Global illumination with many-light methods

Jaroslav Křivánek

Charles University, Prague
Instant radiosity

- Alexander Keller, 1997
- The "original" many-light method
- Probably the first GPU-based GI algorithm
Instant radiosity

- Approximate indirect illumination by Virtual Point Lights (VPLs)

1. Generate VPLs
2. Render with VPLs
Instant radiosity as BDPT

- VPLs = light sub-paths
- VPL contributions = sub-path connections
Instant radiosity

- Works well in diffuse scenes
- 100s of VPLs sufficient for ok-ish images
- Basis of many real-time GI algorithms
Real-time GI with Instant radiosity

- Reflective shadow maps
  [Dachsbacher and Stammbinger 05]
  - Fast VPL generation

- Incremental Instant Radiosity
  [Laine et al. 07]
  - Only a few new VPLs per frame

- Imperfect Shadow Maps
  [Ritschel et al. 08]
  - Faster shadow mapping
Intuition behind VPLs

- There is nothing in global illumination images that a CG artist could not simulate otherwise.
- VPLs "automate" the artist approach.

Slide credit: Miloš Hašan
Clamping & compensation

Kollig and Keller, 2004
• Singularity in light contribution
Biased result with clamping
Unbiased result with compensation
Scalability
Instant radiosity with glossy surfaces

- Large number of VPLs required
  - True even for diffuse scenes
  - Scalability issues
Scalable many-light methods

1. Generate many, many VPLs
2. Use only the most relevant VPLs for rendering

• Choosing the right VPLs
  – Per-image basis
    • Matrix Row Column Sampling [Hašan et al. 07]
  – Per-pixel basis
    • Lightcuts [Walter et al. 05/06]
More lights may not do the trick...

VPL  |  GI reference  |  VPLs w/ clamping

artifacts  |  material change
Dealing with gloss in many-light methods

Approach #1: Virtual Spherical Lights

Hašan, Křivánek & Bala, SIGGRAPH Asia 2009
Instant radiosity

• Approximate indirect illumination by Virtual Point Lights (VPLs)

1. Generate VPLs
2. Render with VPLs
Emission distribution of a VPL

- Cosine-weighted BRDF lobe at the VPL location

Glossy

Diffuse
Glossy VPL emission: illumination spikes

Common solution: Only **diffuse** BRDF at light location
Remaining spikes
Remaining spikes

- VPL contribution =

\[
\text{As } \| p - x \| \to 0, \text{ VSL contribution } \to \infty
\]

- Common solution: **Clamp** VPL contributions
Instant radiosity: The practical version

Clamping and diffuse-only VPLs: Illumination is lost!
Comparison

Clamped VPLs: Illumination loss
Path tracing: Slow
Recall: Emission Distribution of a VPL

Spike!
What happens as \#lights \rightarrow \infty?

Spiky lights converge to a continuous function!
Idea: We want a "virtual area light"

Aggregate incoming illumination

Aggregate outgoing illumination

"Virtual area light"

Problem: What if surface is not flat?
VPL to VSL

Non-zero radius \( (r) \)

Integration over non-zero solid angle

\( \mathbf{x} \)
Light Contribution

Non-zero radius \( (r) \)

Integration over non-zero solid angle

\[ \int_{\Omega} \]
Non-zero radius ($r$)

Integration over non-zero solid angle

Problem: Finding $y$ requires ray-tracing

$$\frac{\Phi}{\pi r^2} \int_{\Omega} f_r(x) \cos \theta_x f_r(y) \left( \|p - y\| < r \right) dl$$
Simplifying Assumptions

- Non-zero radius ($r$)
- Integration over non-zero solid angle

- Constant in $\Omega$:
  - Visibility
  - Surface normal
  - Light BRDF

- Taken from $p$, the light location
Light Contribution Updated

Non-zero radius ($r$)

Integration over non-zero solid angle

\[ V \int_{\Omega} f_r(p) \cos \theta_p \, dl \]
Virtual Spherical Light

- All inputs taken from \( \mathbf{x} \) and \( \mathbf{p} \)
  - Local computation
- Same interface as any other light
  - Can be implemented in a GPU shader
- Visibility factored from the integration
  - Can use shadow maps

\[
V \frac{\Phi}{\pi r^2} \int_{\Omega} f_r(\mathbf{x}) \cos \theta_x \ f_r(\mathbf{p}) \ \cos \theta_p \ \mathrm{d}l
\]
Implementation

- Matrix row-column sampling
  - Shadow mapping for visibility
  - VSL integral evaluated in a GPU shader

- Need more lights than in diffuse scenes
Results: Kitchen

- Most of the scene lit indirectly
- Many materials glossy and anisotropic
Results: Disney concert hall

- Curved walls with no diffuse component
- Standard VPLs cannot capture any reflection from walls

Clamped VPLs:
22 sec (GPU) – 4000 lights

New VSLs:
1 min 26 sec (GPU) – 15000 lights

Path tracing:
30 hours (8 cores)
Results: Anisotropic tableau

- Difficult case
- Standard VPLs capture almost no indirect illumination

Clamped VPLs: 32 sec (GPU) – 1000 lights

Path tracing: 2.2 hours (8 cores)

New VSLs: 1 min 44 sec (GPU) – 5000 lights
Limitations: Blurring

- VSLs can blur illumination
- Converges as number of lights increases
Conclusions

• Many-light methods do not deal well with glossy scenes
  – Artifacts or energy loss
  – Energy loss -> change of material perception

• Handling glossy effects with many-lights
  – Virtual Spherical Lights