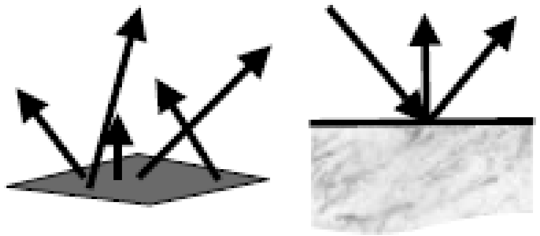
# CS680:
# Rendering Equation

Sung-Eui Yoon
(윤성의)

**Course URL:**
**http://jupiter.kaist.ac.kr/~sungeui/SGA/**

# Light and Material Interactions

- **Physics of light**



- **Radiometry**
- **Material properties**

From kavita's slides

- **Rendering equation**

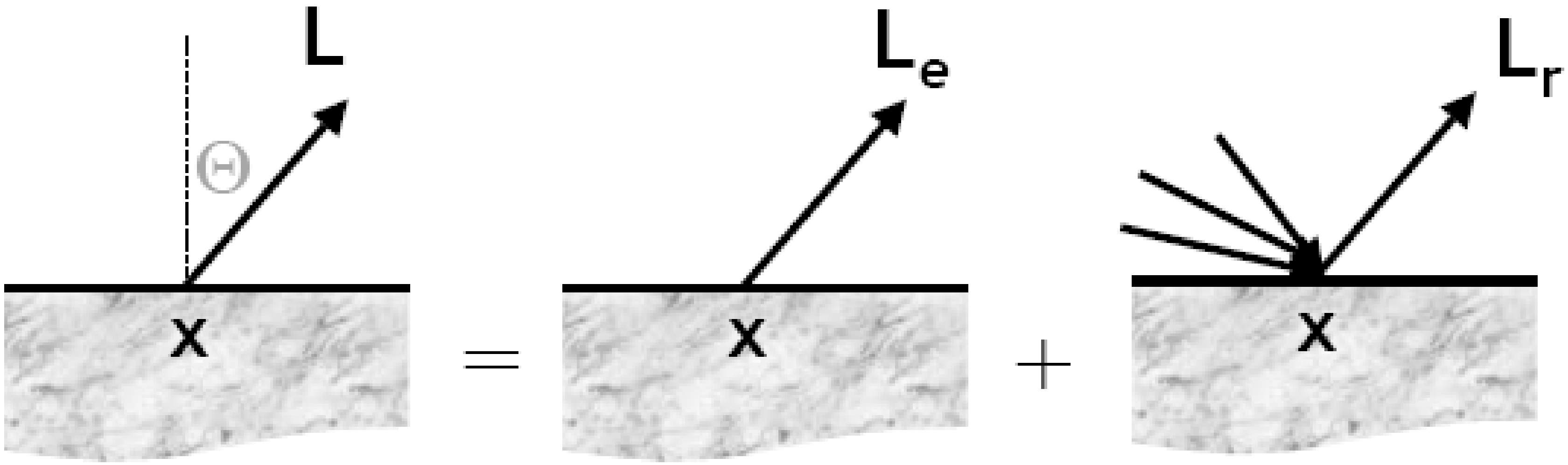
# Light Transport

- **Goal**
  - **Describe steady-state radiance distribution in scene**
- **Assumption**
  - **Geometric optics**
  - **Achieves steady state instantaneously**

# Rendering Equation

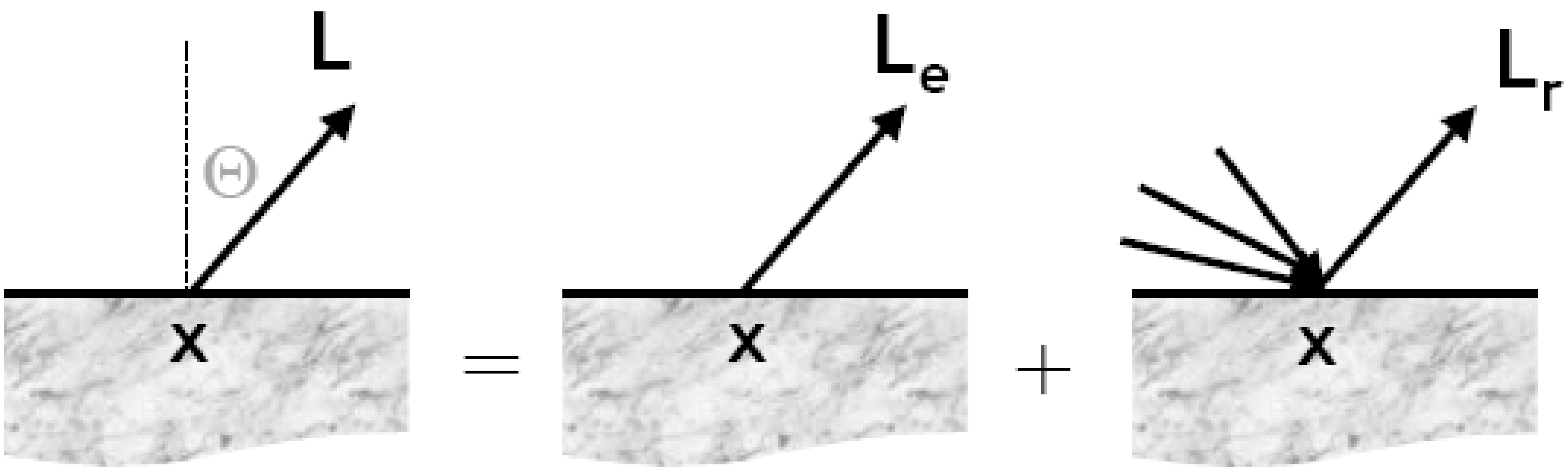
- Describes energy transport in scene
- Input
  - Light sources
  - Surface geometry
  - Reflectance characteristics of surfaces
- Output
  - Value of radiances at all surface points in all directions

# Rendering Equation




$$L(x \to \Theta) = L_e(x \to \Theta) + L_r(x \to \Theta)$$

# Rendering Equation




$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{hemisphere} L(x \leftarrow \Psi) \cdots$$

# Rendering Equation

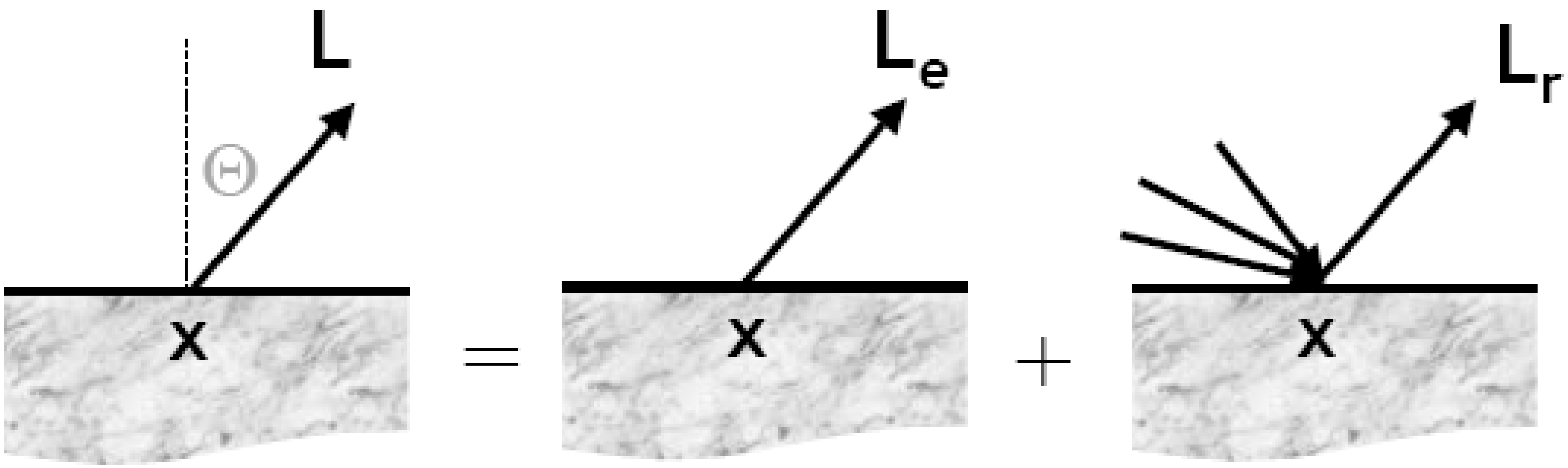
$$f_r(x, \Psi \leftrightarrow \Theta) = \frac{dL(x \to \Theta)}{dE(x \leftarrow \Psi)}$$

$$dL(x \to \Theta) = f_r(x, \Psi \leftrightarrow \Theta) dE(x \leftarrow \Psi)$$

$$dL(x \to \Theta) = f_r(x, \Psi \leftrightarrow \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi$$

$$L_r(x \to \Theta) = \int_{hemisphere} f_r(x, \Psi \leftrightarrow \Theta) L(x \leftarrow \Psi) \cos(N_x, \Psi) d\omega_\Psi$$

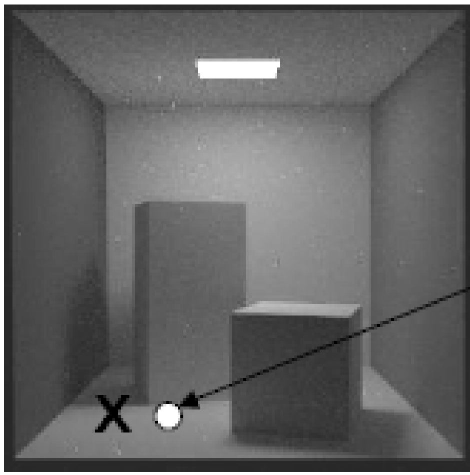
# Rendering Equation




$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{hemisphere} L(x \leftarrow \Psi)\, f_r(x, \Psi \leftrightarrow \Theta) \cos(\mathbf{N}_x, \Psi)\, d\omega_\Psi$$

- **Applicable for each wavelength**

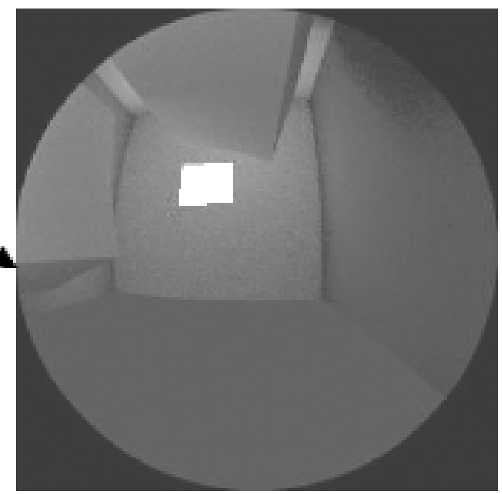
# Rendering Equation



$$L(x \to \Theta) \;=\; L_e(x \to \Theta) +$$

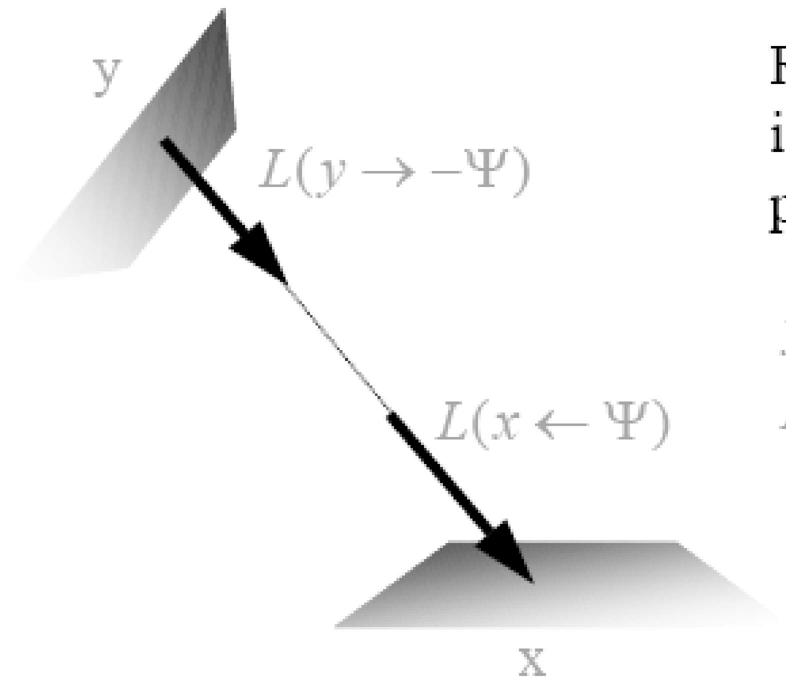
$$\int_{hemisphere} L(x \leftarrow \Psi) f_r(x, \Psi \leftrightarrow \Theta) \cos(N_x, \Psi) d\omega_\Psi$$

incoming radiance



# Rendering Equation: Area Formulation

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos\theta_x \cdot d\omega_\Psi$$

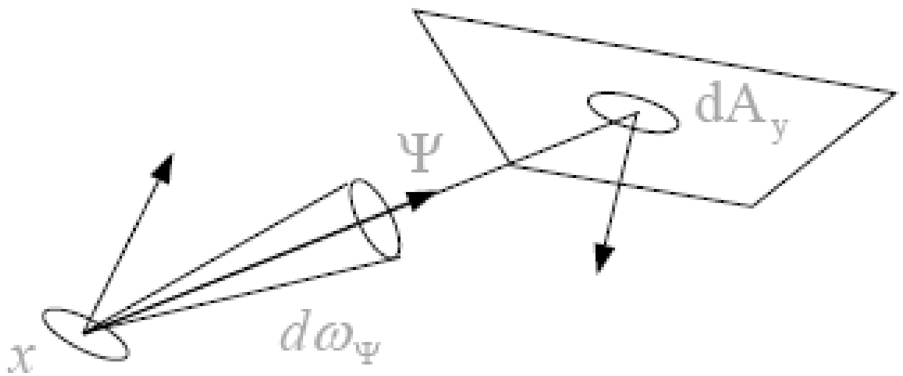



Ray-casting function: what is the nearest visible surface point seen from x in direction $\Psi$?

$$y = vp(x, \Psi)$$
$$L(x \leftarrow \Psi) = L(vp(x, \Psi) \to -\Psi)$$

# Rendering Equation

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos\theta_x \cdot d\omega_\Psi$$




$$y = vp(x, \Psi)$$

$$L(x \leftarrow \Psi) = L(vp(x, \Psi) \to -\Psi)$$

$$d\omega_\Psi = \frac{dA_y \cos\theta_y}{r_{xy}^2}$$

# Rendering Equation: Visible Surfaces

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos\theta_x \cdot d\omega_\Psi$$

Coordinate transform

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\substack{y \text{ on} \\ \text{all surfaces}}} f_r(\Psi \leftrightarrow \Theta) \cdot L(y \to -\Psi) \, \cos\theta_x \cdot \frac{\cos\theta_y}{r_{xy}^2} \cdot dA_y$$

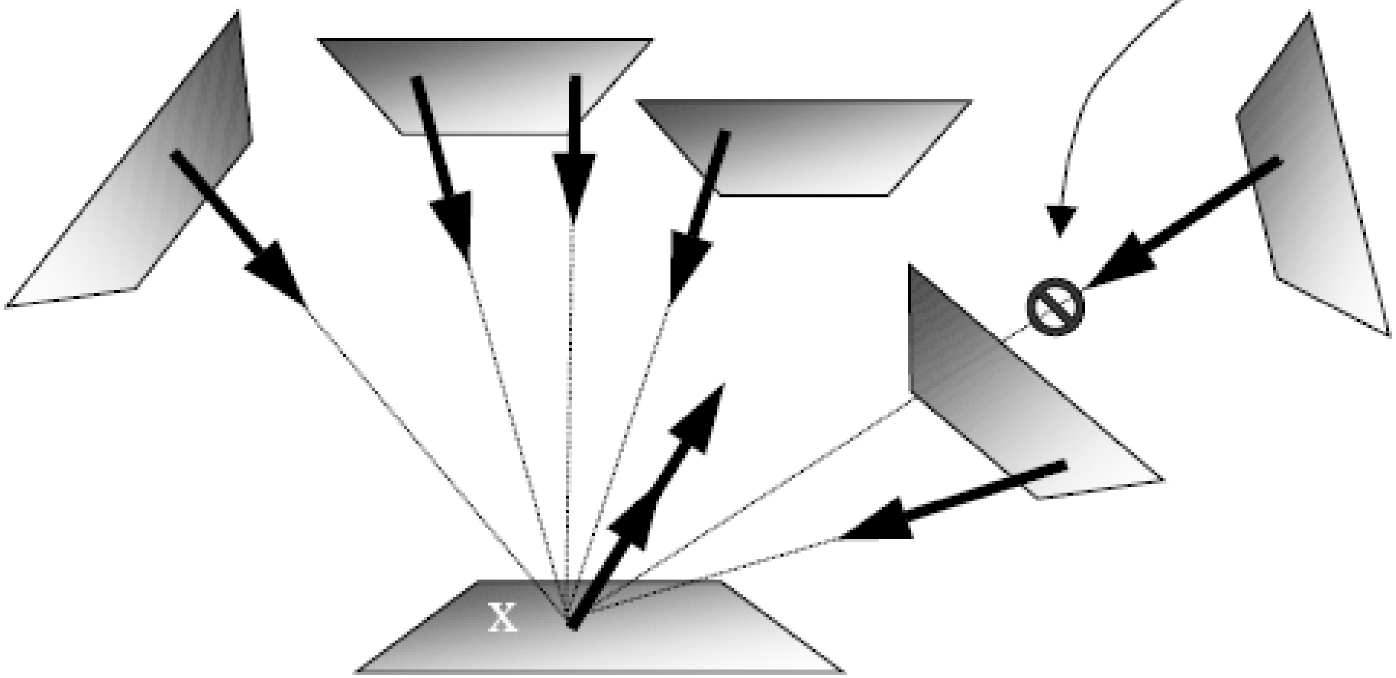
$$y = vp(x, \Psi)$$

Integration domain = visible surface points y

- Integration domain extended to ALL surface points by including visibility function

# Rendering Equation: All Surfaces

$$\underline{L(x \rightarrow \Theta)} = \underline{L_e(\ldots)} + \int_A f_r(\ldots) \cdot \underline{L(y \rightarrow -\Psi)} \cdot \frac{\cos\theta_x \cdot \cos\theta_y}{r_{xy}^2} \cdot \boxed{V(x,y)}\, dA_y$$



# Two Forms of the Rendering Equation

- **Hemisphere integration**

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos\theta_x \cdot d\omega_\Psi$$

- **Area integration**

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_A f_r(\Psi \leftrightarrow \Theta) \cdot L(y \to -\Psi) \cdot \frac{\cos\theta_x \cdot \cos\theta_y}{r_{xy}^2} \cdot V(x,y) \cdot dA_y$$

# Summary

- Geometric optics
- Radiometry
- Rendering equation
  - Mathematical formulation that global illumination algorithms must solve

# Next Time

- **Monte Carlo ray tracing**