Many-light methods – Clamping & compensation

Jaroslav Křivánek

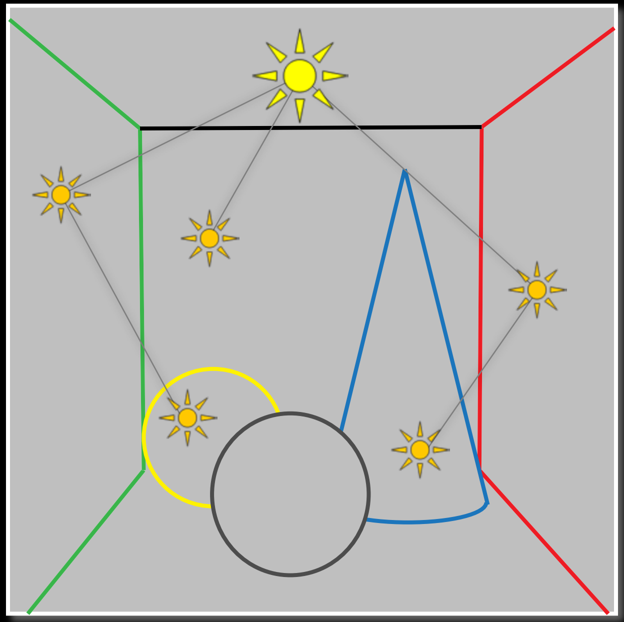
Charles University, Prague

Instant radiosity

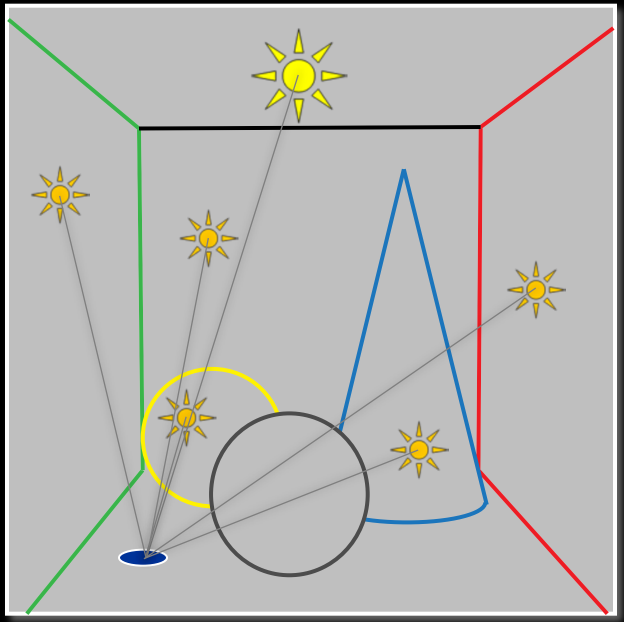
- Approximate indirect illumination by

  Virtual Point Lights (VPLs)

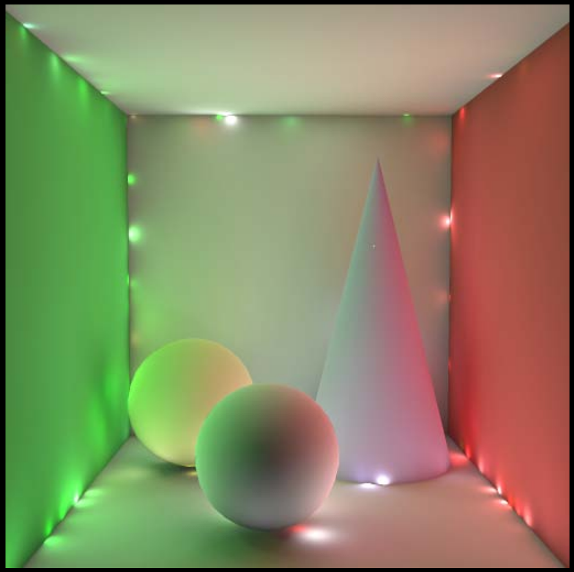
1. Generate VPLs



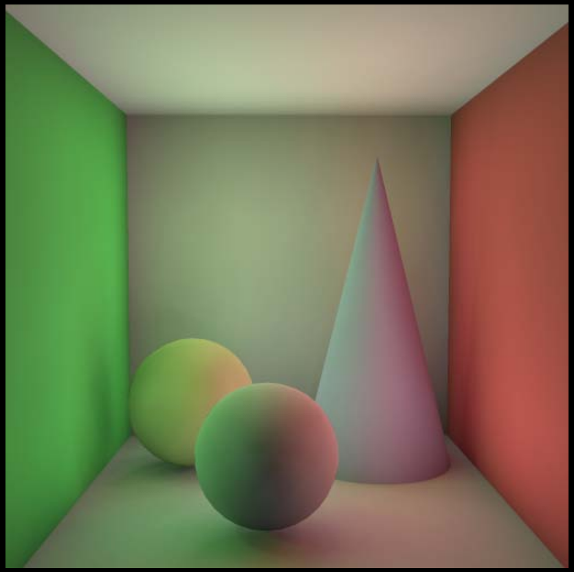
2. Render with VPLs



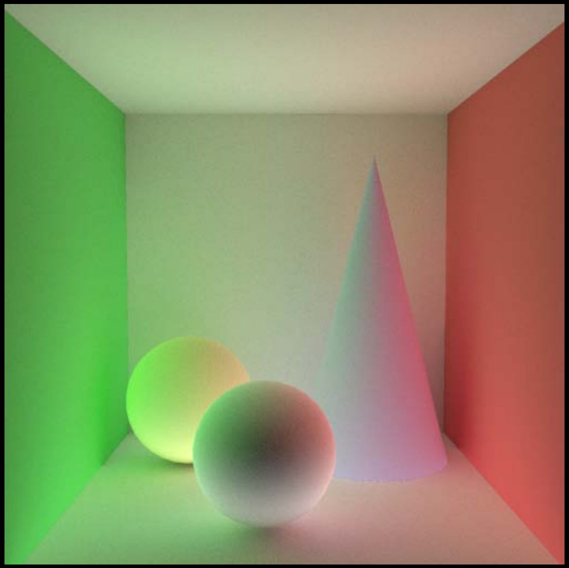
Clamping



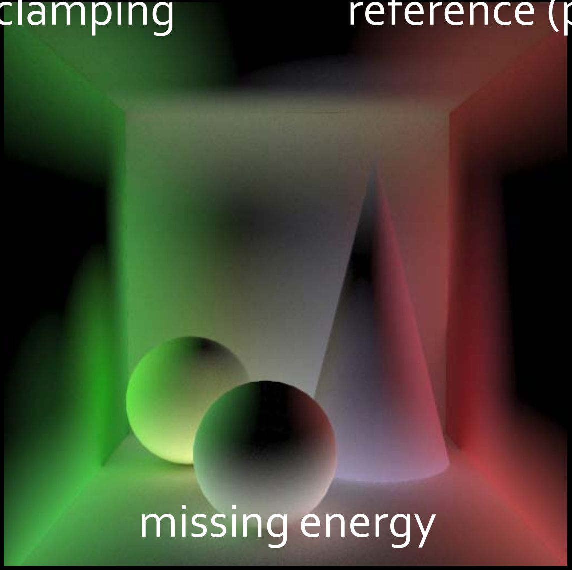
1000 VPLs - no clamping



1000 VPLs - clamping



reference (path tracing)

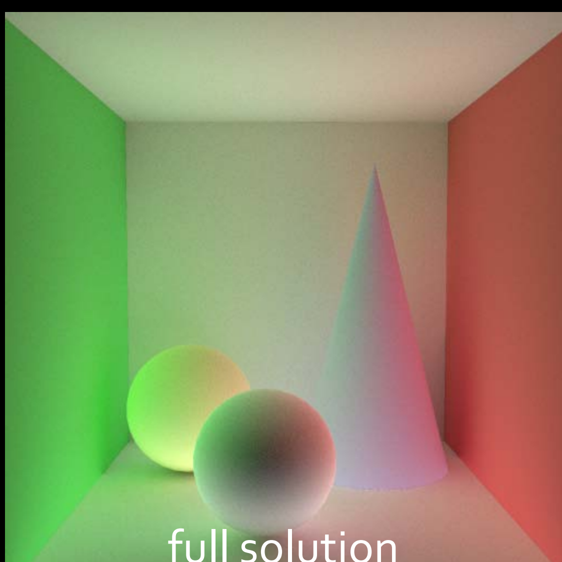


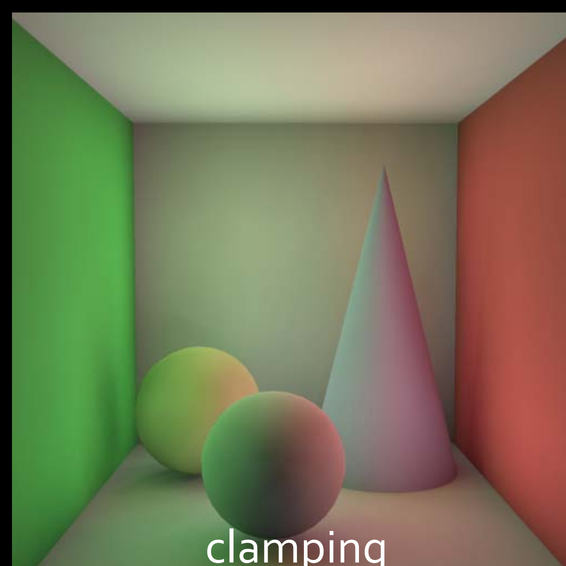
Clamping Compensation

Kollig & Keller, MCQMC 2004

Idea

- Clamping reduces variance but some energy is lost
- Find formula for the lost energy
- Compute the lost energy by selective path tracing


full solution


clamping


path tracing compensation

Clamping

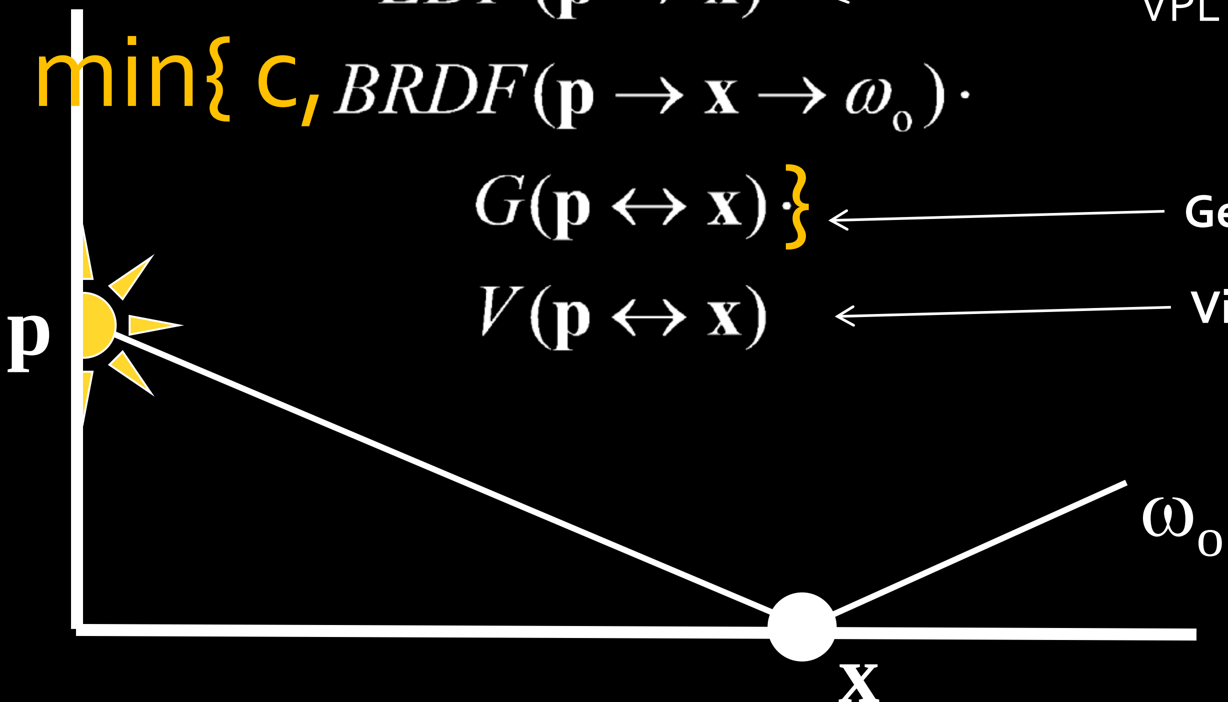
$$\text{VPL contrib} = \Phi \cdot EDF(\mathbf{p} \rightarrow \mathbf{x}) \cdot \min\{\, c,\ BRDF(\mathbf{p} \rightarrow \mathbf{x} \rightarrow \omega_o) \cdot G(\mathbf{p} \leftrightarrow \mathbf{x}) \,\} \cdot V(\mathbf{p} \leftrightarrow \mathbf{x})$$

VPL power

VPL emission distribution
(BRDF lobe at p – for a diffuse VPL can be folded into Φ)

Geometry term

Visibility



Formal derivation

- Clamping evaluates this equation

$$\int_{\mathcal{M}} L_i(\mathbf{y} \to \mathbf{x})\, V(\mathbf{y} \leftrightarrow \mathbf{x})\, \min\{c, G(\mathbf{y} \leftrightarrow \mathbf{x})\, f_r(\mathbf{y} \to \mathbf{x} \to \omega_o)\}\, \mathrm{d}A_{\mathbf{y}}$$

- Can be written as

$$L_w = \int_{\mathcal{M}} L_i \cdot V \cdot G \cdot f_r \cdot w_1 \cdot \mathrm{d}A_{\mathbf{y}}$$

$$w_1 = \min\left\{1, \frac{c}{G(\mathbf{y} \leftrightarrow \mathbf{x}) \cdot f_r(\mathbf{y}_k \to \mathbf{x} \to \omega_o)}\right\}$$

What’s missing?

$$L_{w'} = \int_{\Omega} L_i \cdot f_r \cdot \cos\theta \cdot (1 - w_1) \cdot \mathrm{d}\omega$$

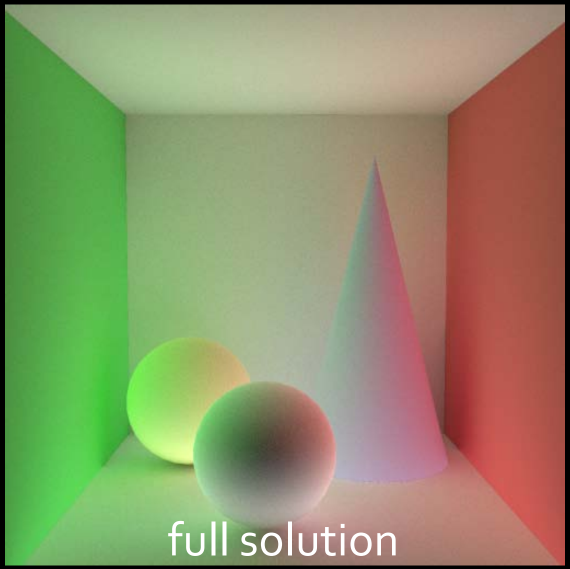
- Unbiased solution

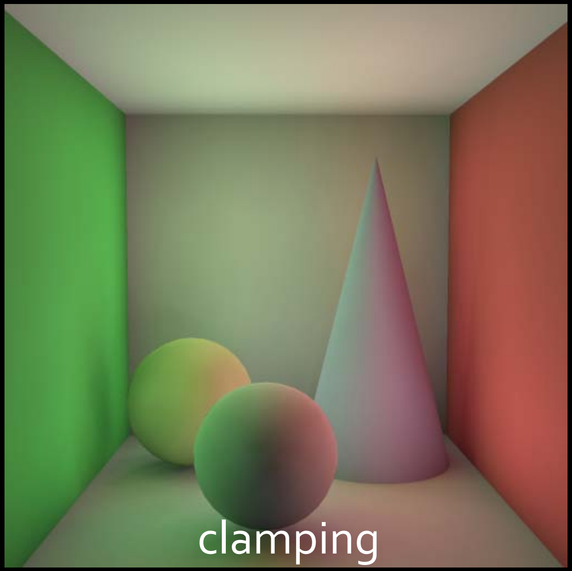
$$L_o = L_w + L_{w'} =$$

$$= \underbrace{\int_{\mathcal{M}} L_i \cdot V \cdot G \cdot f_r \cdot w_1 \cdot \mathrm{d}A_{\mathbf{y}}}_{\text{VPLs w/ clamping}} + \underbrace{\int_{\Omega} L_i \cdot f_r \cdot \cos\theta \cdot (1 - w_1) \cdot \mathrm{d}\omega}_{\substack{\text{Path tracing} \\ \text{compensation of the clamped energy}}}$$

Result

- Compensation faster than path tracing everything (many path terminated early)



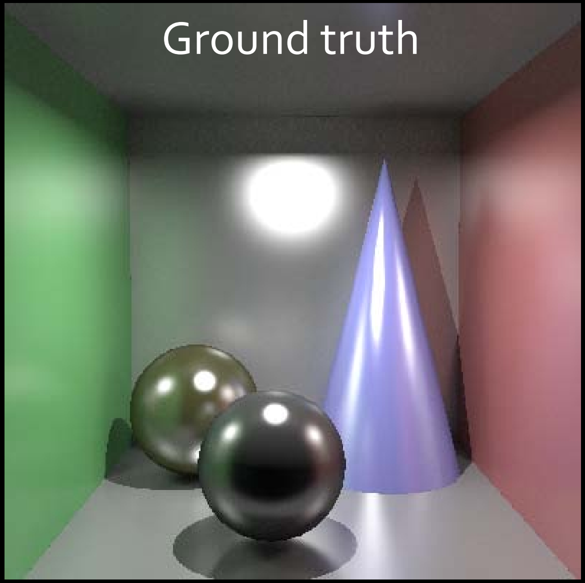


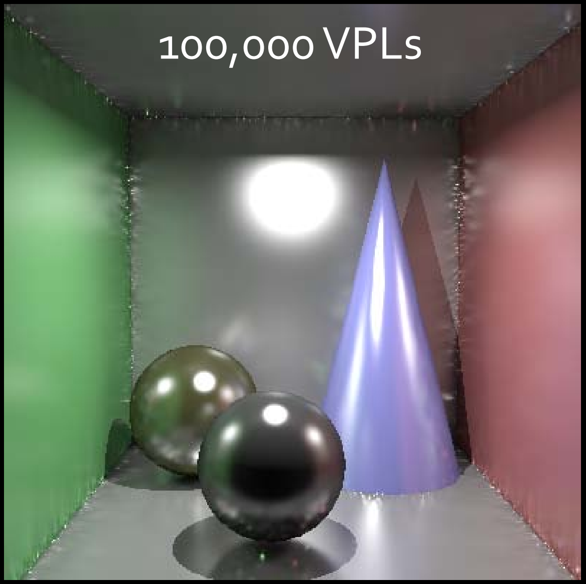
Biased result with clamping

Unbiased result with compensation

Dealing with Glossy Transport

Instant radiosity with glossy surfaces





Effect of clamping

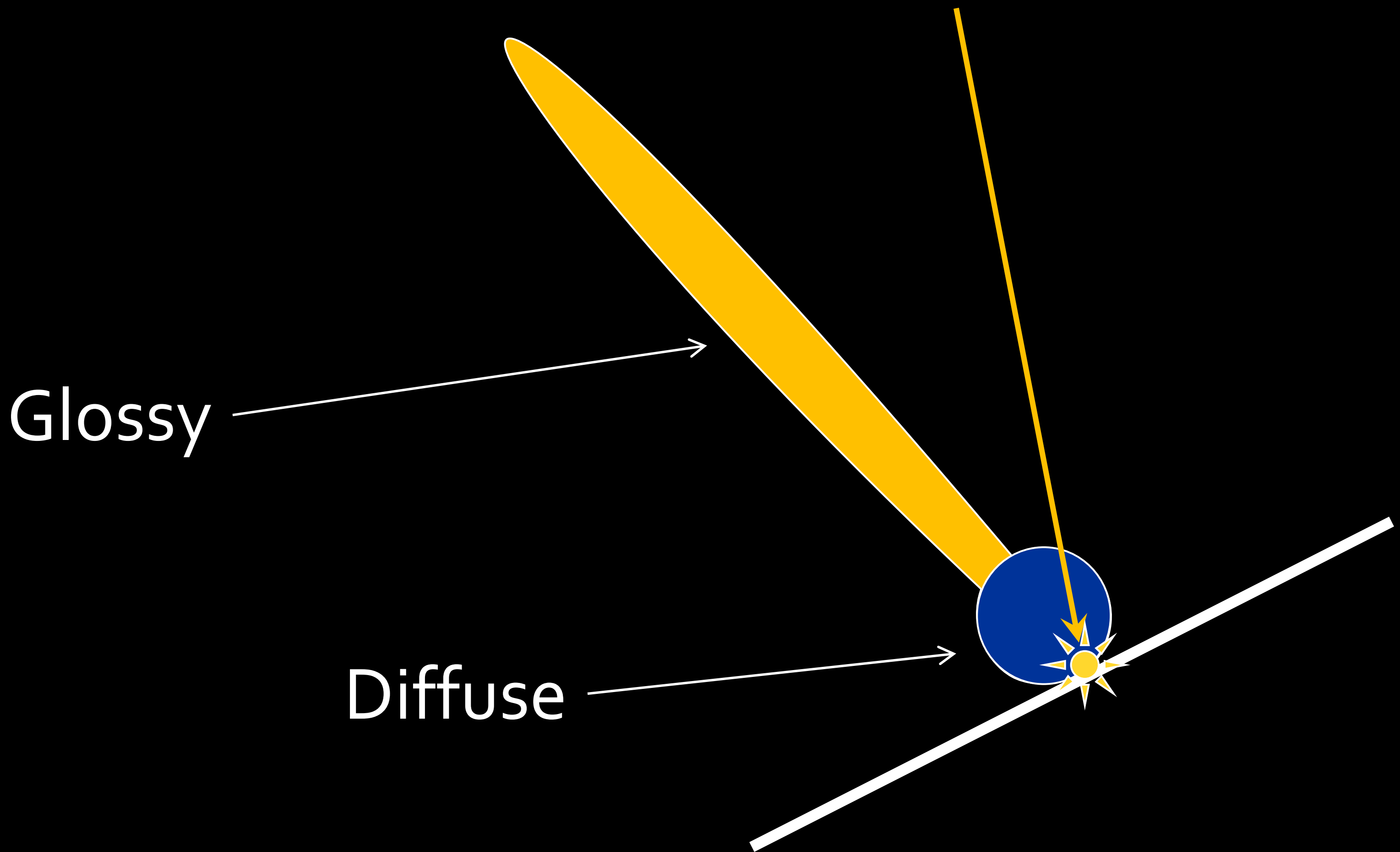
artifacts **material change**

Virtual Spherical Lights

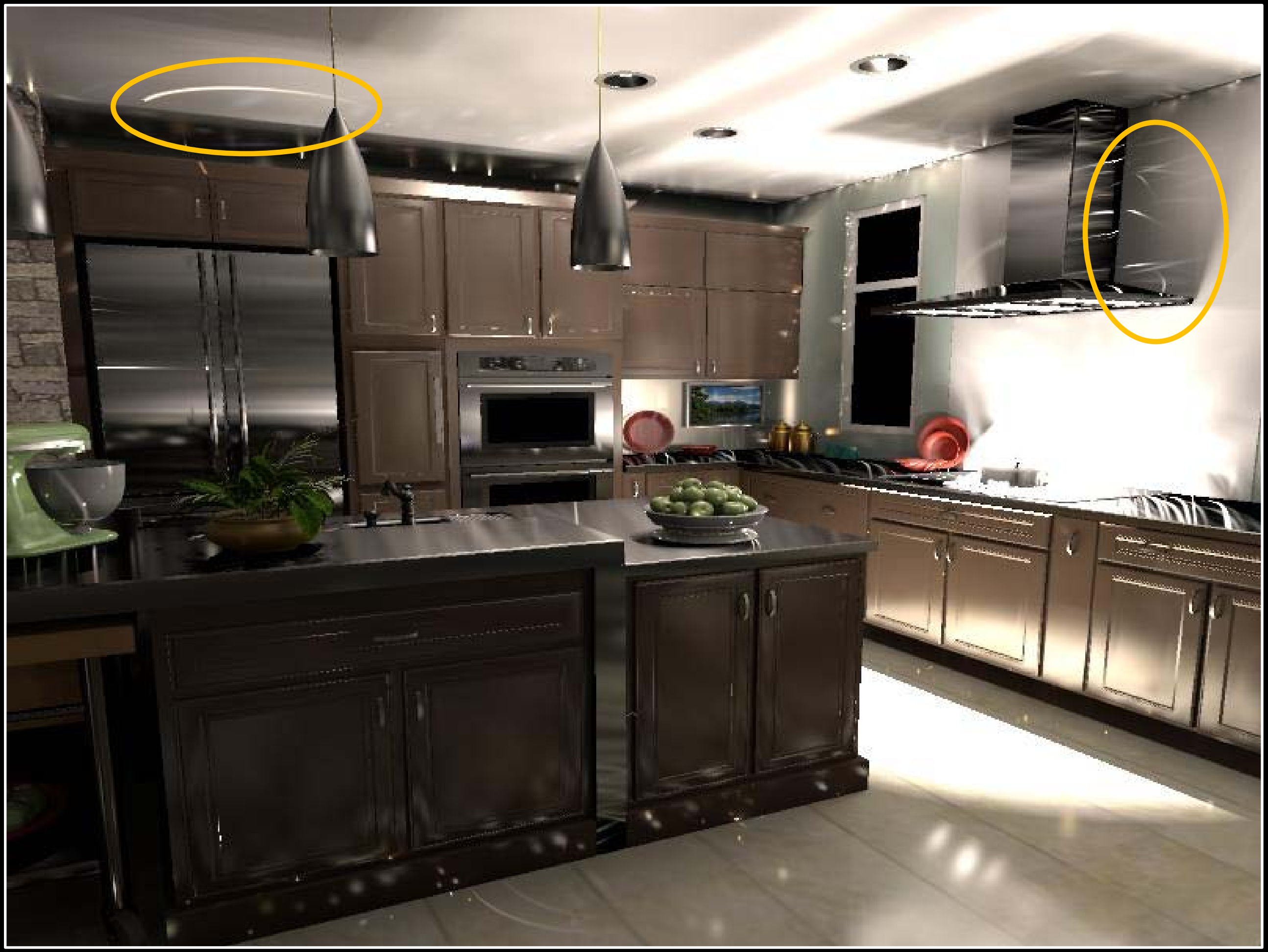
Hašan, Křivánek & Bala, SIGGRAPH Asia 2009

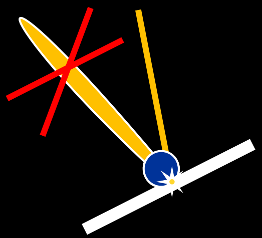
Emission distribution of a VPL

- Cosine-weighted BRDF lobe at the VPL location



Glossy VPL emission: illumination spikes

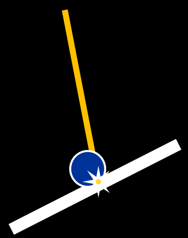


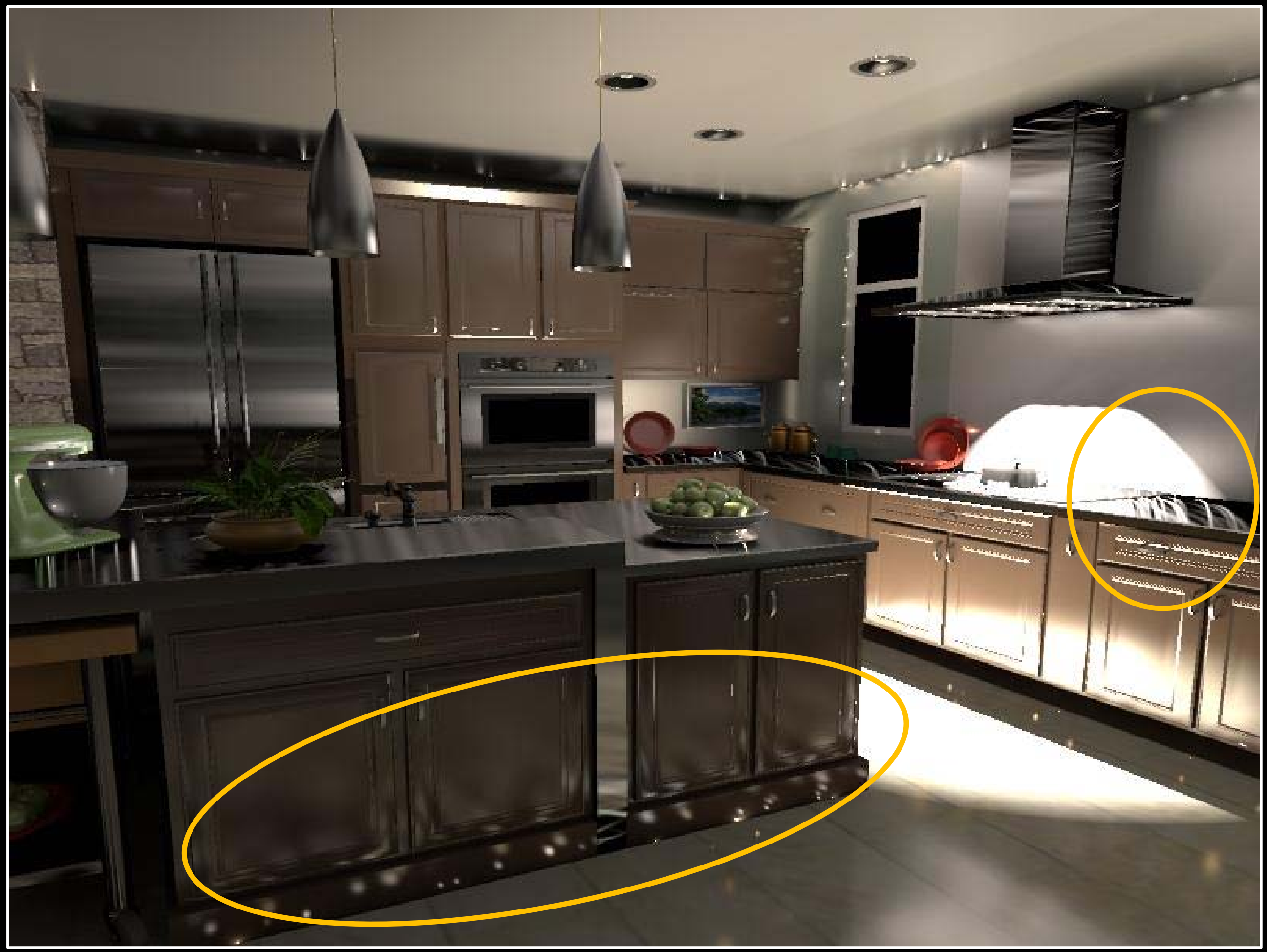


Common solution:

Only **diffuse** BRDF at light location

Remaining spikes

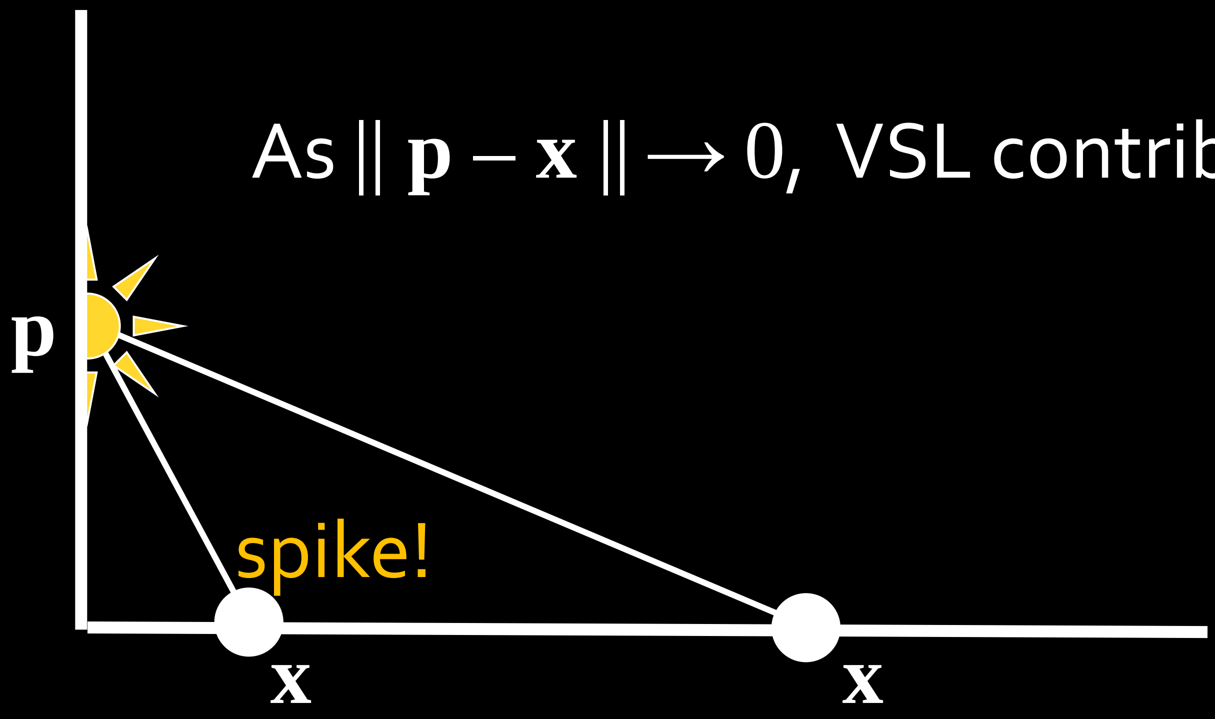




Remaining spikes

- VPL contribution =

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



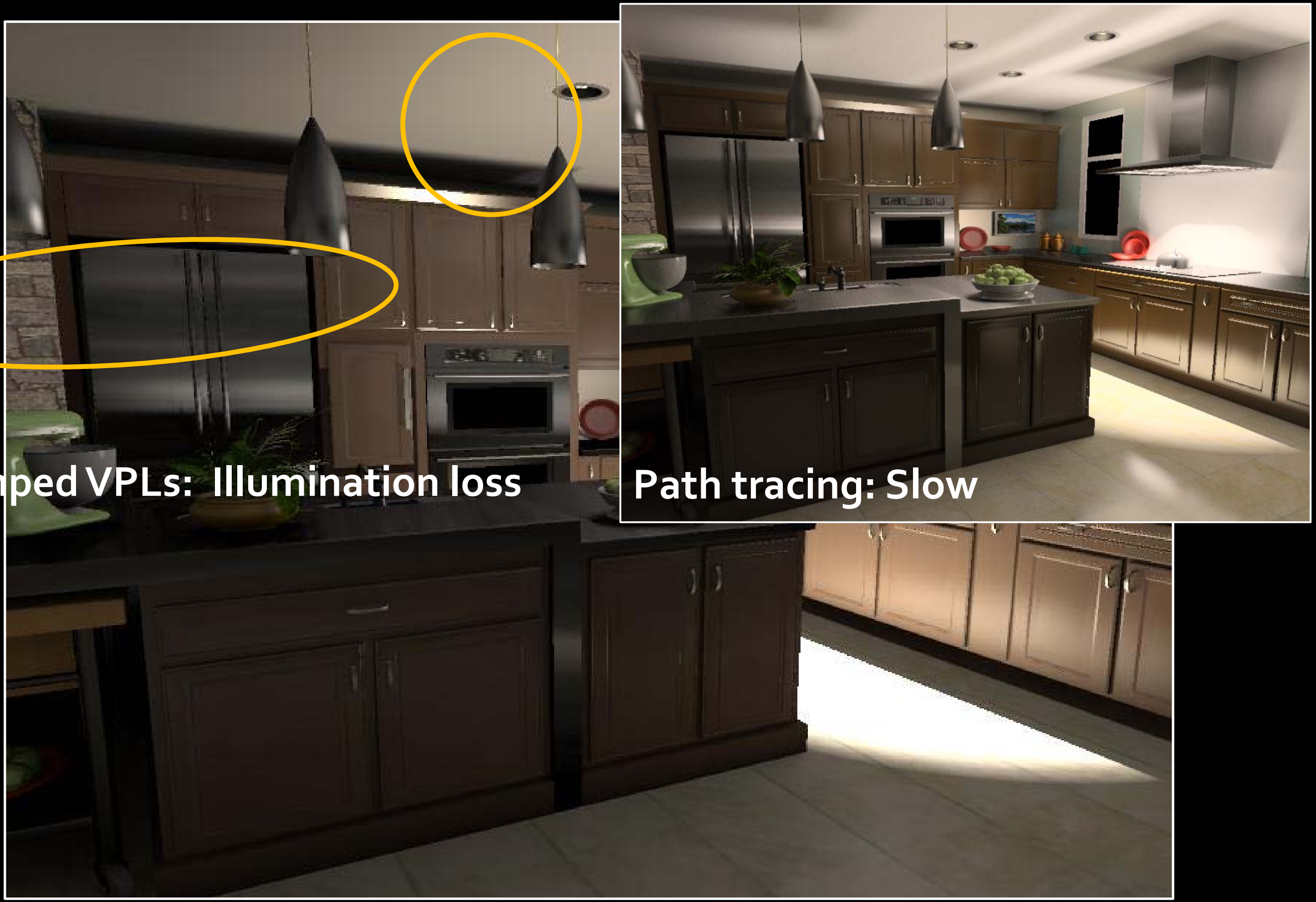
- Common solution: **Clamp** VPL contributions

Instant radiosity: The practical version

Clamping and diffuse-only VPLs:

Illumination is lost!

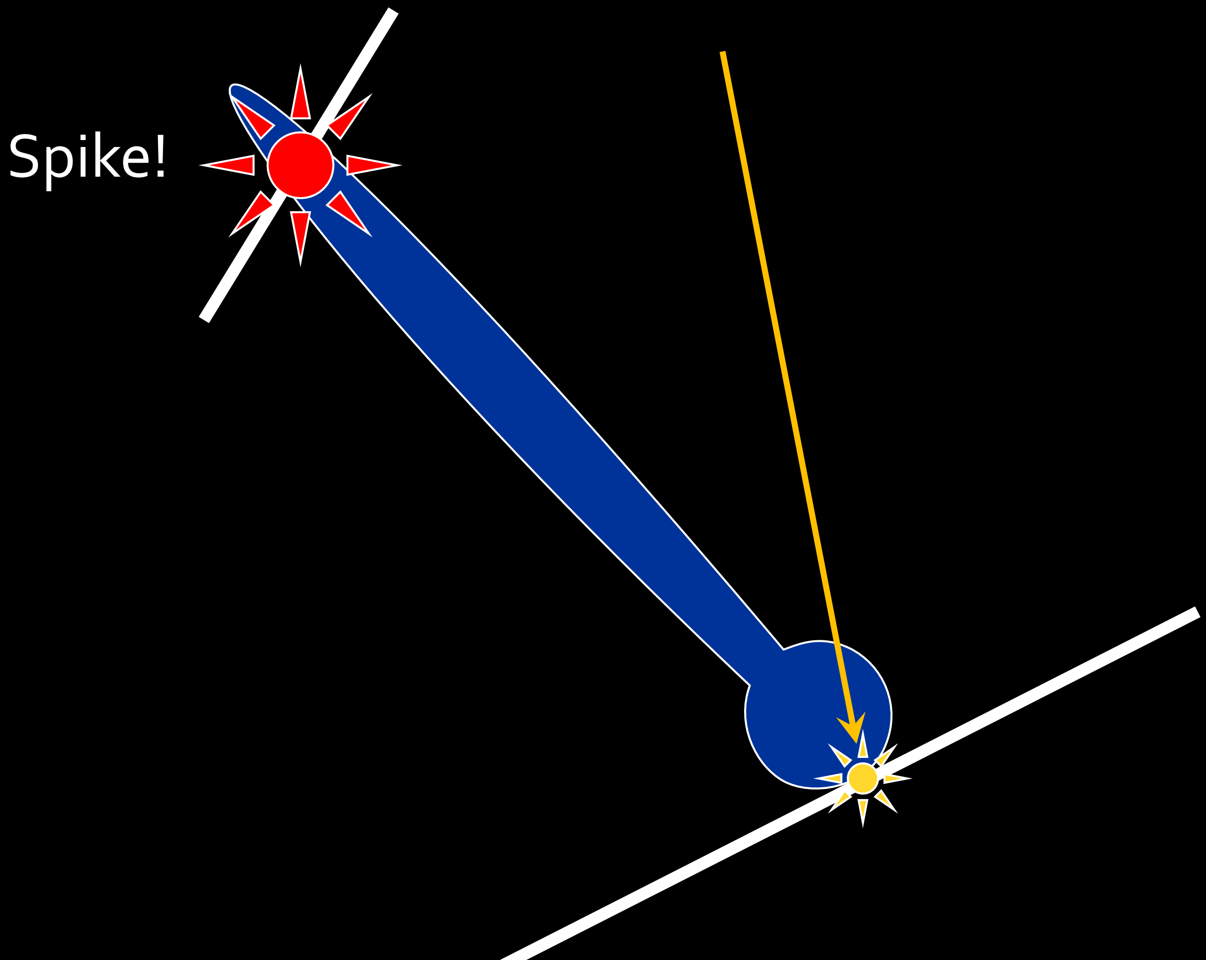
Comparison



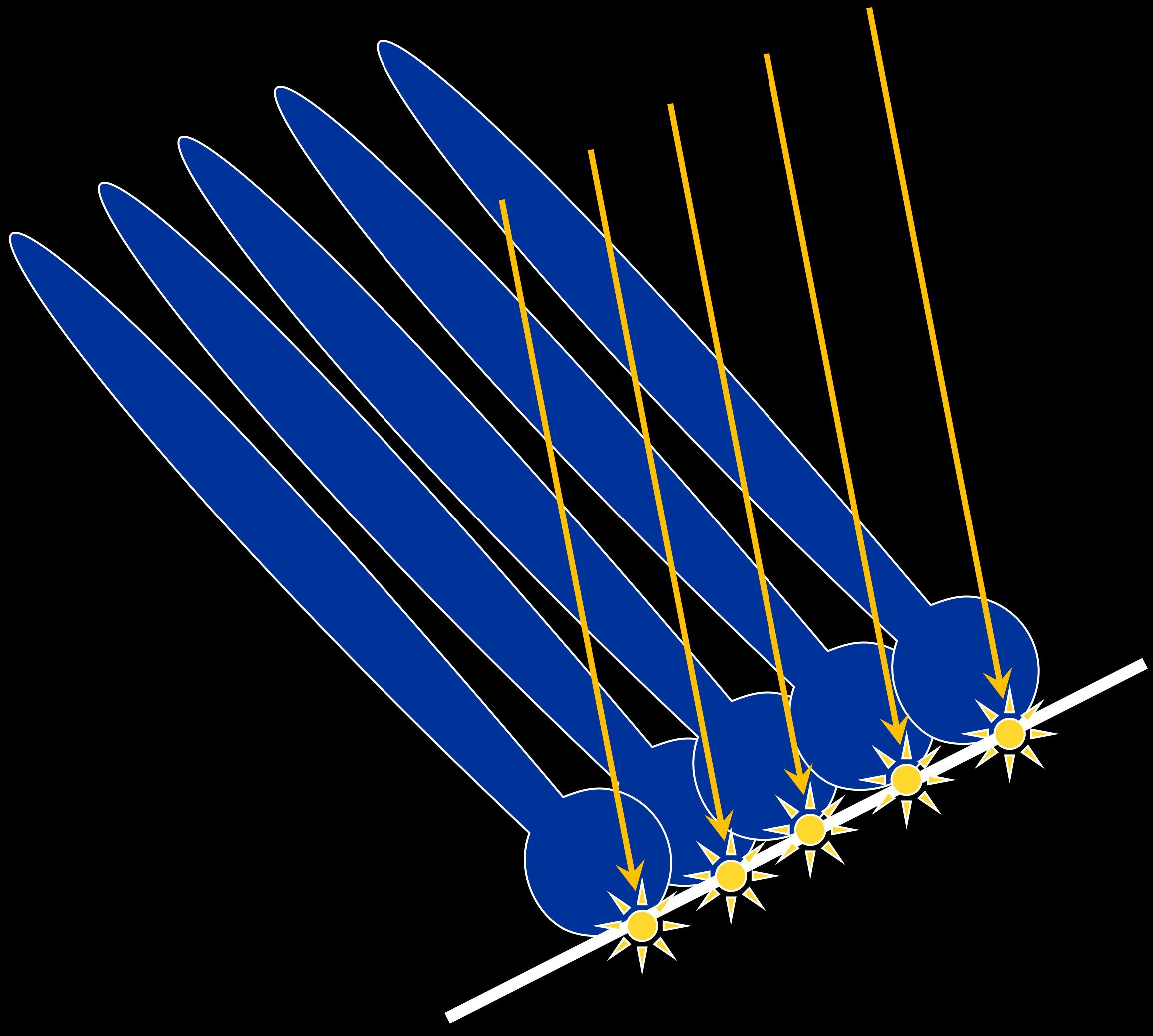
Clamped VPLs: Illumination loss

Path tracing: Slow

Recall: Emission Distribution of a VPL

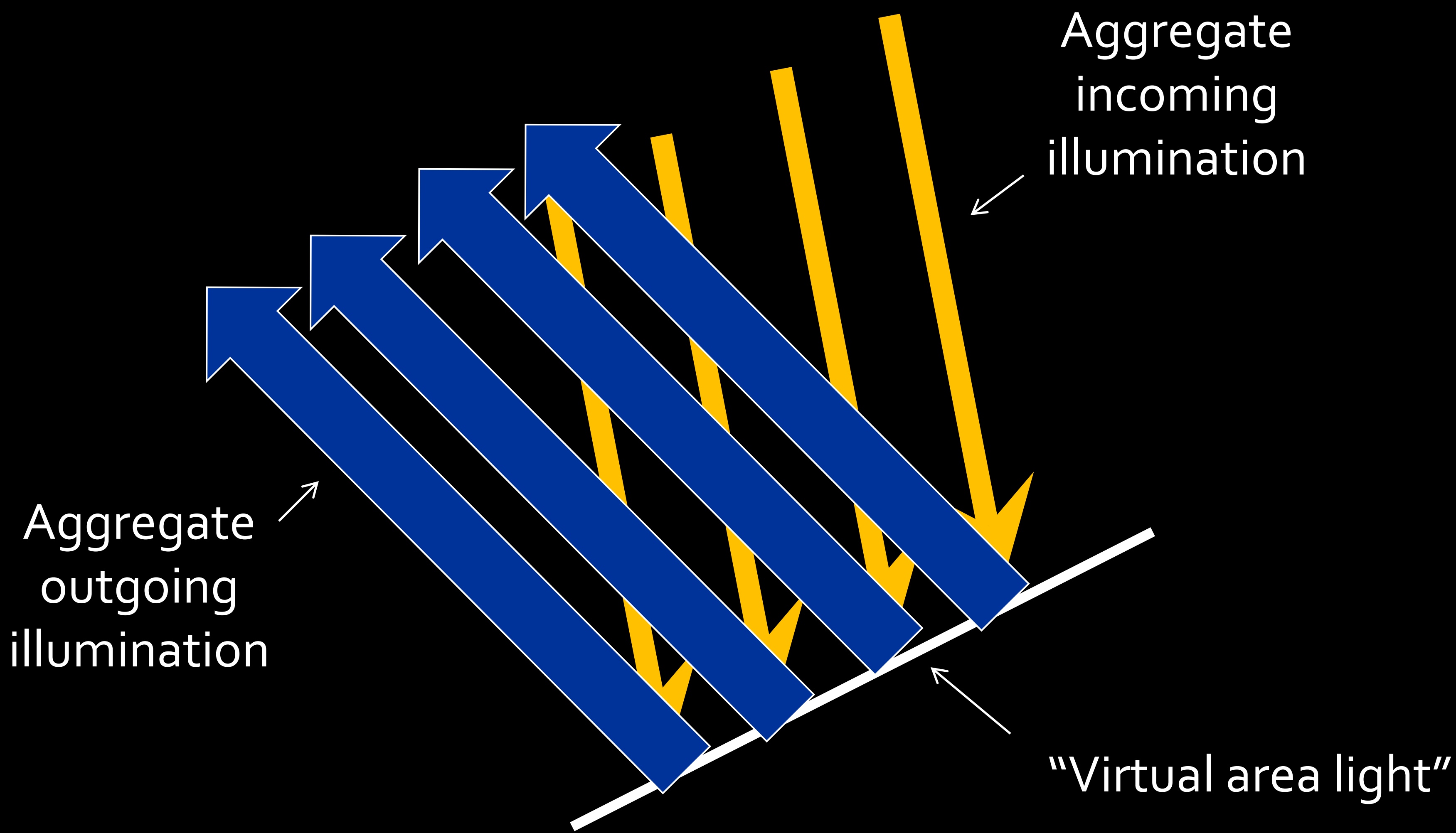


What happens as #lights $\to \infty$?



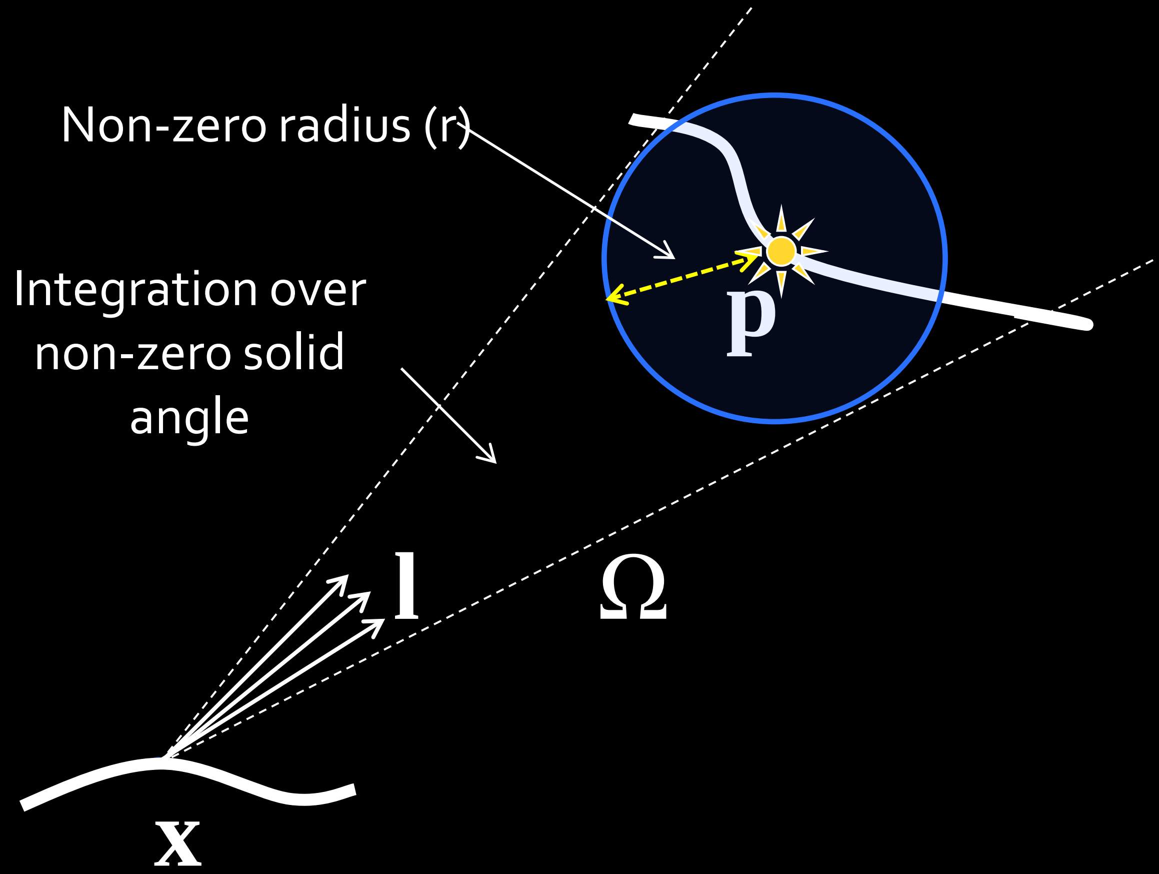
Spiky lights converge to a continuous function!

Idea: We want a “virtual area light”



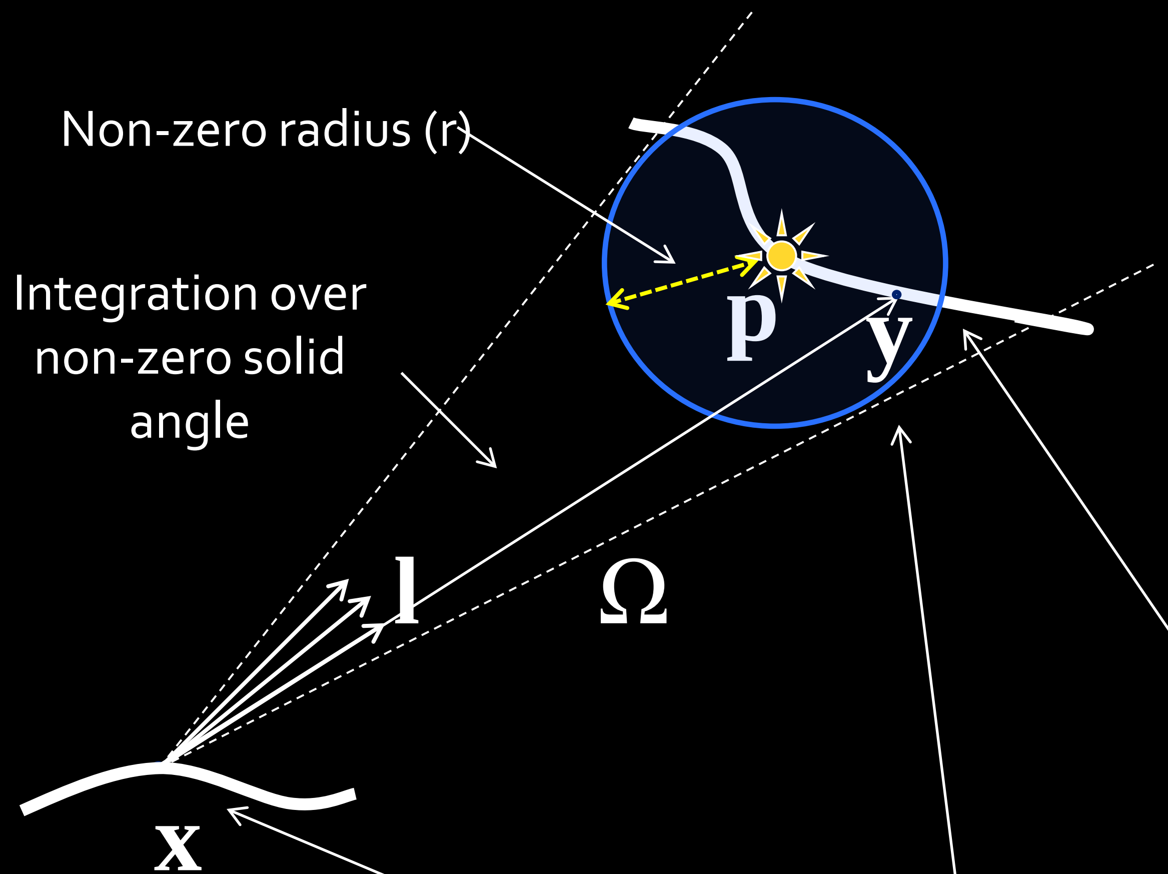
Problem: What if surface is not flat?

VPL to VSL



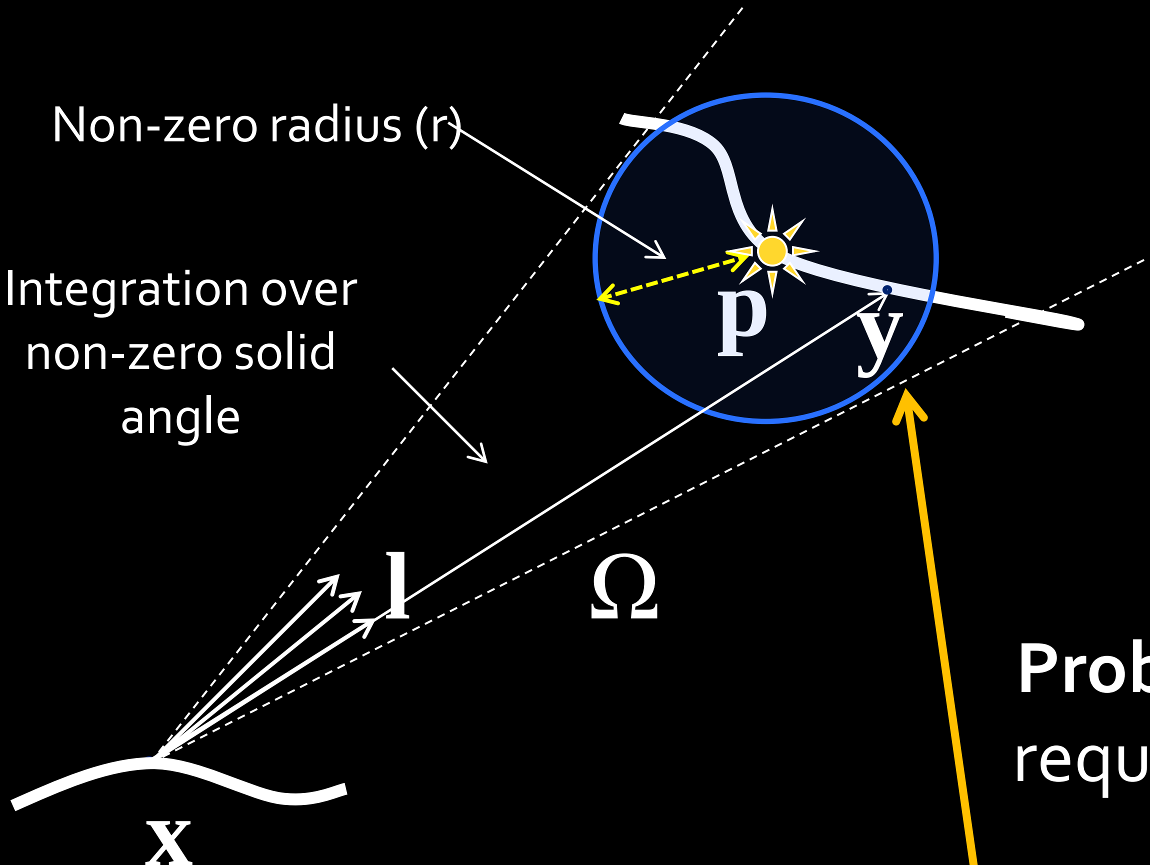
Non-zero radius (r)
Integration over non-zero solid angle
p
l
Ω
x

Light Contribution



$$\int_\Omega \boxed{}\ \boxed{}\ \boxed{}\ d\mathbf{l}$$

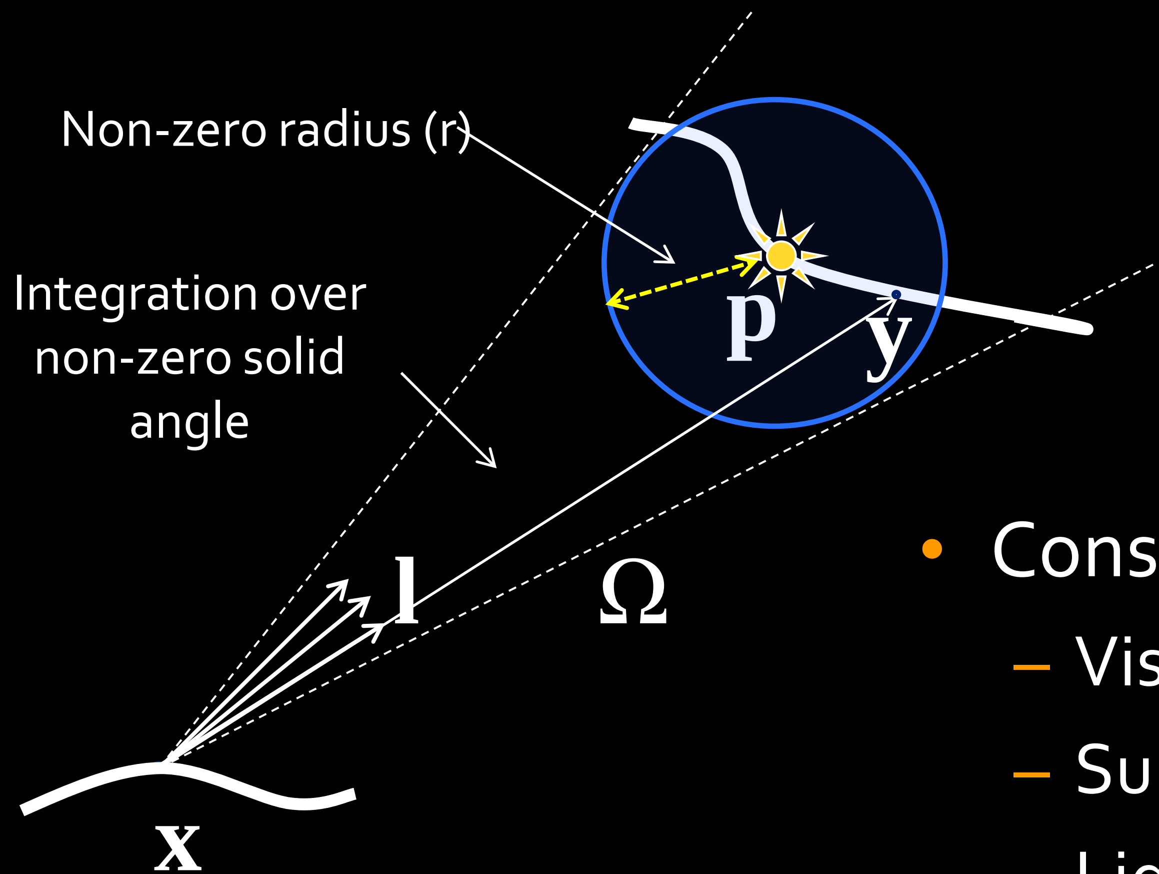
Light Contribution



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

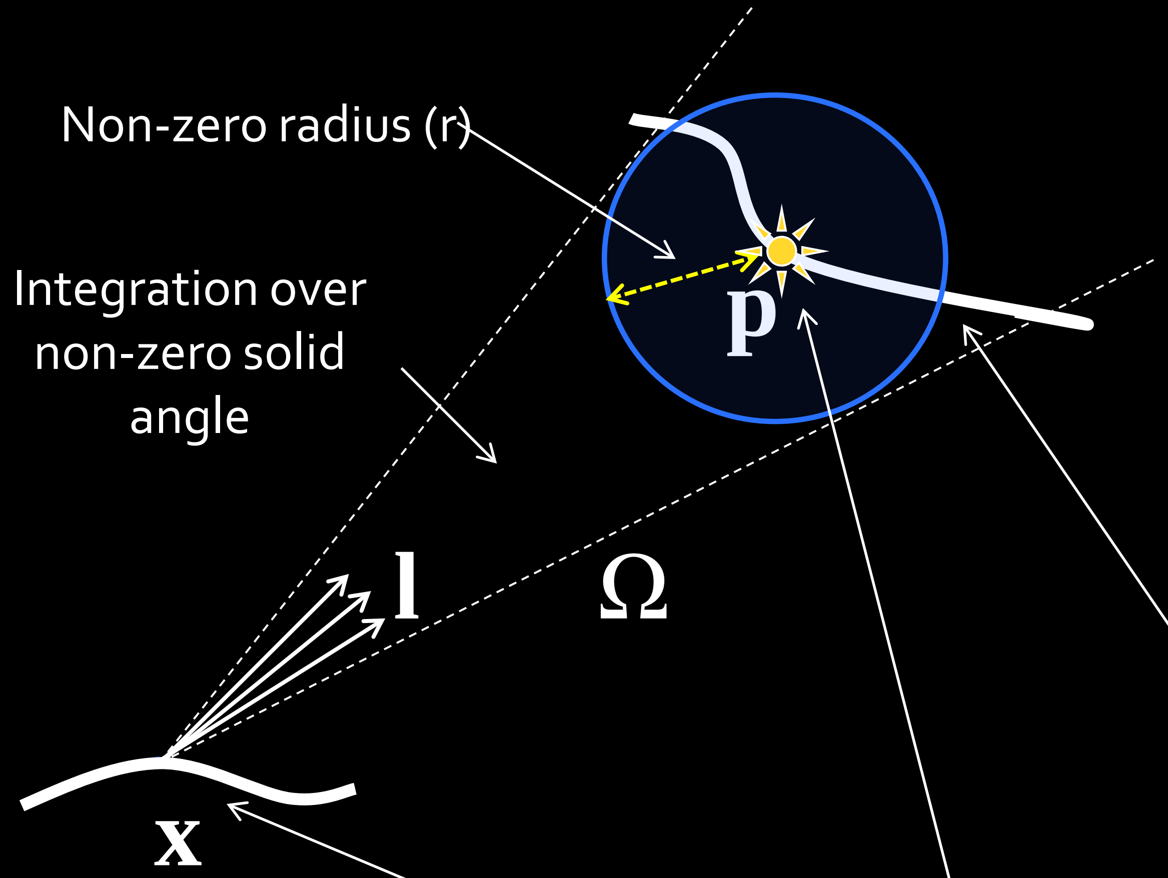
$$\frac{\Phi}{\pi r^2} \int_{\Omega} f_r(\mathbf{x}) \cos\theta_{\mathbf{x}} \; f_r(\boxed{\mathbf{y}}) \; (\|\mathbf{p} - \boxed{\mathbf{y}}\| < r) \mathrm{d}\mathbf{l}$$

Simplifying Assumptions



- Constant in Ω:
  - Visibility
  - Surface normal
  - Light BRDF
- Taken from $\mathbf{p}$, the light location

Light Contribution Updated



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

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

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

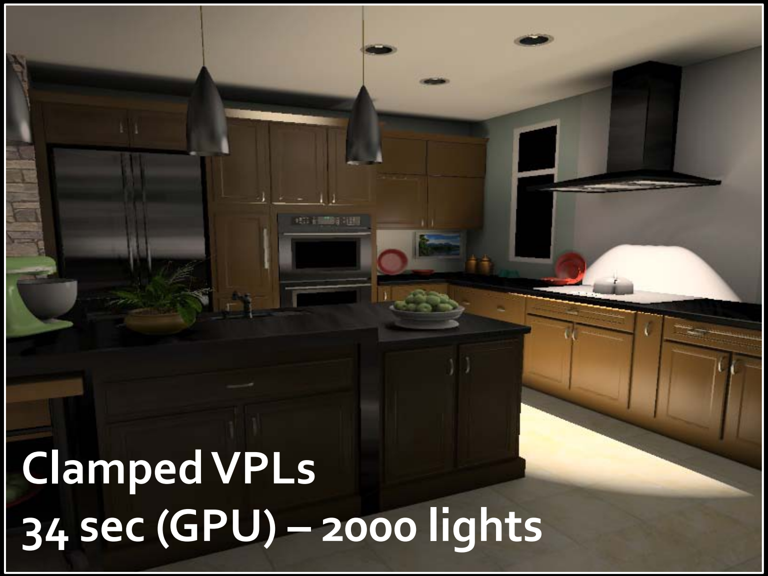
Implementation

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

- Need more lights than in diffuse scenes

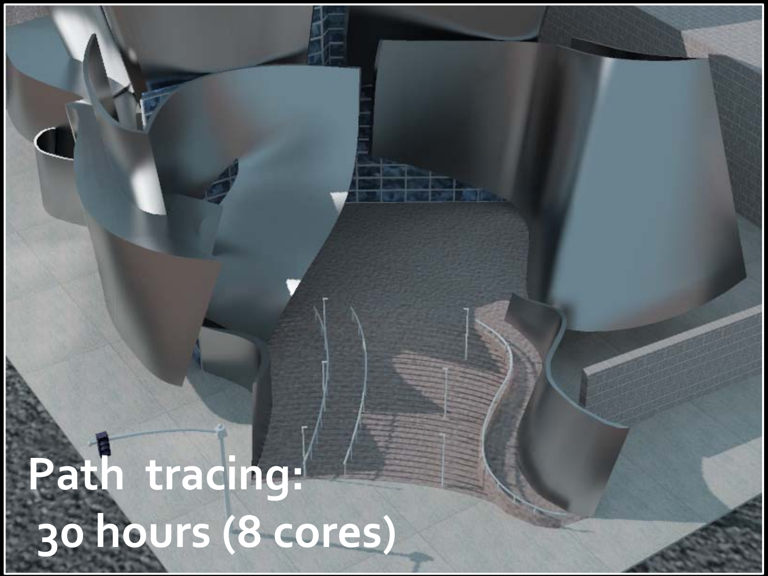
Results: Kitchen

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

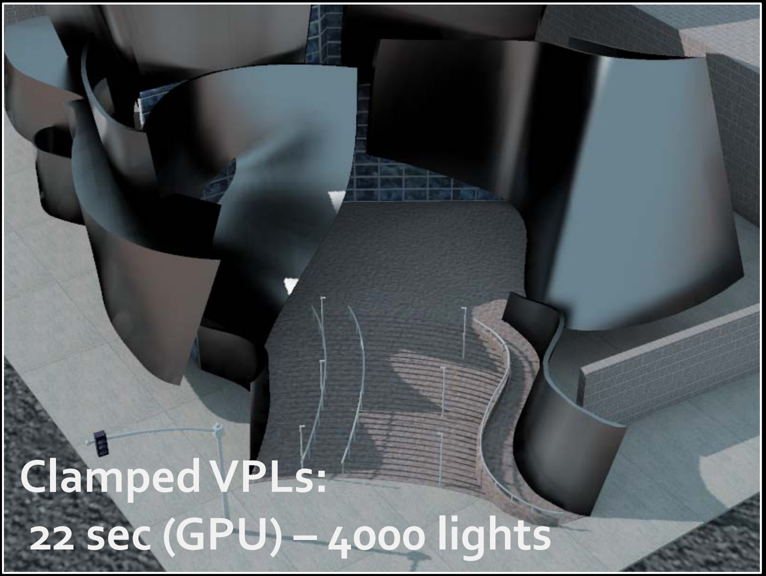


Results: Disney concert hall

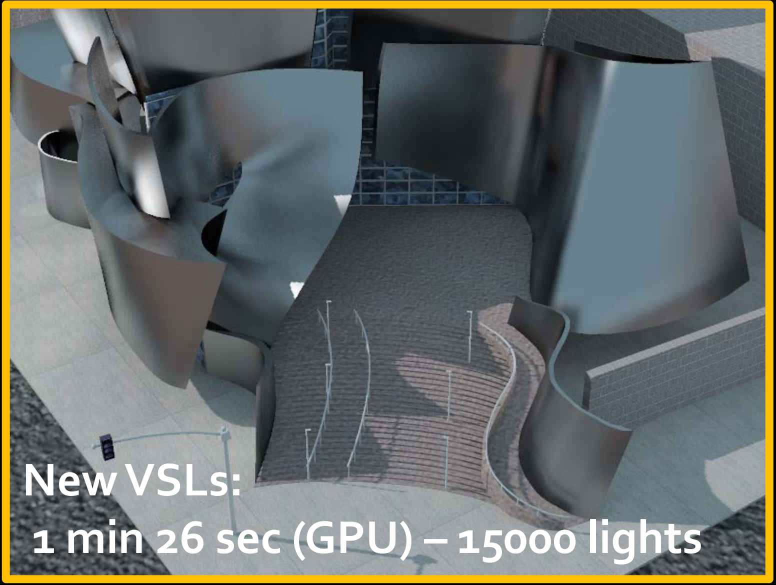
- Curved walls with no diffuse component
- Standard VPLs cannot capture any reflection from walls



Path tracing:
30 hours (8 cores)



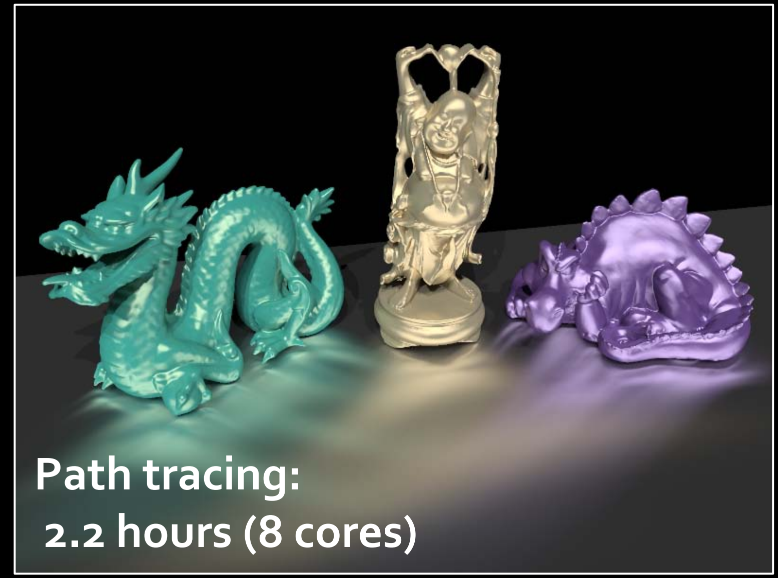
Clamped VPLs:
22 sec (GPU) – 4000 lights



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

Results: Anisotropic tableau

- Difficult case
- Standard VPLs capture almost no indirect illumination



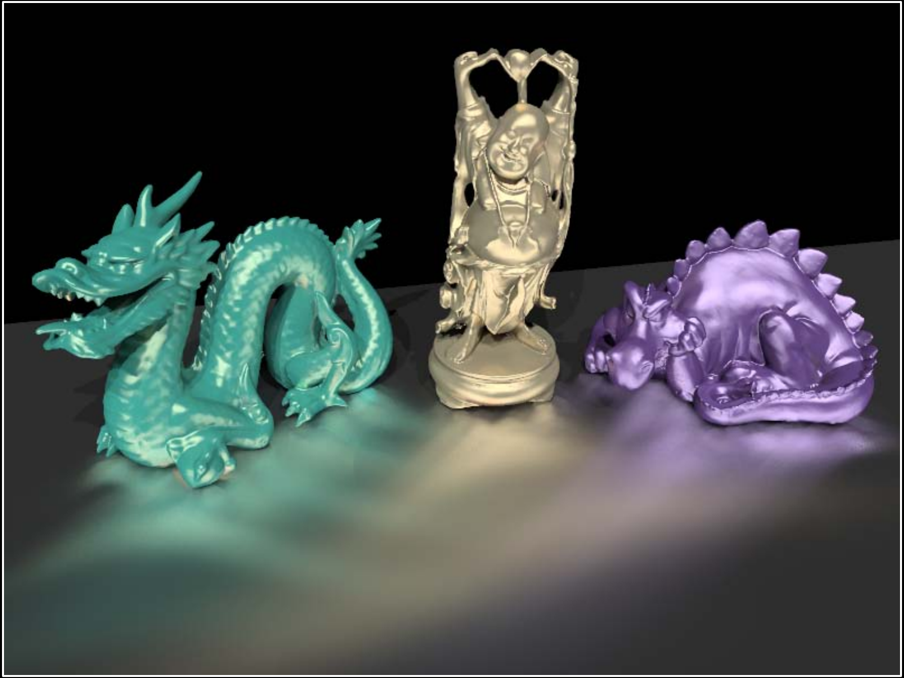
Path tracing:
2.2 hours (8 cores)

Clamped VPLs:
32 sec (GPU) – 1000 lights

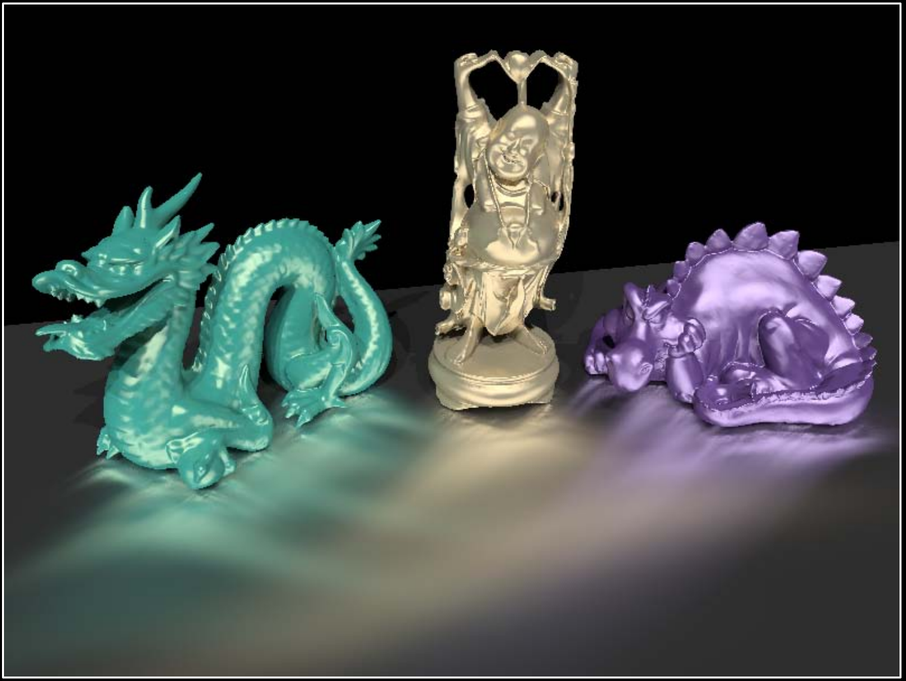
New VSLs:
1 min 44 sec (GPU) – 5000 lights

Limitations: Blurring

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



5,000 lights - blurred



1,000,000 lights - converged

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
  - [Davidovič et al. 2010]