Eurographics 2013
May 6-10, Girona (Spain)

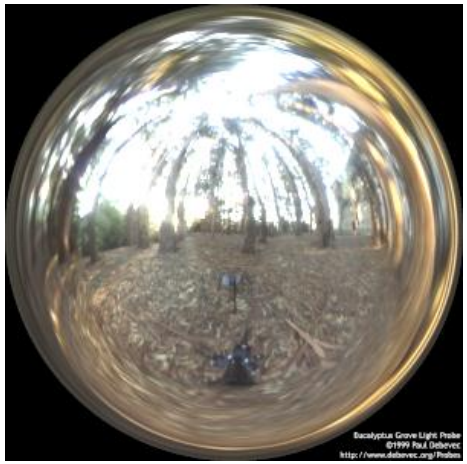
Real-Time Importance Sampling of Dynamic Environment Maps

Heqi Lu Romain Pacanowski Xavier Granier
Inria – Univ. Bordeaux - IOGS - CNRS

Motivation

Captured HDR EM Stream

3D Models



Real-Time Rendering

Unbiased

Direct rendering at the same time as EM capturing

Previous Work

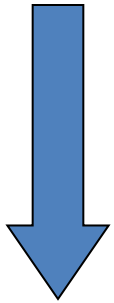
- Static environment map
  - Fast Hierarchical Importance Sampling [e.g., ODJ04]
- Dynamic environment map
  - Precomputed Radiance Transfer. [e.g., WTL06]
  - Point light sources [e.g., HSK*05]
    - Precomputation ☹
    - Additonal storage ☹
  - Real-Time Shading with Filtered IS.[e.g. KC08]
    - Biased☹

Contributions

- Dynamic Sample Distribution for cube map
  - Reduce useless samples
- Unbiased Monte Carlo estimator built on MIS
  - Low number of samples
  - High quality
- GPU Algorithm without precomputations
  - Fully Dynamic EMs as light sources

Problem Statement

$$L(p,o) = \int_{\Omega} \rho(o,\omega) < n,\omega > L(\omega) d\omega$$



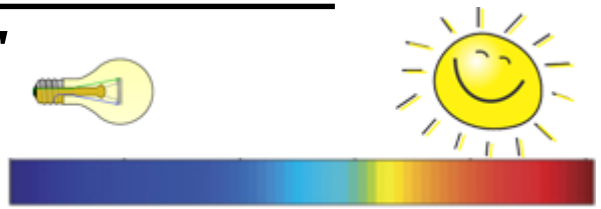
Monte Carlo

$$L(p,o) \coloneqq \frac{1}{N} \sum \frac{\rho(o,\omega) < n,\omega > L(\omega)}{PDF}$$

The better PDF the smaller N

Problem Statement

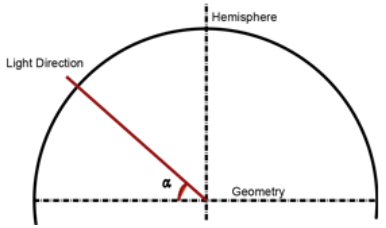
$$L(p,o) := \frac{1}{N}\sum \frac{\rho(o,\omega) < n,\omega > L(\omega)}{PDF}$$



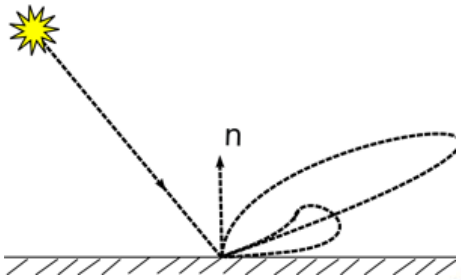
PDF?

$\rho(o,\omega)$

$< n,\omega >$



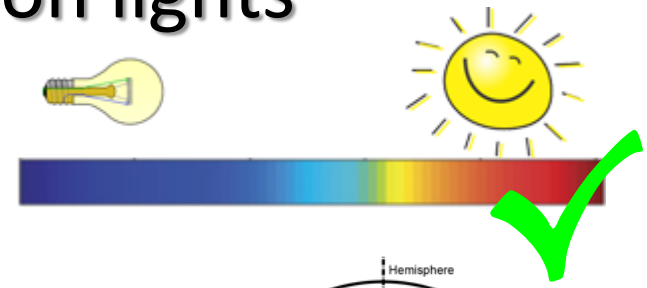
$L(\omega)$



Our Method Overview

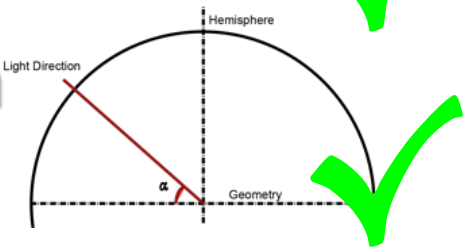
For each frame

- Tabulated CDF construction on lights
- Light sample generation

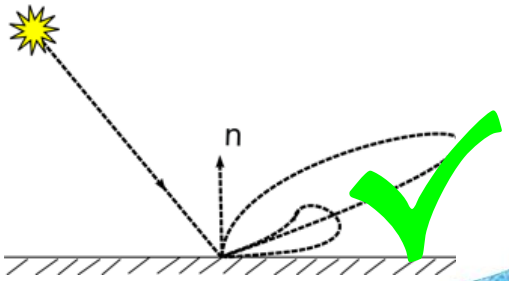


For each pixel (*Deferred shading*)

- Dynamic light sample distribution
  - ***Light + cosine***



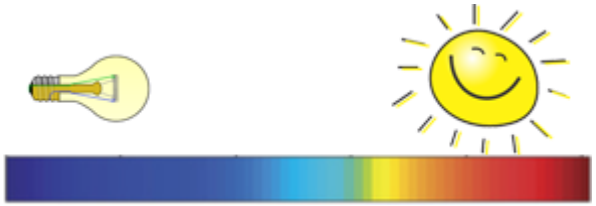
- BRDF sample generation
- Shading

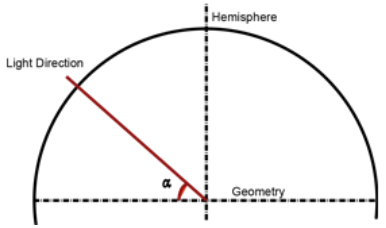


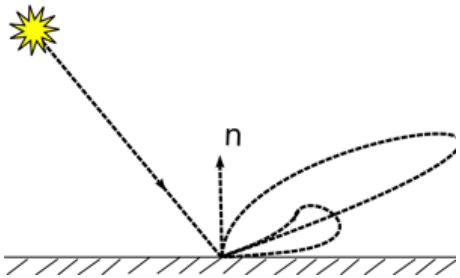
Stream

For each frame:

Load current EM





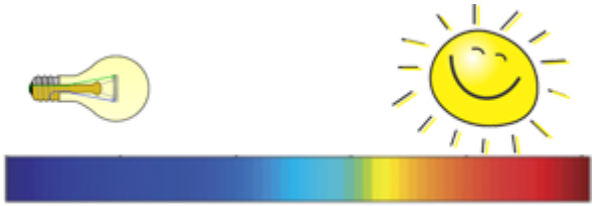


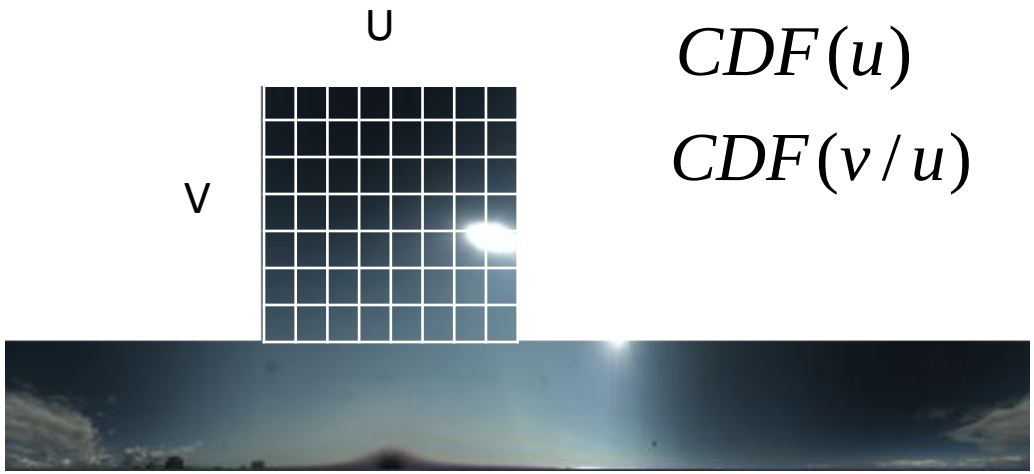
Tabulated CDF Construction

For each frame:

Load current EM

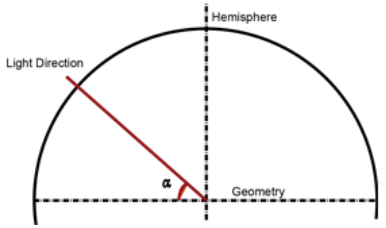
Build CDFs of Light Intensity

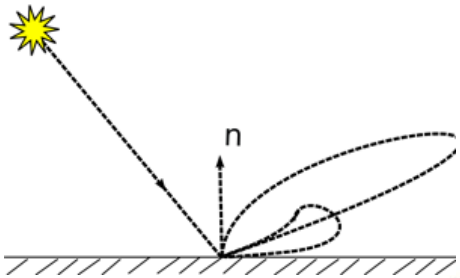




$$CDF(u)$$

$$CDF(v/u)$$

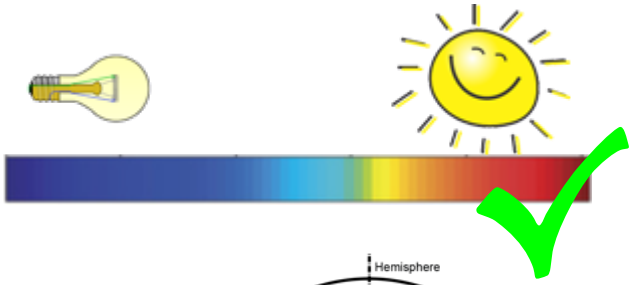




Light Source Sampling

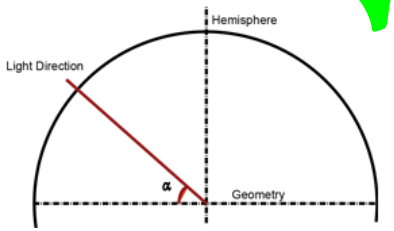
For each frame:

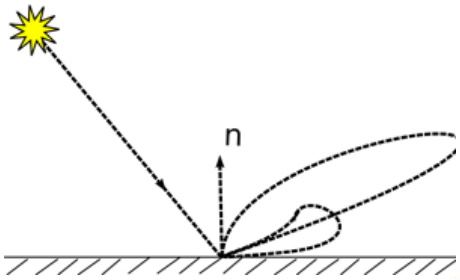
Same number of samples on each face



$$CDF(u)$$

$$CDF(v/u)$$

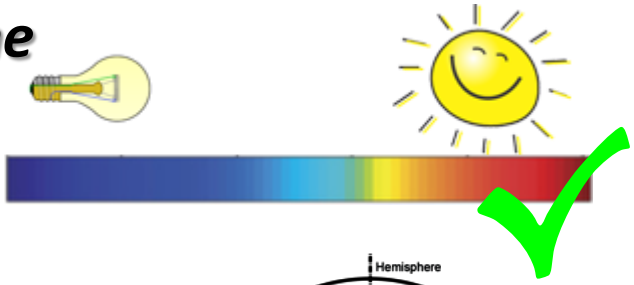


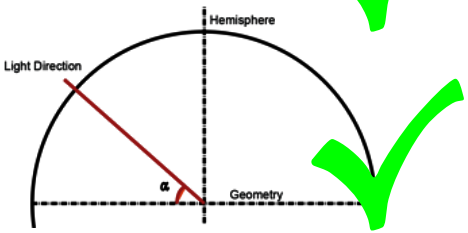


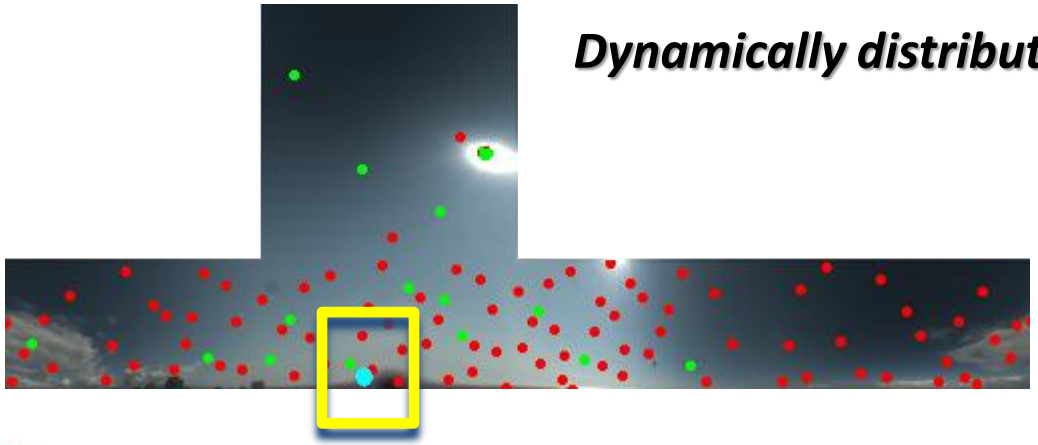
Dynamic Sample Distribution

For each pixel:

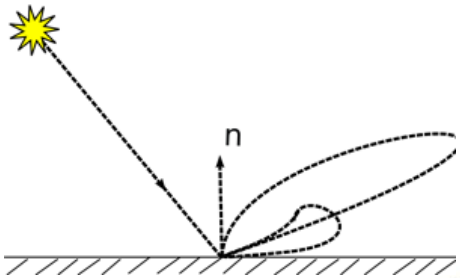
Select samples wrt. ***Light + Cosine***





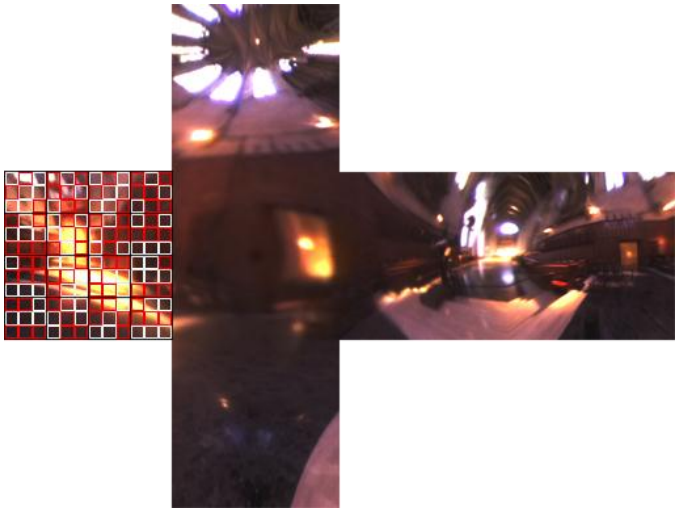


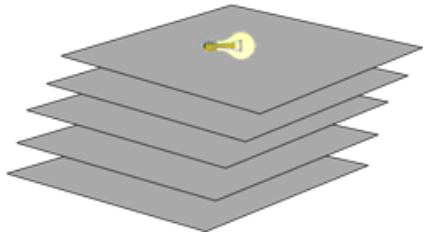
Dynamically distributed



Dynamic Distribution of Samples

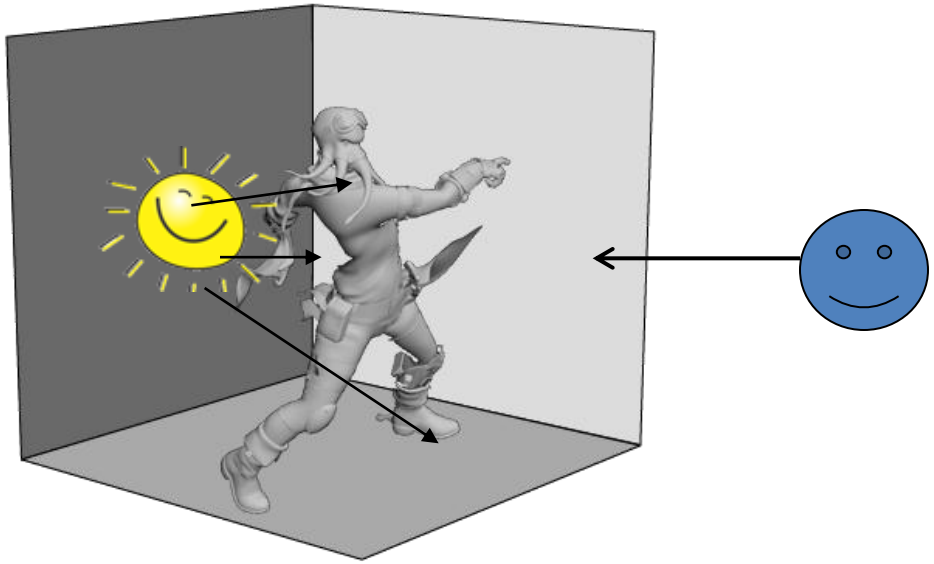
- According to ***Contribution*** *of* ***Faces***
  - Light Energy $L(\omega)$ → I_f

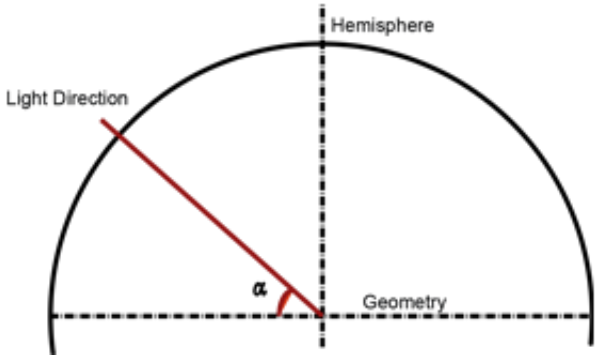




Dynamic Sample Distribution

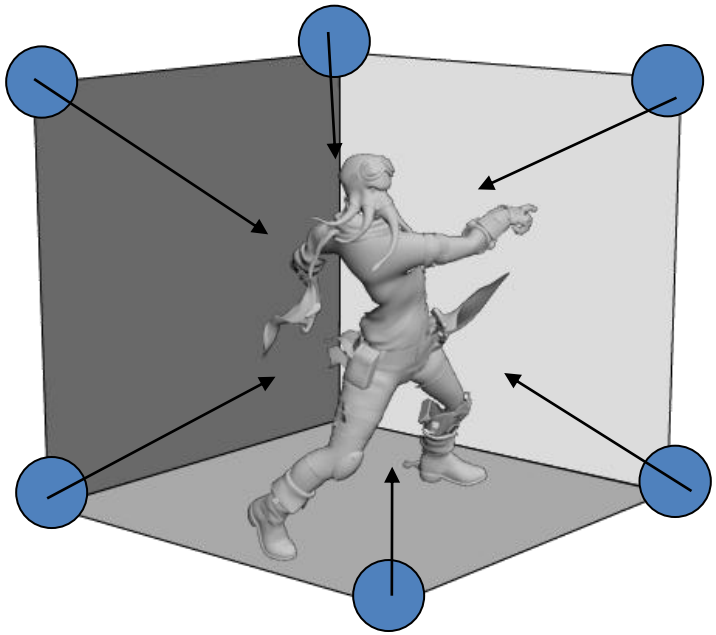
- According to ***Contribution*** *of* ***Faces***
  - Light Energy $L(\omega)$ → I_f
  - Cosine term $< n, \omega >$





Dynamic Sample Distribution

- According to ***Contribution*** *of* ***Faces***
  - Light Energy $L(\omega)$ ⟹ I_f
  - Cosine term $< n, \omega >$



$$C_f = \sum_{corner}^{4} < n, \omega >_{corner}$$

Pseudo Form Factor(point to face):

$$F_f(n) = \frac{C_f}{\sum_{f}^{6} C_f}$$

Dynamic Sample Distribution

- According to ***Contribution*** *of* ***Faces***
  - Light Energy $L(\omega)$ → I_f
  - Cosine term $< n, \omega >$ → $F_f(n)$

Face contribution weight:

$$\mu_f(n) = \frac{F_f(n) I_f}{\sum_f^6 F_f(n) I_f}$$

Smooth Transitions

Number of samples for a face:

$$N_f = \mu_f(n) N_L$$

– Use only Integer?

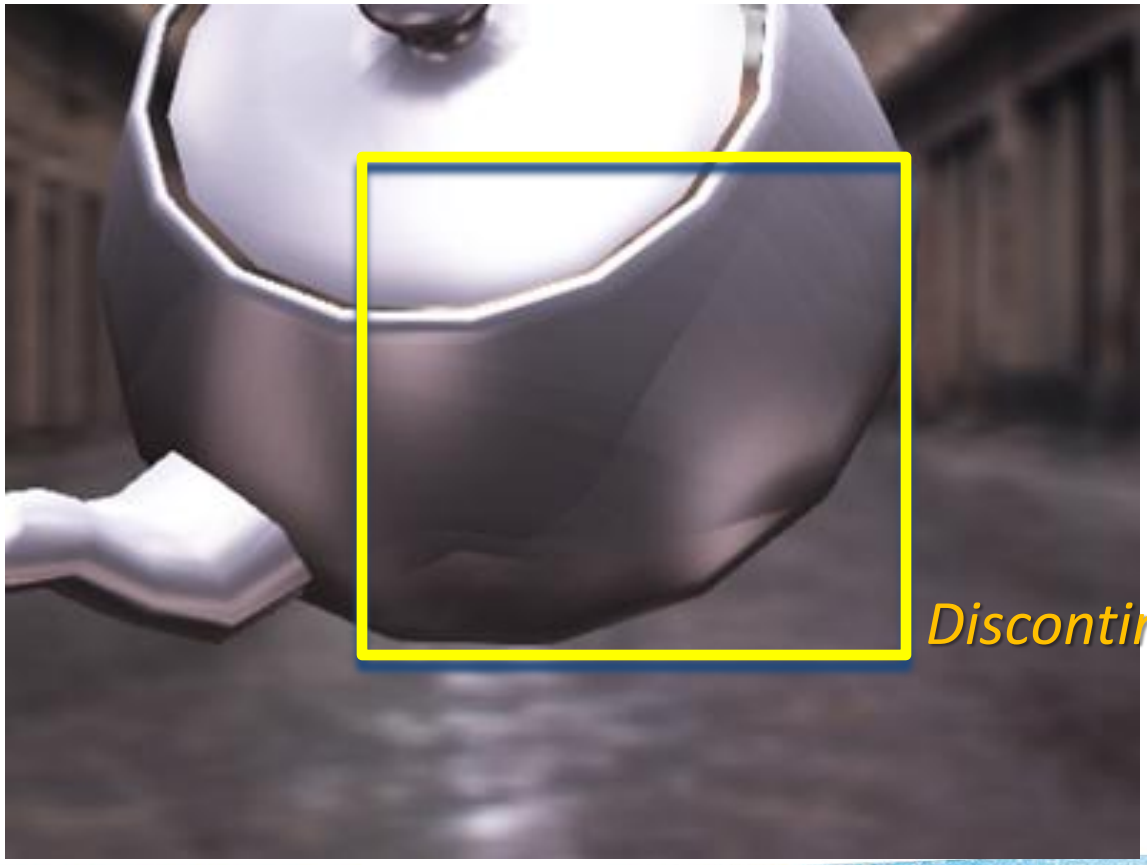
$$N_f = N + frac$$

$$L(p,o) := \frac{1}{N}\left(\sum \frac{p(o,\omega) < n,\omega > L(\omega)}{PDF}\right)$$

$$L(p,o) := \frac{1}{N}\left(\sum_{i}^{N} F_i\right)$$

Smooth Transitions

Integer number of samples:



Smooth Transitions

Number of samples for a face:

$$N_f = \mu_f(n) N_L$$

- Different for each pixel
- Integer number ?

$$N_f = N + frac$$

$$L(p,o) := \frac{1}{N}\left(\sum_i^N F_i\right)$$

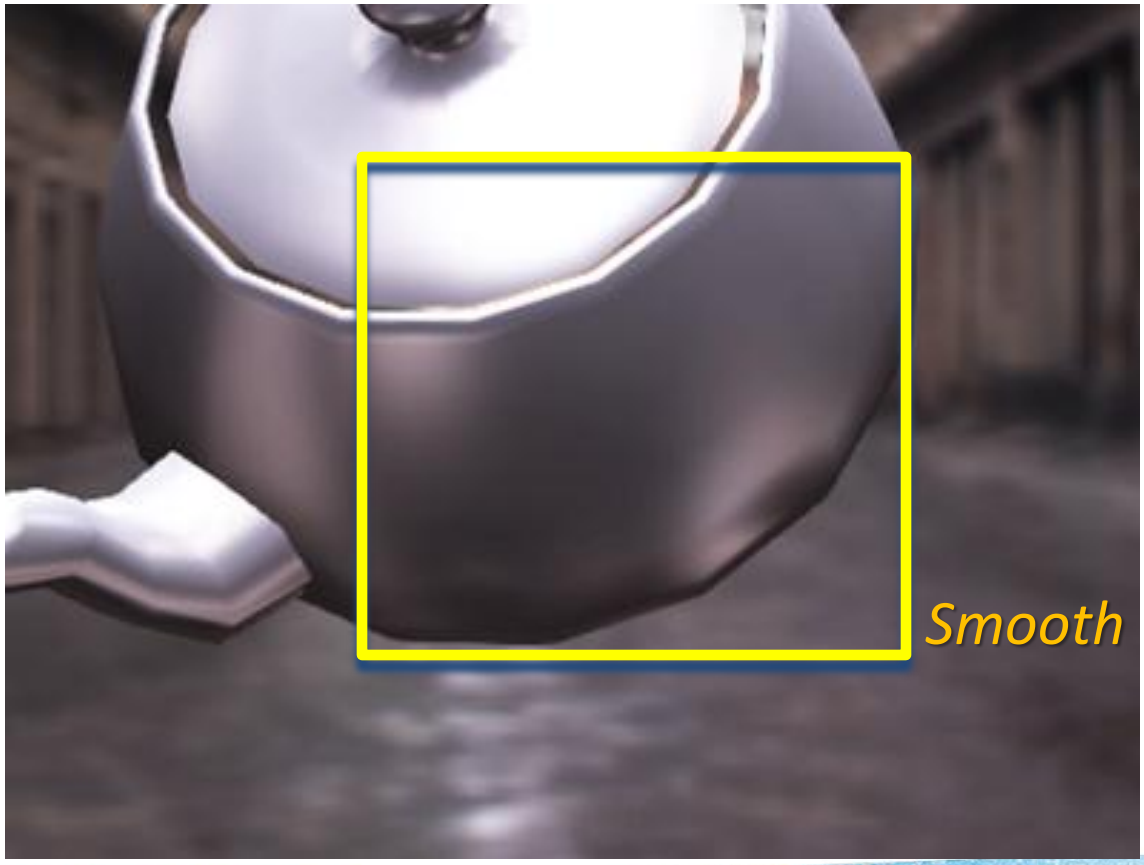
- Floating point number

Unbiased

$$L(p,o) := \frac{1}{N_f}\left(\sum_i^N F_i + frac \times F_{N+1}\right)$$

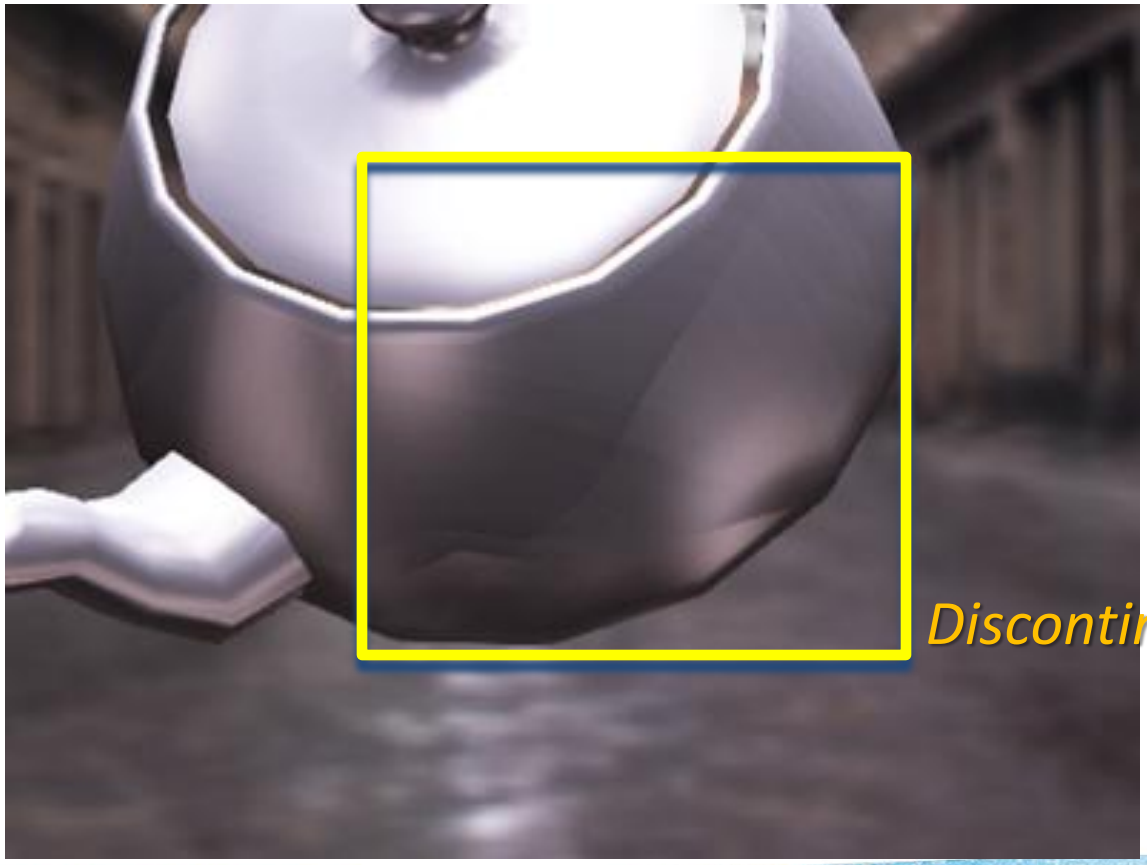
Smooth Transitions

Floating point number of samples :



Smooth Transitions

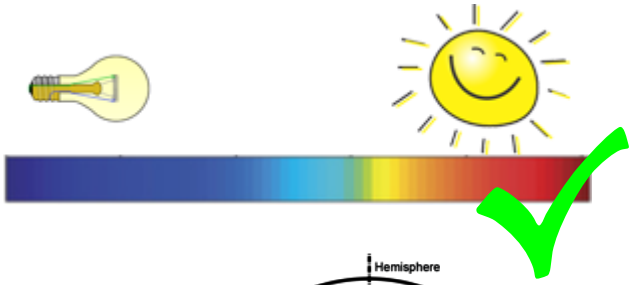
Integer number of samples:

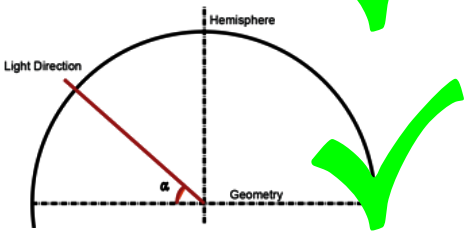


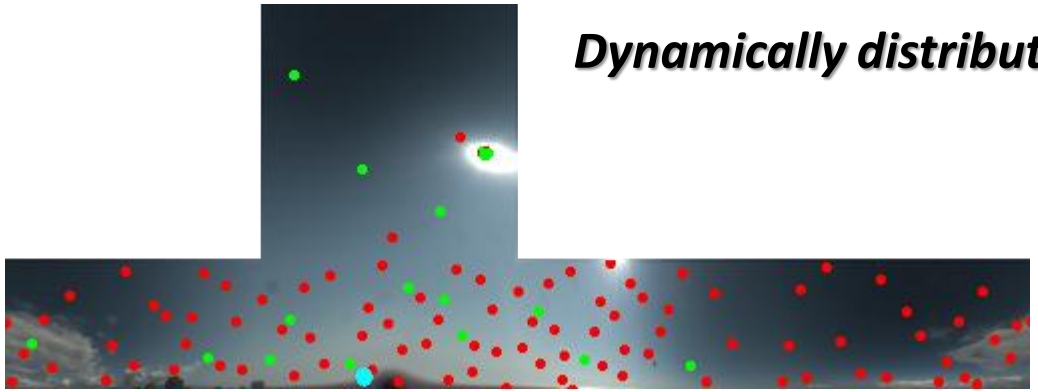
Dynamic Sample Distribution

For each pixel:

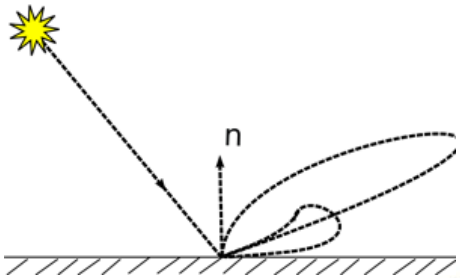
Samples wrt. ***Light + Cosine***



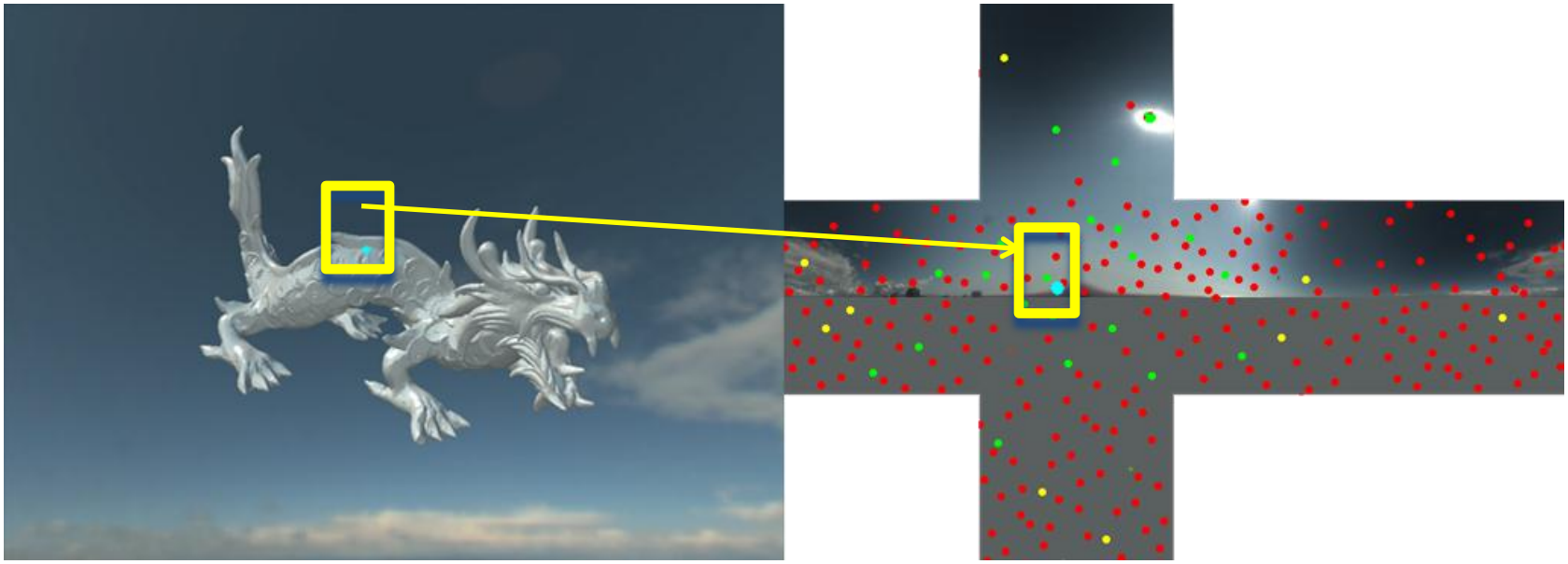




Dynamically distributed

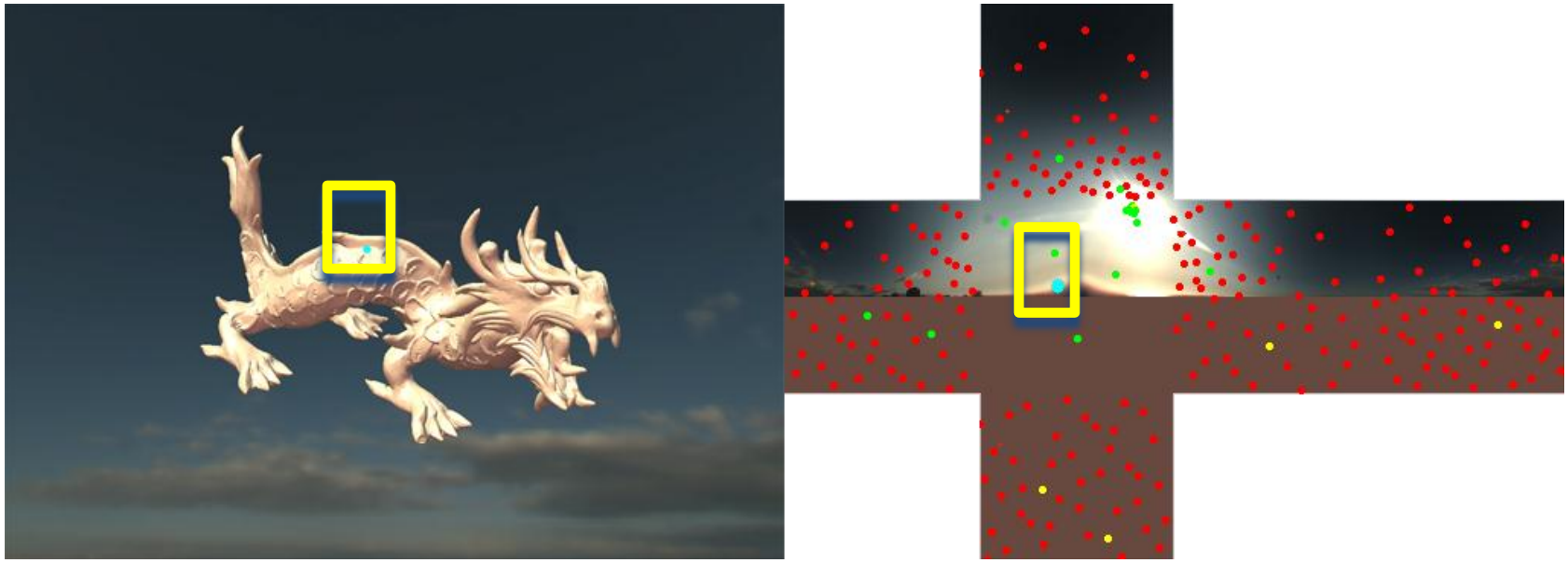


Results



Red : Samples discarded by ***face weighting***
Green: Samples with positive cosine
Yellow: Samples with negative cosine

Results



Red : Samples discarded by ***face weighting***
Green: Samples with positive cosine
Yellow: Samples with negative cosine

BRDF Sampling

- Multiple Importance Sampling
  - Select N_L samples from *Light Sources*
  - Select N_B samples from *BRDFs*

$$N_s = N_b + N_L$$

Why MIS ?

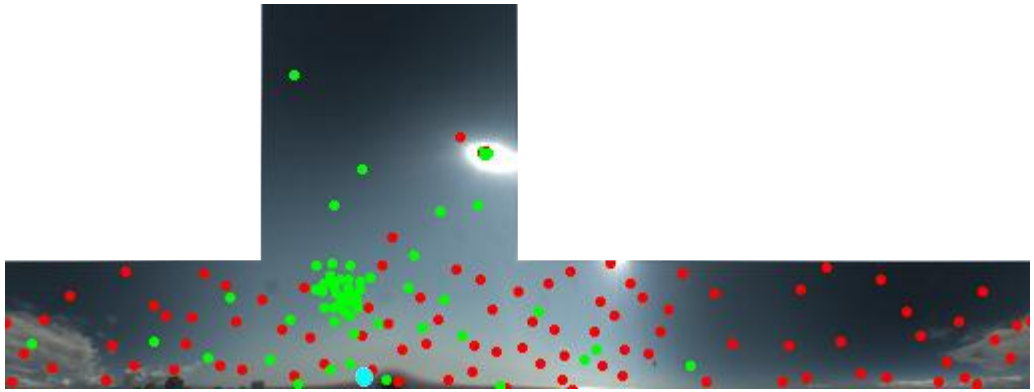
Why not sample the product

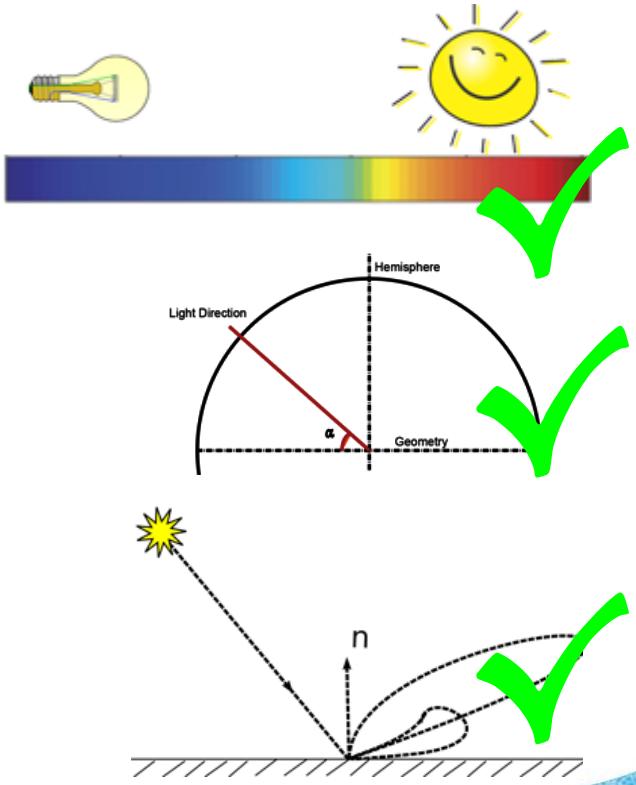
$$\rho(o,\omega) < n,\omega > L(\omega)$$

BRDF Sampling

For each pixel:

Generate samples from BRDF





GPU Implementation

- For each frame
  - CDFs Construction
  - Cube map sampling
    - Consider only light intensity
    - Static number of samples per face
  - Face weighting
    - Dynamic number of samples
  - BRDF samples combination
  - Shading

Conclusion

- Unbiased Estimator
  - Limits the number of useless samples
  - Reduces popping artifacts
    - ➔ Floating point number of samples
  - Integrates easily with MIS
- Real-time Solution without precomputations
  - Dynamic scenes
  - Dynamic environment maps
- Future work
  - Visibility and indirect lighting effects


Eurographics 2013
May 6-10, Girona (Spain)
Real-Time Importance Sampling of Dynamic Environment Maps
Thank you for your attention