# Instant Radiosity

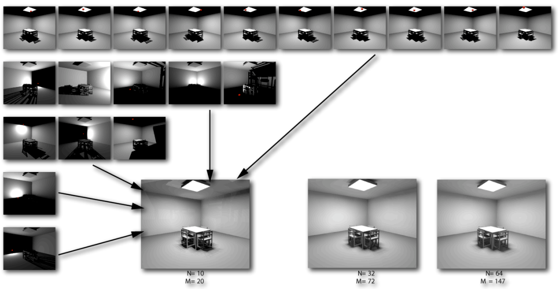
M074/GV14



## Overview

- Goal is to enable "interactive" global illumination in purely diffuse environments
- Approximate direct and indirect lighting with *virtual point lights (VPL)*.
- Render scene with VPLs (with shadowing)
  - Sum up contributions from all VPLs.

## Overview



## Rendering Equation

- Defined as:

$$L(p,\omega_o) = L_e(p,\omega_o) + \int_\Omega f(p,\omega_i,\omega_o) L(p^*,-\omega_i)\cos\theta_i \,\mathrm{d}\omega_i$$

- Using operator R:

$$(\mathbf{R}L)(p,\omega) = \int_\Omega f(p,\omega_i,\omega) L(p^*,-\omega_i)\cos\theta_i \,\mathrm{d}\omega_i$$

- Equation becomes: $L = L_e + \mathbf{R}L$

## Rendering Equation

- Solving for L:

$$L = L_e + \mathbf{R}L$$
$$(1-\mathbf{R})L = L_e$$
$$L = (1-\mathbf{R})^{-1} L_e$$
$$L = (1 + \mathbf{R} + \mathbf{R}^2 + \mathbf{R}^3 + \ldots) L_e$$

- Radiance towards eye =
  - direct light from light source
  - plus light reflected once,
  - plus light reflected twice, …

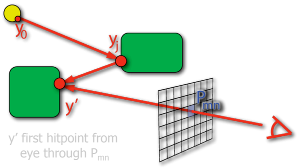
## Instant Radiosity

- Assume BRDF is diffuse

$$f(p) = f(p,\omega_i,\omega_o) = \frac{k_d}{\pi}$$

- Rewrite rendering equation with *explicit* sampling of ***all*** possible paths (with length j=0, j=1, j=2, …)

## Instant Radiosity

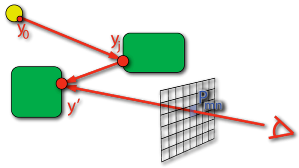



- Integrate over pixel $P_{mn}$
- Sum over all paths with length j
- Integrate over all paths $\Omega^j$ ($\Omega \times \Omega \times \ldots$) with length j
- Integrate over light source $S_e$ (positions $y_0$)

$$L(m,n) = \frac{1}{|P_{mn}|}\int_{P_{mn}} L_e(y', P - y') \; + \frac{1}{|P_{mn}|}\int_{P_{mn}} \sum_{j=0}^{\infty} \int_{\Omega^j} \int_{S_e} p_j(y_0, \omega_0, \ldots, \omega_j) \cdot V(y_j, y') f_d(y') \frac{\cos\theta_j \cos\theta'}{|y_j - y'|^2} dy_0 d\omega_0 \ldots d\omega_j dP$$

7

## Instant Radiosity



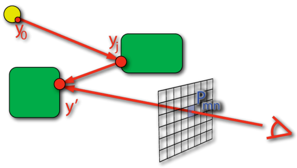

- Radiance after j reflections

$$p_j(y_0, \omega_0, \ldots, \omega_j) := L_e(y_0) \prod_{l=1}^{j} (\cos\theta_{l-1} f_d(y_l))$$

[Assumes valid paths (all $y_l$ are mutually visib.)]

8

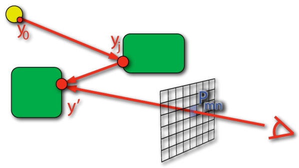
## Instant Radiosity




- Now, we importance sample those paths
  - Create paths of different lengths ($y_0$ to $y_j$)
  - All $y_j$ act as virtual point lights

9

## Instant Radiosity




- Sampling the paths
  - Start with random point $y_0$ on light source
  - Follow them (using IS) through the scene
  - Every hitpoint $y_j$ becomes a VPL

10

## When and how to end a path?

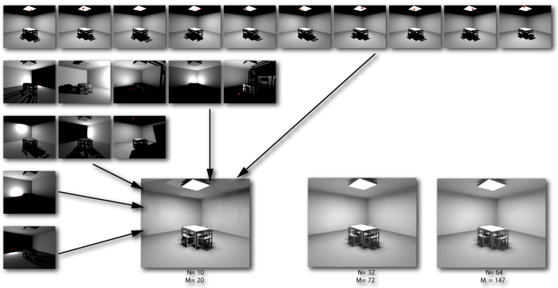
- We start with *N* point lights on the area light
  - Reflect all *N* of them? Paths never end then…
  - Russian Roulette?
- Assume surfaces are not far from average reflectivity $\bar{\rho}$
  - Enables use of fractional absorption
  - Of the initial *N*: $\bar{\rho}N$ get reflected (1st bounce)
  - 2nd bounce: $\bar{\rho}^2 N$ get reflected, etc…
  - Average path length: $\frac{1}{1-\bar{\rho}} N$

## Code

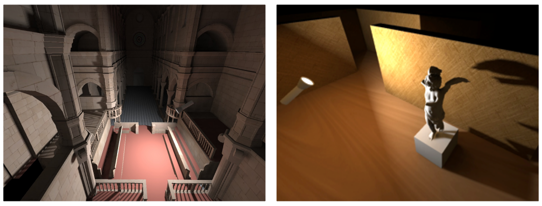
- First, render all paths that immediately end on the light source (all $(1-\bar{\rho})N$ of them)
- Then, render all paths that are reflected ones, etc.

```
void InstantRadiosity(int N, double ρ̄)
{
    double w, Start; int End, Reflections = 0;
    Color L; Point y; Vector ω⃗;

    Start = End = N;

    while(End > 0)
    {
        Start *= ρ̄;

        for(int i = (int) Start; i < End; i++)
        {
            // Select starting point on light source
            y = y0(Φ2(i), Φ3(i));
            L = Le(y) * supp Le;
            w = N;

            // trace reflections
            for(int j = 0; j <= Reflections; j++)
            {
                glRenderShadowedScene(N/⌊w⌋ L, y);
                glAccum(GL_ACCUM, 1/N);
                // diffuse scattering
                ω⃗ = ω⃗d(Φb_{2j+2}(i), Φb_{2j+3}(i));
                //trace ray from y into direction ω⃗
                y = h(y, ω⃗);
                // Attenuate and compensate
                L *= fd(y);
                w *= ρ̄;
            }
        }

        Reflections++;
        End = (int) Start;
    }

    glAccum(GL_RETURN, 1.0);
}
```

## Overview (again)

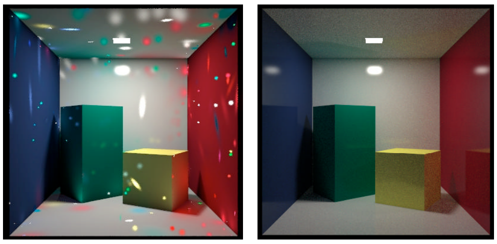



## More Results



## Problems

- Difficult to extend to specular surfaces




## Summary

- Instant Radiosity
  - Assume diffuse scenes (like Radiosity)
  - Approximate indirect illumination with VPLs
  - Similarities to photo mapping (path sampling)
  - Easy to implement on GPUs