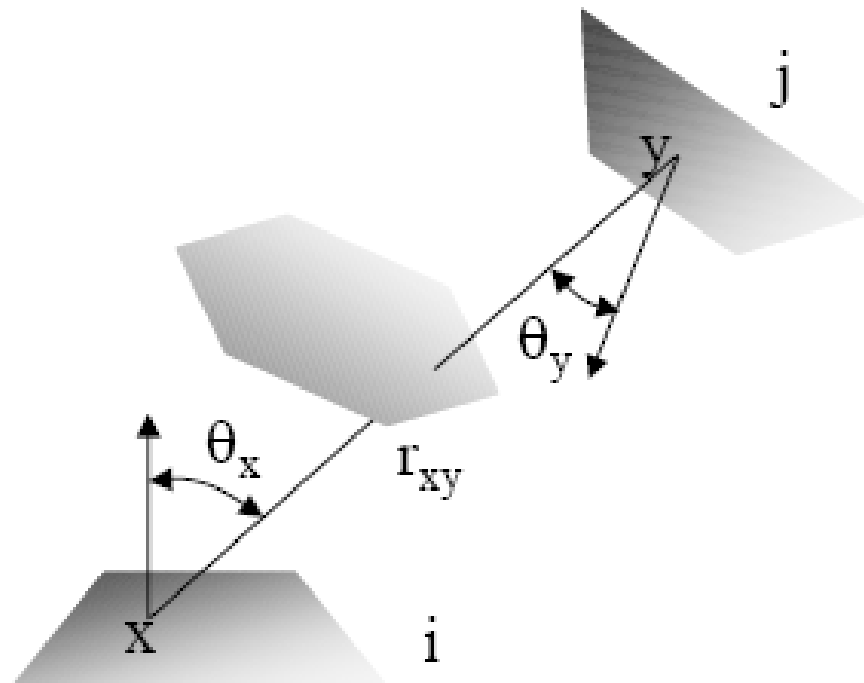


Theatre Scene



Compute Form Factors

$$F(j \rightarrow i) = \frac{1}{A_j} \int_{A_i} \int_{A_j} \frac{\cos \theta_x \cdot \cos \theta_y}{\pi \cdot r_{xy}^2} \cdot V(x, y) \cdot dA_y \cdot dA_x$$



Radiosity Equation

- **Radiosity for each polygon i**

$$B_i = B_{e,i} + \rho_i \sum_j B_j F(i \rightarrow j)$$

- **Linear system**

- B_i : radiosity of patch i (unknown)
- $B_{e,i}$: emission of patch i (known)
- ρ_i : reflectivity of patch i (known)
- $F(i \rightarrow j)$: form-factor (coefficients of matrix)

Linear System of Radiosity

$$\begin{array}{c} \text{Known} \end{array}
 \begin{bmatrix} 1 - \rho_1 F(1 \rightarrow 1) & -\rho_1 F(1 \rightarrow 2) & \dots & -\rho_1 F(1 \rightarrow n) \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F(n \rightarrow 1) & -\rho_n F(n \rightarrow 2) & \dots & 1 - \rho_n F(n \rightarrow n) \end{bmatrix}
 \begin{array}{c} \text{Known} \end{array}
 \begin{bmatrix} B_1 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} B_{e,1} \\ \vdots \\ B_{e,n} \end{bmatrix}$$

$\uparrow$
Unknown

How to Solve Linear System

- **Matrix inversion**
 - Takes $O(n^3)$
- **Gather methods**
 - Jacobi iteration
 - Gauss-Seidel
- **Shooting**
 - Southwell iteration

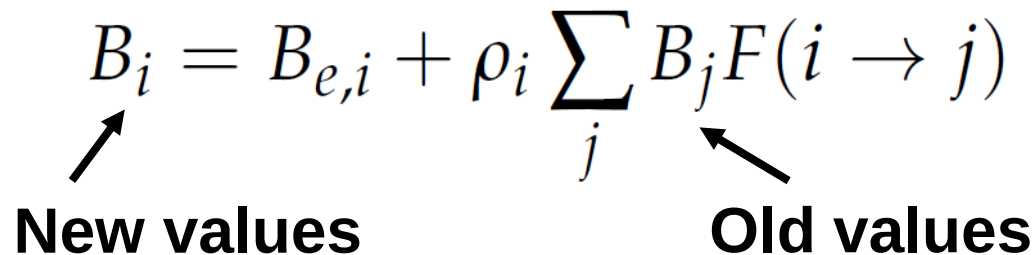
Iterative Approaches

- **Jacobi iteration**

- Start with initial guess for energy distribution (light sources)
- Update radiosity of all patches based on the previous guess

$$B_i = B_{e,i} + \rho_i \sum_j B_j F(i \rightarrow j)$$

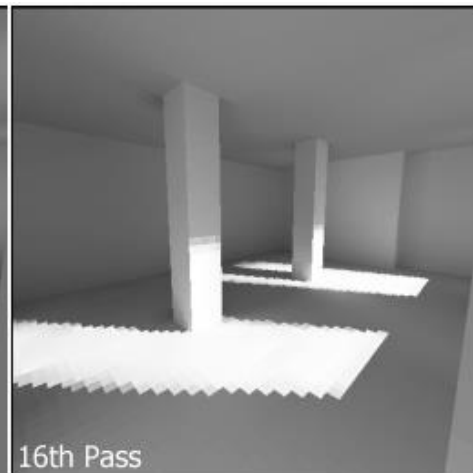
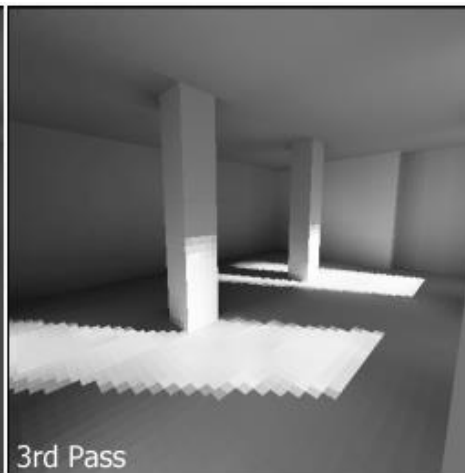
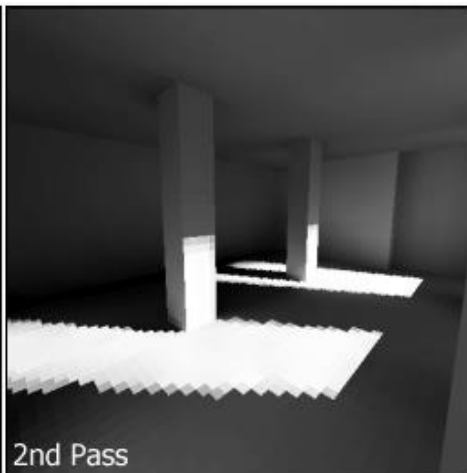
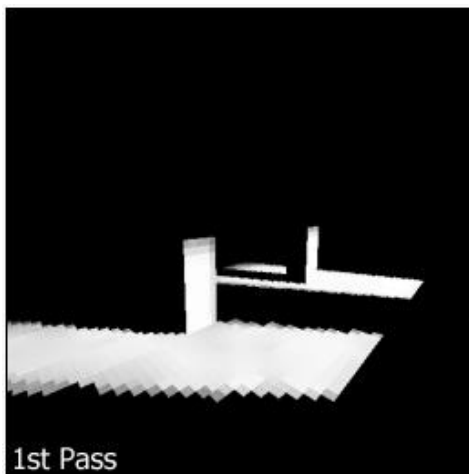
New values Old values



- Repeat until converged
- **Guass-Seidel iteration**
 - New values used immediately

Progress of Update Steps

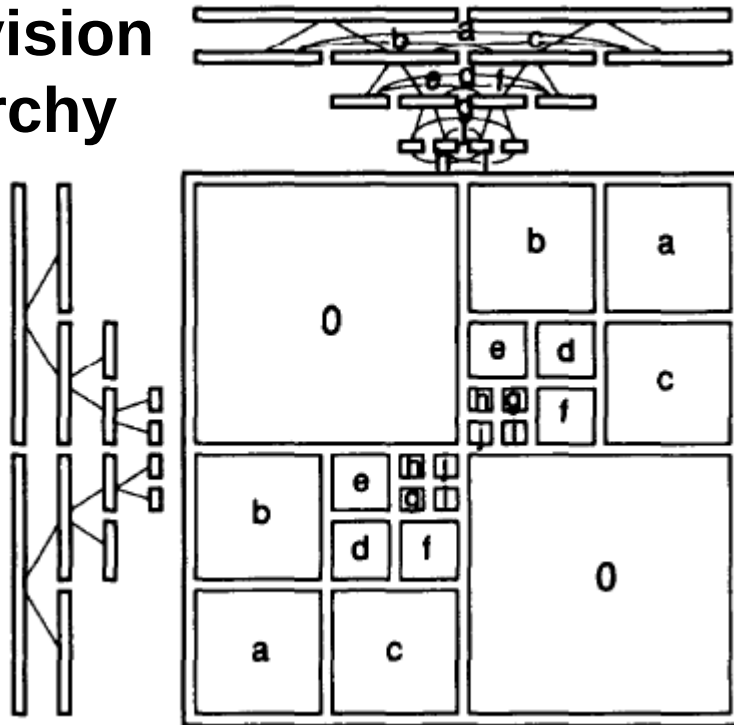
- Update step supports the light bounce



Multi-Resolution Approach

- **A Rapid Hierarchical Radiosity Algorithm, Hanrahan, et al, SIGGRAPH 1991**

Subdivision hierarchy



- **Refine triangles only if doing so improves the foam factor accuracy above a threshold**

Block diagram of the form factor matrix

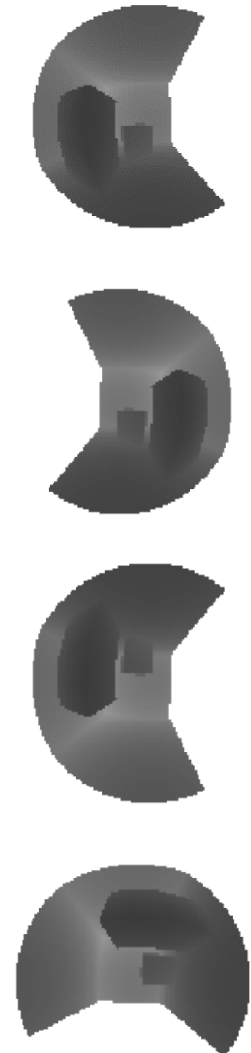
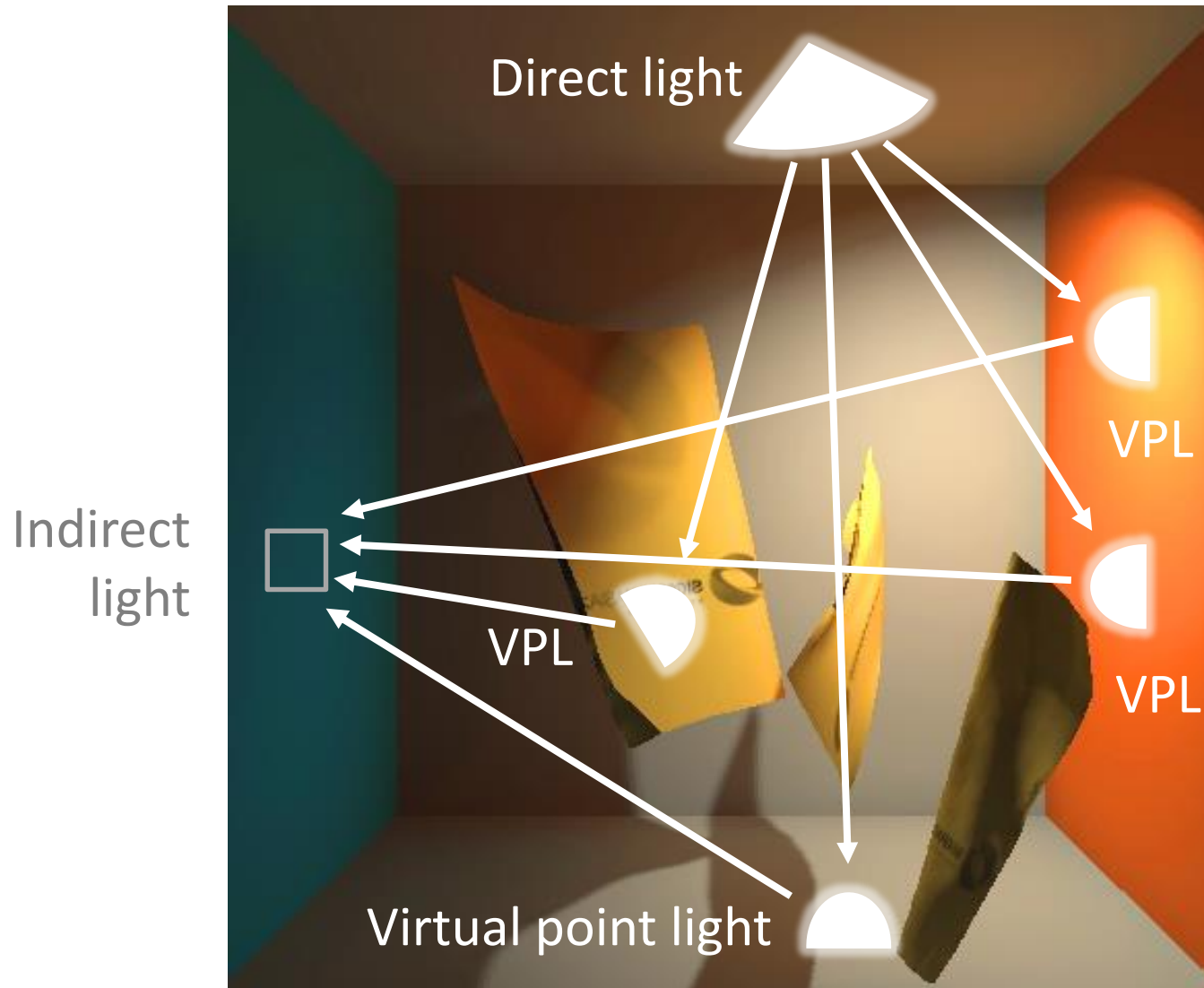
Hybrid and Multipass Methods

- **Ray tracing**
 - **Good for specular and refractive indirect illumination**
 - **View-dependent**
- **Radiosity**
 - **Good for diffuse**
 - **Allows interactive rendering**
 - **Does not scale well for massive models**
- **Hybrid methods**
 - **Combine both of them in a way**

Instant Radiosity

- **Use the concept of radiosity**
- **Map its functions to those of classic rendering pipeline**
 - **Utilize fast GPU**
- **Additional concepts**
 - **Virtual point lights**
 - **Shadow maps**

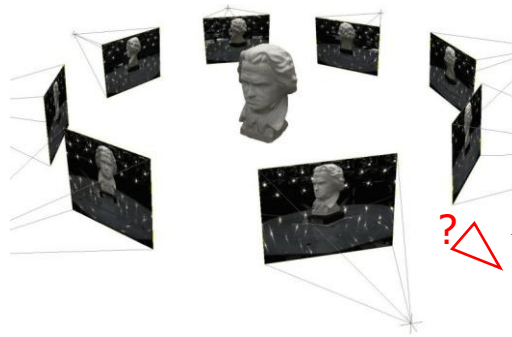
Instant Radiosity



NeRF: Neural Radiance Fields ECCV 2020 Oral - Best Paper Honorable Mention

The goal of NeRF is to synthesize photorealistic images from novel camera viewpoints.

Input: images from various camera viewpoints



Output: images from novel camera viewpoints

Source: <https://theaisummer.com/nerf/>

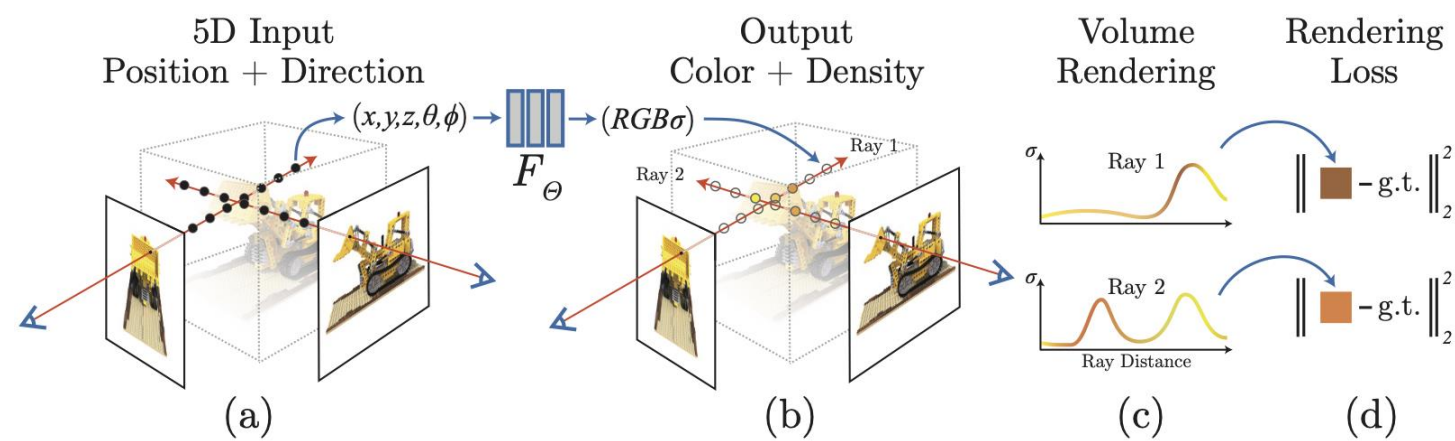
Examples (synthesized from novel views)



Videos: <https://www.youtube.com/watch?v=JuH79E8rdKc&t=191s>

<https://www.matthewtancik.com/nerf>

Neural Radiance Fields ECCV 2020 Oral - Best Paper Honorable Mention



$$\frac{C(\mathbf{r})}{T(t)} = \int_{t_n}^{t_f} \frac{T(t)\sigma(\mathbf{r}(t))\mathbf{c}(\mathbf{r}(t), \mathbf{d})dt}{T(t)}, \text{ where } T(t) = \exp\left(-\int_{t_n}^t \sigma(\mathbf{r}(s))ds\right)$$

Ray color

Transmittance

Density

Color

$\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$

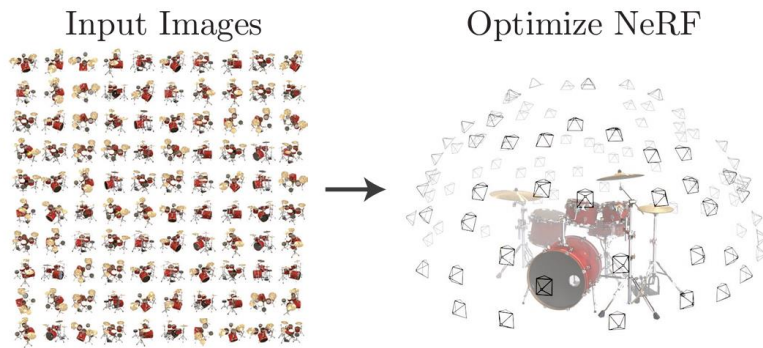
Position

Direction

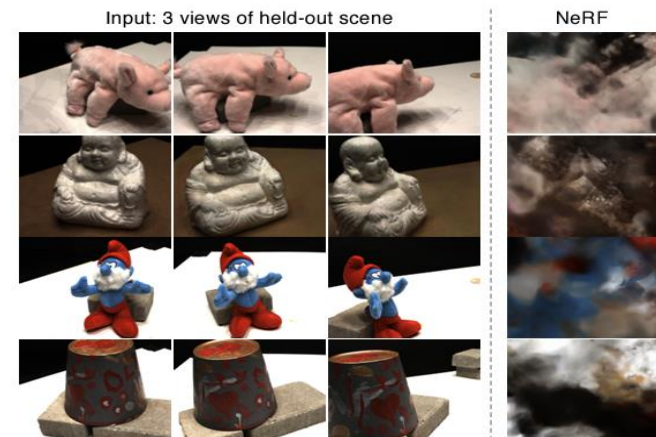
Cons of NeRF

Requires Dense camera viewpoints

Since NeRF is **under-constrained** it produces blurred or distorted results with sparse-view inputs.
=> Learning accurate 3D representation of an object requires dense views.

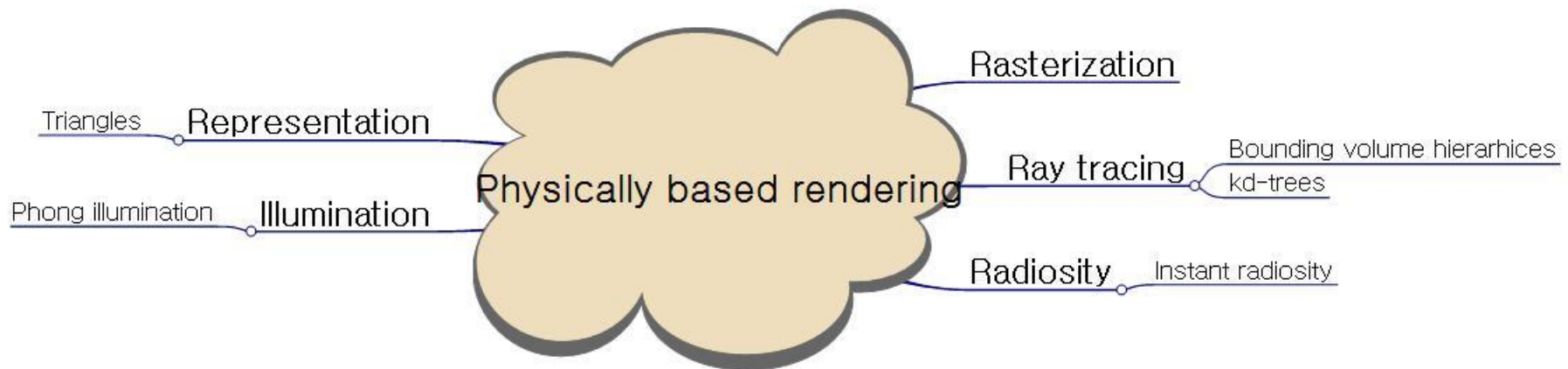


Dense camera viewpoints are required (e.g., 50+ source images)



Class Objectives were:

- **Understand radiosity**
 - Radiosity equation
 - Solving the equation



Homework

- **Go over the next lecture slides before the class**
- **Watch 2 paper videos and submit your summaries every Mon. class**
 - **Just one paragraph for each summary**

Example:

Title: XXX XXXX XXXX

Abstract: this video is about accelerating the performance of ray tracing. To achieve its goal, they design a new technique for reordering rays, since by doing so, they can improve the ray coherence and thus improve the overall performance.

Any Questions?

- **Submit four times in Sep./Oct.**
- **Come up with one question on what we have discussed in the class and submit at the end of the class**
 - **1 for typical questions**
 - **2 for questions that have some thoughts or surprise me**

Next Time

- **Radiometry and rendering equation**