Adaptive Sampling and Filtering for Fast Monte-Carlo Rendering

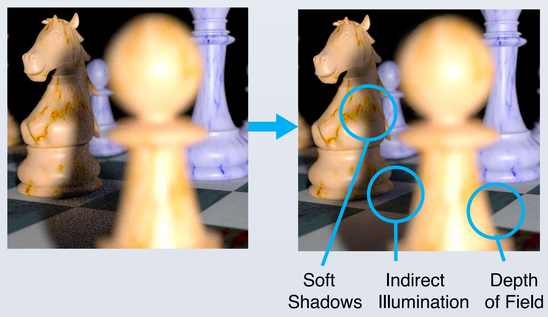
Soham U. Mehta[1] Ravi Ramamoorthi[2] Fredo Durand[3]

[1] UC Berkeley [2] UC San Diego [3] MIT CSAIL

ABSTRACT

- Distribution effects (defocus and motion blur, soft shadows and indirect illumination) are important for photo-realistic rendering
- Integrate radiance over lens, time, light, angle
- Monte-Carlo sampling converges very slowly, produces noise with fewer samples
- We propose a fast and adaptive sampling and filtering algorithm

INTRODUCTION



We make the following **contributions**:

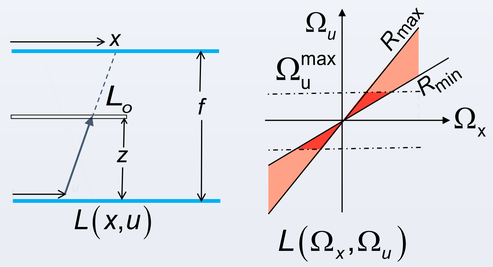
- Combined 6D frequency analysis
- Factoring texture and irradiance
- Two-level adaptive sampling strategy
- DOF + Soft shadows + Indirect Illum. in ~5 sec

Previous work on similar problems includes:

- Adaptive multi-level denoising [Kalantari & Sen 13]
- Indirect LF Reconstuction [Lehtinen et al 12]
- Sheared Filtering [Egan et al 09,10,11]
- Covariance Tracing [Belcour et al 13]
- Layered Reconst. for DOF/MB [Munkberg et al 14]
- They are either not general enough, or have high reconstruction overheads
- Our filtering scheme is very fast and accurate

THEORY

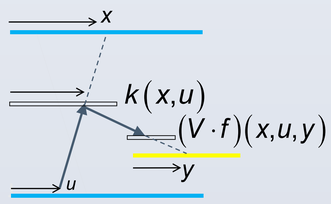
Axis-aligned filter for defocus blur



- For defocus blur only, axis-aligned bandwidth is:

$$\Omega_x^d = \frac{\Omega_u^{max}}{R_{min}}$$

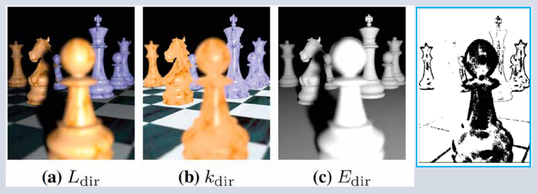
Radiance Factoring



- Defocus blur couples Texture and Illumination:

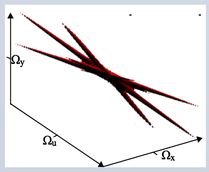
$$L_{dir}(x) = \int k(x,u)\left[\int (V \cdot f)(x,u,y) I(y)\, dy\right] A(u)\, du$$
$$\approx k_{dir}(x) \times E_{dir}(x)$$

- Filter separately whenever possible:



(a) L_{dir} (b) k_{dir} (c) E_{dir}

Axis-aligned filter for irradiance with DOF

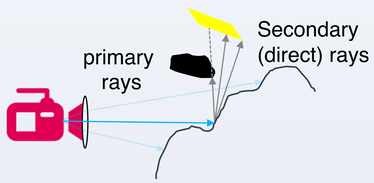


$$\Omega_x^s = \min\left(\frac{\Omega_y^{max}}{s_{min} / l_p}, \frac{\Omega_u^{max}}{R_{min}}\right)$$

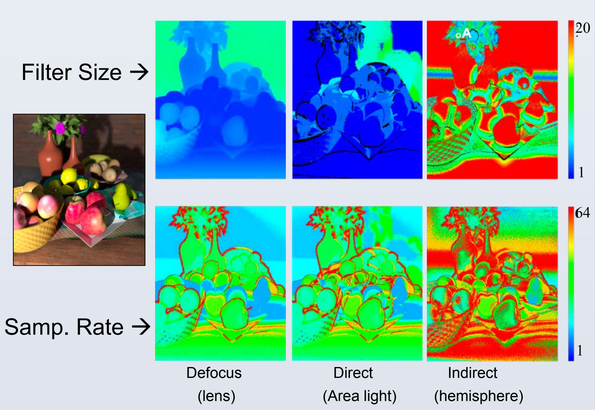
$V(\Omega_x,\Omega_u,\Omega_y)$

IMPLEMENTATION

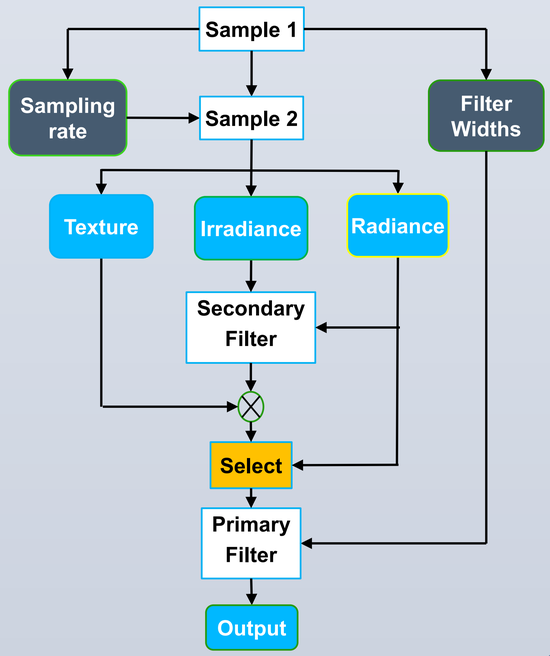
Adaptive Sampling



- Analyze packing of spectral aliases in 3D
- Separate sampling rate for primary and secondary

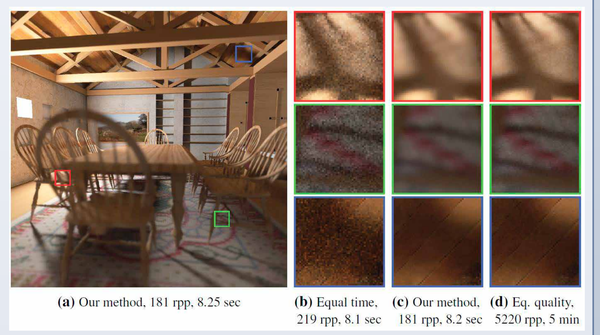


Algorithm

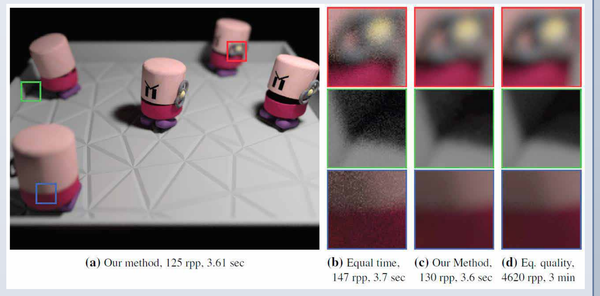


RESULTS

- We reduce sampling rate by 30x, filtering in 80 ms
- Results using Nvidia GTX Titan GPU
- Ray-tracing with Nvidia's OptiX, filtering with CUDA

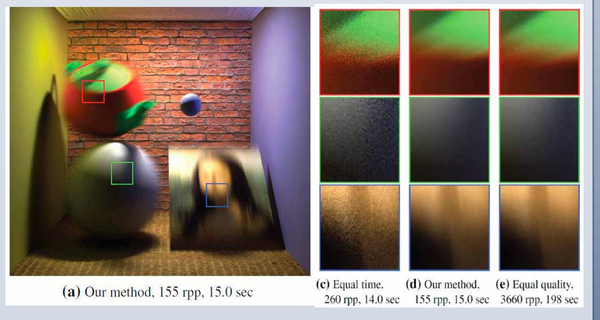


(a) Our method, 181 rpp, 8.25 sec (b) Equal time, 219 rpp, 8.1 sec (c) Our method, 181 rpp, 8.2 sec (d) Eq. quality, 5220 rpp, 5 min



(a) Our method, 125 rpp, 3.61 sec (b) Equal time, 147 rpp, 3.7 sec (c) Our Method, 130 rpp, 3.6 sec (d) Eq. quality, 4620 rpp, 3 min

- We can also handle motion blur (CPU based)



(a) Our method, 155 rpp, 15.0 sec (c) Equal time. 260 rpp, 14.0 sec (d) Our method. 155 rpp, 15.0 sec (e) Equal quality. 3660 rpp, 198 sec

REFERENCES

Mehta et al 2014. Factored Axis-Aligned Filtering for Rendering Multiple Distribution Effects. Soham U. Mehta, Ravi Ramamoorthi, Fredo Durand. In Proc. of Siggraph 2014.

This work was sponsored by NVIDIA fellowship.