EUROGRAPHICS 2016

THE 37th ANNUAL CONFERENCE OF THE EUROPEAN ASSOCIATION FOR COMPUTER GRAPHICS

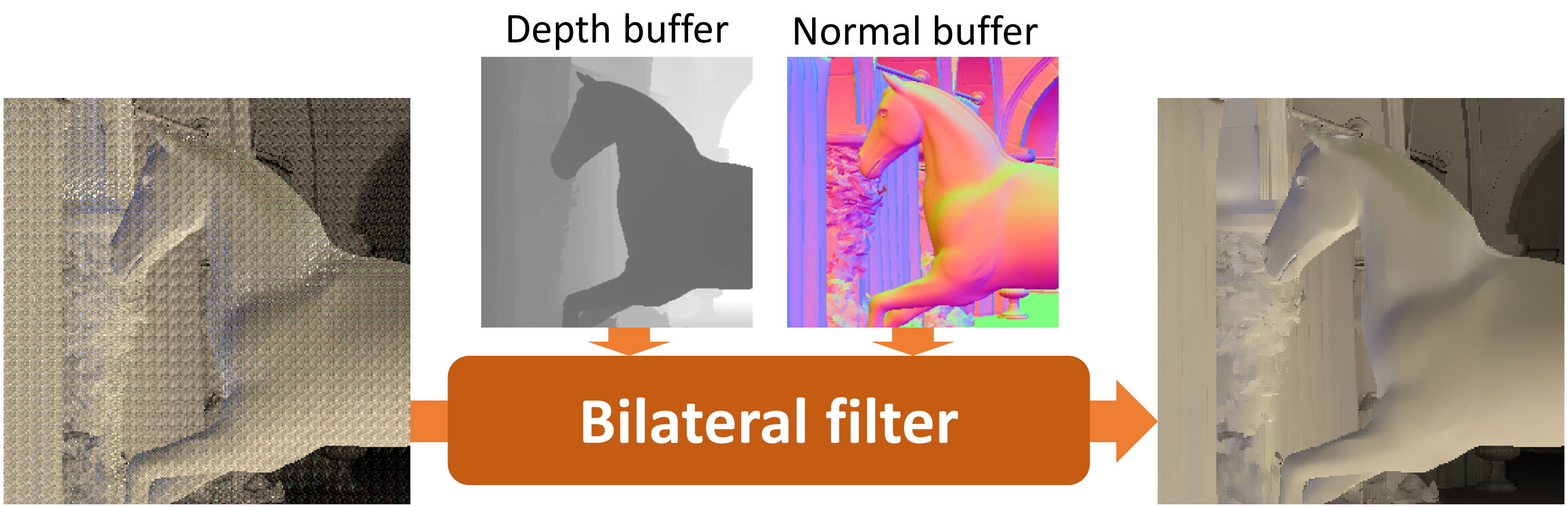
Specular Lobe-Aware Filtering and Upsampling for Interactive Indirect Illumination

Yusuke Tokuyoshi

Square Enix Co., Ltd.

Post-processing for Deferred Shading

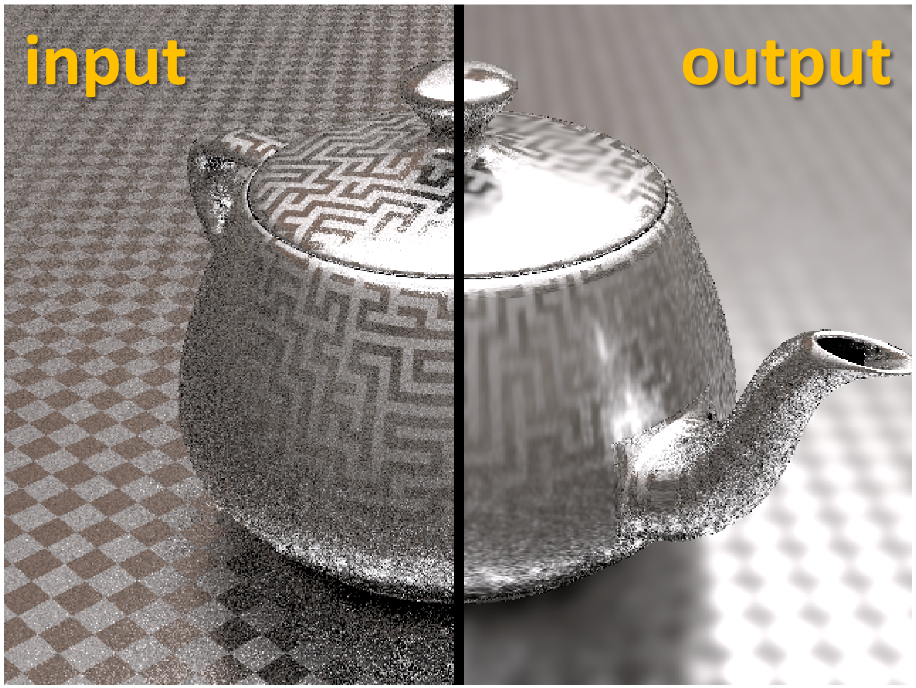
- Cross bilateral filter using depth & normal buffers
- Effective for diffuse surfaces



Noisy indirect illumination

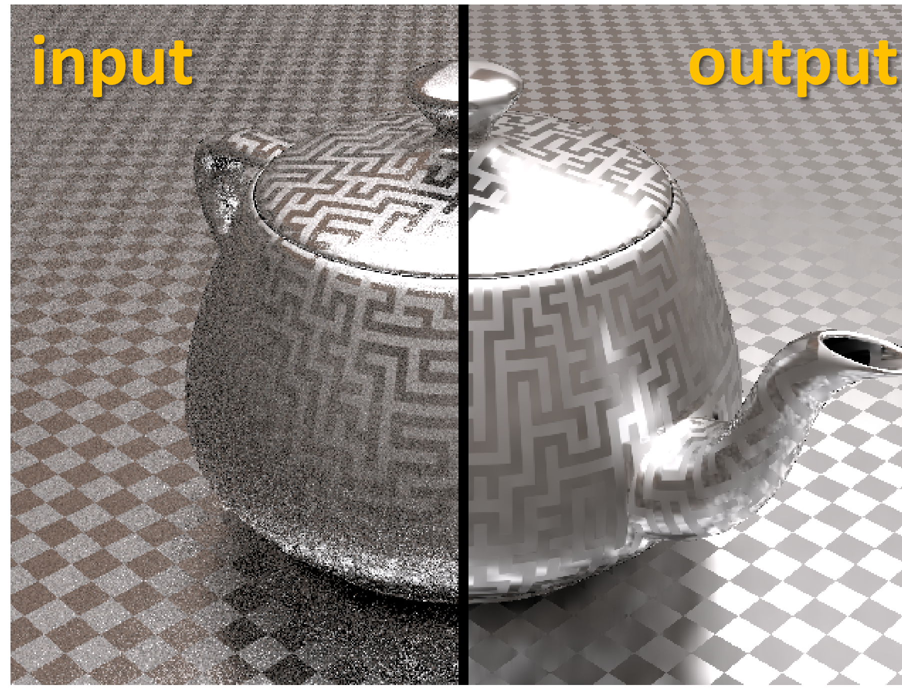
result

Specular Surfaces



Geometry-aware filtering

MSE: 0.1380



Our filtering

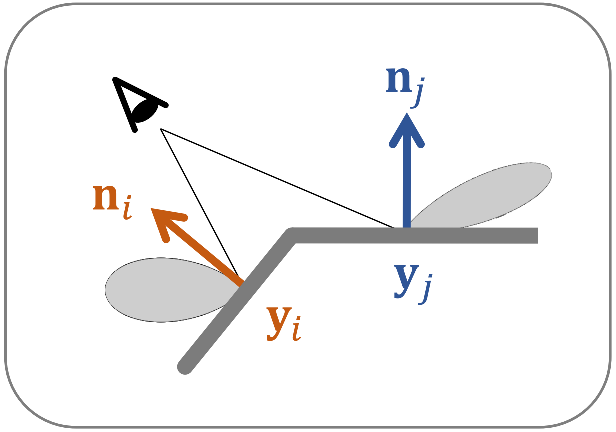
MSE: 0.0157

Similarity of pixels

Pixel values are represented using the rendering equation

$$I(i) = \epsilon_i + \int_{S^2} L(\mathbf{y}_i, \boldsymbol{\omega}) \boxed{\rho_i(\boldsymbol{\omega})} \max(\mathbf{n}_i \cdot \boldsymbol{\omega}, 0) \mathrm{d}\boldsymbol{\omega}$$

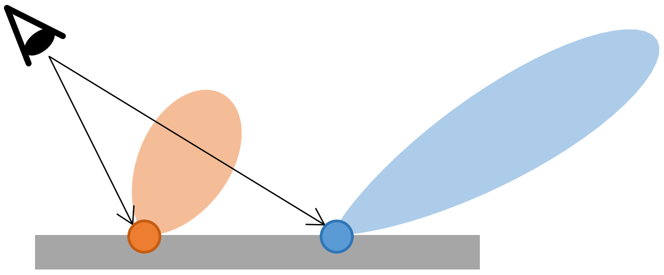
$$I(j) = \underbrace{\epsilon_j}_{\text{numerical error}} + \int_{S^2} L(\mathbf{y}_j, \boldsymbol{\omega}) \underbrace{\boxed{\rho_j(\boldsymbol{\omega})}}_{\text{BRDF}} \max(\mathbf{n}_j \cdot \boldsymbol{\omega}, 0) \mathrm{d}\boldsymbol{\omega}$$



Our contribution: similarity of reflection lobes

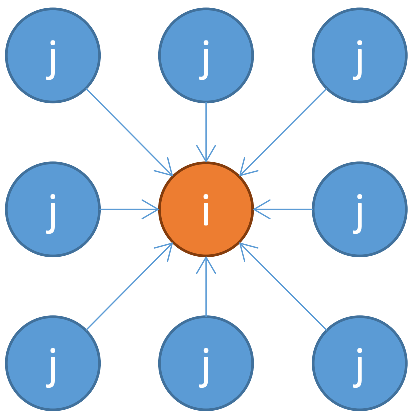
Similarity of Reflection Lobes

- Use shapes of reflection lobes instead of shading parameters
- More suitable to represent visual differences than using the difference of shading parameters
- Can measure the similarity of different BRDF models
- Inputs: G-buffer and camera position



Specular Lobe-Aware Filtering

Cross Bilateral Filtering



Weighted average

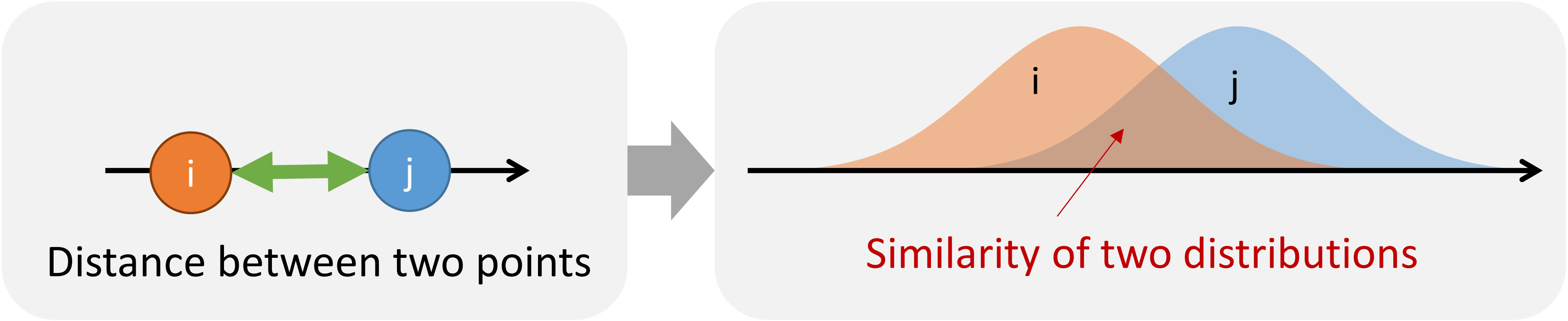
$$\hat{I}(i) = \frac{\sum_j w(i,j) I(j)}{\sum_j w(i,j)}$$

Similarity between pixels i and j

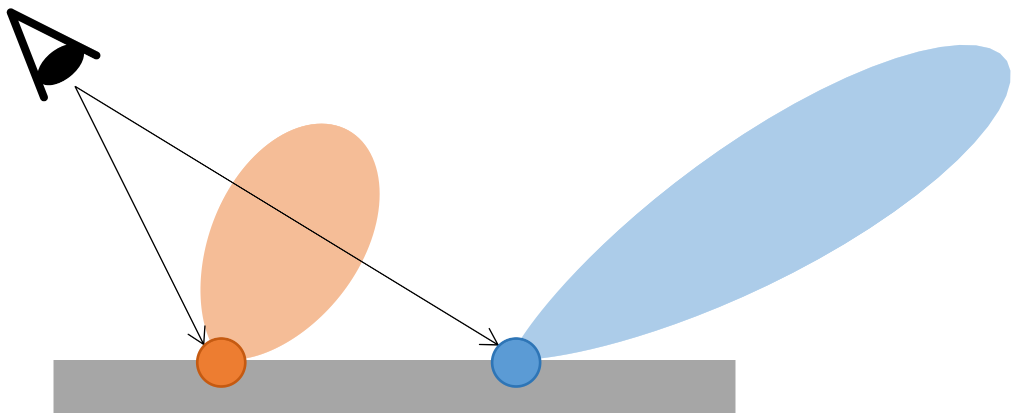
- Feature = pixel value of a guidance image
- Gaussian-based weighting function $w(i,j)$
- Use the Euclidean distance between two pixel values

Reformulation of Bilateral Filtering

- Feature = **distribution function** for each pixel
- Weighting function: **product integral** of two distribution functions
- Can use an arbitrary distribution function in a non-Euclidean space for each pixel



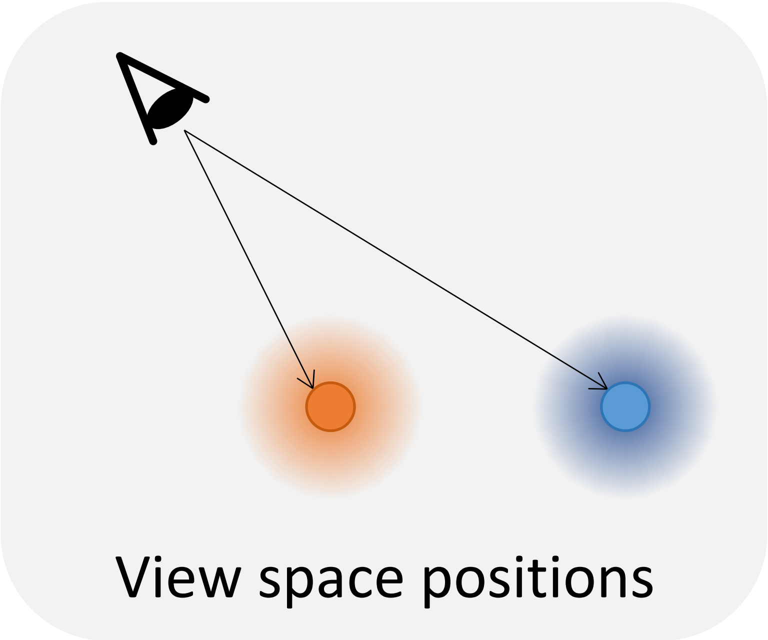
Reflection Lobe = Feature



$$I(i) = \epsilon_i + \int_{\mathbb{R}^3} \int_{S^2} L(\mathbf{y}, \boldsymbol{\omega}) \boxed{\rho_i(\boldsymbol{\omega}) \max(\mathbf{n}_i \cdot \boldsymbol{\omega}, 0) \delta(\mathbf{y} - \mathbf{y}_i)} \mathrm{d}\boldsymbol{\omega} \mathrm{d}\mathbf{y}$$

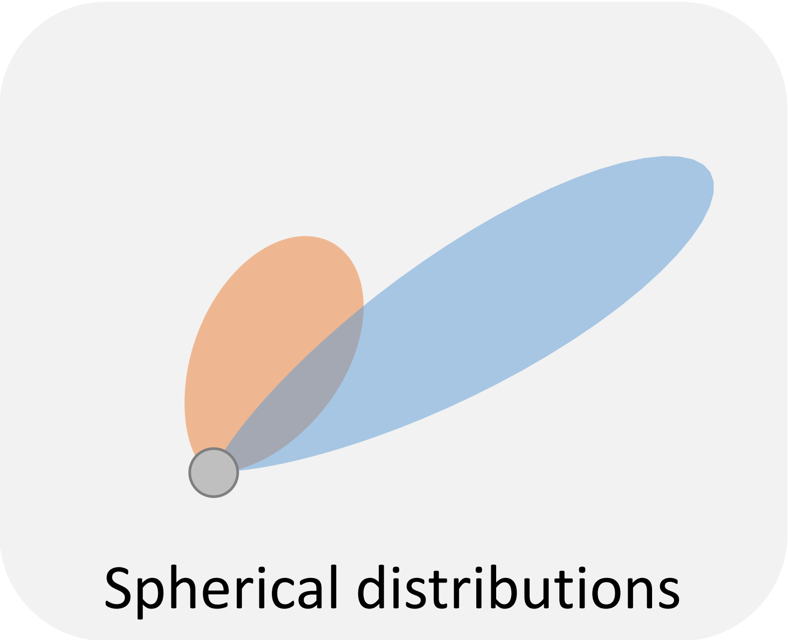
Feature of the pixel

Reflection Lobe Similarity



Gaussian function using
- Depth buffer
- Screen-space positions

×



Product integral of
two reflection lobes
(No analytical solution)

Spherical Gaussian Approximation

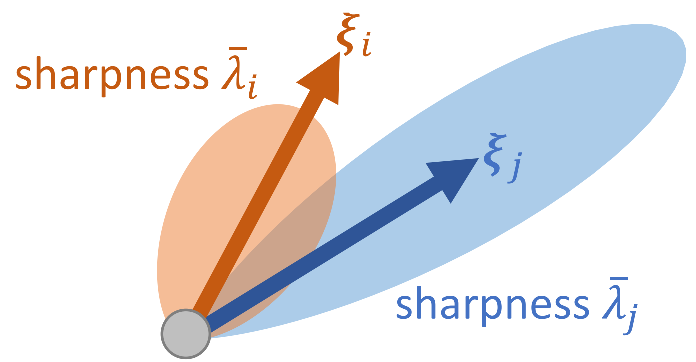
$$p_i(\boldsymbol{\omega})\max(\mathbf{n}_i, \boldsymbol{\omega}, 0) \approx \sum_{l}^{N_i} \mu_{i,l} G_{i,l}(\boldsymbol{\omega})$$

- Approximate specular lobes using SGs [Wang09]
  - Analytical approximation for parametric BRDFs
  - Fitting in preprocessing for measured BRDFs
- Product integral is closed in SG basis [Iwasaki12]
  - Complexity: $O\left(N_i N_j\right)$ ☹
  - Use a small number of SGs

Single Lobe Case

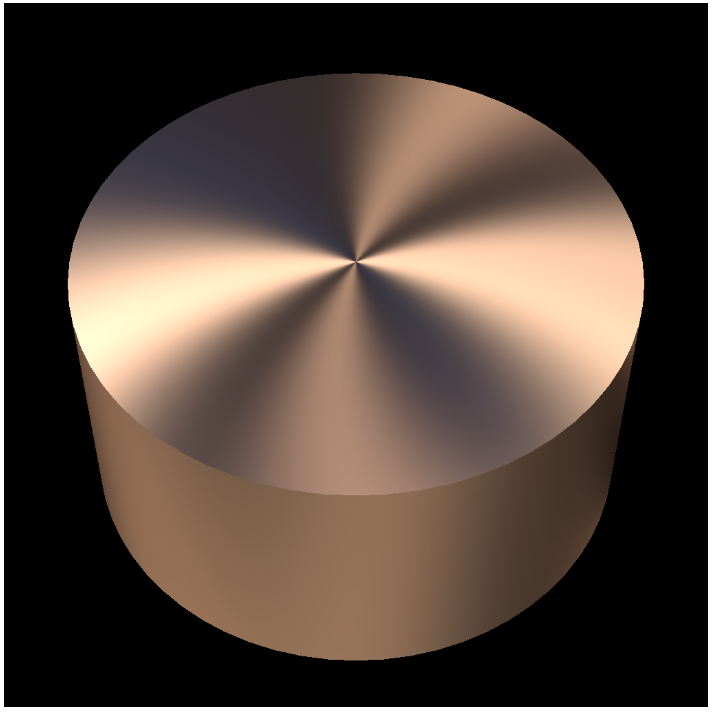
- Simple ☺
- Extended version of the existing geometry-aware filtering

$$w(i,j) \approx \underbrace{\left(\frac{2\sqrt{\bar{\lambda}_i \bar{\lambda}_j}}{\bar{\lambda}_i + \bar{\lambda}_j}\right)^{\beta}}_{\text{Similarity of lobe sharpness}} \underbrace{G\left(\xi_i, \xi_j, \frac{\beta \bar{\lambda}_i \bar{\lambda}_j}{\bar{\lambda}_i + \bar{\lambda}_j}\right)}_{\text{Similarity of lobe axes}}$$



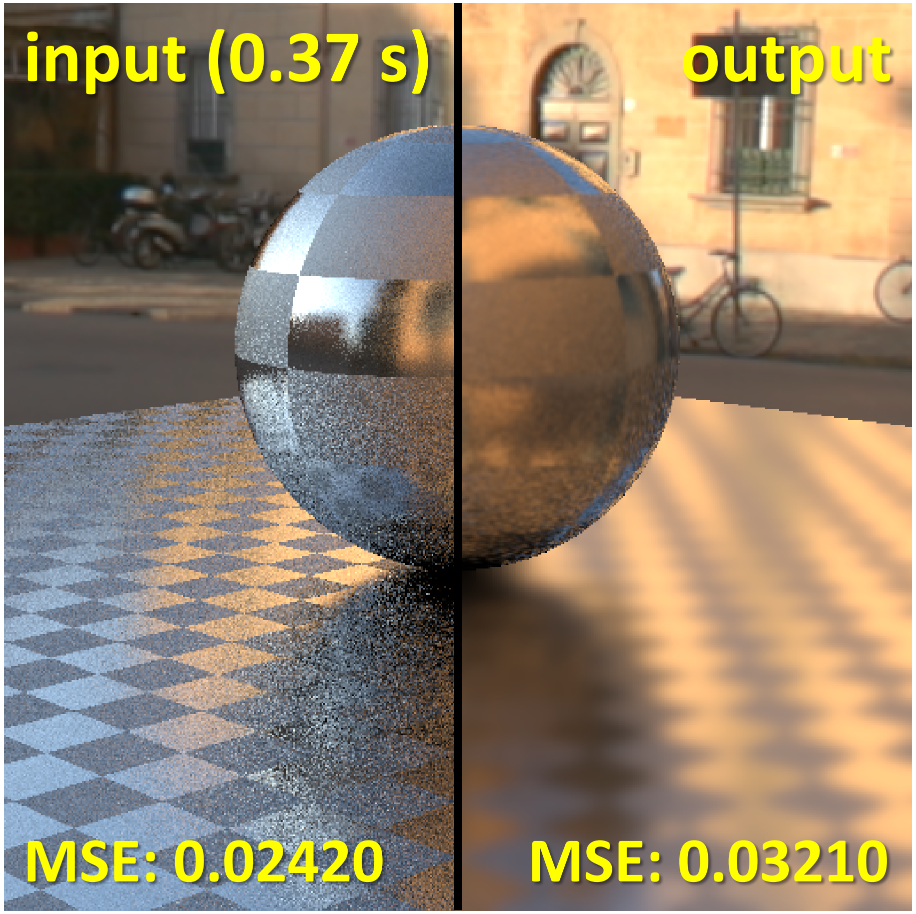
Anisotropic BRDFs

- Use anisotropic spherical Gaussians [Xu13]

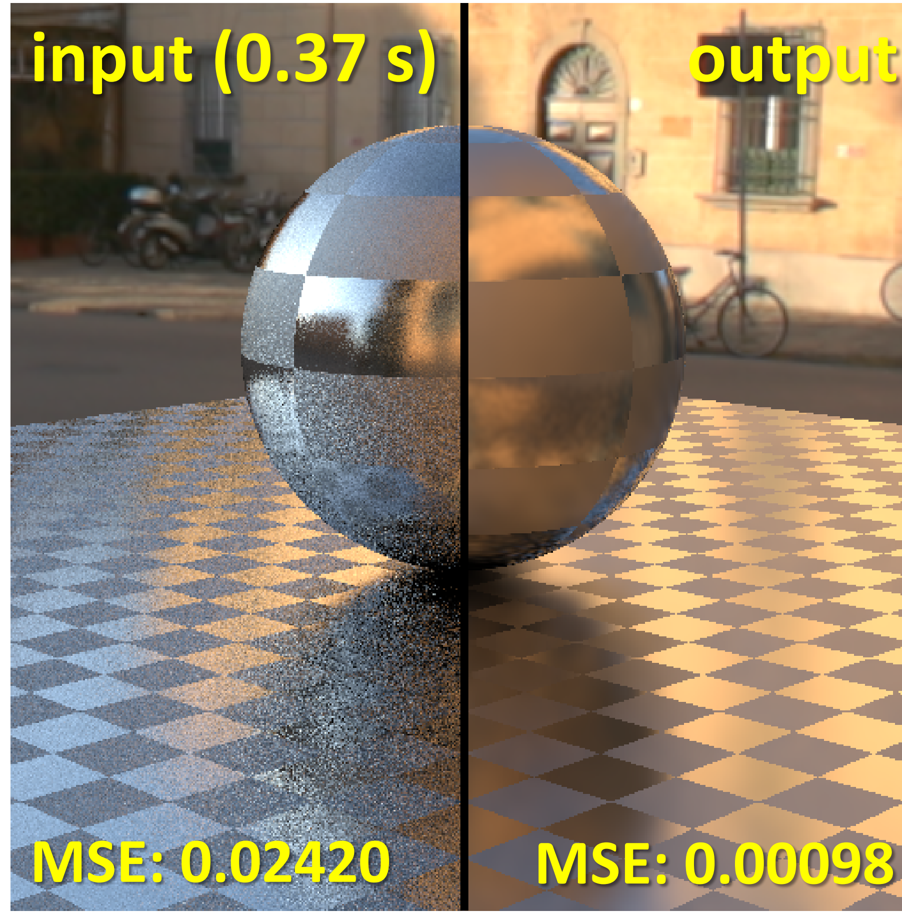


Experimental Results

Denoising for Path Tracing (1)



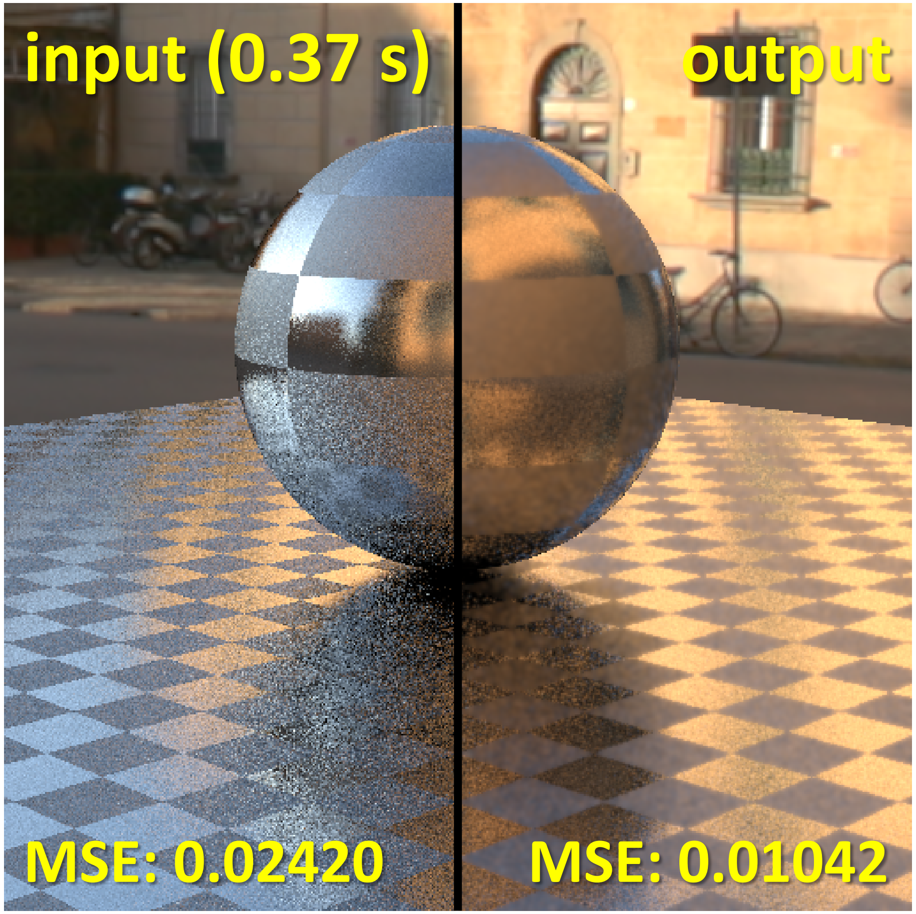
Normal-aware filtering



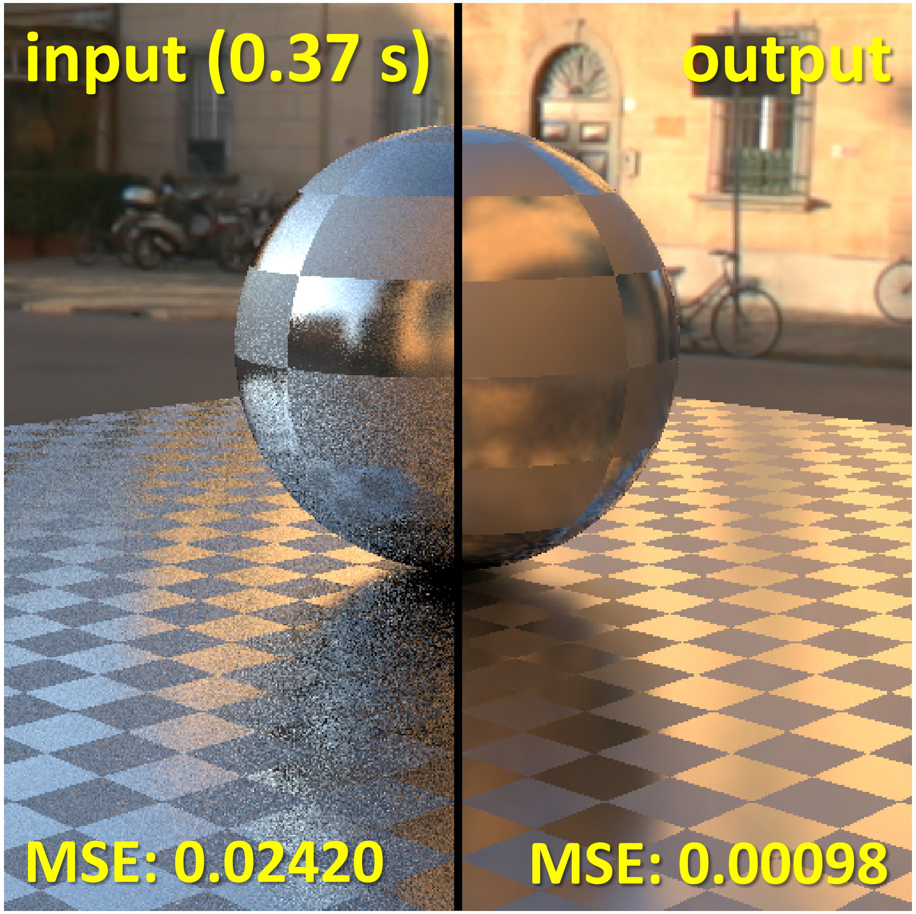
Specular lobe-aware filtering

Resolution: 512×512, CPU: Intel Xeon W5590

Denoising for Path Tracing (2)



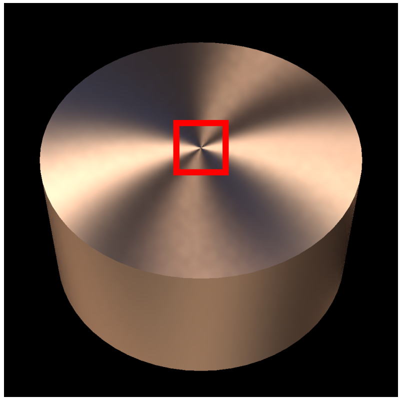
Normal-aware filtering
with screen-space bandlimit
according to the BRDF parameter

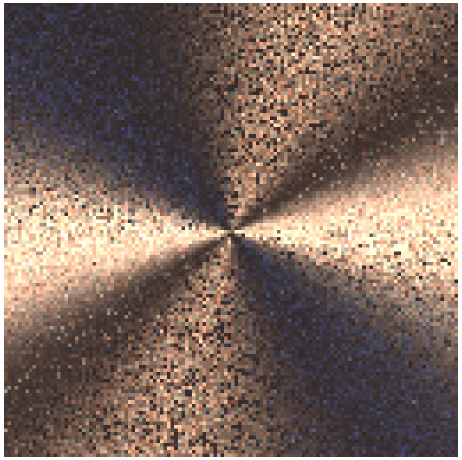


Specular lobe-aware filtering

Resolution: 512×512, CPU: Intel Xeon W5590

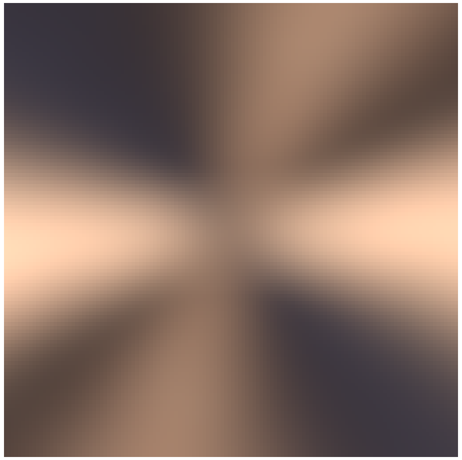
Anisotropic BRDFs





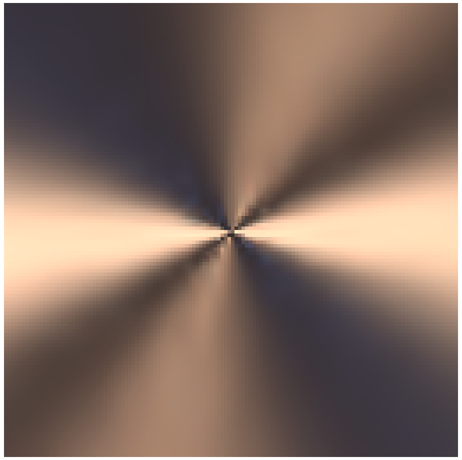
Noisy input

MSE: 0.08286



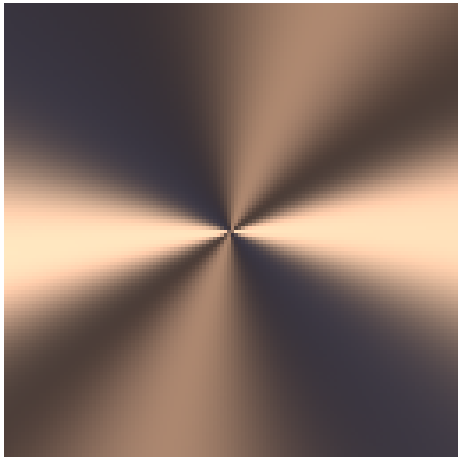
Normal-aware filtering

MSE: 0.00128



Lobe-aware filtering

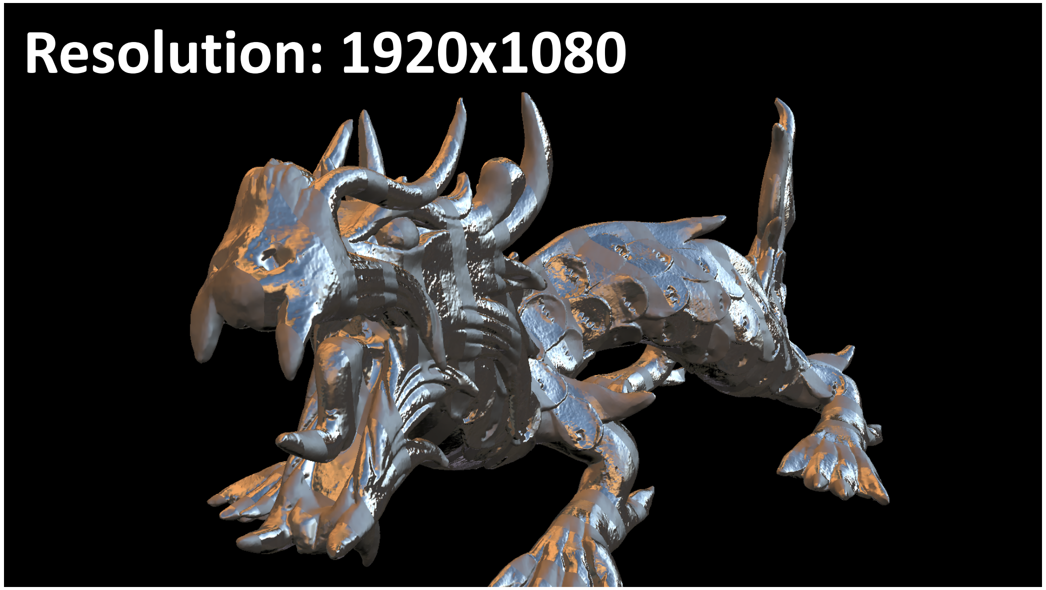
MSE: 0.00027



Reference

Resolution: 1024×1024, GPU: AMD Radeon™ HD 6990

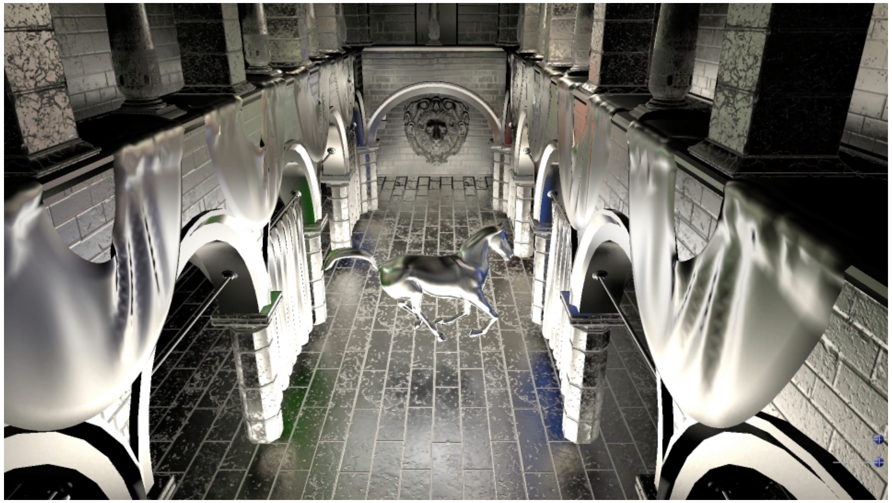
Denoising Performance



kernel radius	Normal-aware filtering	Ours (1 SG)	Ours (3 SGs)	Ours (1 ASG)
8 pixels	2.43 ms	3.29 ms	15.2 ms	17.7 ms
16 pixels	4.56 ms	6.03 ms	27.6 ms	30.8 ms

GPU: AMD Radeon™ HD 6990

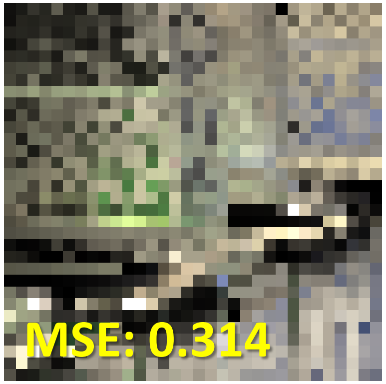
Spatio-Temporal Upsampling [Herzog10]



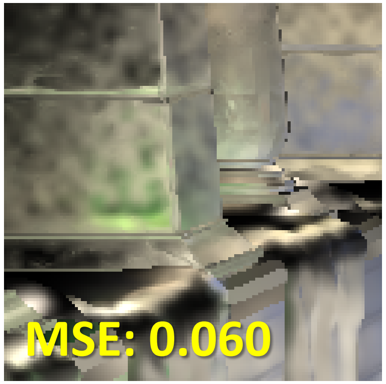
Voxel cone tracing [Crassin12]

480×270

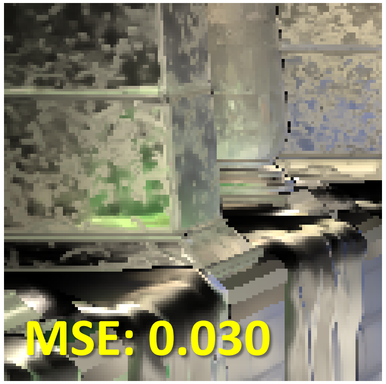
1920×1080



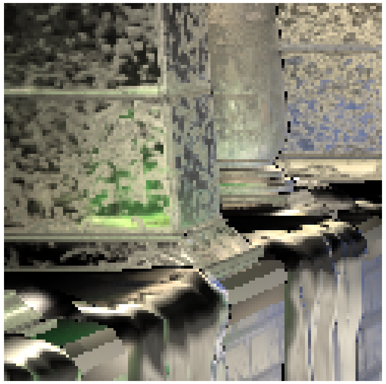
w/o upsampling
(1.6 ms)



Normal-aware
(1.9 ms)



Ours
(2.1 ms)

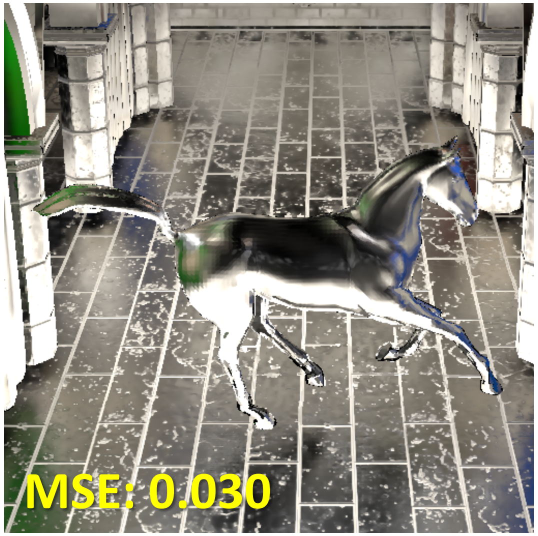


Reference
(18.1 ms)

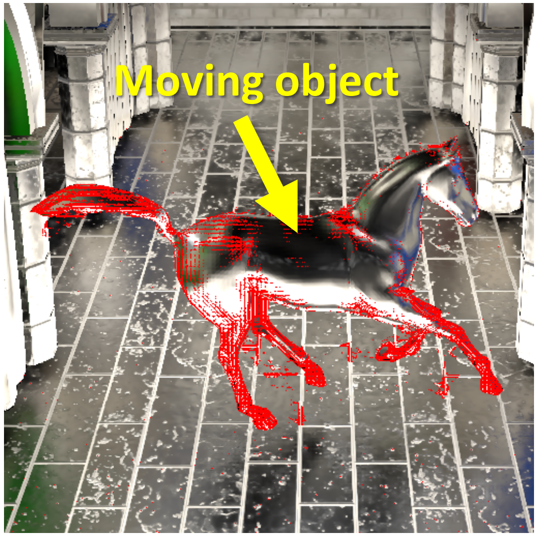
Total rendering time: 32.1 ms, GPU: AMD Radeon^{TM} HD 6990

Adaptive Sampling

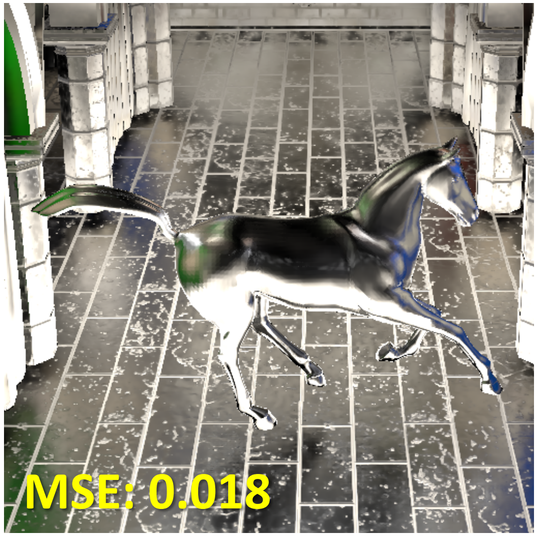
- Undersampling error on discontinuous regions
- Solution: re-computation for low-weight pixels



Upsampled image



Low-weight pixels (red)

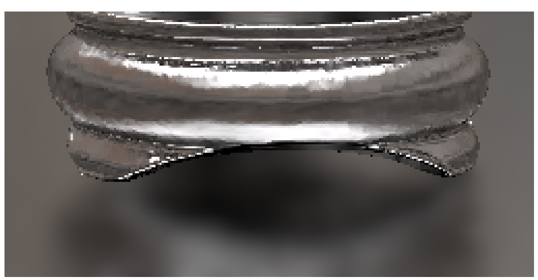


Recompute pixel values (3.6 ms)

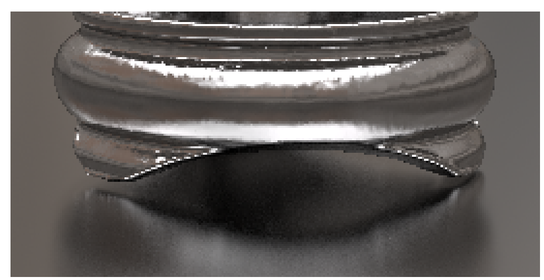
GPU: AMD Radeon™ HD 6990

Limitations (1)

- Undersampling on discontinuous regions
  - Common limitation of bilateral filtering-based methods
  - Solved by temporal filtering & adaptive sampling
- Neglect high-frequency changes of incident radiance
  - Similar to existing geometry-aware filtering
  - Possible solution: use secondary-ray information [Metha13]



Our filtering

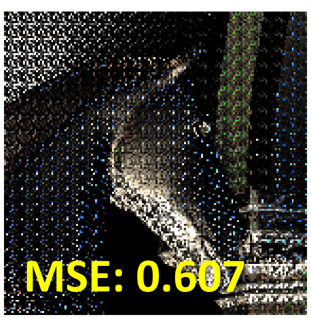


Reference

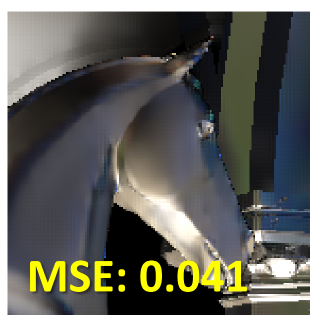
Limitations (2)

- Our method assumes high-frequency BRDFs produce high-frequency illumination appearance
  - Higher-frequency filtering bandwidth for sharper reflection lobes
- Importance sampling according to BRDFs should be used

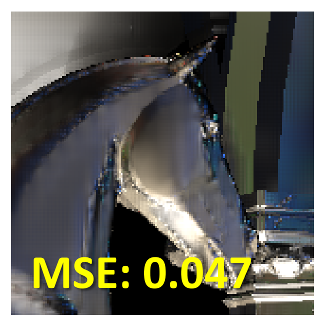
Results w/o importance sampling according to BRDFs (instant radiosity [Kellar97])



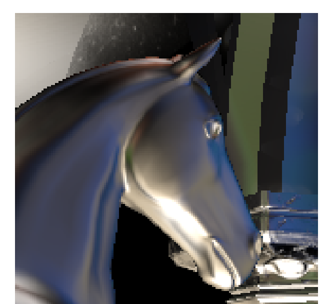
Noisy input



Normal-aware



Ours


Reference

Conclusions

- Specular lobe similarity for cross bilateral filtering
  - Product integral of two reflection lobes
  - Can detect the difference of BRDFs
- Simple implementation
  - Analytical solution of the weighting function
  - Only a G-buffer and camera position are used

Thank you for your attention