Radiance Caching

Jaroslav Křivánek

Charles University, Prague

Jaroslav.Krivanek@mff.cuni.cz

SEVENTH FRAMEWORK PROGRAMME

MARIE CURIE ACTIONS

Glossy Surfaces

Frank Gehry: Walt Disney Concert Hall, Los Angeles, CA

What Does ‘Glossy’ Mean Here?

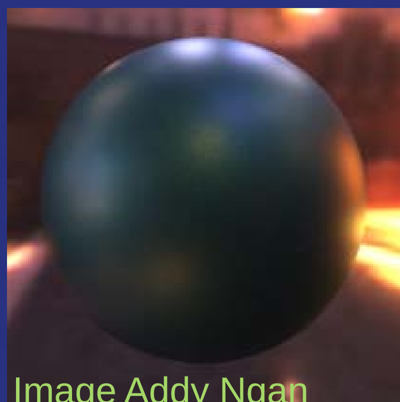


Image Addy Ngan

Low-frequency BRDF

High-frequency BRDF

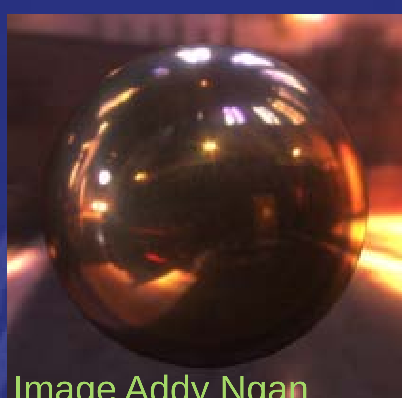


Image Addy Ngan

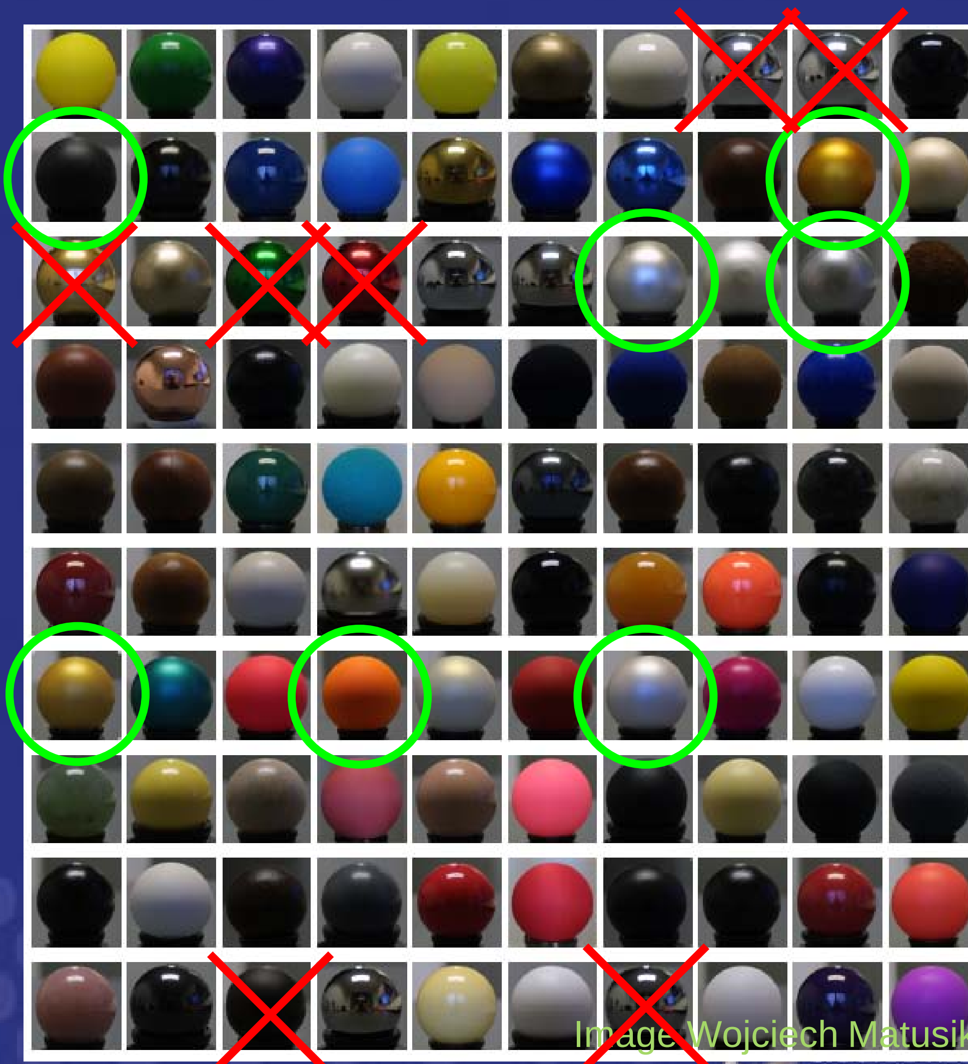


Image Wojciech Matusik

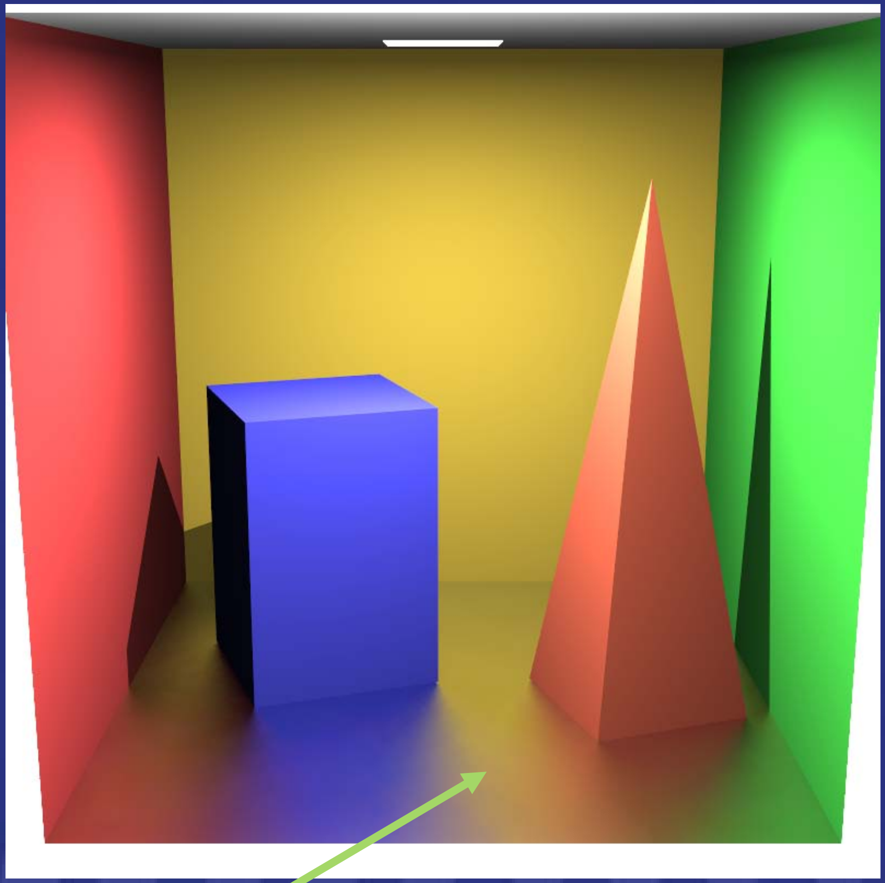
Do We Need Indirect Illumination on Glossy Surfaces?

Yes!

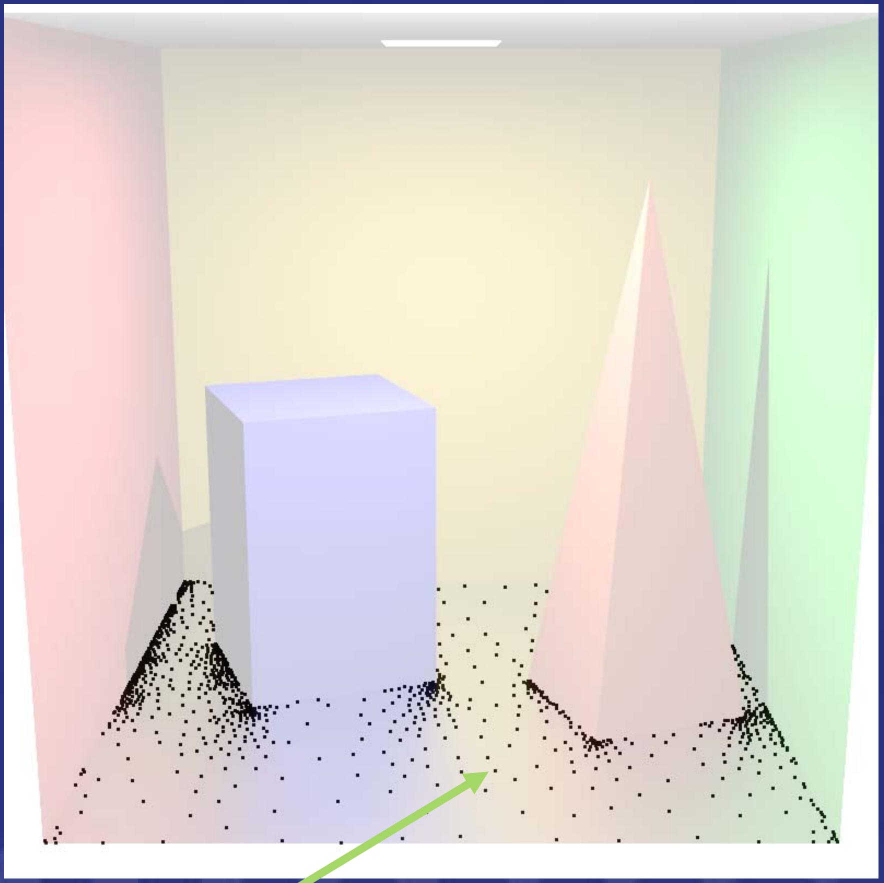
With indirect

Without indirect

Caching Works on Glossy Surfaces

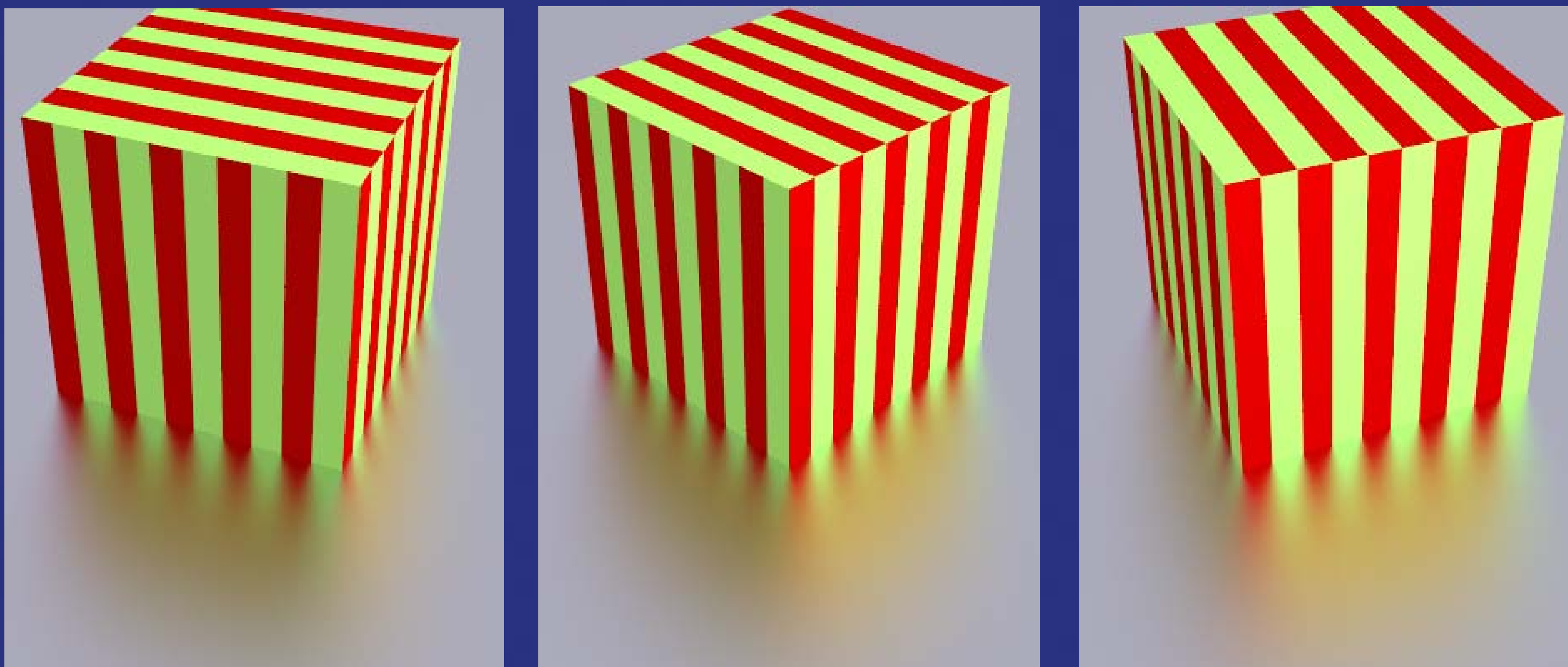


Smooth indirect term



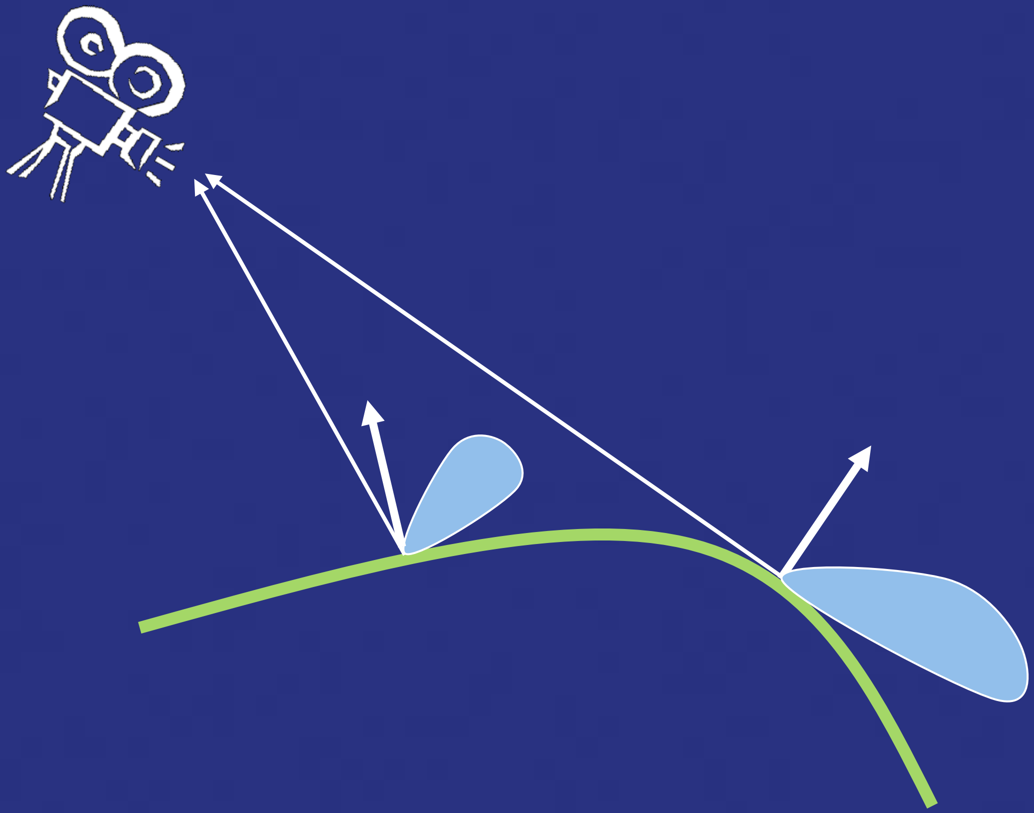
Sparse computation & interpolation

View-Dependence



- Appearance of glossy surfaces is view-dependent

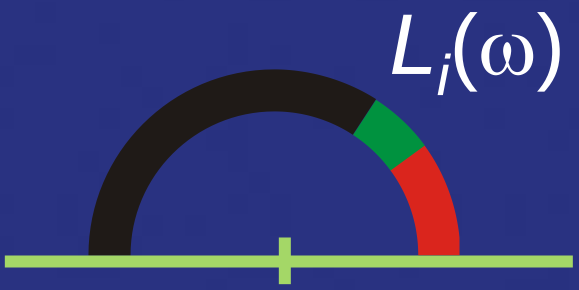
View-Dependence



- Different BRDF lobe for different viewing directions.
- Need to cache directional distribution of incident light.

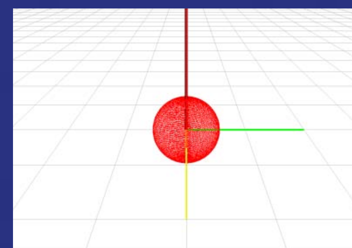
Incoming Radiance Representation

- Directional distribution of incoming radiance
  - Function on the hemisphere

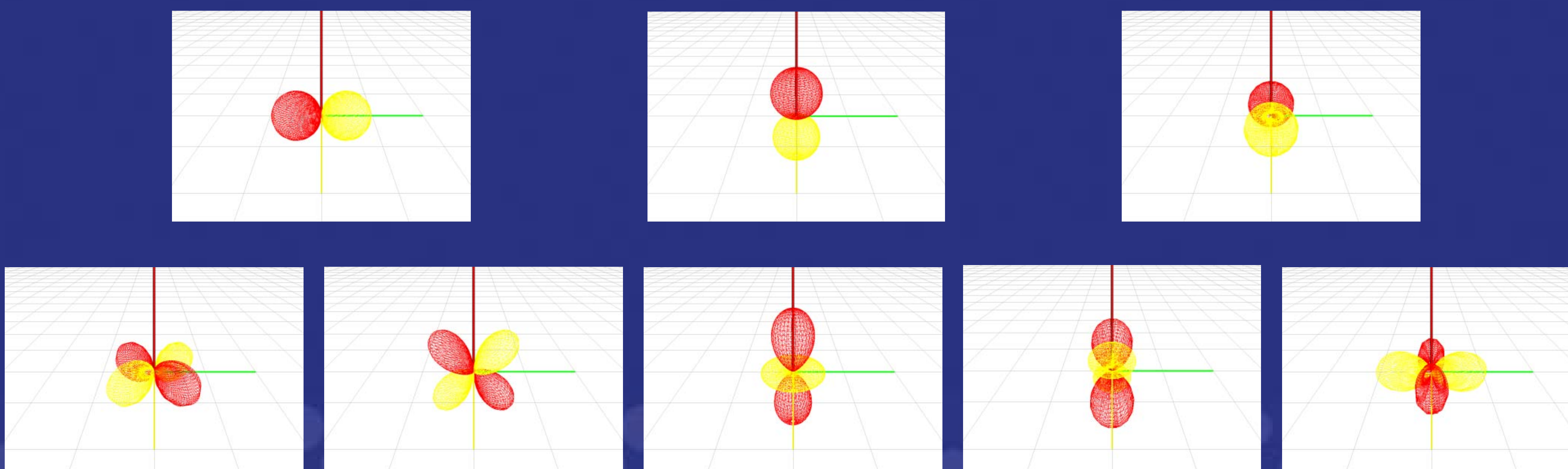


Incoming Radiance Representation

- Spherical harmonics



- Basis functions on the sphere

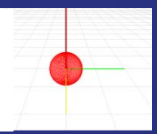


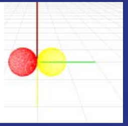
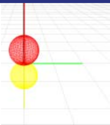
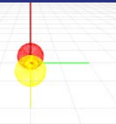
- Intro to Spherical harmonics [Green 2003]

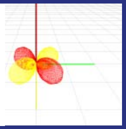
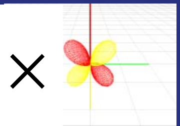
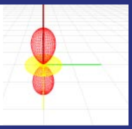
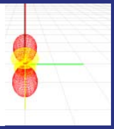
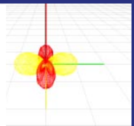
Incoming Radiance Representation

- Linear combination of basis functions

$$L_i(\omega) =$$

$\lambda_0^0 \times$  $+$

$\lambda_1^{-1} \times$  $+$ $\lambda_1^{0} \times$  $+$ $\lambda_1^{1} \times$  $+$

$\lambda_2^{-2} \times$  $+$ $\lambda_2^{-1} \times$  $+$ $\lambda_2^{0} \times$  $+$ $\lambda_2^{1} \times$  $+$ $\lambda_2^{2} \times$ 

$$L_i(\omega) = \sum_{l=0}^{n-1} \sum_{m=-l}^{m=l} \lambda_l^m Y_l^m(\omega)$$

Incoming Radiance Computation

- How to find the coefficients for $L_i(\omega)$?

- Project $L_i(\omega)$ onto the basis

$$\lambda_l^m = \int_{\Omega} L_i(\omega) Y_l^m(\omega) \mathrm{d}\omega$$

Incoming Radiance Computation

`SampleHemisphere(p)`

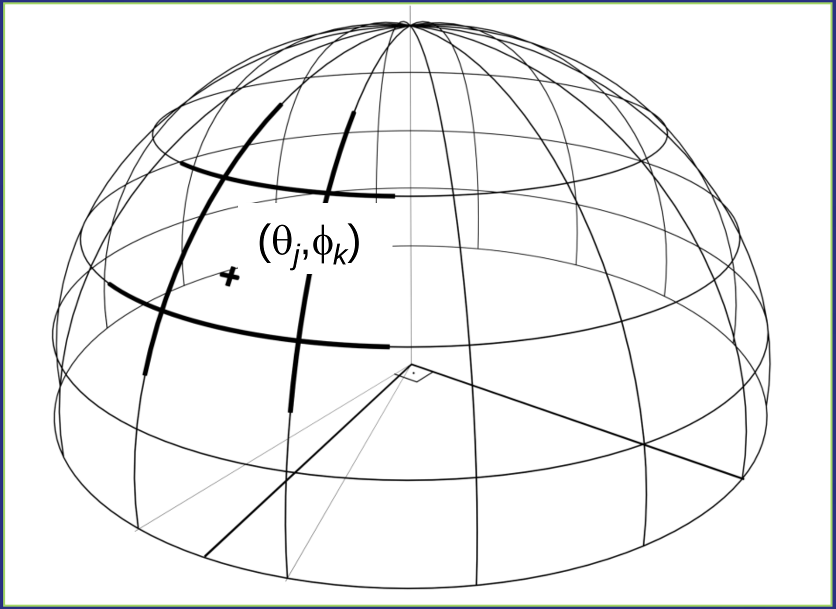
- Practice: Uniform hemisphere sampling

Sum over all cells

$$\lambda_l^m = \frac{2\pi}{NM} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} L_{j,k} Y_l^m(\theta_j, \phi_k)$$

Incoming radiance from the direction (θ_j,ϕ_k)

Multiplied by the basis function

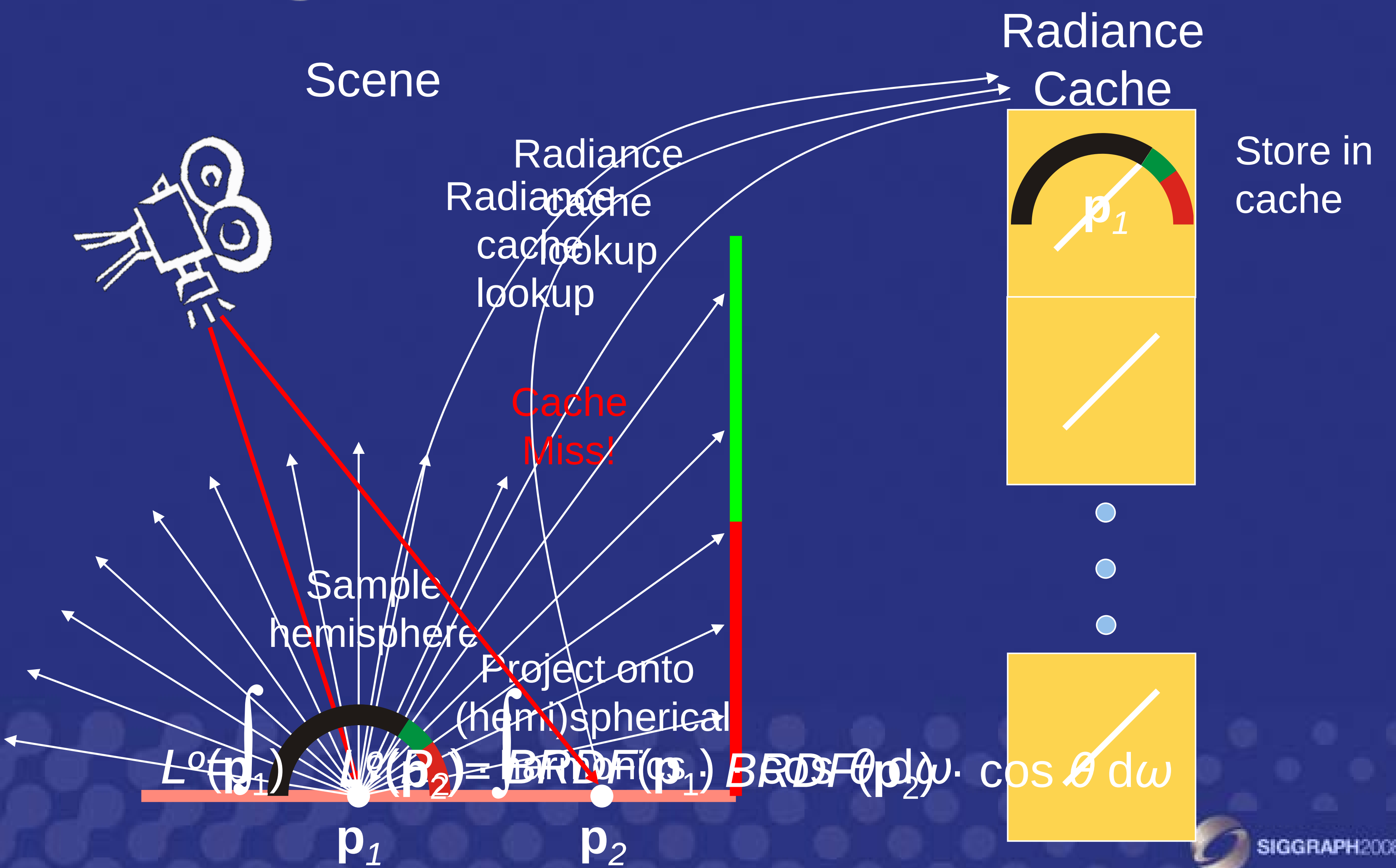


$$(\theta_j, \phi_k) = \left(\arcsin \frac{j+\zeta_{j,k}^1}{M}, 2\pi \frac{k+\zeta_{j,k}^2}{N} \right)$$

Spherical Harmonics

- Pros
  - Efficient rotation
  - Smooth – no aliasing
  - Little memory
  - Easy to use
- Cons
  - Only low-frequency BRDFs
  - Alternative – Wavelets

Caching Scheme



Caching Scheme

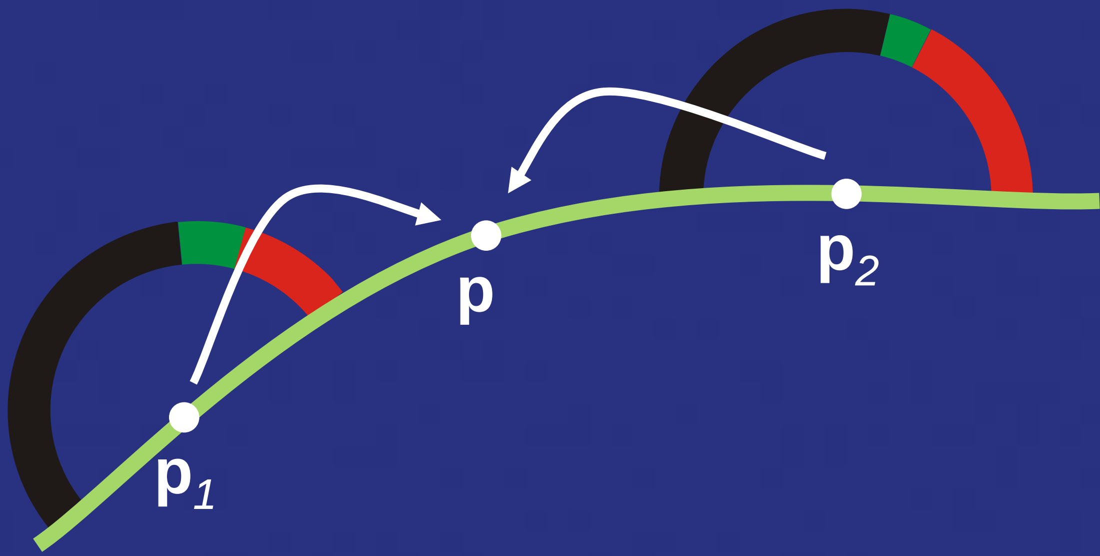
```
GetOutRadiance(p,wo):
    CoeffVector Λ;
    if( ! InterpolateFromCache(p, Λ) ) {
        Λ = SampleHemisphere(p);
        InsertIntoCache(Λ,p);
    }
    return ComputeOutRadiance(Λ, BRDF(p,wo));
```

Radiance Interpolation

`InterpolateFromCache(p, Λ)`

- Weighted average of coefficient vectors (borrowed from irradiance caching)

$$\Lambda_{\text{intp}}(\mathbf{p}) = \frac{\sum_{i \in S} \Lambda_i w_i(\mathbf{p})}{\sum_{i \in S} w_i(\mathbf{p})}$$



- Same weight as in IC

Radiance Interpolation

```
InterpolateFromCache(p, Λ)
```

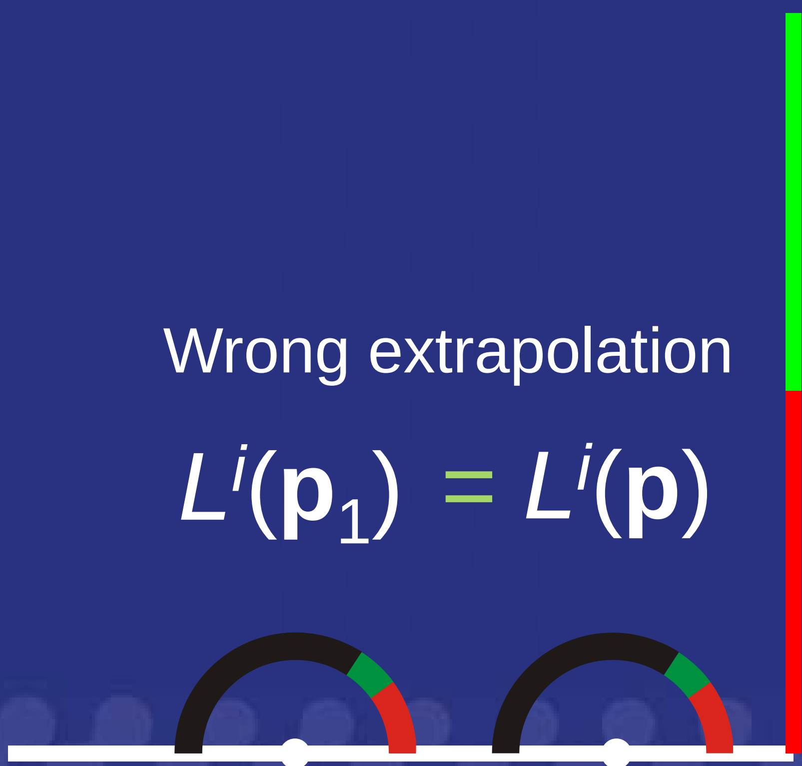
- For each nearby record
  - Adjust by gradient
  - Rotate
  - Update the weighted average

Translational Gradients

With radiance caching

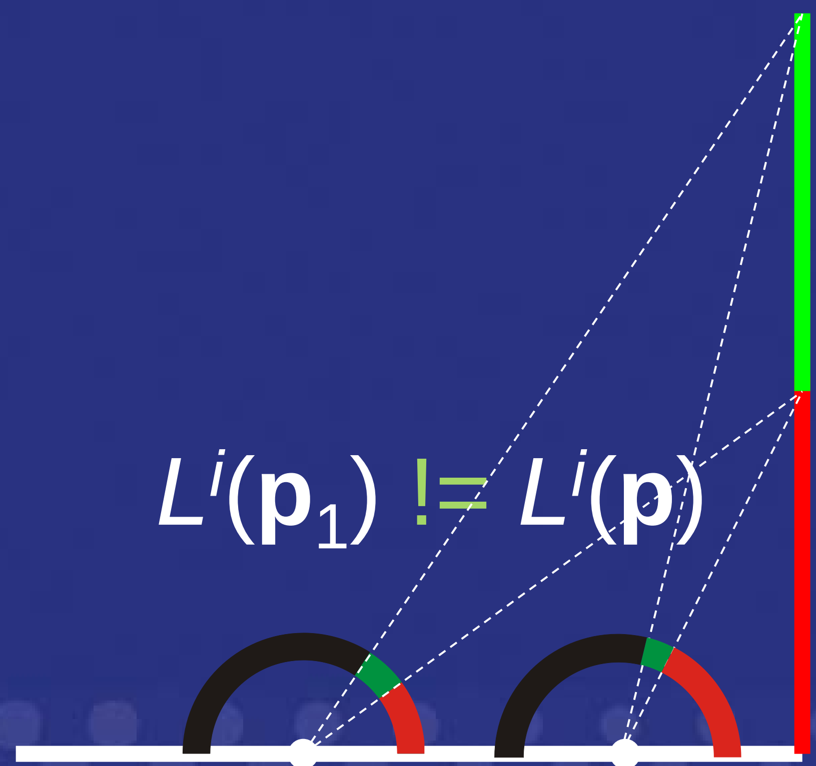
Wrong extrapolation

$$L^i(\mathbf{p}_1) = L^i(\mathbf{p})$$



Reality

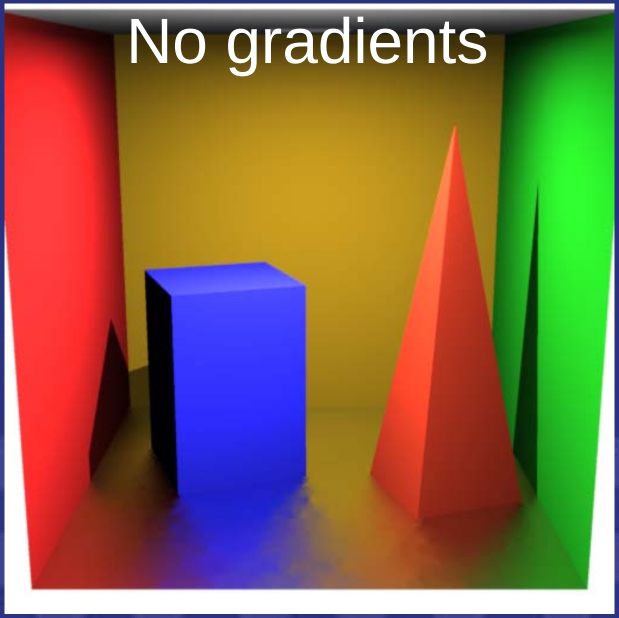
$$L^i(\mathbf{p}_1) \mathrel{!=} L^i(\mathbf{p})$$

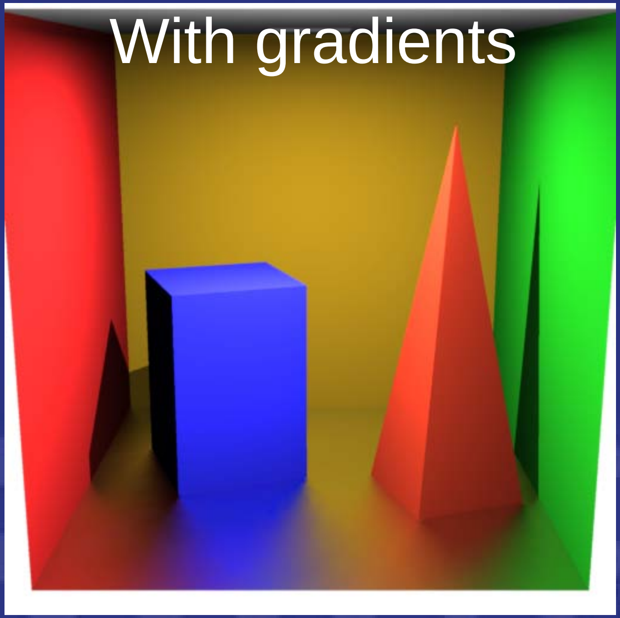


Translational Gradients

- How does $L_i(\mathbf{p})$ change with $\mathbf{p}$?
- First order approximation:

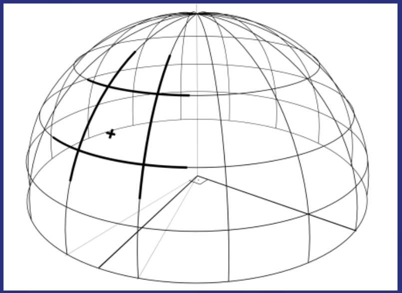
Translational radiance gradient





Gradient Computation

- For free – in hemisphere sampling
- Gradient for each coefficient



$$\vec{\nabla}\lambda_l^m = \sum_{k=0}^{N-1}\left[\hat{u}_k \frac{2\pi}{N}\sum_{j=1}^{M-1}\frac{\cos\theta_{j\_}\sin\theta_{j\_}}{\min\{r_{j,k},r_{j-1,k}\}}(L^i_{j,k}-L^i_{j-1,k})H_l^m(\theta_{j,k},\phi_{j,k}) + \hat{v}_{k\_}\frac{1}{M}\sum_{j=0}^{M-1}\frac{1}{\sin\theta_{j,k}\min\{r_{j,k},r_{j,k-1}\}}(L^i_{j,k}-L^i_{j,k-1})H_l^m(\theta_{j,k},\phi_{j,k})\right],$$

Cell area change

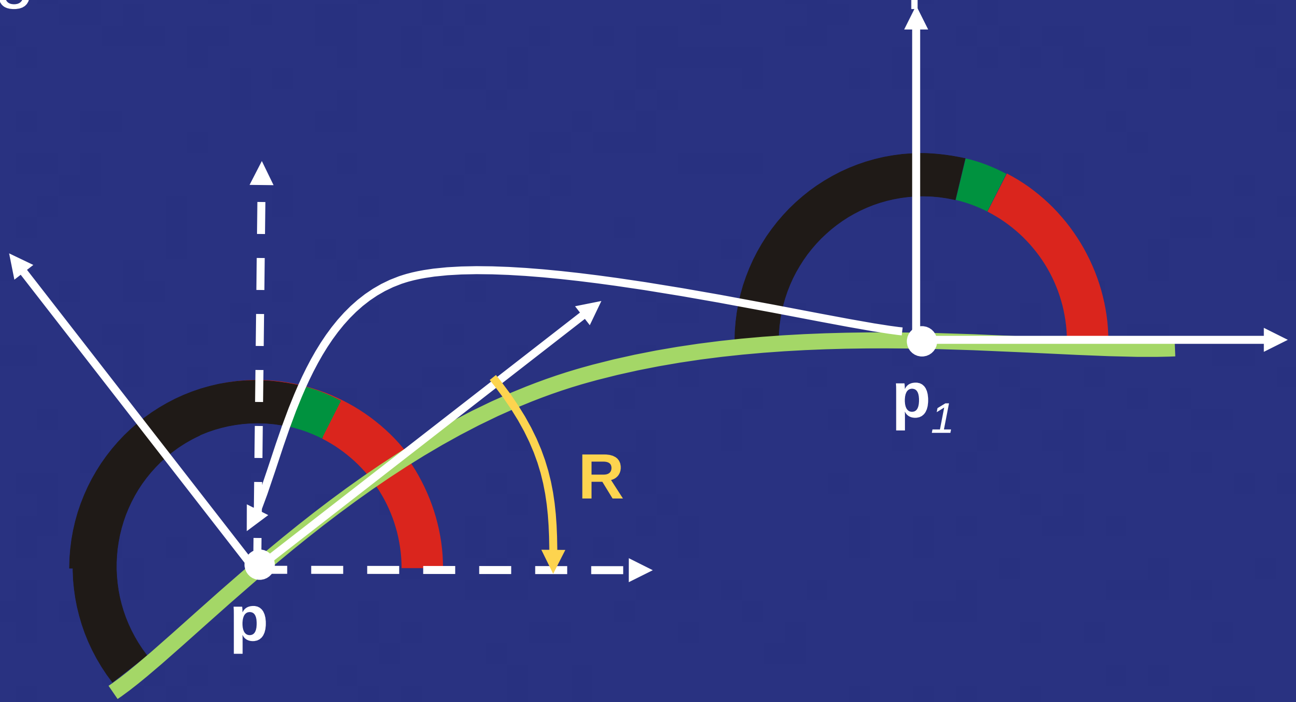
Sum together

Incoming radiance change

Weight by the basis function

Rotation

- Align coordinate frames in interpolation



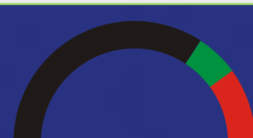
- Needs fast SH rotation (code on the web)
  - [Kautz et al. 2002, Křivánek et al. 2006]

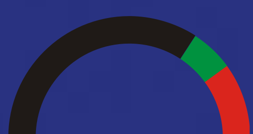
Outgoing Radiance Computation

```
ComputeOutRadiance(Λ, BRDF(p,wo))
```

- $L_o(\omega_o)$ is the final color
- Given by the Illumination Integral

$$L_o(\omega_o) = \int_{\Omega} L_i(\omega_i) \cdot BRDF(\omega_i, \omega_o) \cdot \cos\theta_i \cdot d\omega_i$$

  – *i.e.* Integrate  x BRDF

-  is interpolated
- BRDF is known

Outgoing Radiance Computation

```
ComputeOutRadiance(Λ, BRDF(p,wo))
```

- BRDF represented by spherical harmonics – orthonormal basis

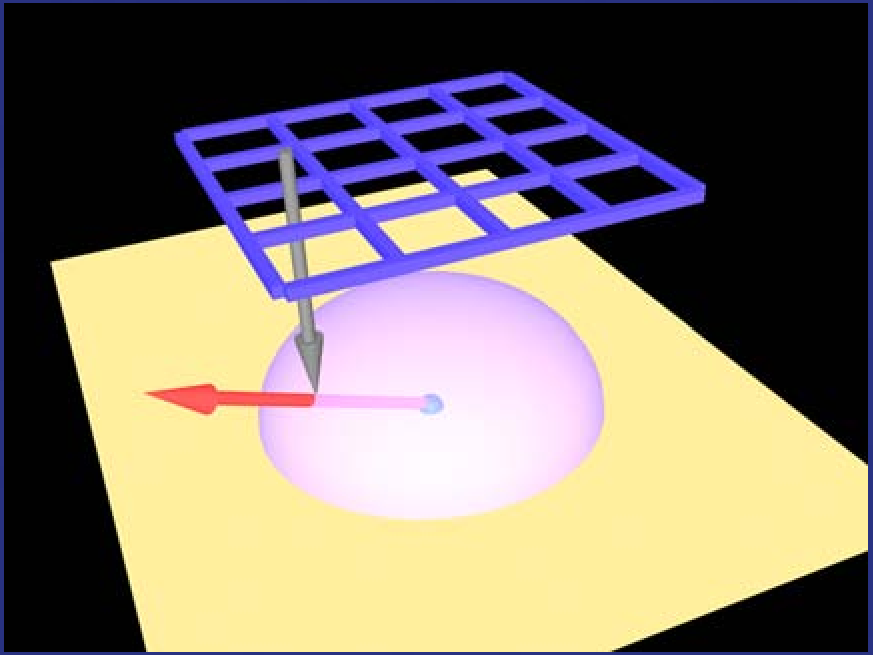
= coeff. dot product

$$L_o(\omega_o) = \Lambda_{\text{intp}}(\mathbf{p}) \bullet F(\mathbf{p}, \omega_o)$$

BRDF Representation

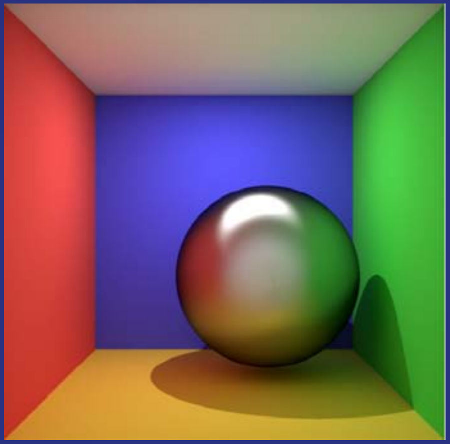
`BRDF(p,wo)`

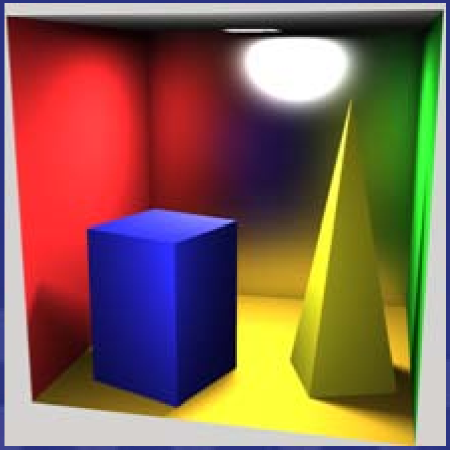
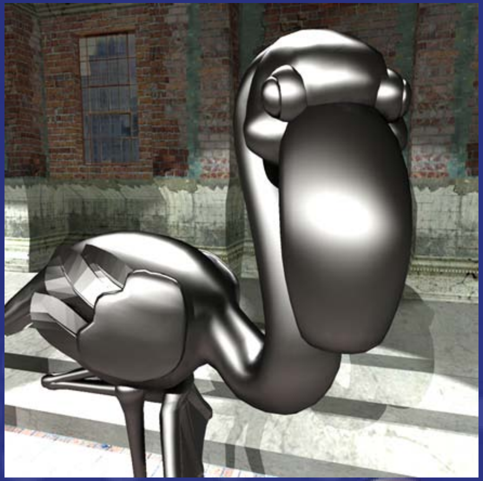
- Paraboloid mapping
- BRDF coefficients pre-computed



- BRDF coefficient vector for a given ω_o, looked up from a texture

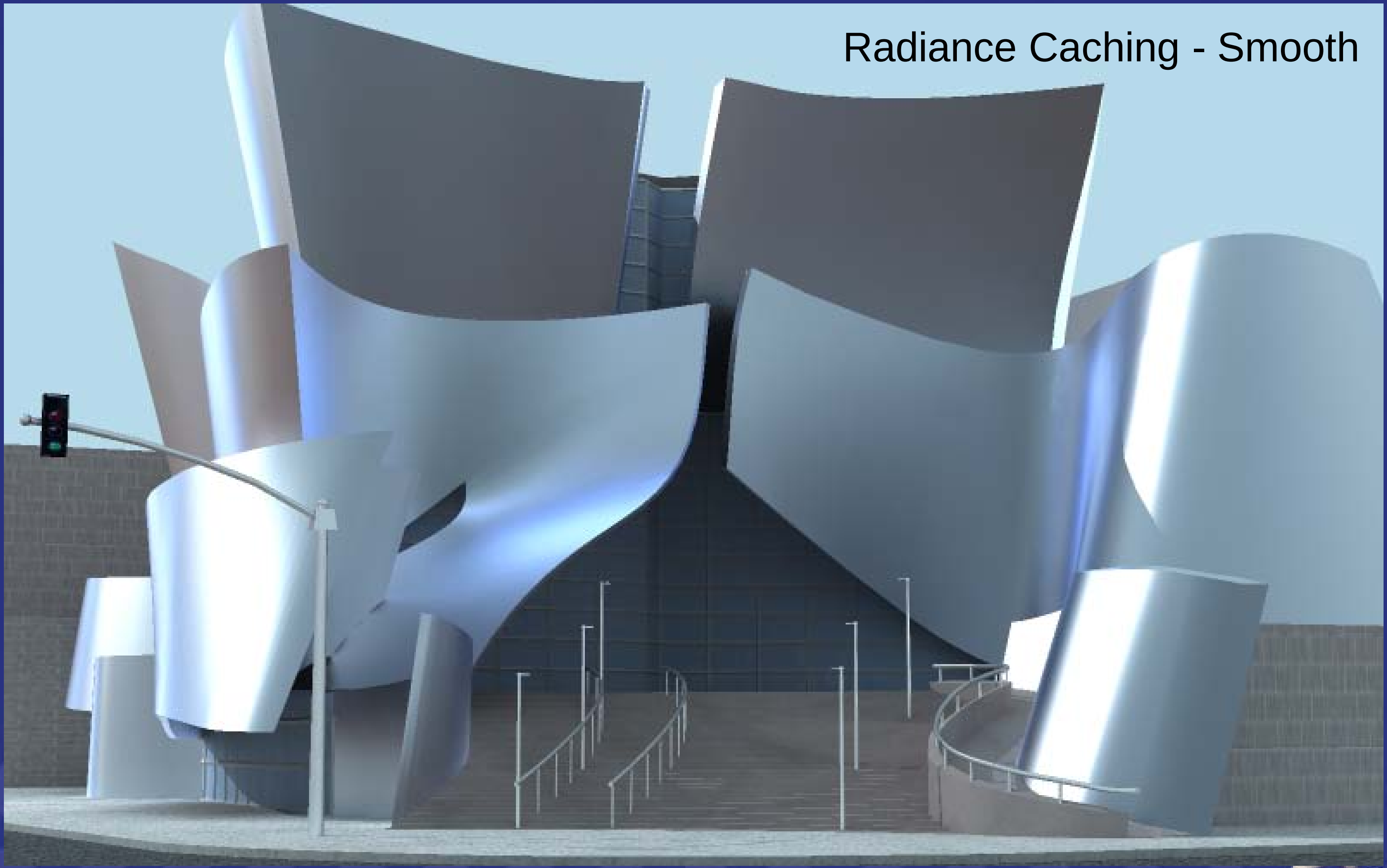
Readiance Caching Results

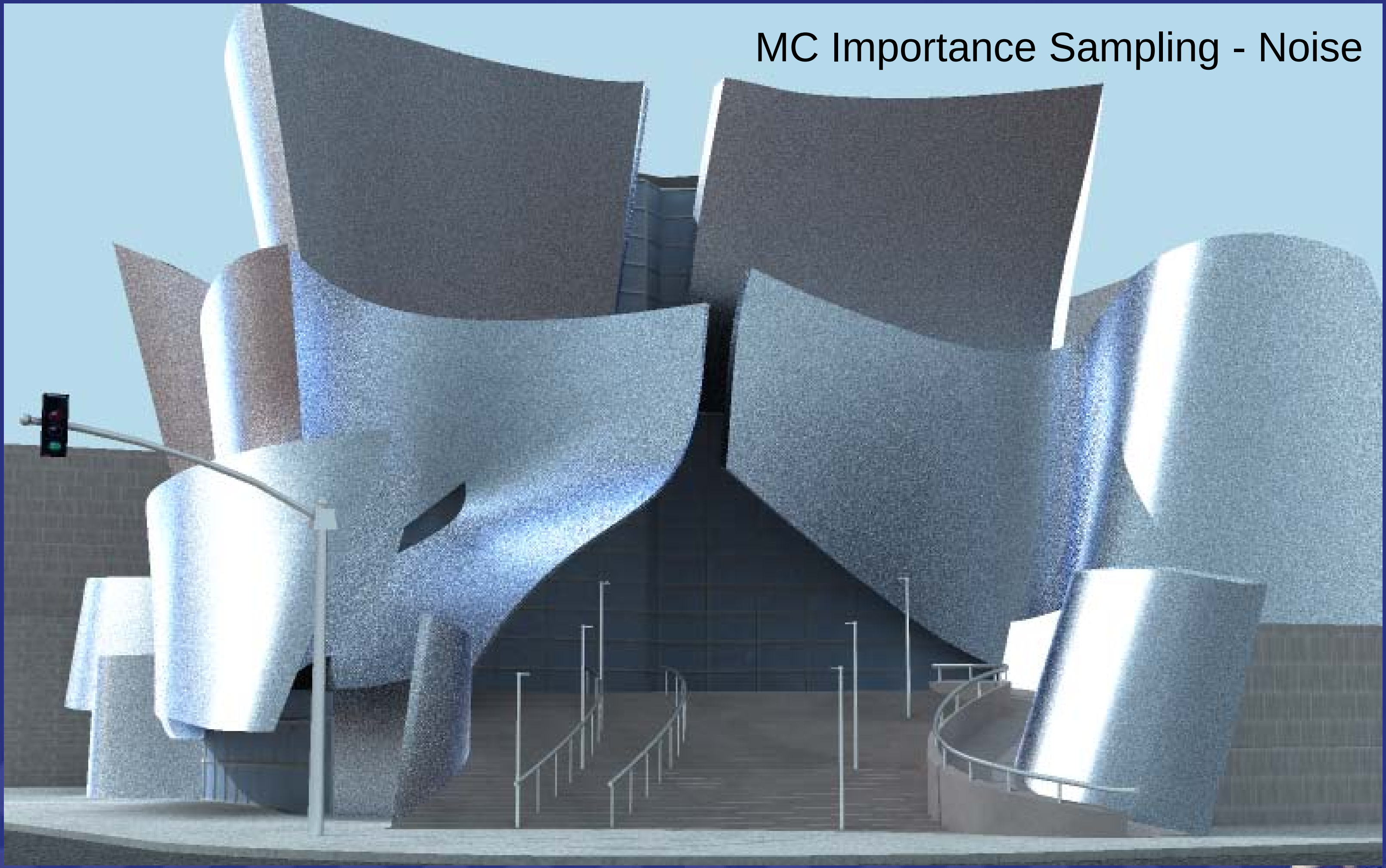
Radiance Caching vs. Monte Carlo

Same rendering time



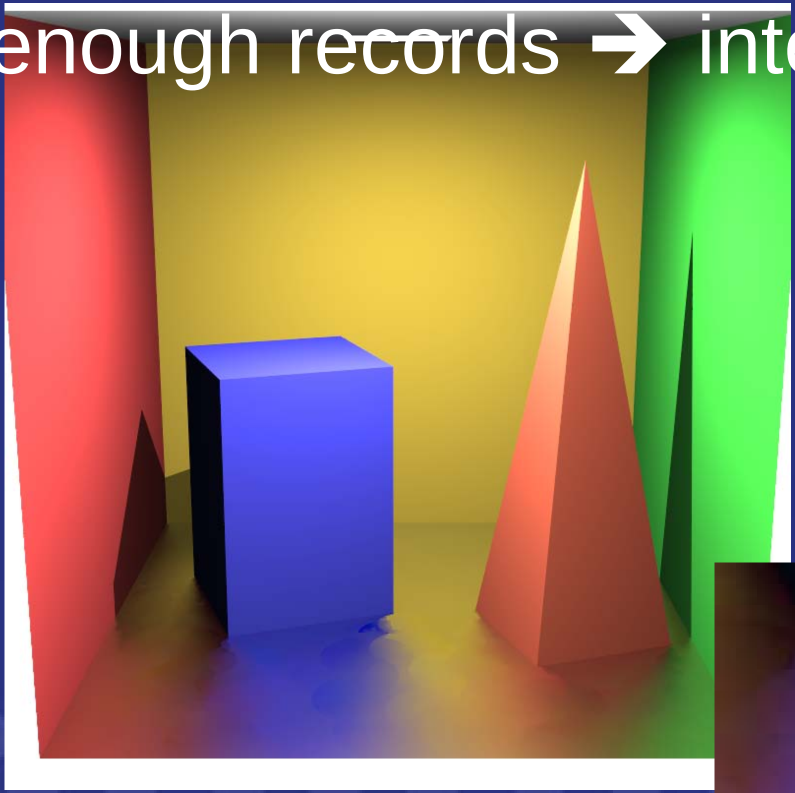
Radiance Caching vs. Monte Carlo

Same rendering time



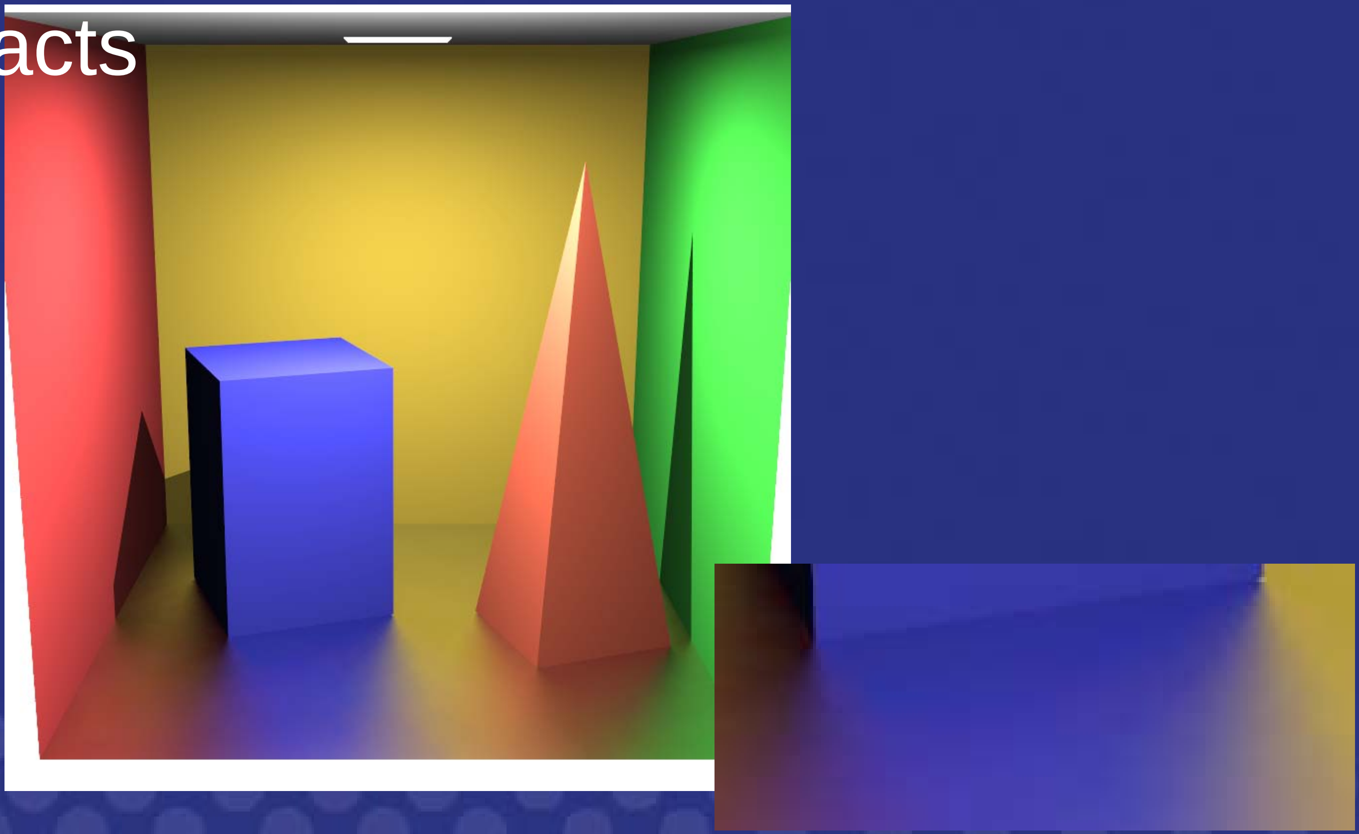
Adaptive Radiance Caching

- If rate of change of illumination is high and not enough records → interpolation artifacts



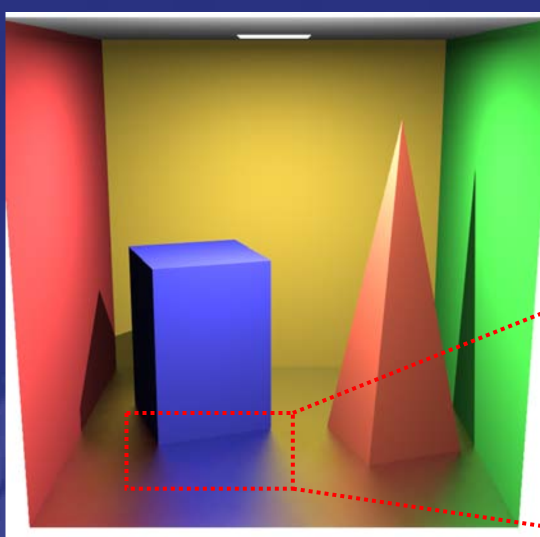
Adaptive Radiance Caching

- Adaptive caching prevents interpolation artifacts

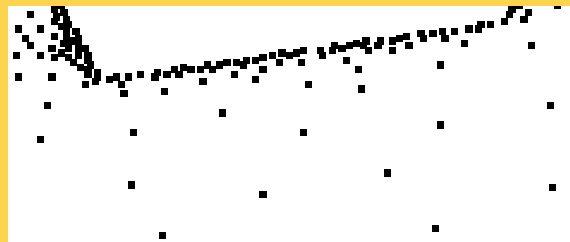


Adaptive Caching

- Adapt sampling to illumination



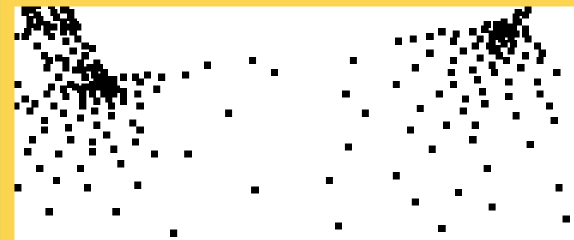
Old Approach



Adopted from [Ward88]

Artifacts on glossy surfaces

New Approach



Adaptive caching

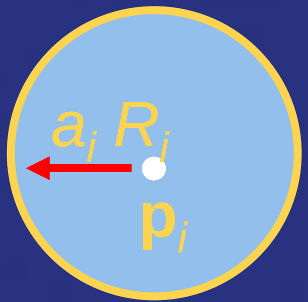
Artifacts-free image

Adaptive Radiance Caching

- Rate of change of illumination on glossy surfaces depends on
  - Actual illumination conditions
  - BRDF sharpness
  - Viewing direction
- Geometry-based criterion cannot take these into account → interpolation artifacts

Adaptive Radiance Caching

- a_i … now modulated per-record



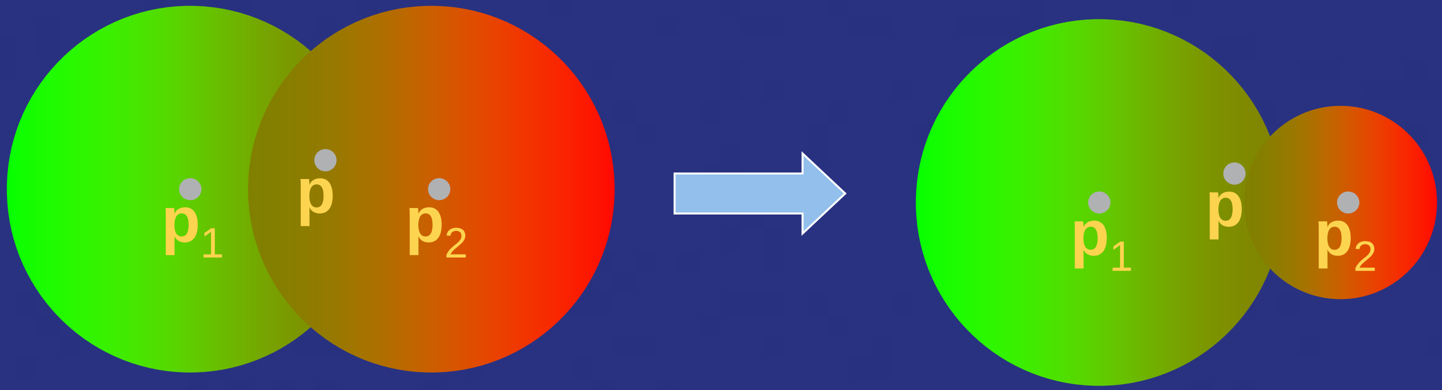
- influence area radius: $a_i \, . \, R_i$

New: illumination-based (a_i)

From irradiance caching: geometry-based (R_i)

Adaptive Radiance Caching

- Our approach
  - If discontinuity detected in the overlap area
  - Decrease radius



If | $L1(\mathbf{p}) - L2(\mathbf{p})$ | $> \tau$
then decrease radius

τ based on the Weber law

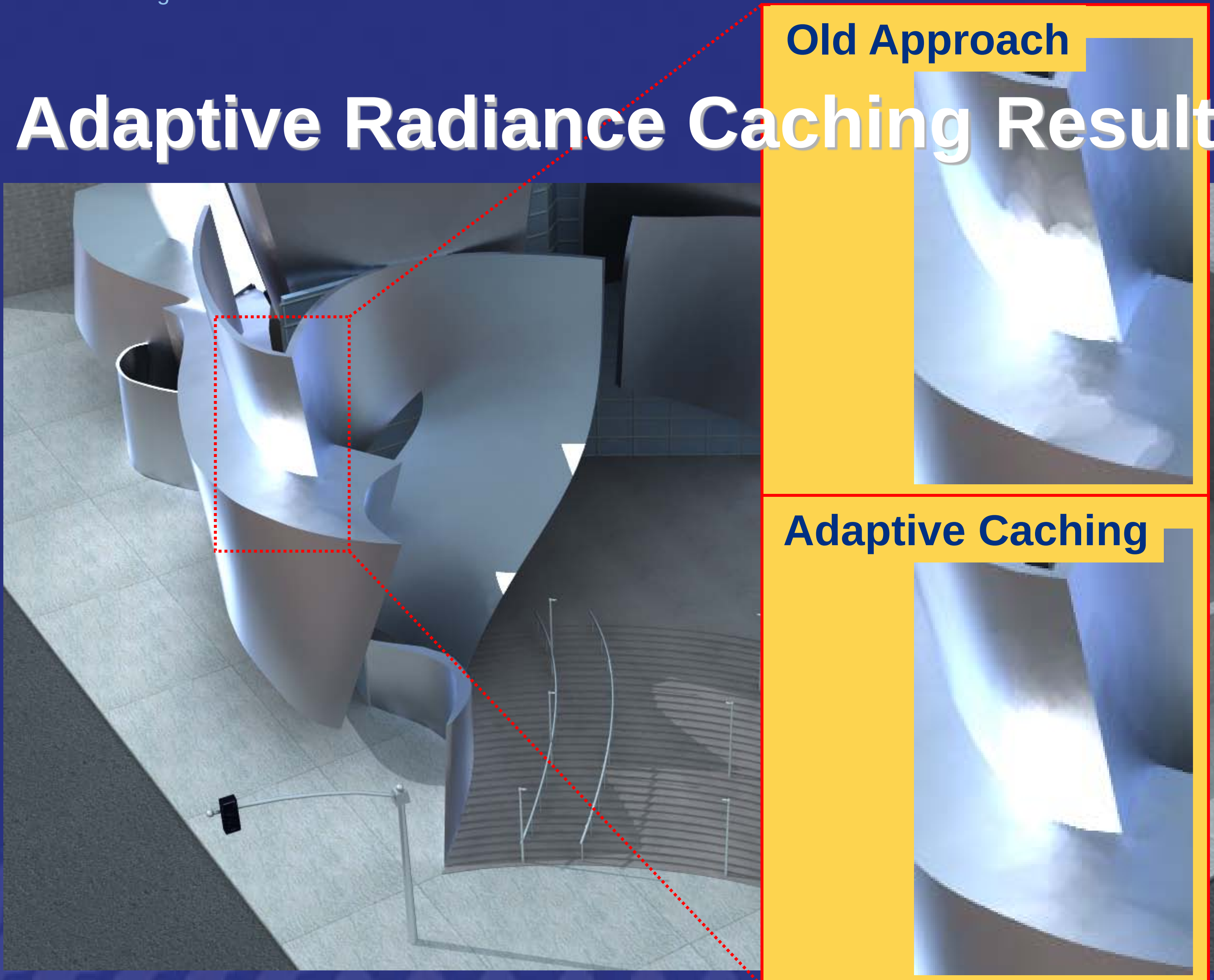
Adaptive Radiance Caching

- Radius decreases →

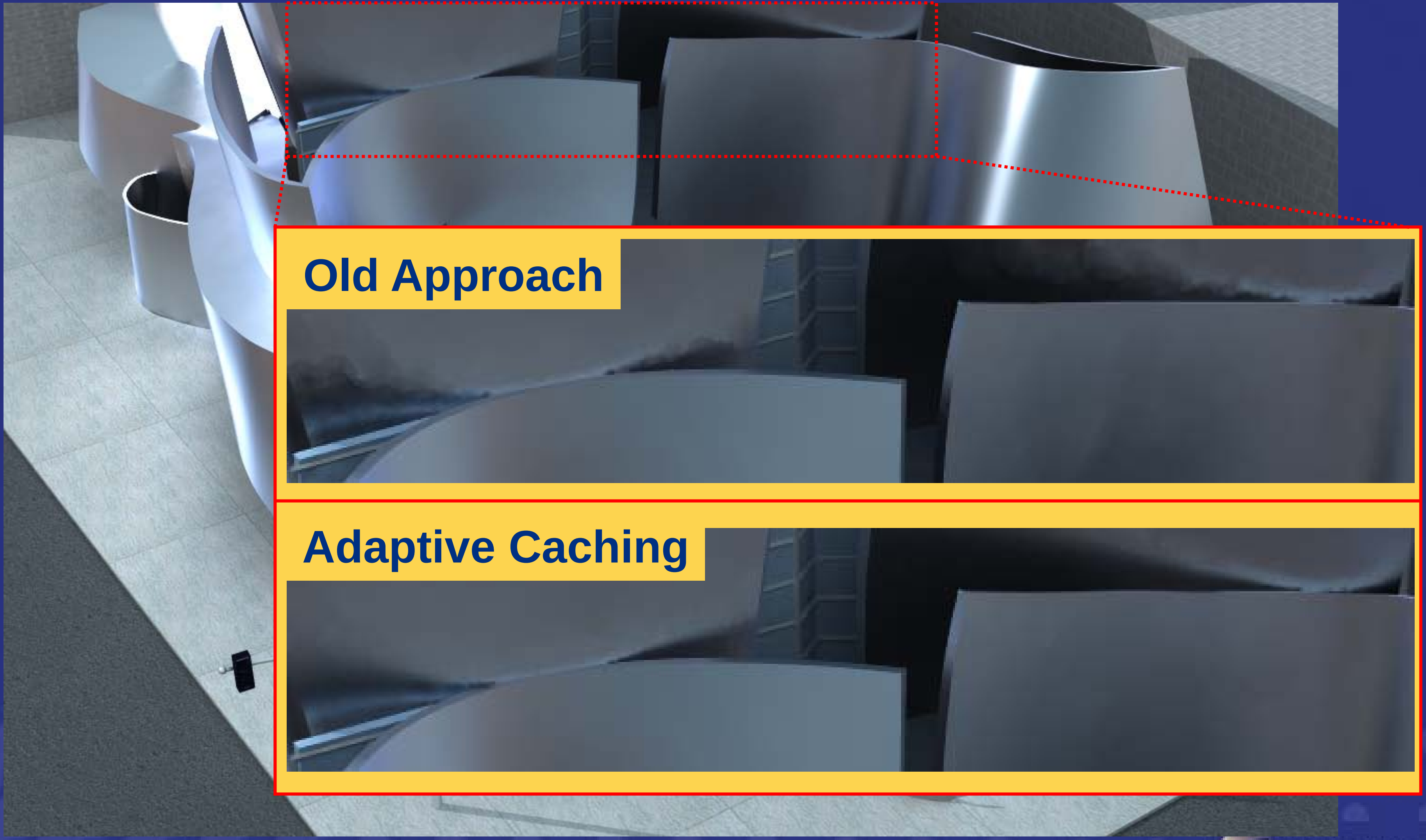
  local record density increases →

  better sampling

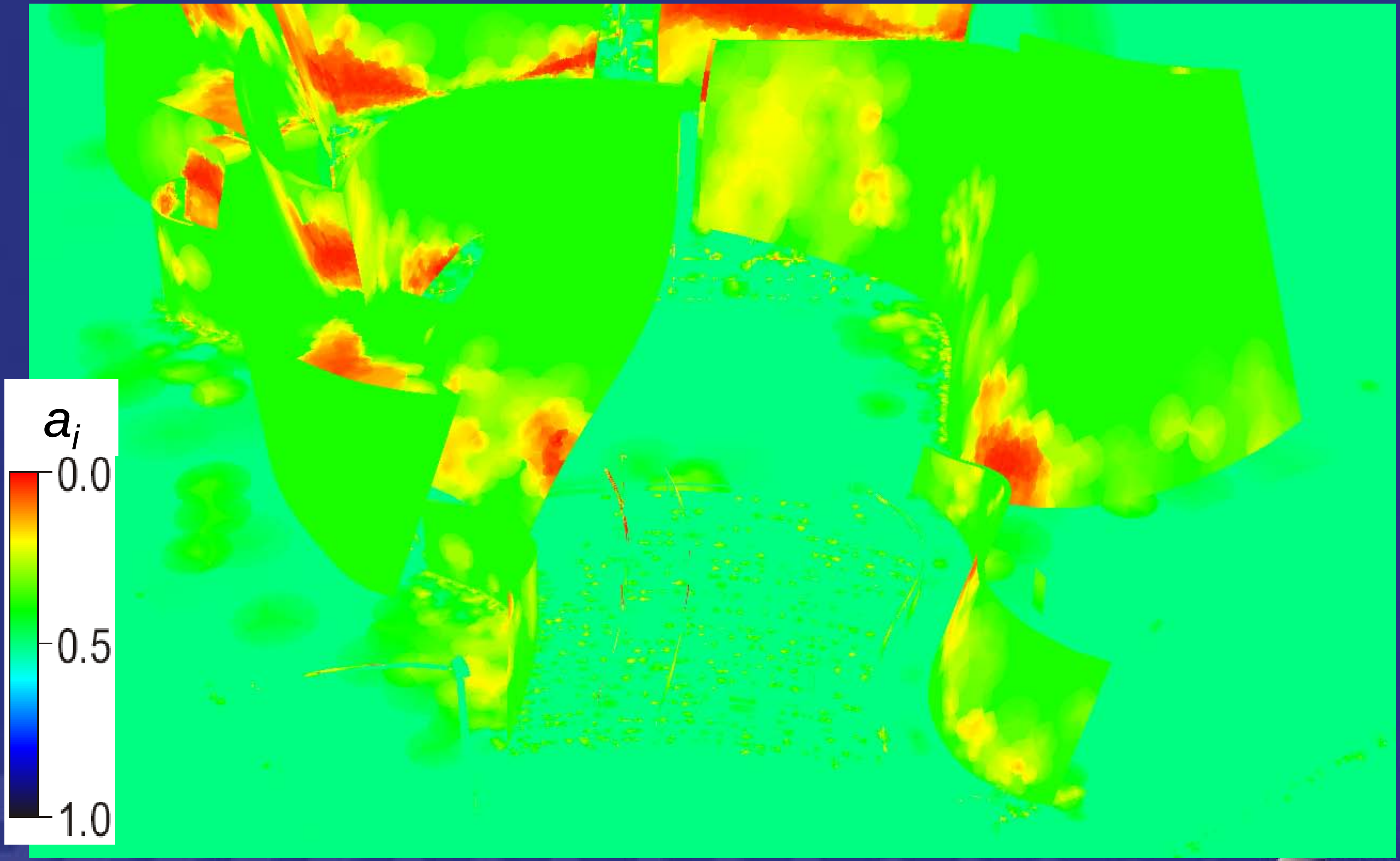
Adaptive Radiance Caching Results



Adaptive Radiance Caching Results

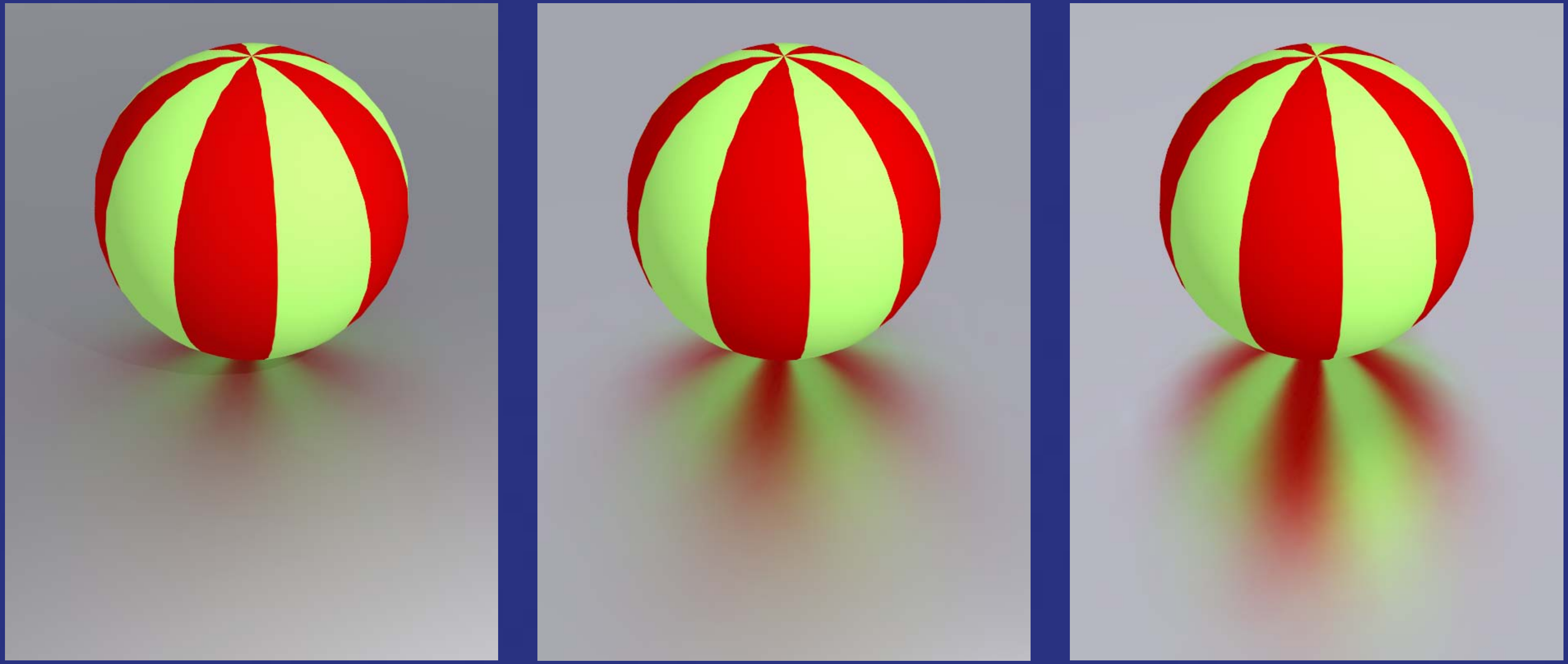


Adaptive Radiance Caching Results



Adaptive Radiance Caching Results

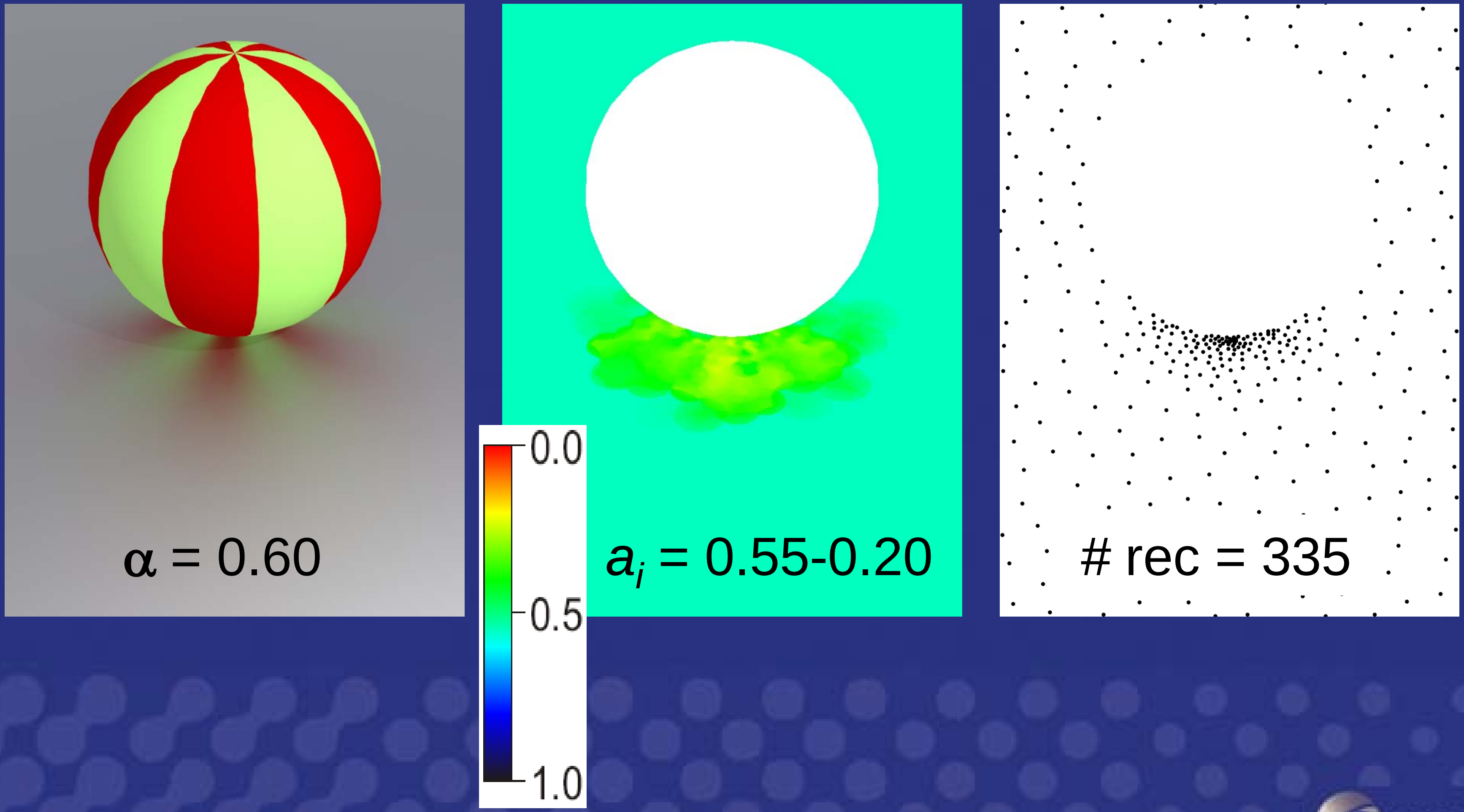
Adaptation to BRDF sharpness



Sharper BRDF → sharper reflections → Higher gradients

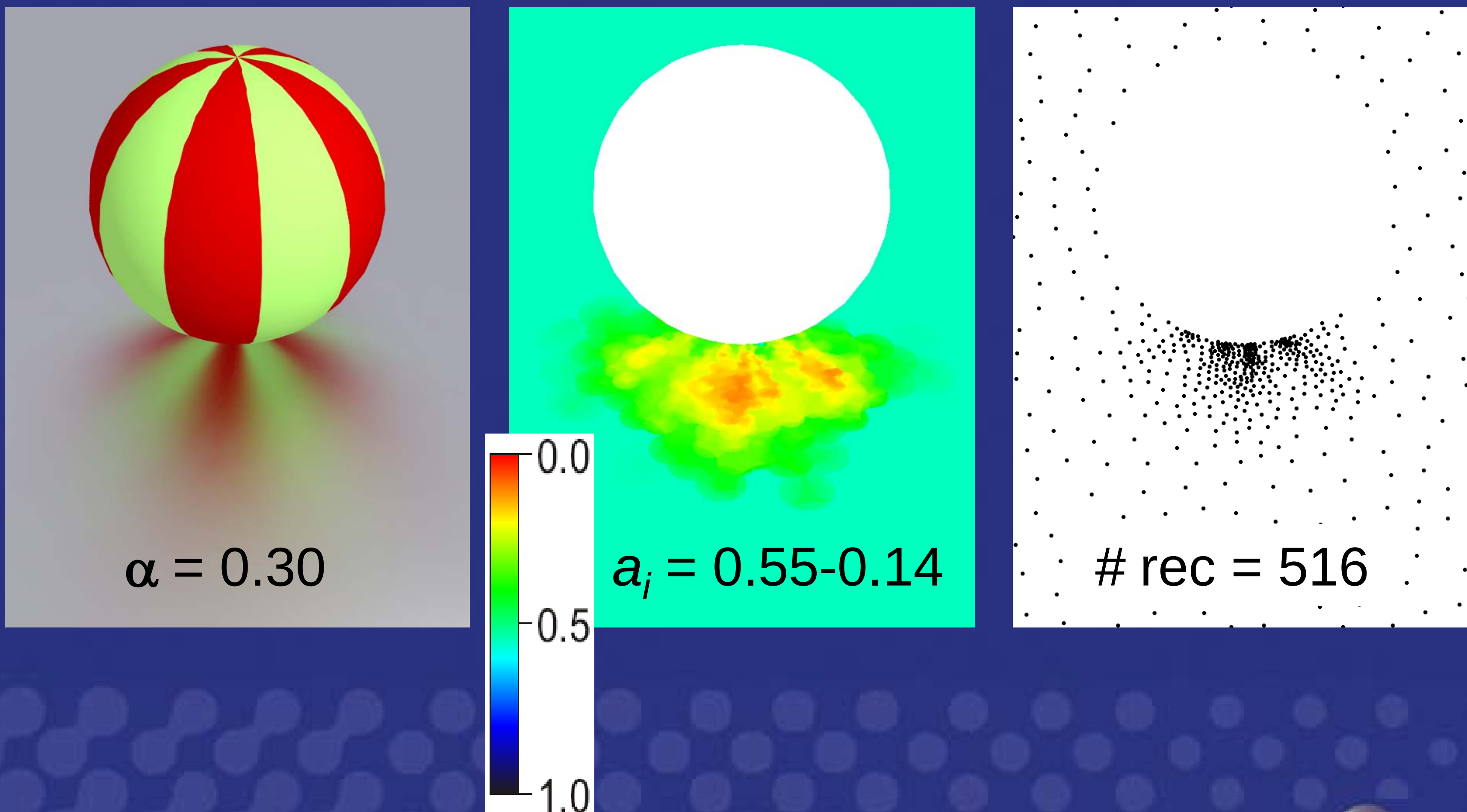
Adaptive Radiance Caching Results

Adaptation to BRDF sharpness



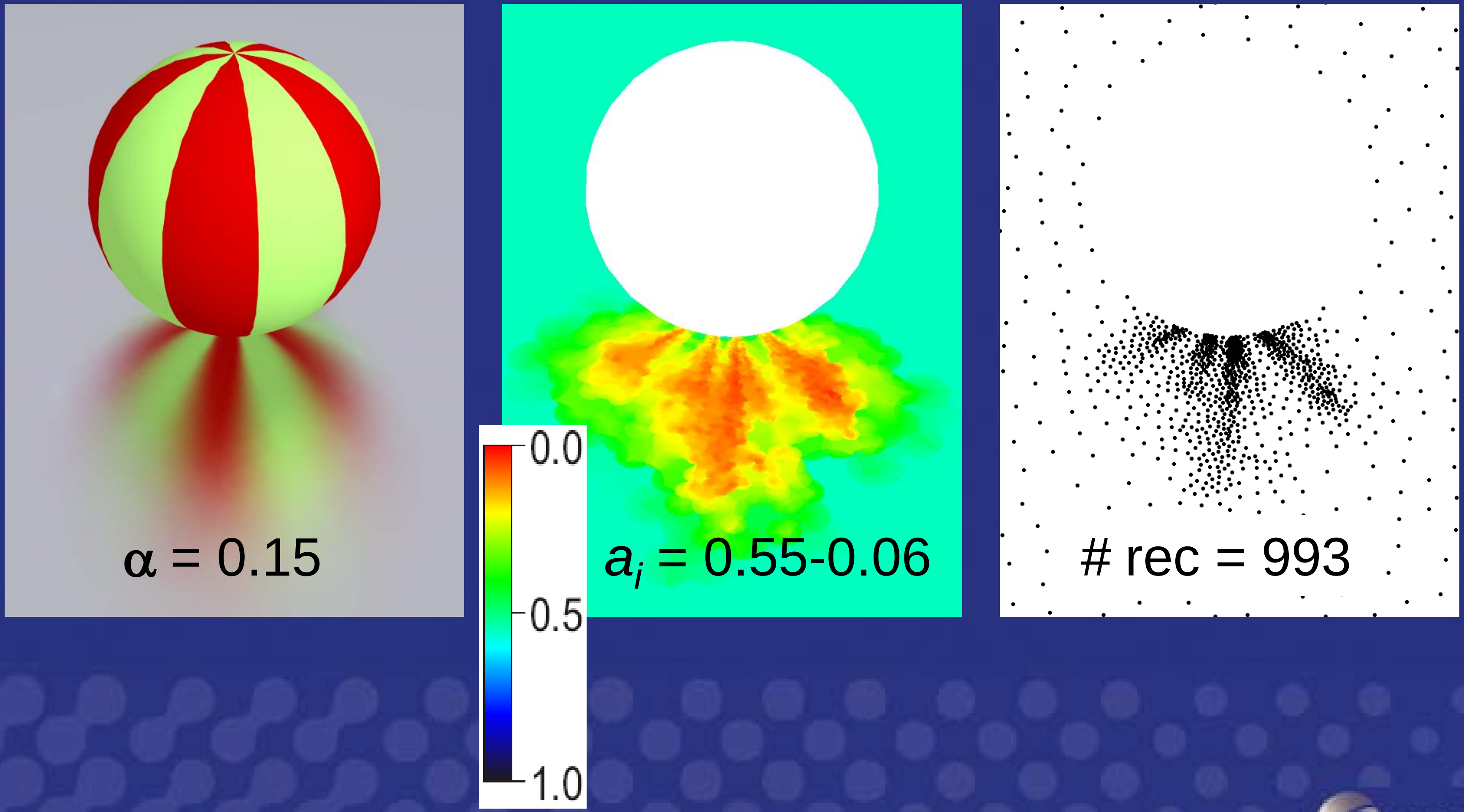
Adaptive Radiance Caching Results

Adaptation to BRDF sharpness



Adaptive Radiance Caching Results

Adaptation to BRDF sharpness



Video

- Disney Hall

Radiance Caching – Summary

- Caching works for glossy surfaces
- Gain not as good as for diffuse surfaces
- Well suited for measured reflectance
- Adaptive caching helps a lot
- For complex geometry and sharp reflections, importance sampling is better

Simple Approximation

- [Tabellion and Lamorlette 2004]
- Dominant incoming light direction with each record
- Interpolate over the surface
- Use as directional light
- Physically incorrect, but works fine in many cases