Virtual Spherical Lights for Many-Light Rendering of Glossy Scenes

Miloš Hašan

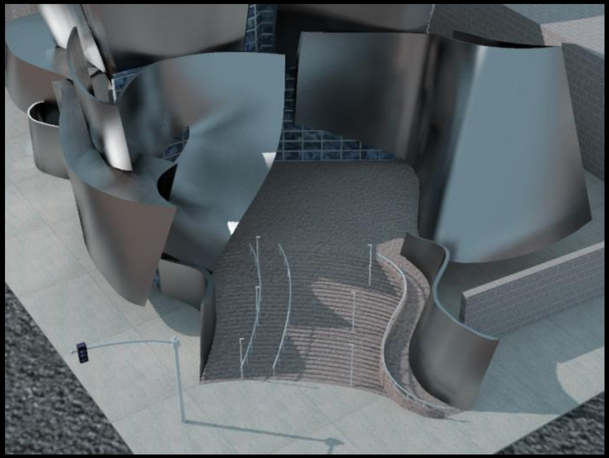
Jaroslav Křivánek *

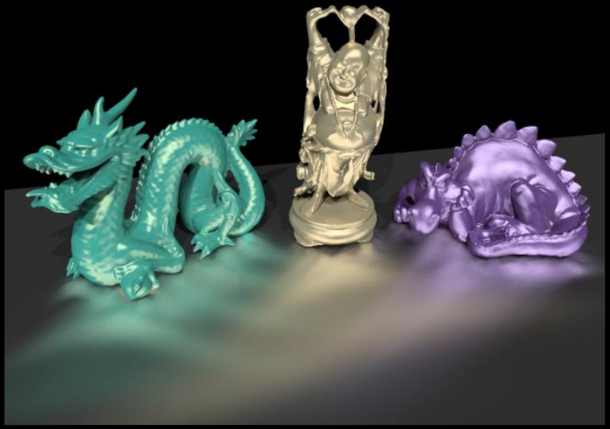
Bruce Walter

Kavita Bala

Cornell University

* Charles University in Prague

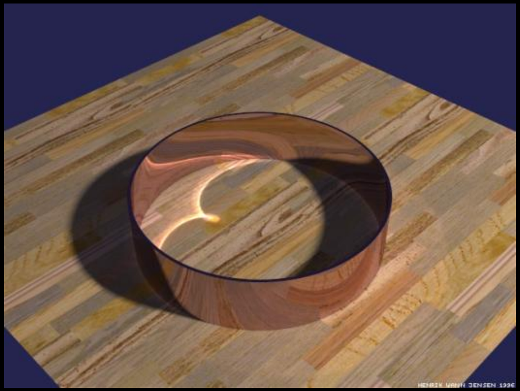




Global Illumination Effects

Soft shadows

Color bleeding



Caustics

Mirror reflection

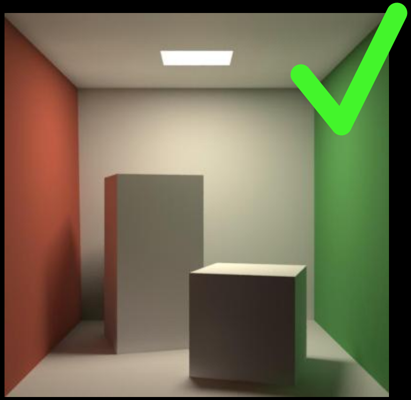
Refraction

Glossy inter-reflection

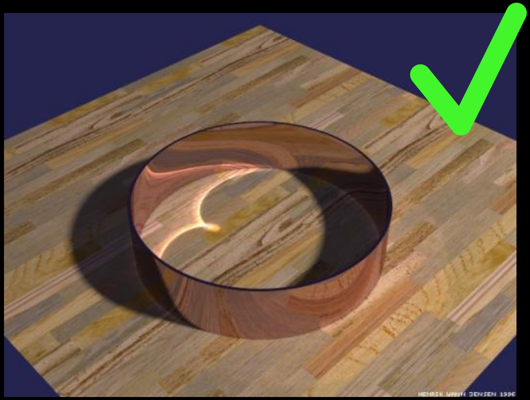
Monte Carlo can handle them all… but is very slow

Faster algorithms exist...

Soft shadows

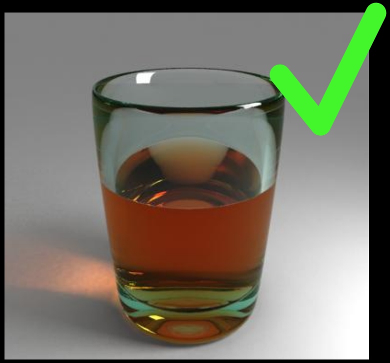


Color bleeding



Caustics

Mirror reflection



Refraction

Glossy inter-reflection

But no satisfying solution for glossy inter-reflection

Glossy Inter-reflections

Previous Work

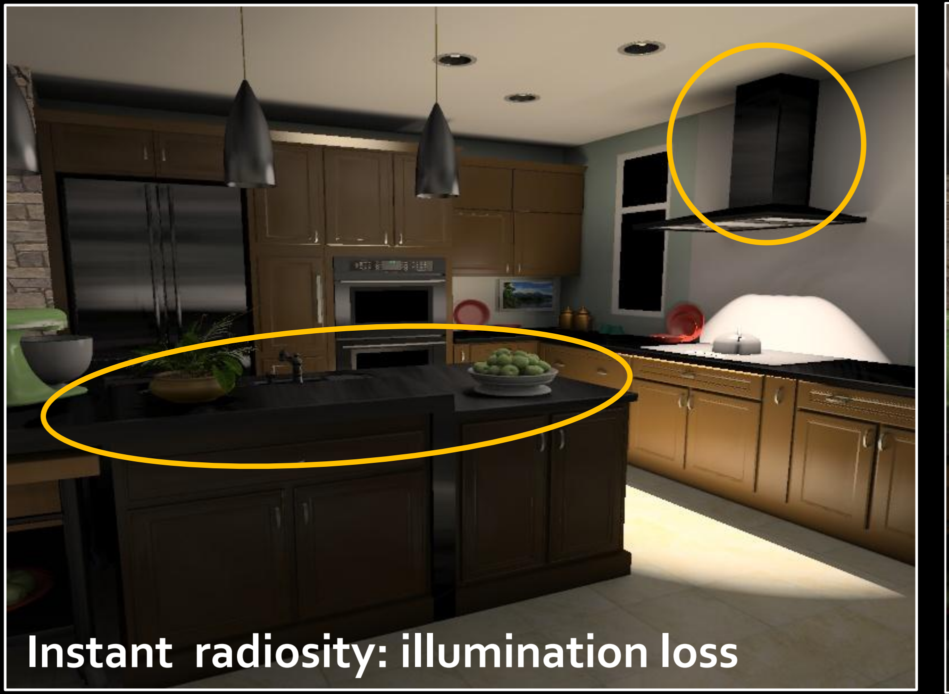
- Unbiased methods
  - (Bidirectional) Path tracing
    [Kajiya 1985, Lafortune et al. 1993]
  - Metropolis Light Transport
    [Veach and Guibas 1997]
- Biased methods
  - Photon Mapping
    [Jensen 2001]
  - Radiance caching
    [Křivánek 2005]

Previous Work – Instant Radiosity

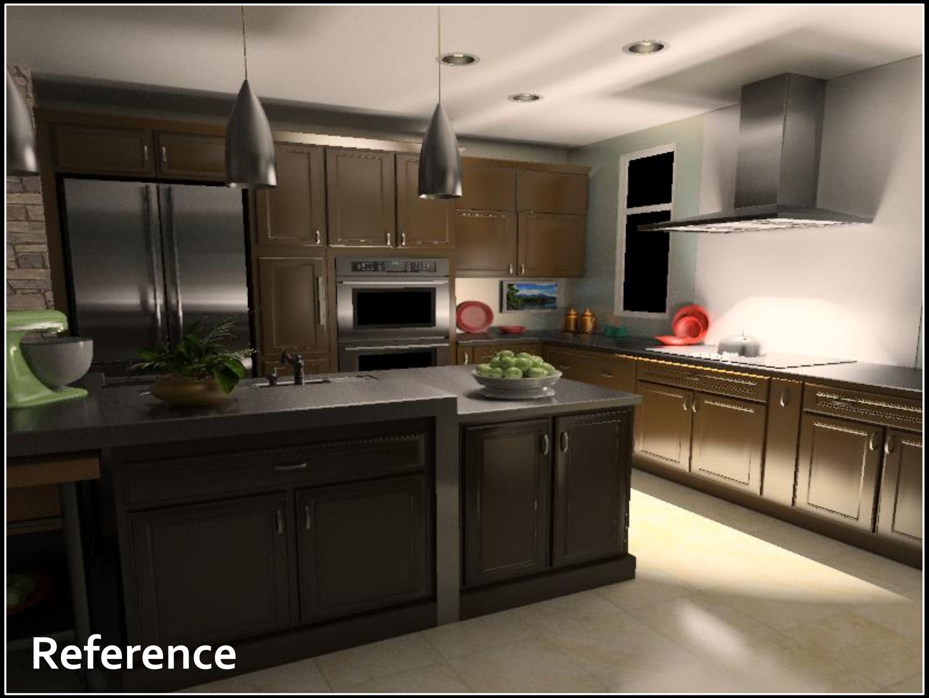
- Virtual Point Lights (VPLs)
- Very efficient in mostly diffuse scenes
  - Real-time global illumination
    [Wald et al. 2002, Segovia et al. 2006, 2007, Laine et al. 2007, Ritschel et al. 2008, Dong et al. 2009]
- Scalability to many lights
  [Walter et al. 2005, 2006, Hašan et al. 2007]

Limitations of Instant Radiosity

- So far: ~~Instant radiosity & Glossy inter-reflections~~



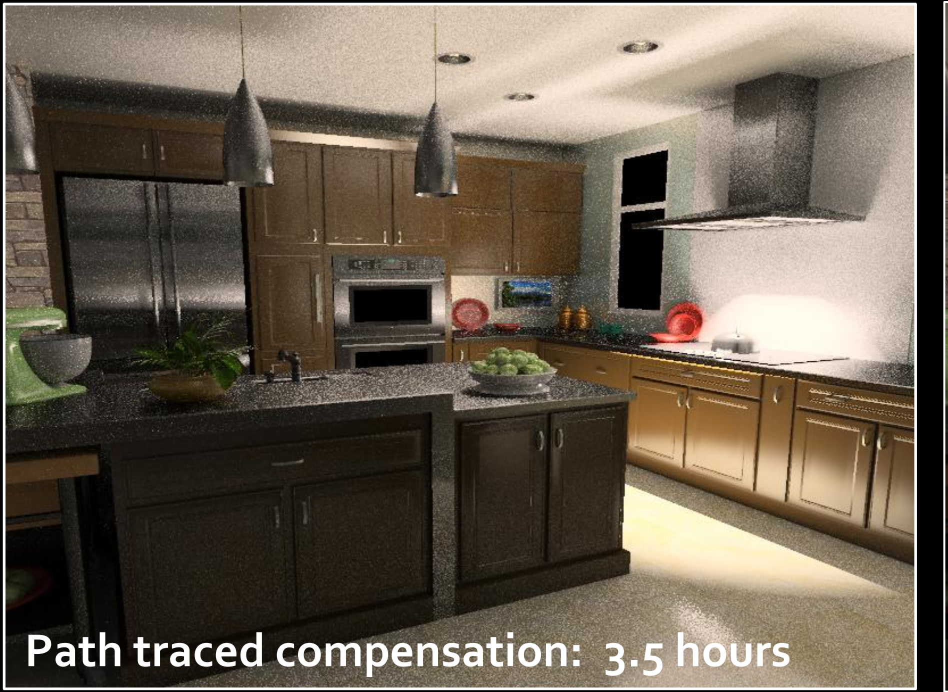
Instant radiosity: illumination loss

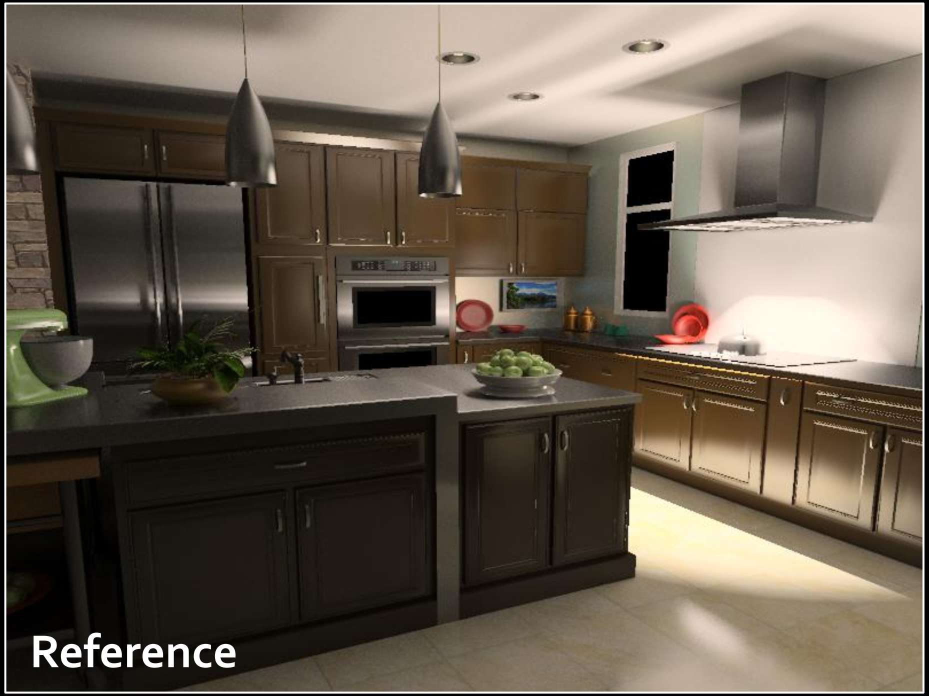


Reference

Previous Work on Compensation

- Compute the missing components by path tracing [Kollig and Keller 2004]

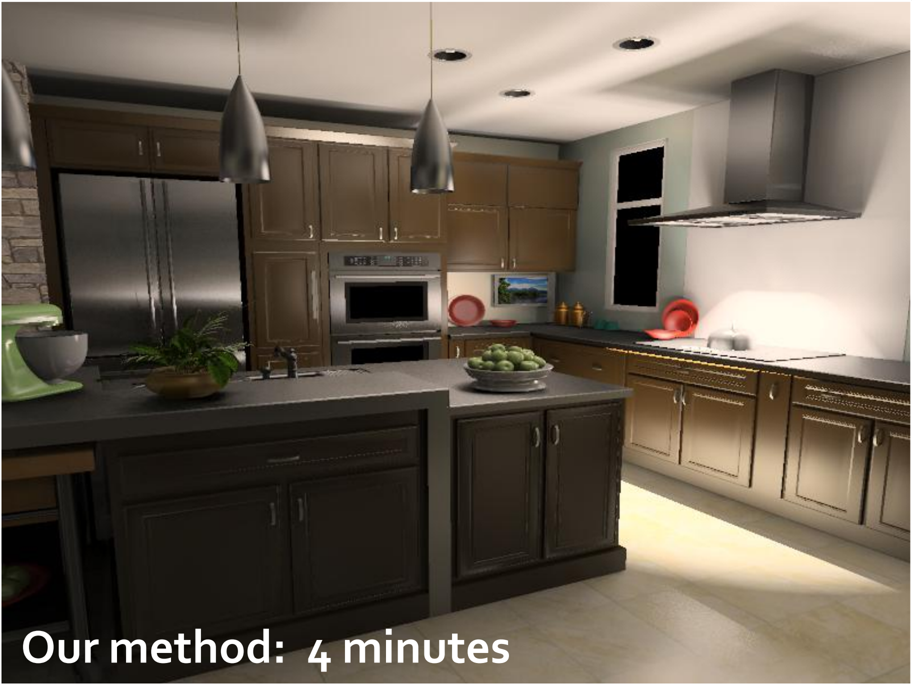




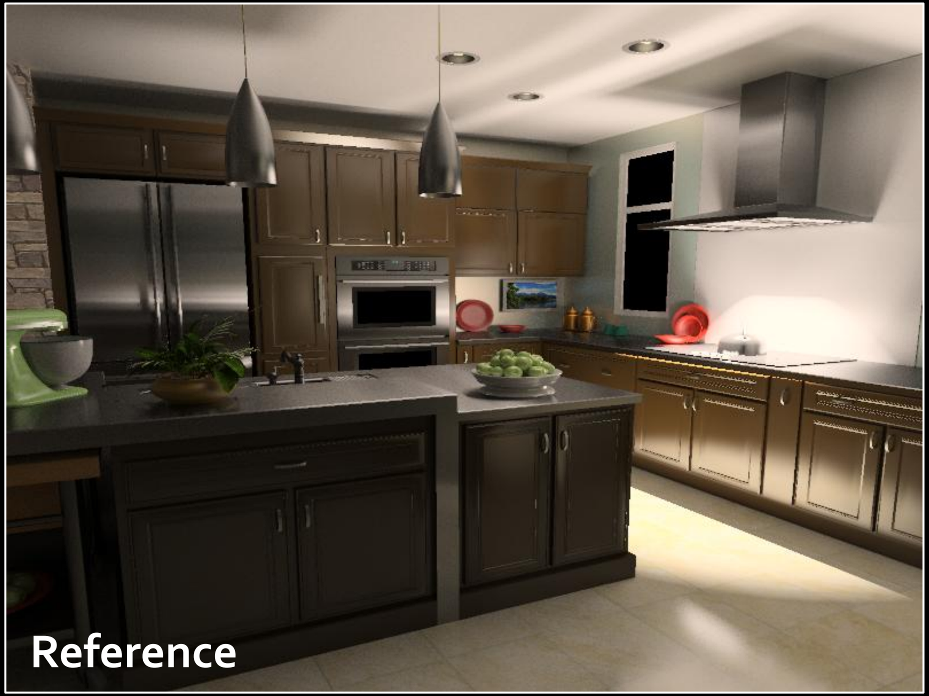
- Glossy scenes
  - As slow as path-tracing everything

Our Method

- New type of light: Virtual Spherical Light



Our method: 4 minutes



Reference

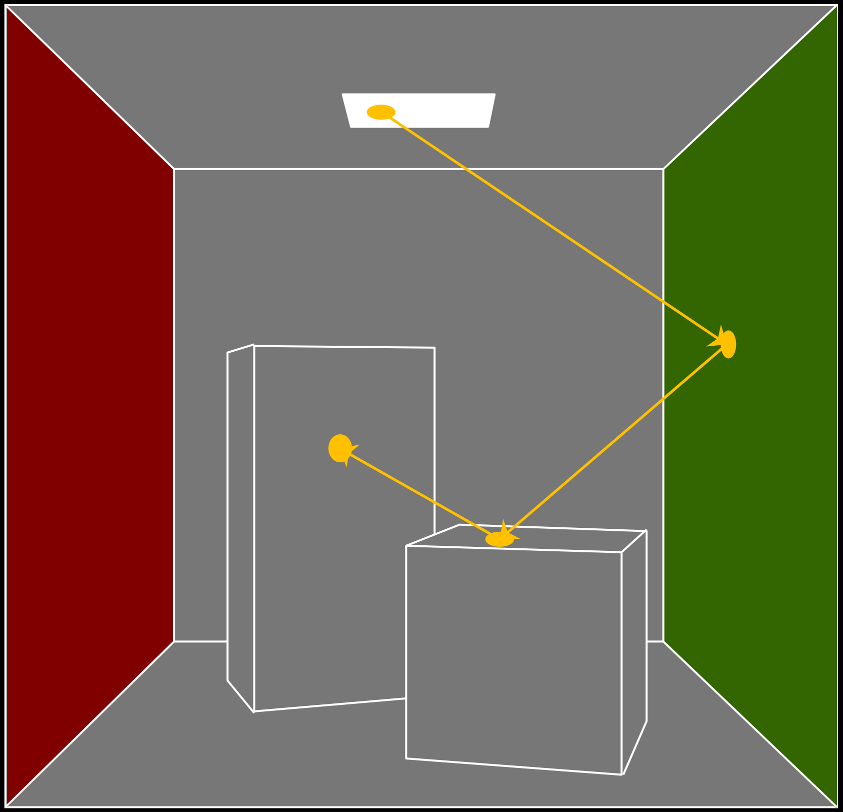
Outline

- Problems with Virtual Point Lights (VPLs)
- Our solution: Virtual Spherical Lights (VSLs)
- Implementation
- Results

Outline

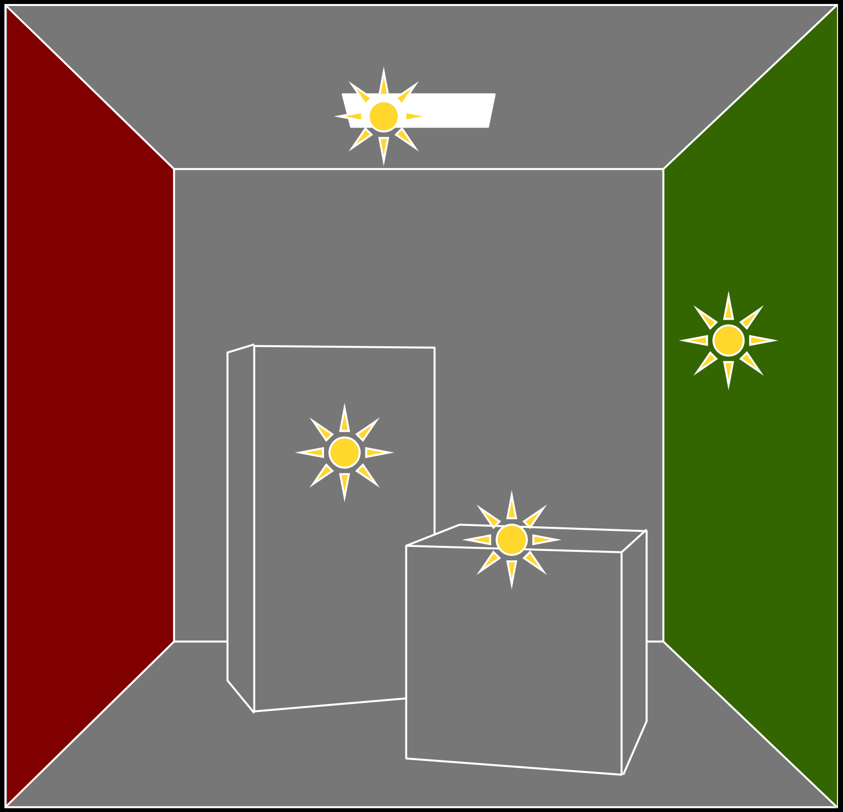
- Problems with Virtual Point Lights (VPLs)
- Our solution: Virtual Spherical Lights (VSLs)
- Implementation
- Results

Instant Radiosity



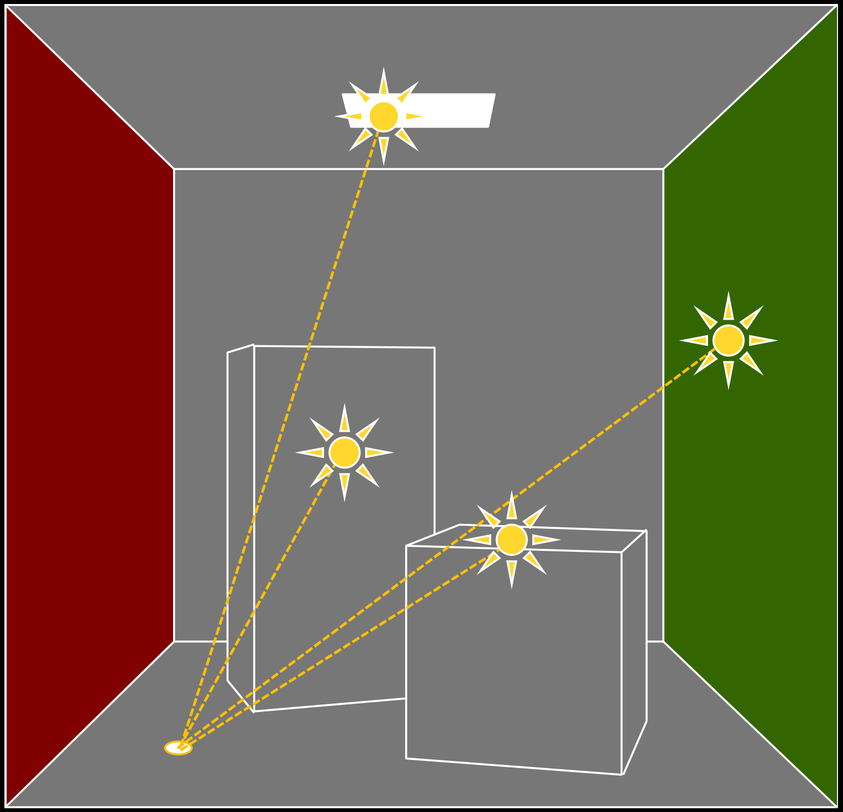
- **STEP 1**
  - Trace paths from the light

Instant Radiosity



- **STEP 1**
  - Trace paths from the light
  - Treat path vertices as Virtual Point Lights (VPLs)

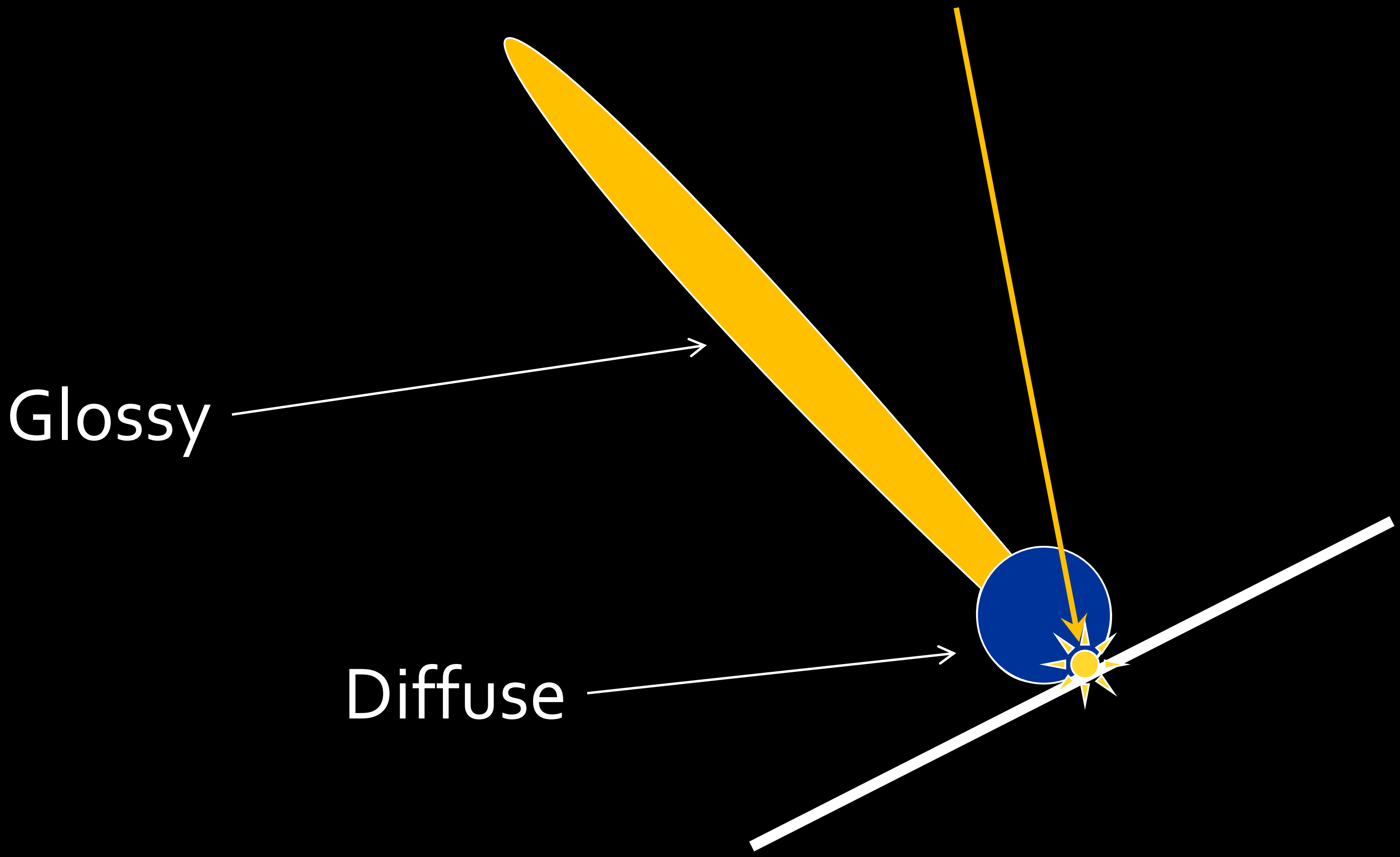
Instant Radiosity



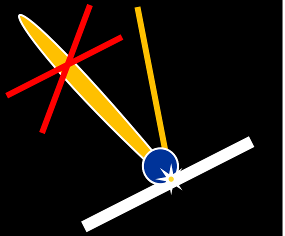
- STEP 1
  - Trace paths from the light
  - Treat path vertices as Virtual Point Lights (VPLs)
- STEP 2
  - Render scene with VPLs

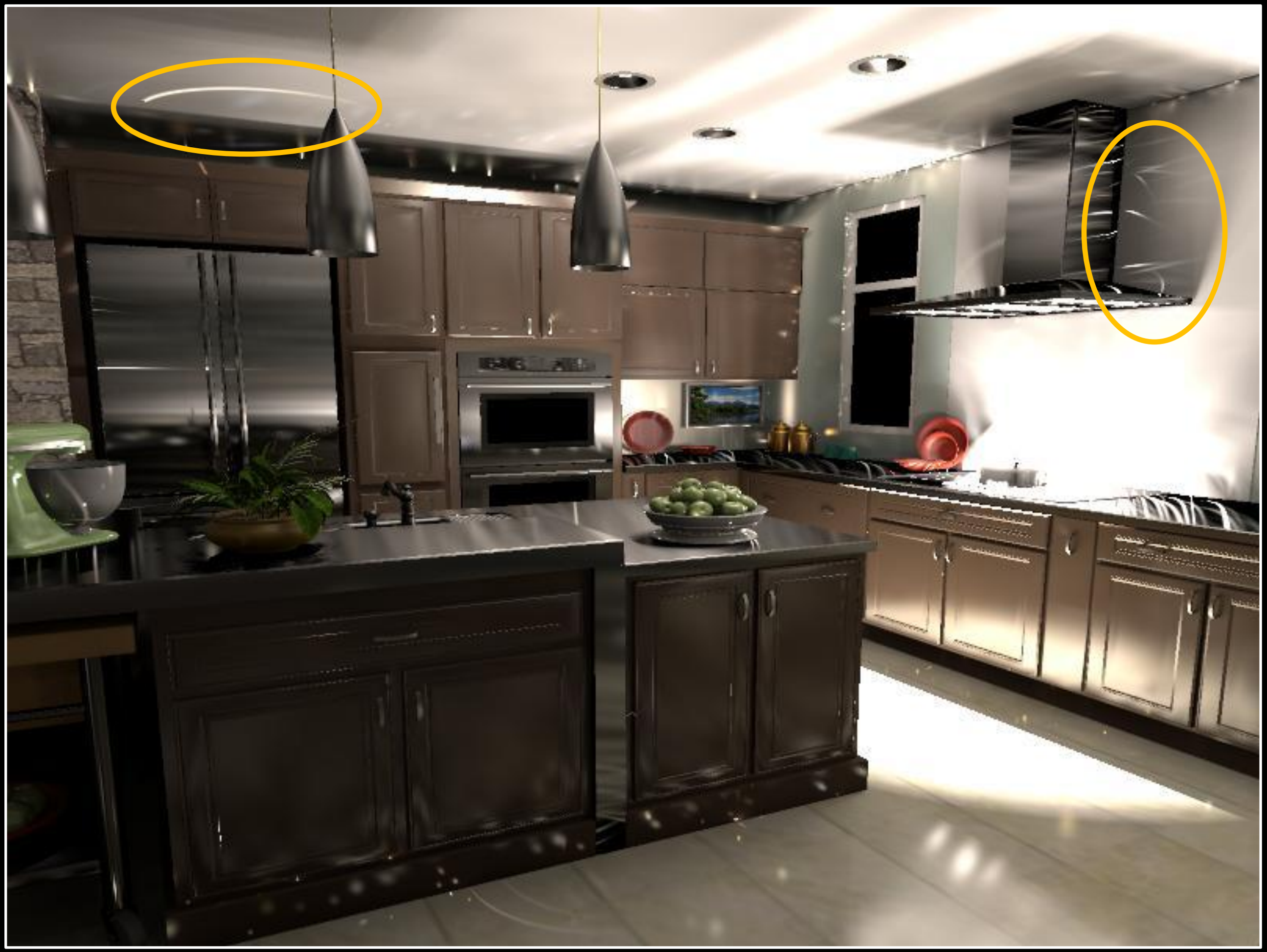
Emission Distribution of a VPL

- Cosine-weighted BRDF lobe at the VPL location



Glossy VPL Emission: Illumination Spikes

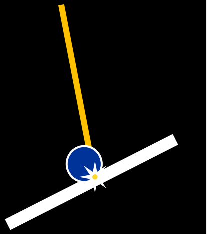


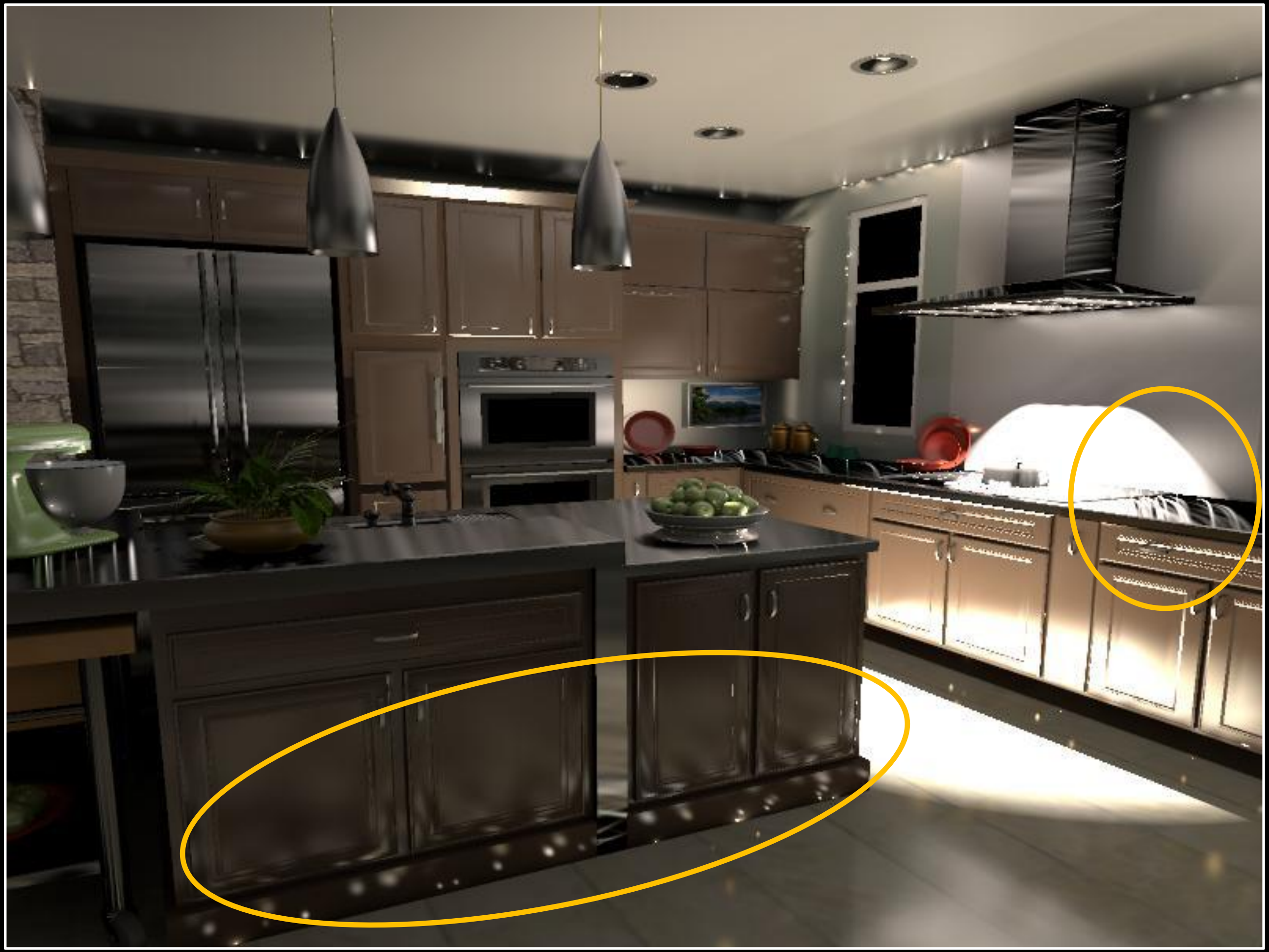


Common solution:

Only **diffuse** BRDF at light location

Remaining Spikes

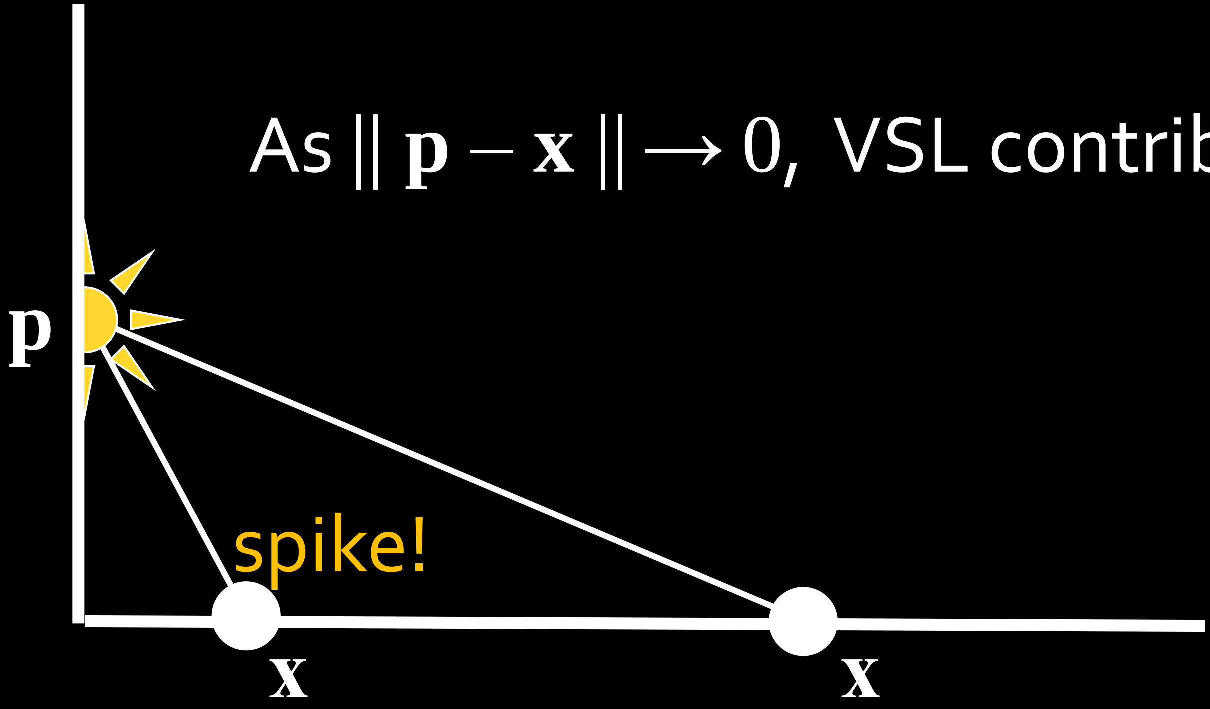




Remaining Spikes

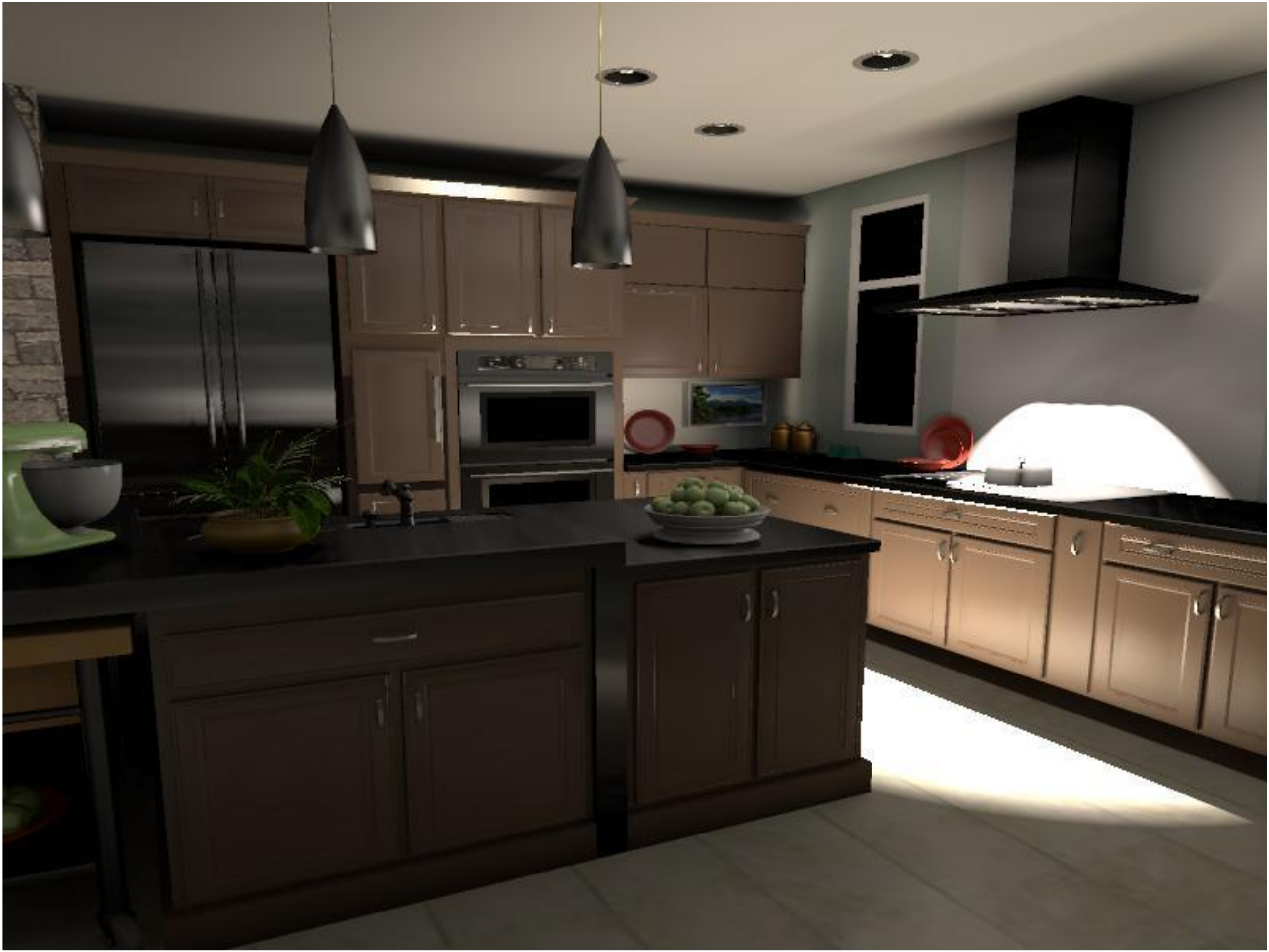
- VPL contribution =

$$\text{As } \| \mathbf{p} - \mathbf{x} \| \to 0, \text{ VSL contribution} \to \infty$$



- Common solution: **Clamp** VPL contributions

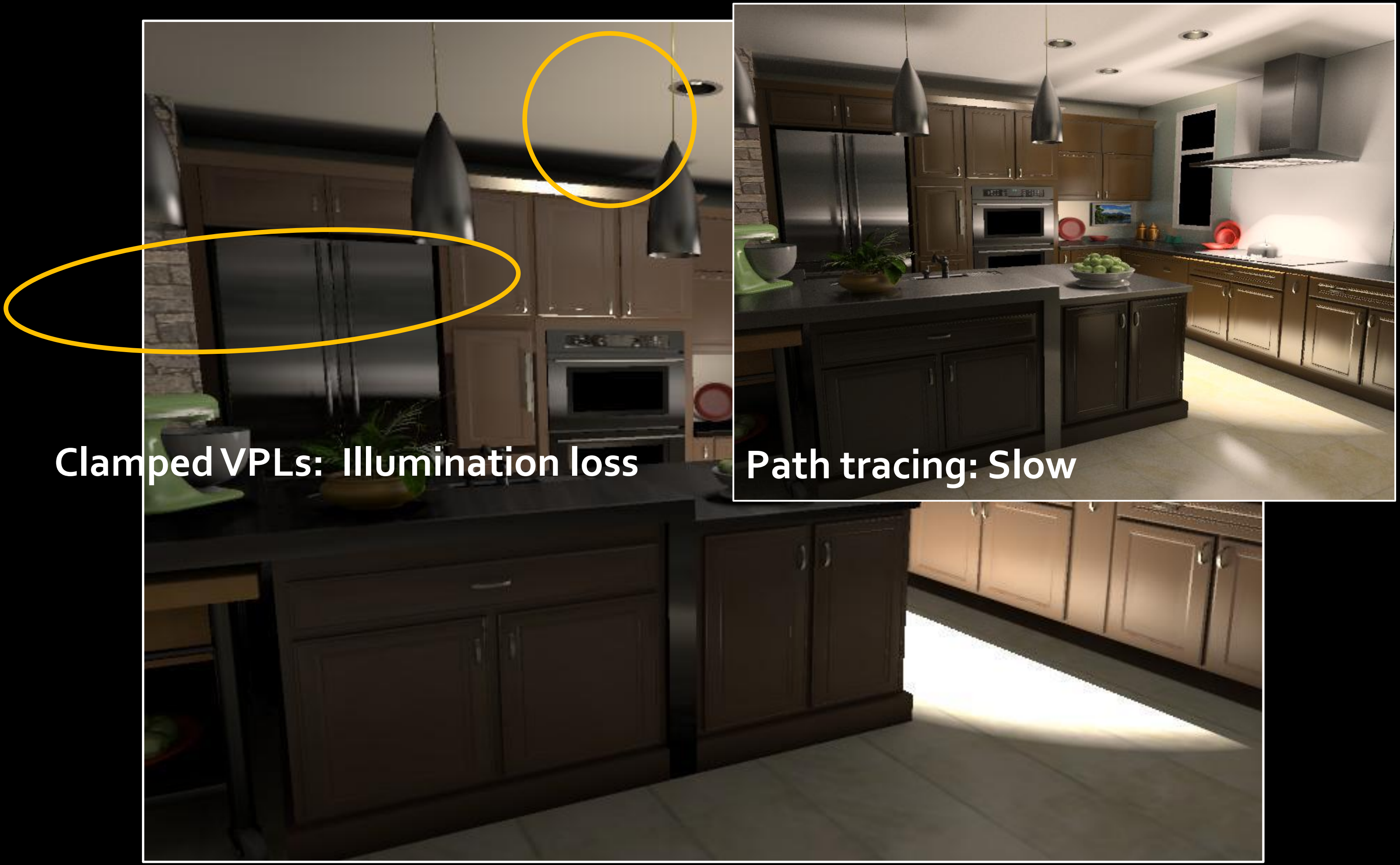
Instant Radiosity: The Practical Version



Clamping and diffuse-only VPLs:

Illumination is lost!

Comparison



Clamped VPLs: Illumination loss

Path tracing: Slow

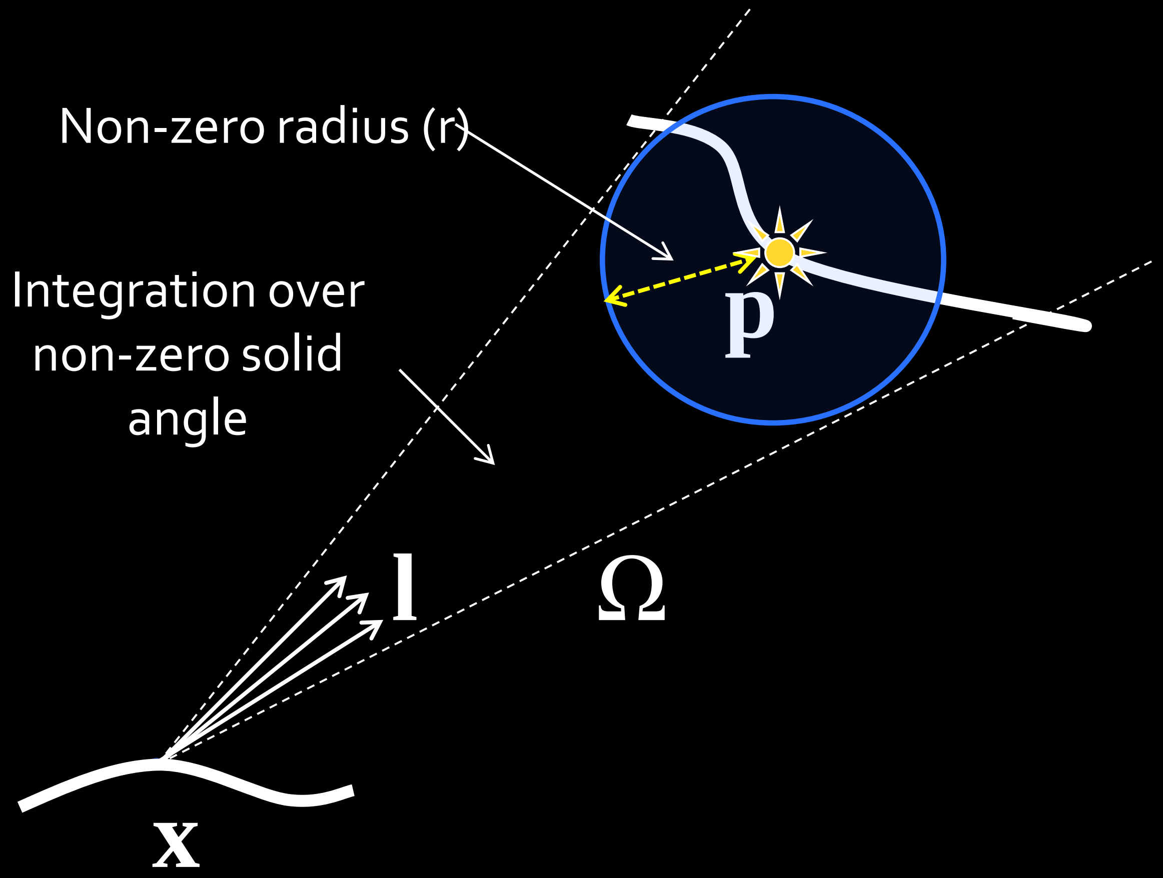
Outline

- Problems with Virtual Point Lights (VPLs)
- Our solution: Virtual Spherical Lights (VSLs)
- Implementation
- Results

Motivation

- VPLs: image splotches due to
  - Spikes in the VPL emission distibution
  - $1 \,/\, \| \mathbf{p} - \mathbf{x} \|$ term
- Idea
  - Spread VPL energy over a finite surface
  - Compute contribution as solid angle integral

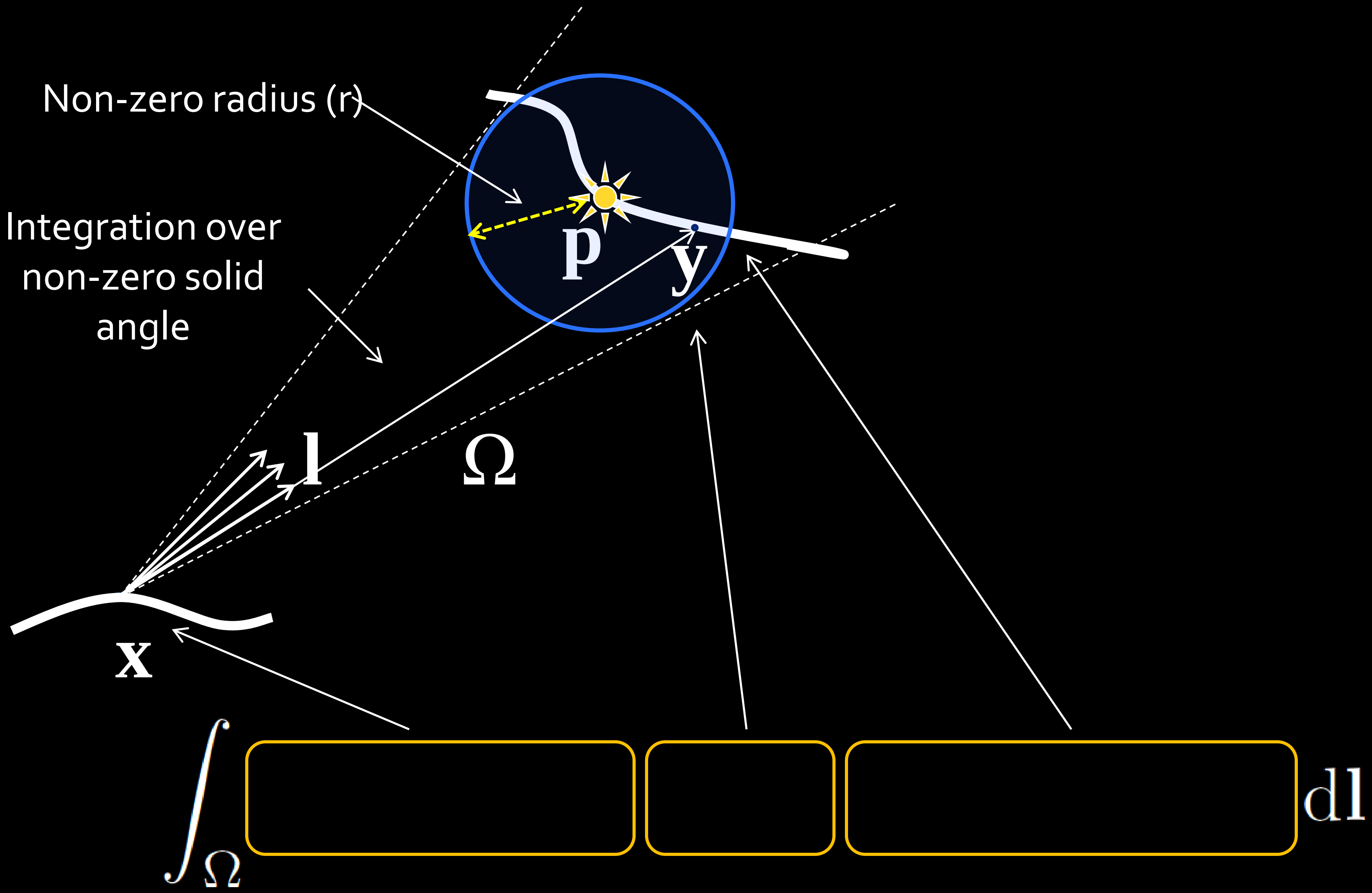
VPL to VSL



Non-zero radius (r)

Integration over non-zero solid angle

Light Contribution

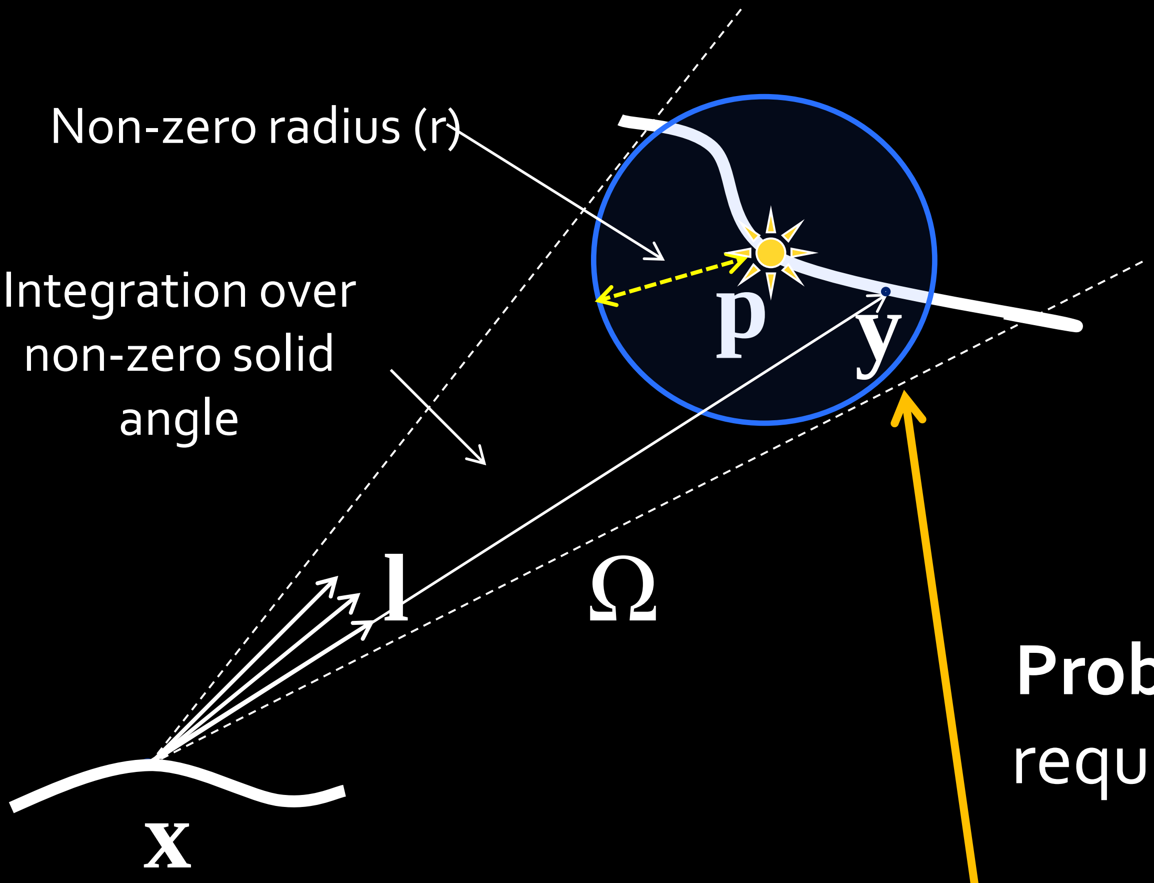


Non-zero radius (r)

Integration over non-zero solid angle

$$\int_\Omega \boxed{}\;\boxed{}\;\boxed{}\;\mathrm{dl}$$

Light Contribution



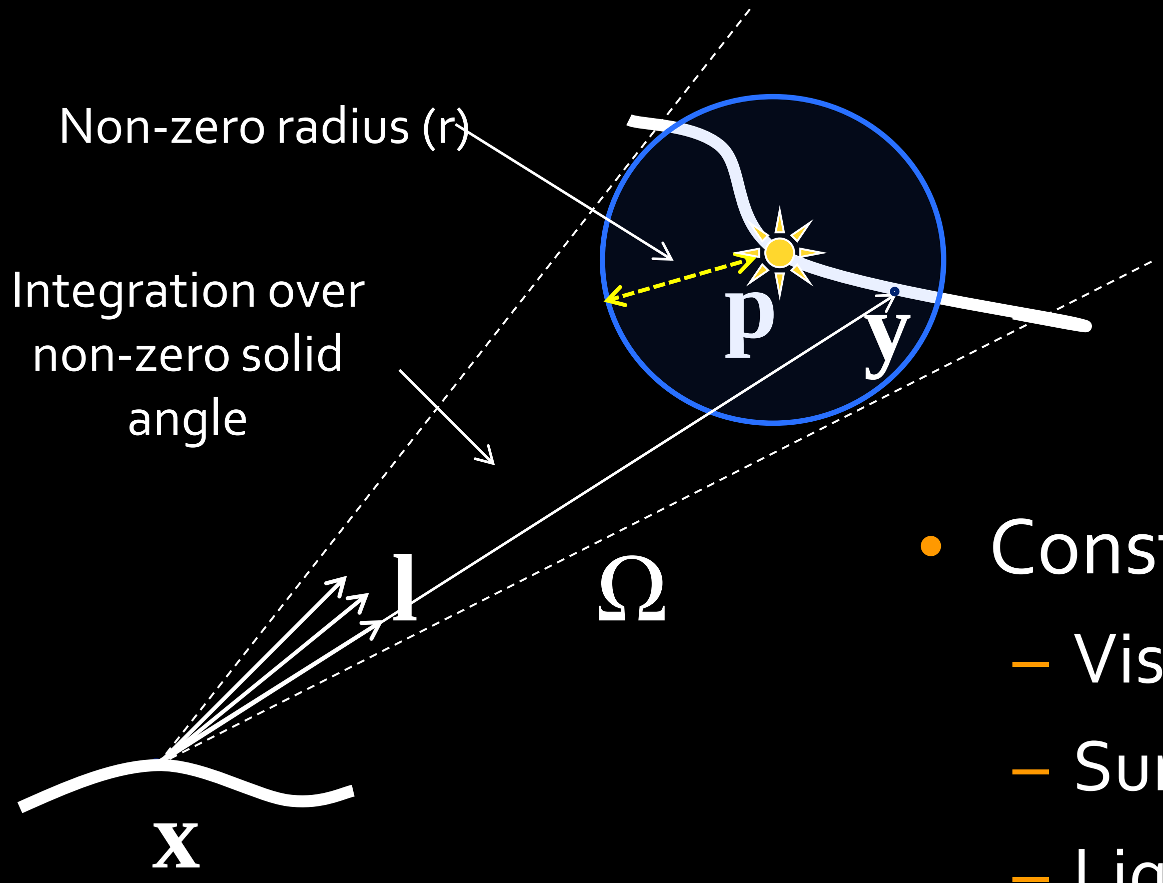
Non-zero radius (r)

Integration over non-zero solid angle

Problem: Finding **y** requires ray-tracing

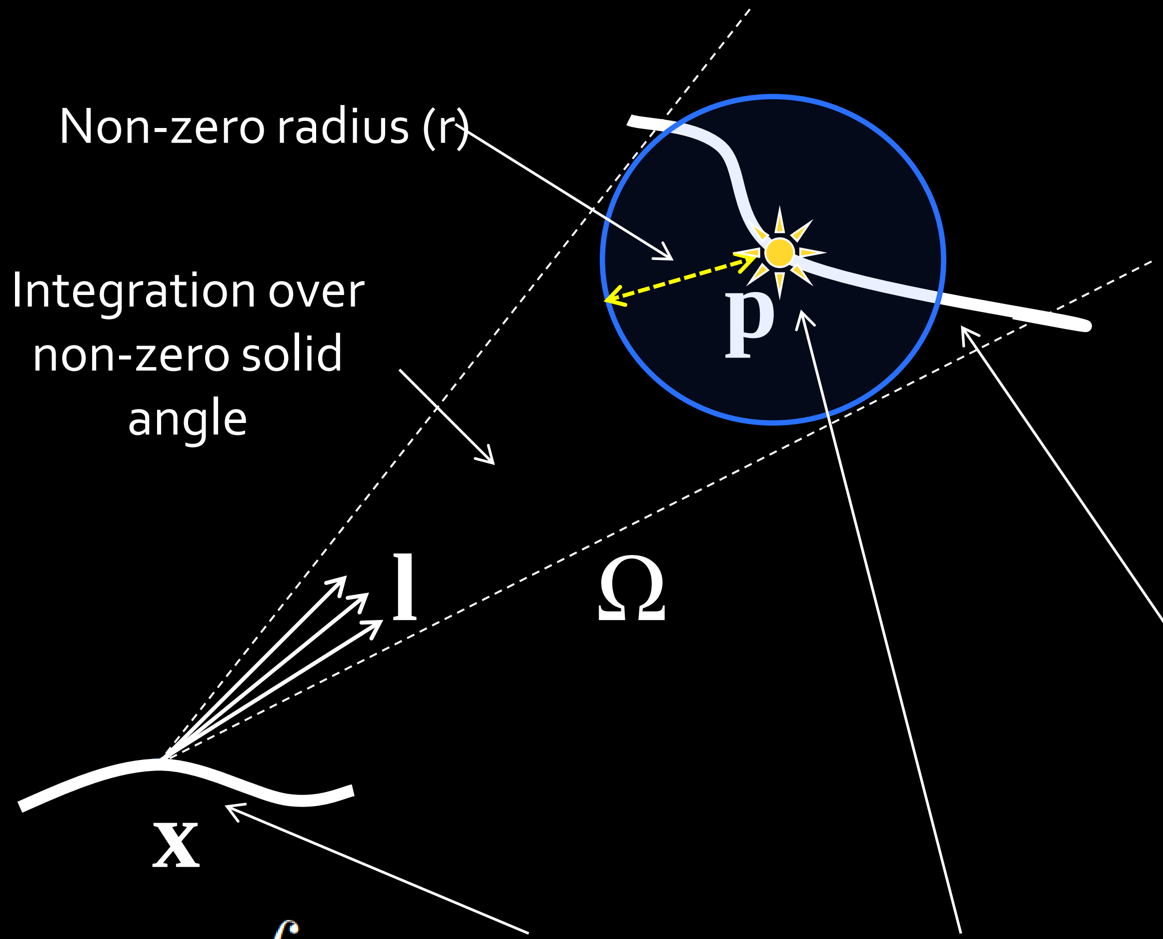
$$\frac{\Phi}{\pi r^2}\int_{\Omega} f_r(\mathbf{x})\cos\theta_{\mathbf{x}}\; f_r(\boxed{\mathbf{y}})\;(\|\mathbf{p}-\boxed{\mathbf{y}}\|<r)\mathrm{dl}$$

Simplifying Assumptions



- Constant in Ω:
  - Visibility
  - Surface normal
  - Light BRDF
- Taken from **p**, the light location

Light Contribution Updated



Non-zero radius (r)

Integration over non-zero solid angle

$$V \int_{\Omega} \boxed{} \boxed{f_r(\mathbf{p})} \boxed{\quad \cos\theta_{\mathbf{p}} \quad} \mathrm{d}\mathbf{l}$$

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

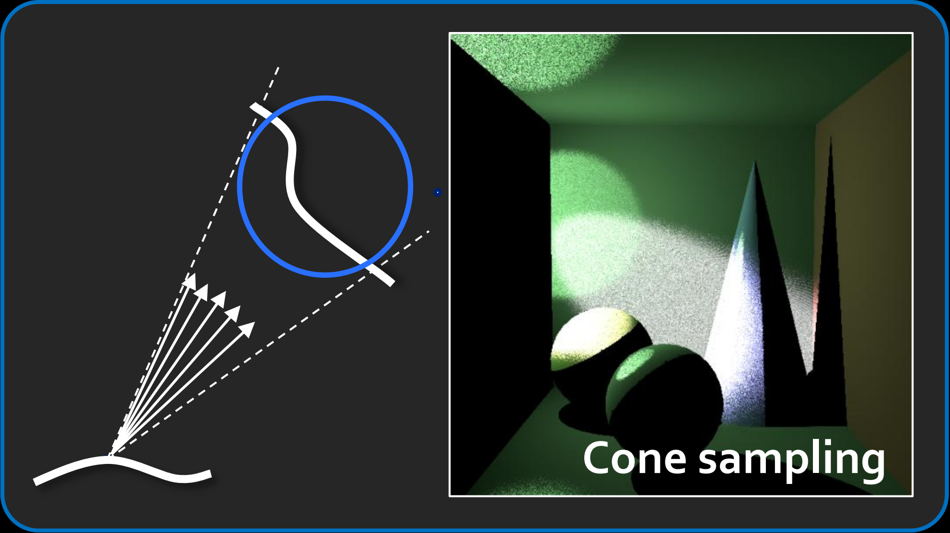
$$V \frac{\Phi}{\pi r^2} \int_{\Omega} f_r(\mathbf{x}) \cos\theta_{\mathbf{x}} \; f_r(\mathbf{p}) \;\; \cos\theta_{\mathbf{p}} \; \mathrm{d}\mathbf{l}$$

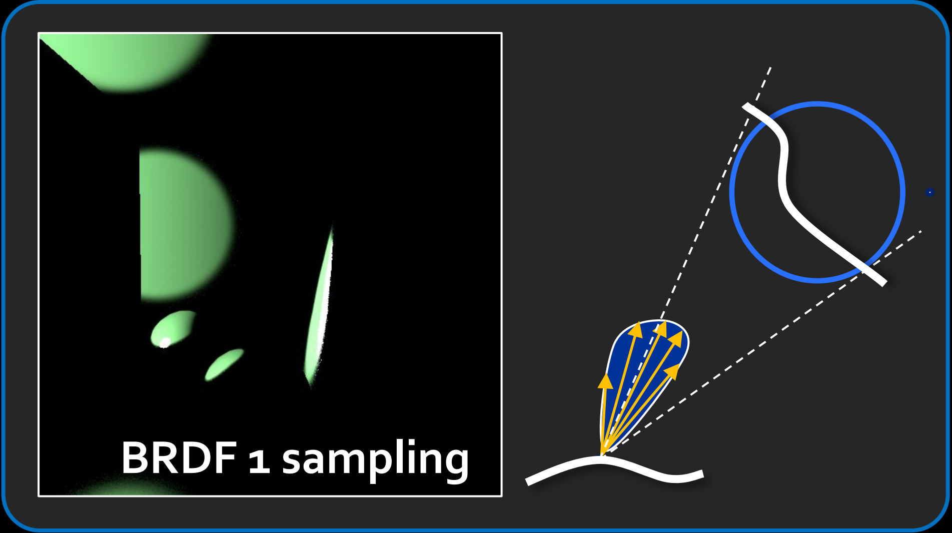
Outline

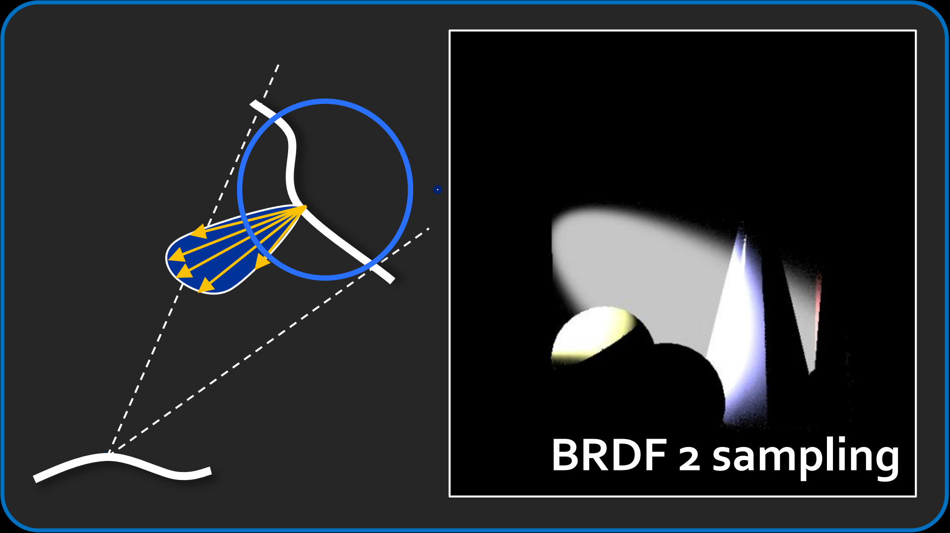
- Problems with Virtual Point Lights (VPLs)
- Our solution: Virtual Spherical Lights (VSLs)
- Implementation
- Results

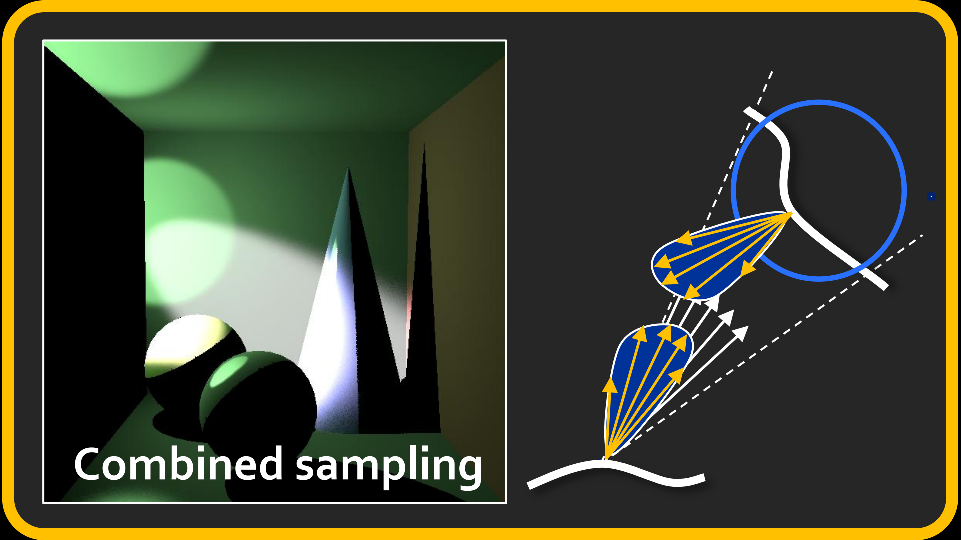
Computing the VSL integral

- Monte Carlo quadrature


Cone sampling


BRDF 1 sampling


BRDF 2 sampling


Combined sampling

Implementation

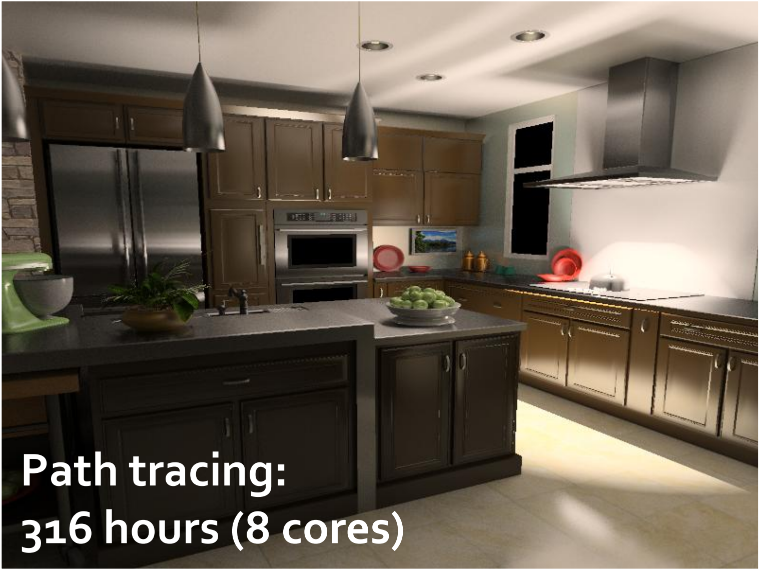
- Matrix row-column sampling [Hašan et al. 2007]
  - Shadow mapping for visibility
  - VSL integral evaluated in a GPU shader
- Need more lights than in diffuse scenes
- VSL radius proportional to local VSL density
  - determined by k-NN queries

Outline

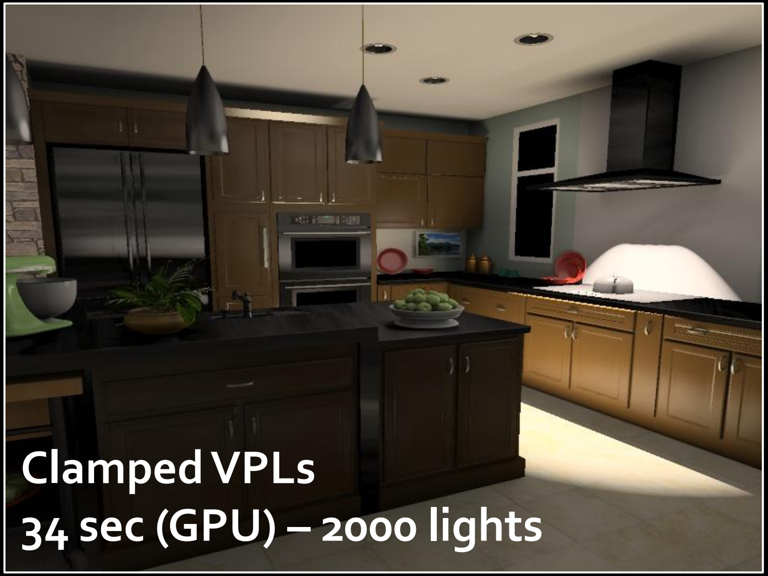
- Problems with Virtual Point Lights (VPLs)
- Our solution: Virtual Spherical Lights (VSLs)
- Implementation
- Results

Results: Kitchen

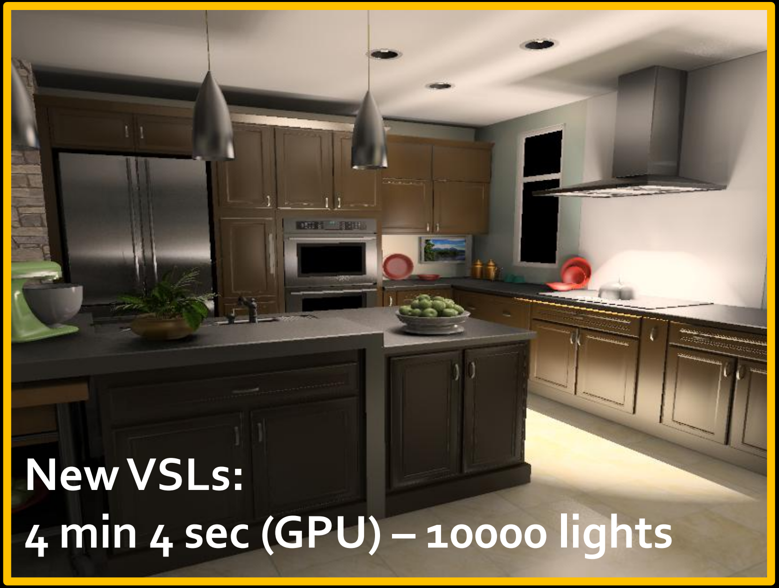
- Most of the scene lit indirectly
- Many materials glossy and anisotropic



Path tracing:
316 hours (8 cores)



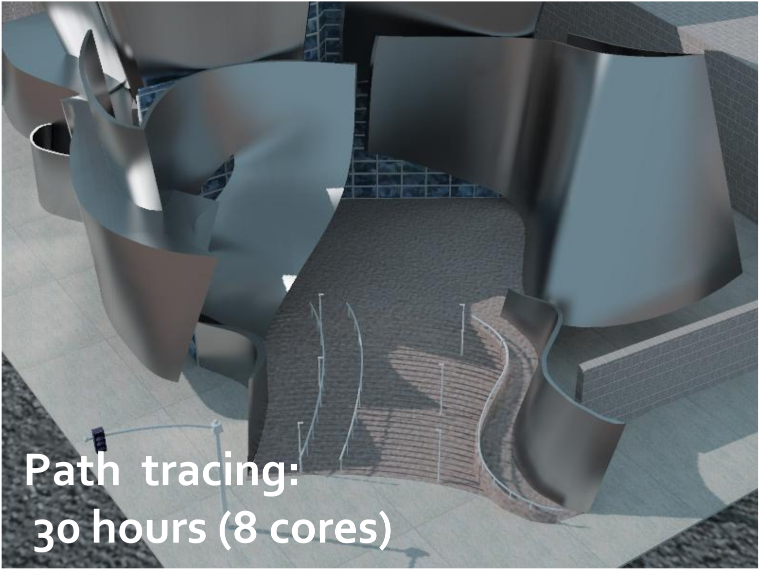
Clamped VPLs
34 sec (GPU) – 2000 lights



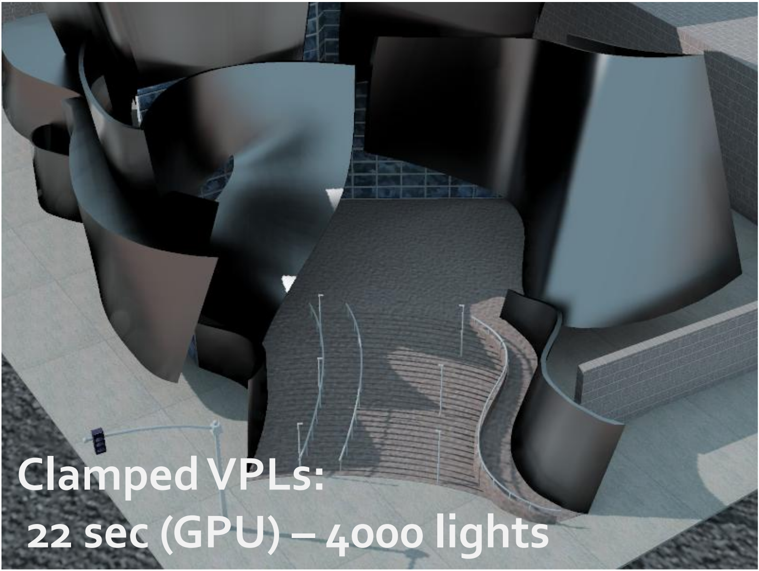
New VSLs:
4 min 4 sec (GPU) – 10000 lights

Results: Disney Concert Hall

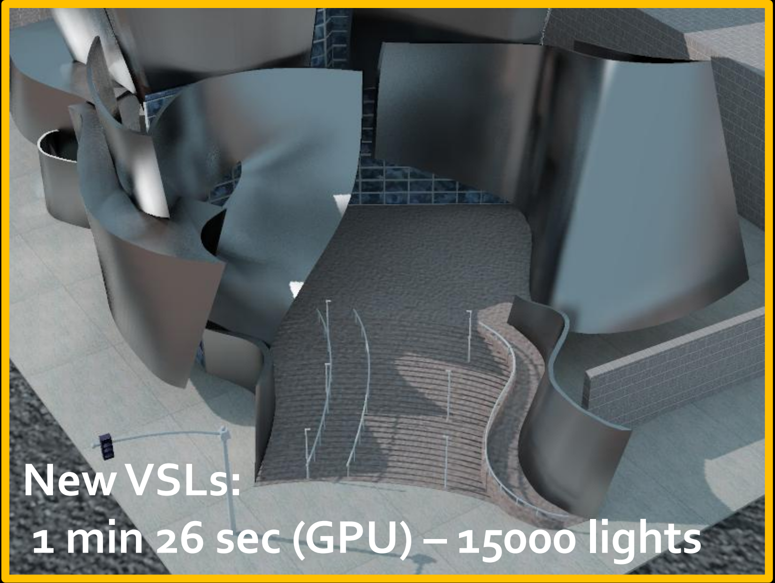
- Curved walls with no diffuse component
- Standard VPLs cannot capture any reflection from walls



Path tracing:
30 hours (8 cores)



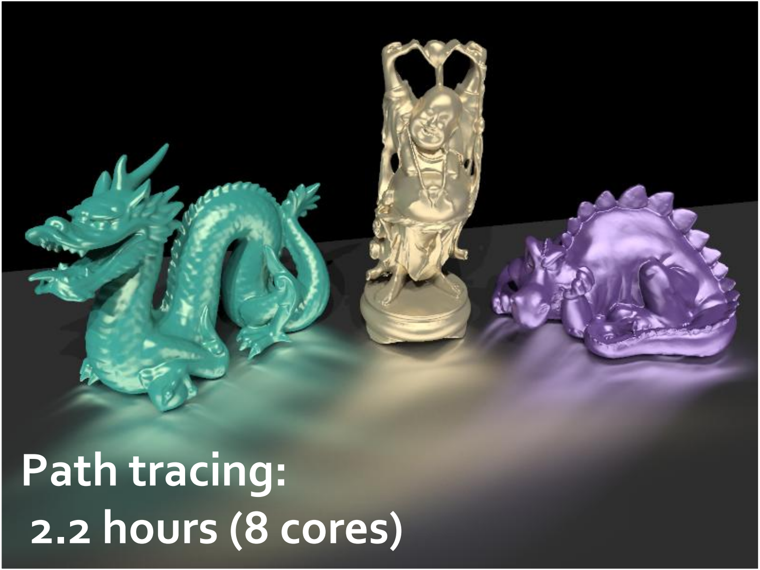
Clamped VPLs:
22 sec (GPU) – 4000 lights



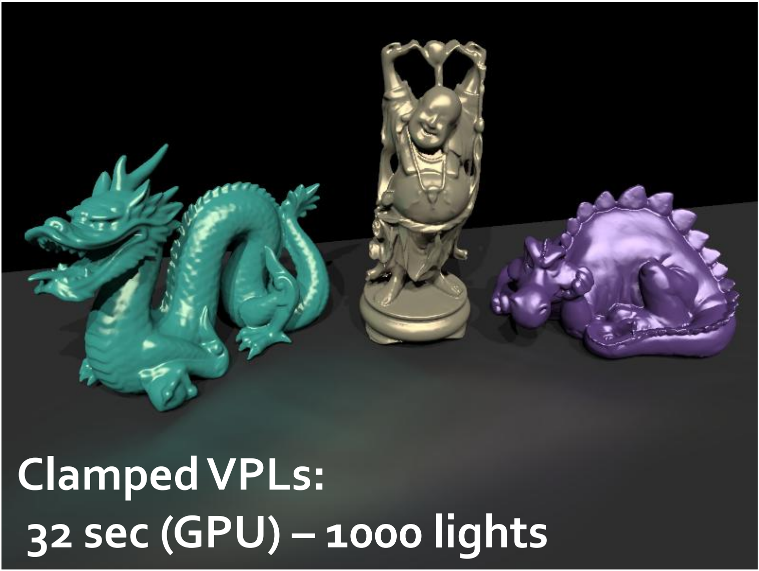
New VSLs:
1 min 26 sec (GPU) – 15000 lights

Results: Anisotropic Tableau

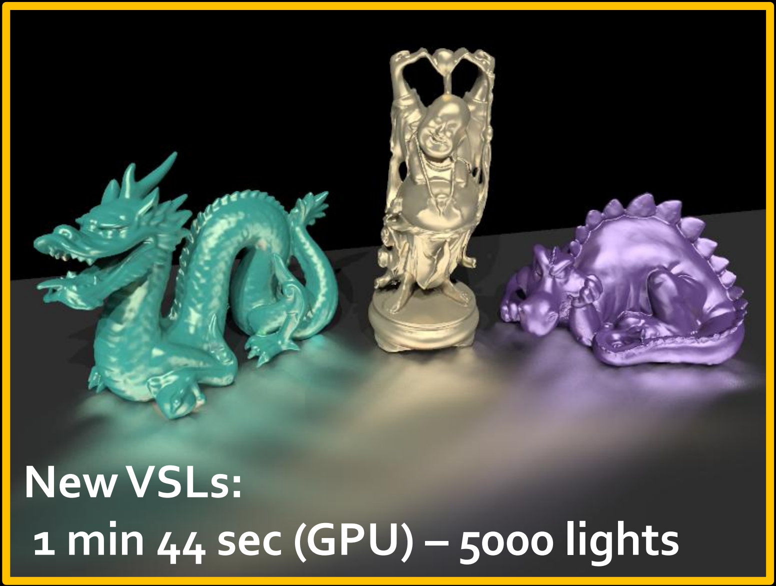
- Difficult case
- Standard VPLs capture almost no indirect illumination



Path tracing:
2.2 hours (8 cores)

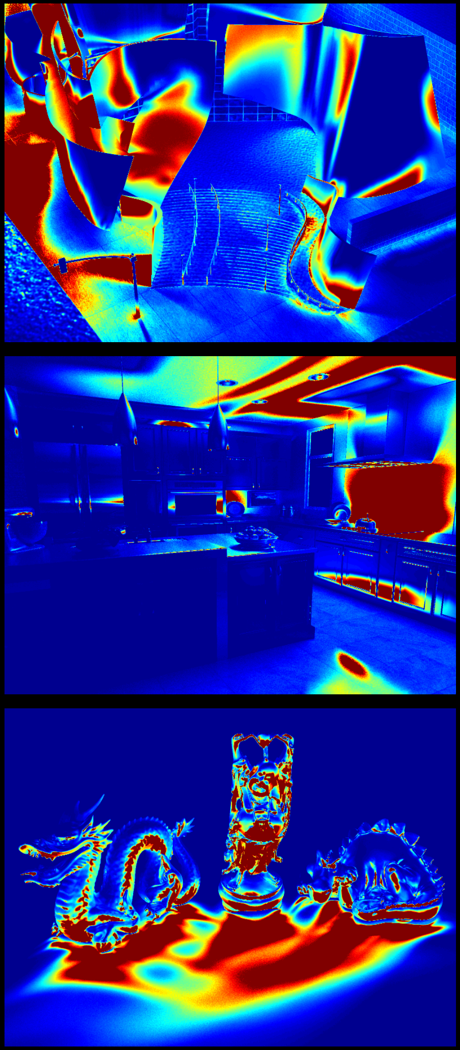


Clamped VPLs:
32 sec (GPU) – 1000 lights

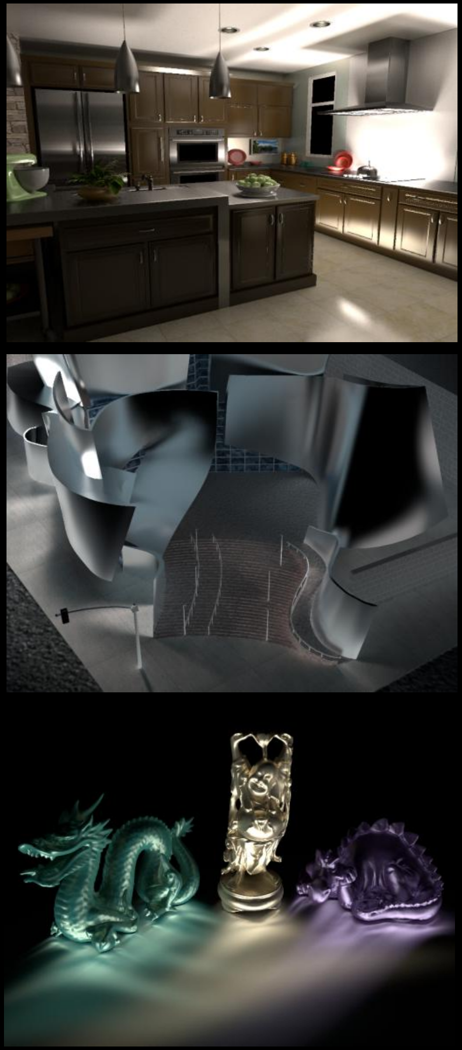


New VSLs:
1 min 44 sec (GPU) – 5000 lights

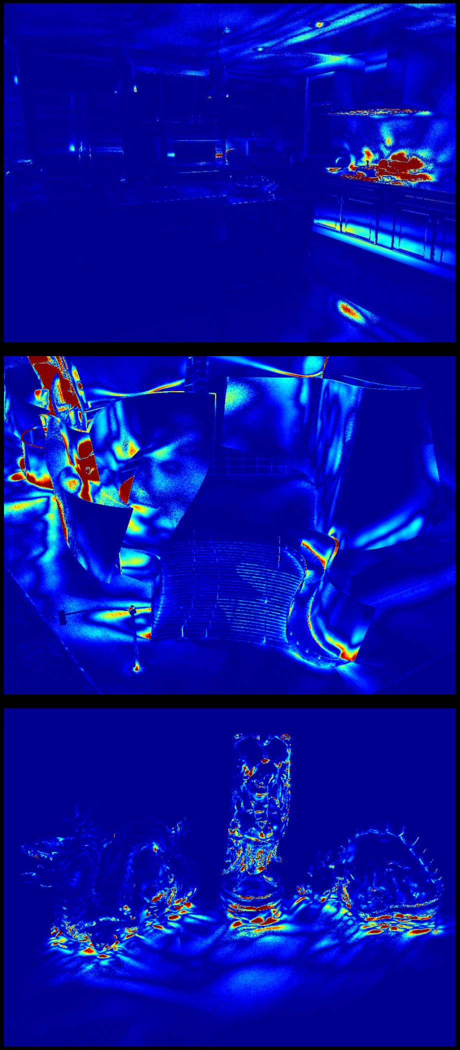
Error Images (Indirect Only)



VPL error
(previous work)



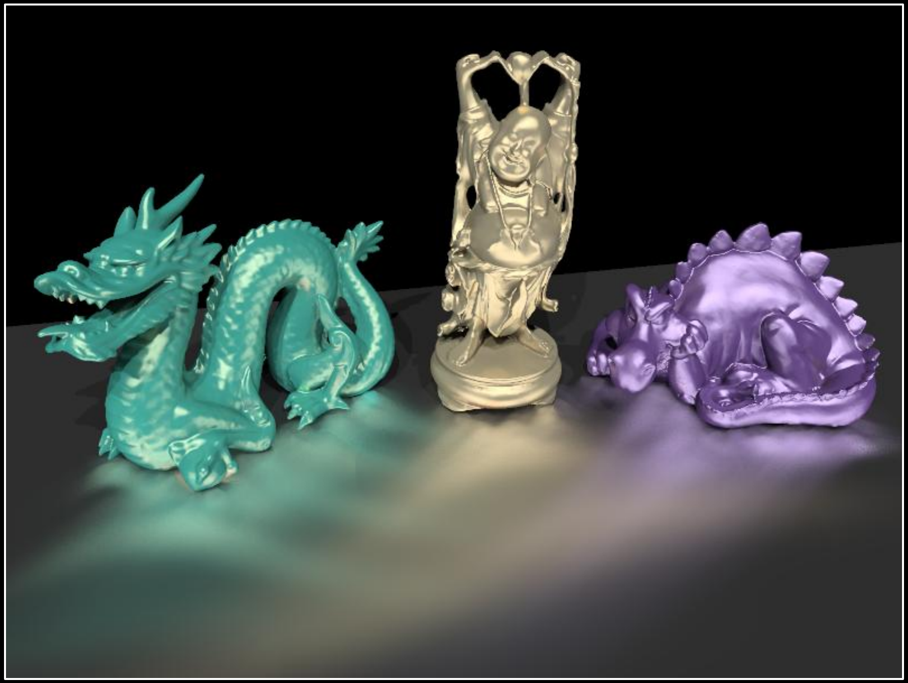
Ground truth



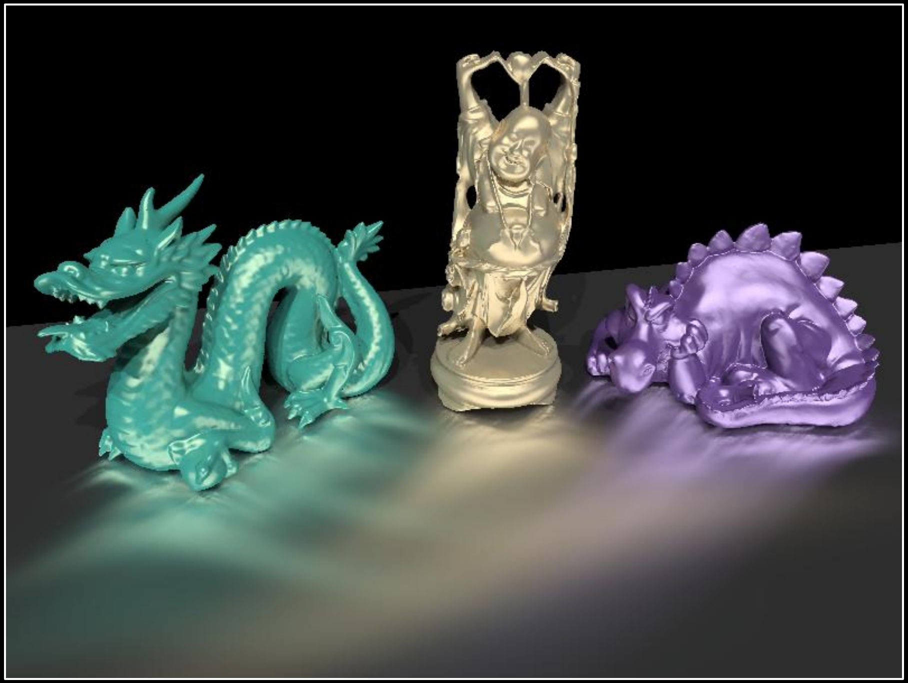
VSL error
(our method)

Limitations: Blurring

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



5,000 lights - blurred



1,000,000 lights - converged

Other Limitations

- Some remaining corner darkening
- Computation overhead

Conclusion

- Virtual Spherical Lights
  - No spikes, no clamping necessary
  - Address illumination loss
- Many-light methods + VSLs:
  - A step to solve the glossy inter-reflection problem
- Future Work
  - More lights: improve scalability

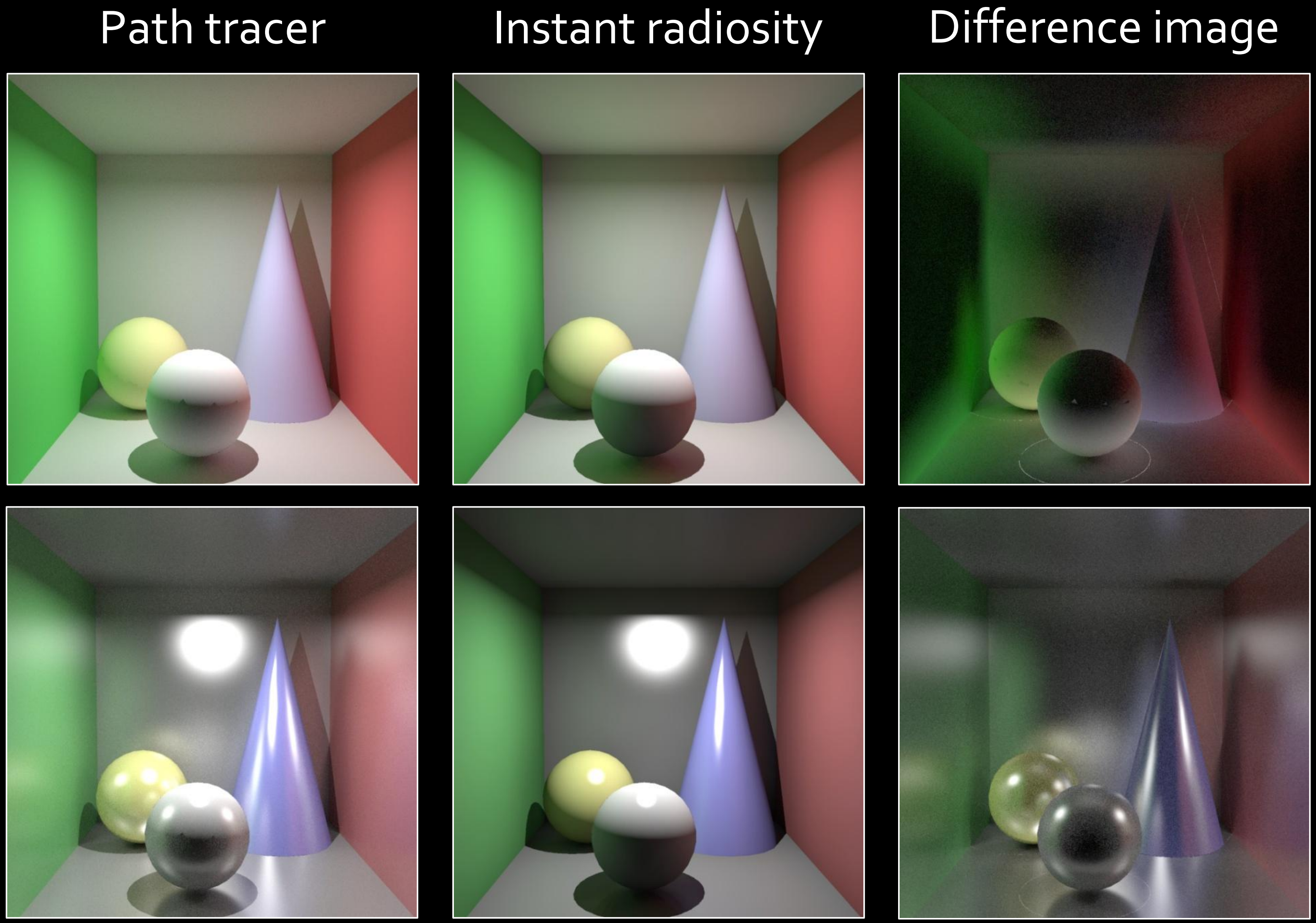
The Problem, Numerically

- Recall: Integration over paths, use Monte Carlo

$$\int_{\Omega} f_j(\bar{x})d\mu(\bar{x}) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

- The contribution *$f(x_i)$ contains:*
  - Inverse distance-squared term
  - Material term at surface location
  - Material term at VPL location
- What if *$f(x_i)$* becomes locally large?
  - “Spikes”

The Problem Revision

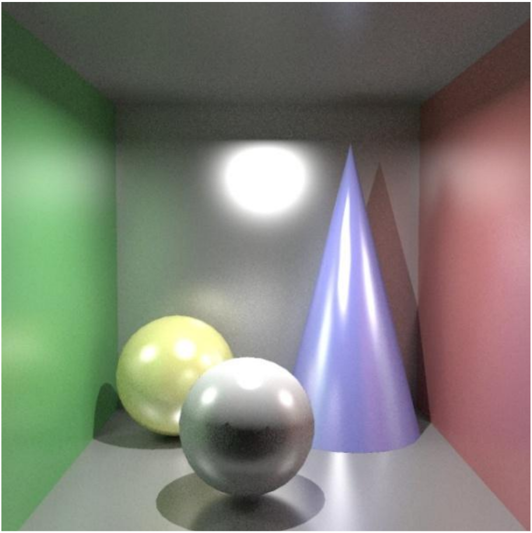


Path tracer

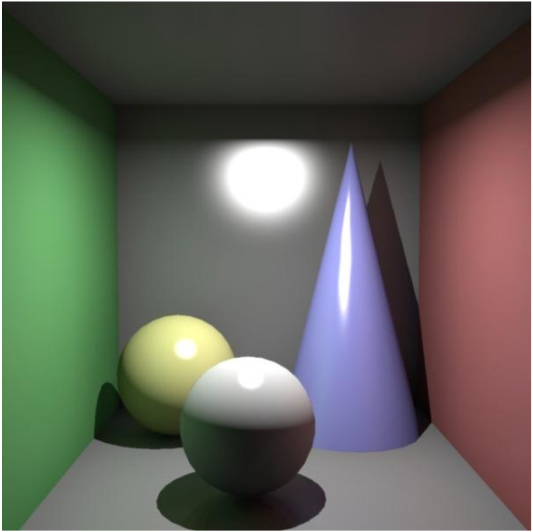
Instant radiosity

Difference image

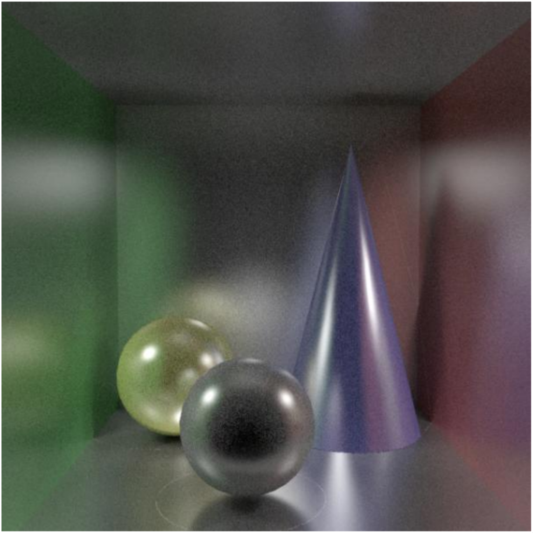
Another Example



Path tracer

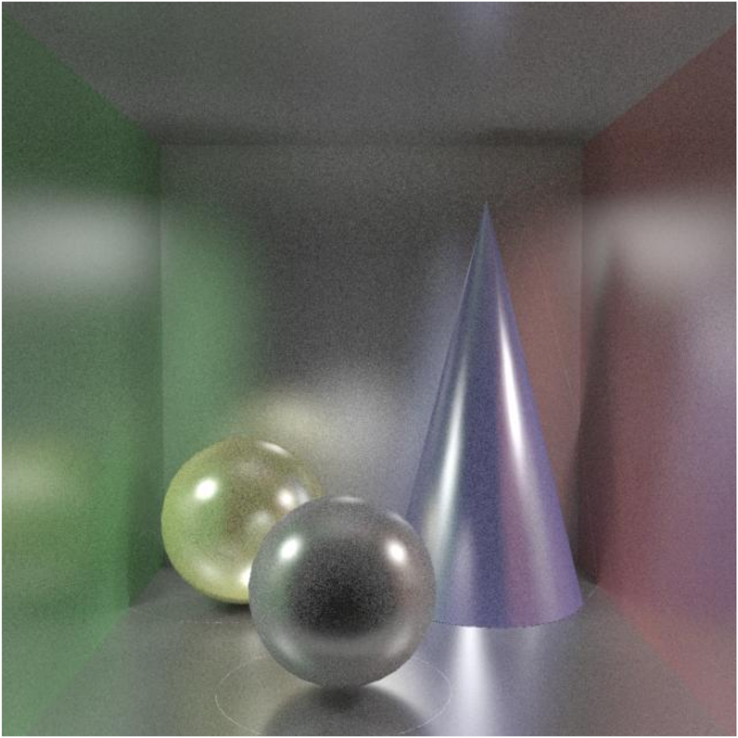


Instant radiosity

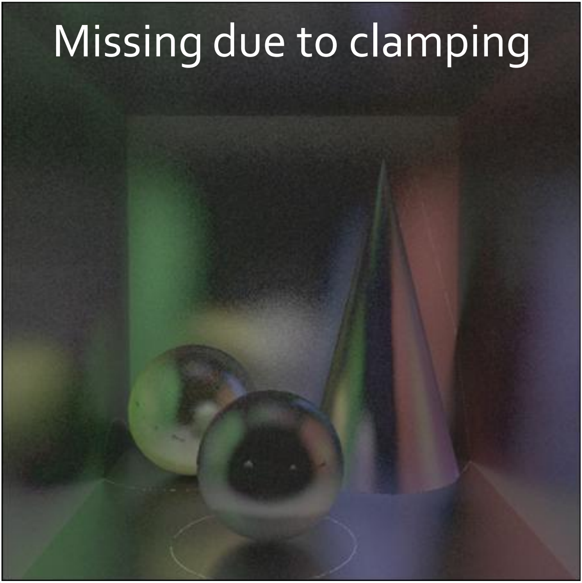


Difference image

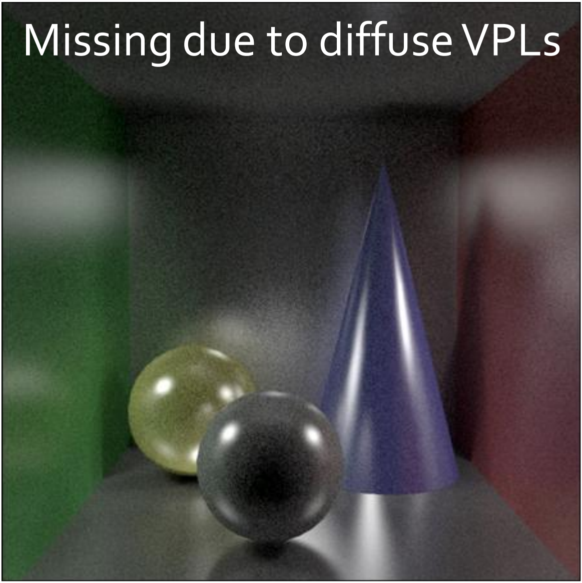
The Missing Components



Missing energy



Missing due to clamping



Missing due to diffuse VPLs