CT5202: Photorealistic Rendering

Adaptive Sampling and Reconstruction

Lecturer: Bochang Moon

Background

- Monte Carlo estimator
  - $\hat{\theta} = \frac{1}{\mathrm{N}} \Sigma_{\mathrm{i}} \mathrm{f}(\mathrm{X}_{\mathrm{i}})$
  - $var(\hat{\theta}) = \frac{\sigma^2}{\mathrm{N}}$
- Homogeneous error
  - $\mathrm{var}\big(\mathrm{f}(\mathrm{X}_{\mathrm{i}})\big) = \sigma^2$
  - Uniform sampling rate would make sense for the homogeneous error
- Heterogeneous error
  - Variance can vary across local regions
  - Light transport function is typically nonlinear and variance varies across regions

Classical Adaptive Sampling

- Image-space adaptive sampling using per-pixel variance
  - Per-pixel estimator
  - Imagine you have an unknown image-space function per pixel
  - $\widehat{\theta_p} = \frac{1}{\mathrm{N_p}} \Sigma_\mathrm{i} \mathrm{f_p}(\mathrm{X_i})$
  - $var\left(\widehat{\theta_p}\right) = \frac{\sigma_p^2}{\mathrm{N_p}}$
  - Assumption:
    - Random samples $\mathrm{f_p}(\mathrm{X_i})$ are iid and share a variance σ_p^2
    - Note that σ_p^2 can vary per pixel p
  - A simple approach
    - Control the number of samples per pixel $\mathrm{N_p}$ to be proportional to σ_p^2
    - High-error regions have high sample counts
  - How do we estimate σ_p^2?
    - $s_p^2 = \frac{1}{N_p - 1} \Sigma_i \left(f_p(X_i) - \overline{f_p}(X)\right)^2$

Classical Adaptive Sampling

- Two-step approach using the sample variance
  - Allocate a small number of samples (equal sample count) per pixel
  - Estimate the sample variance and decide the number of samples per pixel for the next stage
  - Repeat until we use a target sample count

- Potential problem of the simple approach
  - Per-pixel sample variances can be typically noisy unless a large number of samples are used.
  - Sample counts can be noisy and it typically leads to a noisy rendered image
  - A simple trick
    - Apply an image filter to the sample variance and use the smoothed variance

Multi-dimensional Adaptive Sampling

- The classical approach only controls the number of samples in image space
  - Sampling density for the other dimensions (lens, time, …) is still uniform.
- Multi-dimensional sampling
  - The random sample is in a high-dimensional space
  - Maintain a high-dimensional structure (kd-trees) that stores the samples
  - This idea was proposed initially in [Kajiya 86], but it was fully accomplished in [Hachisuka et al. 08]

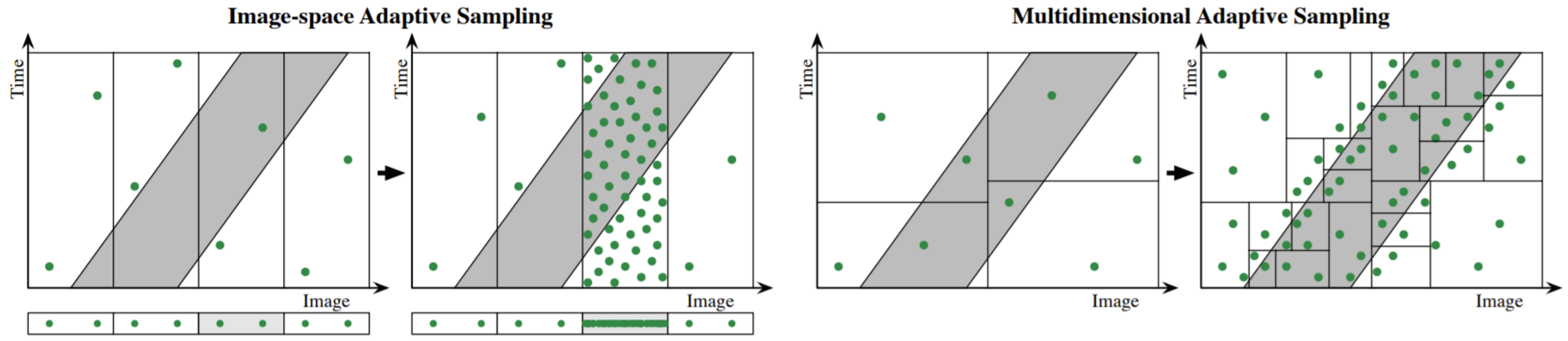


Image from [Hachisuka et al. 08]

Multi-dimensional Adaptive Sampling

- Iteratively allocate more samples to the kd-tree leafs whose variances are high
- Kd-tree leafs are subdivided if its number of samples are higher than a user-defined threshold
- The final image is reconstructed by projecting the sub-regions to the image plane (sub-regions are assumed to be constant)

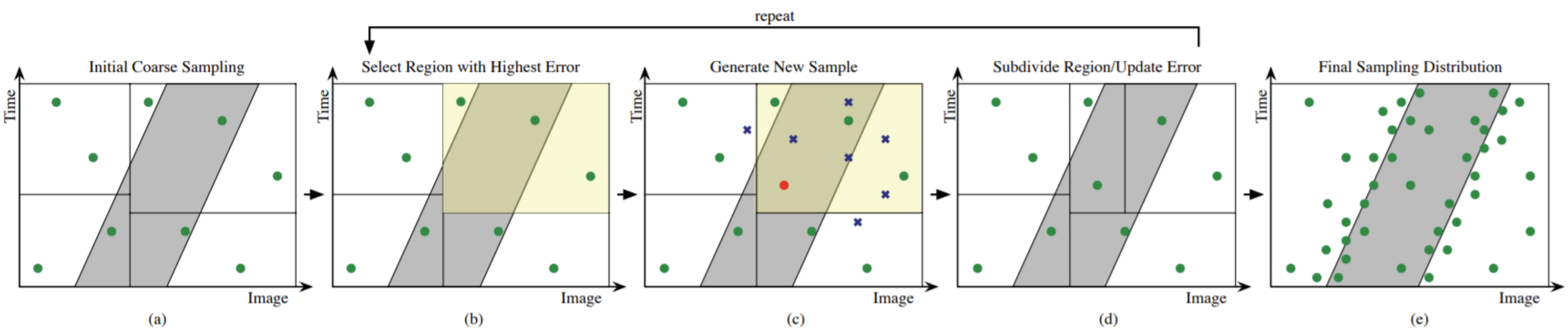
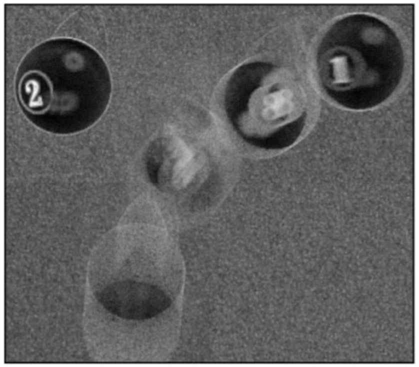


Image from [Hachisuka et al. 08]

Multi-dimensional Adaptive Sampling



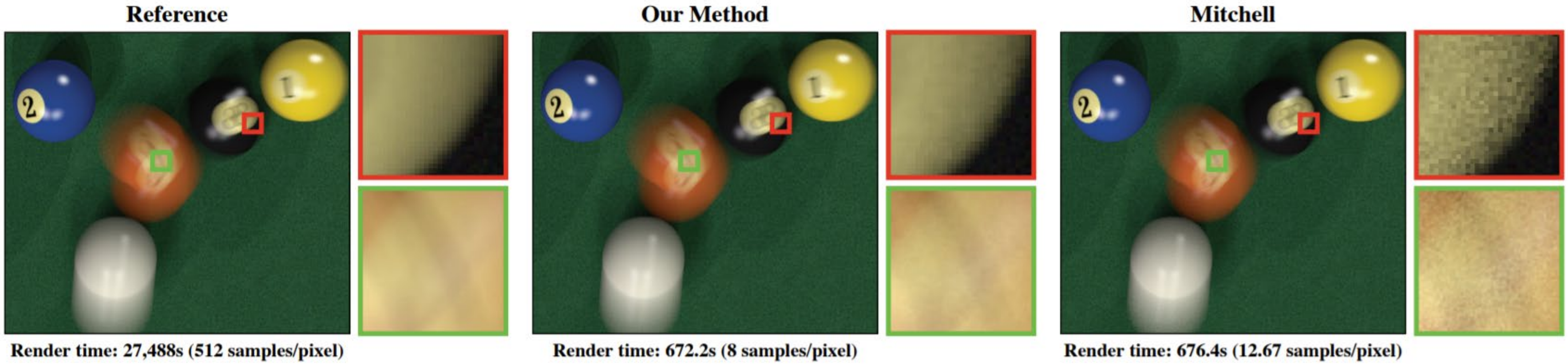


Image from [Hachisuka et al. 08]

Multi-dimensional Adaptive Sampling

- Pros.
  - Very effective for a moderate dimension (direct illumination, motion blur, and depth-of-fields)
- Cons.
  - Sample density becomes very sparse as increasing the dimensionality of samples (e.g., for global illumination)

Adaptive Sampling using Wavelet Space

- Adaptive Wavelet Rendering [Overbeck et al. 09]
- Properties
  - Image-space adaptive sampling
  - More robust than the classical approach
- Wavelet thresholding using variances

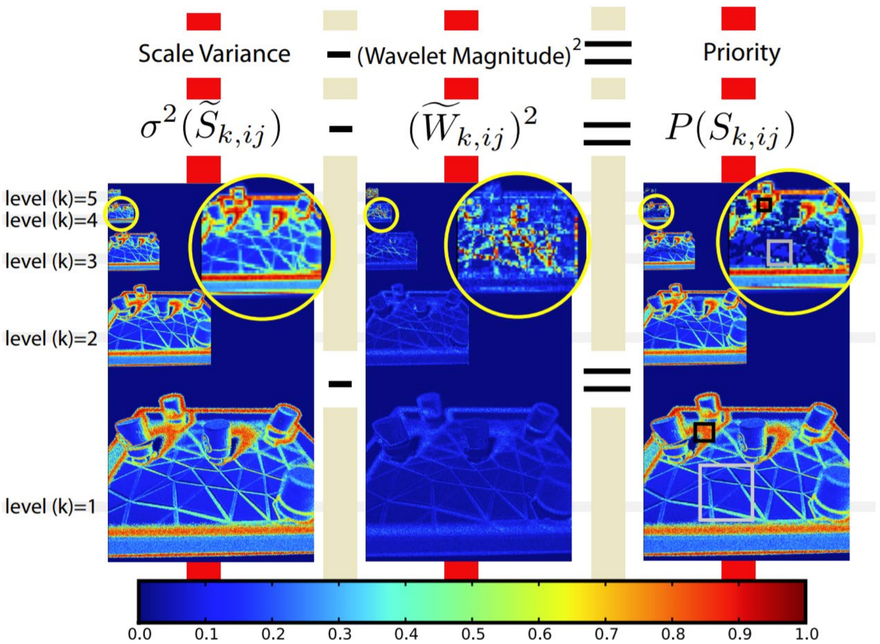


Image from [Overbeck et al. 09]

Adaptive Sampling using Wavelet Space

- Pros.
  - Works well for depth-of-field or motion blur
- Cons.
  - Produces ringing artifacts when GI simulates

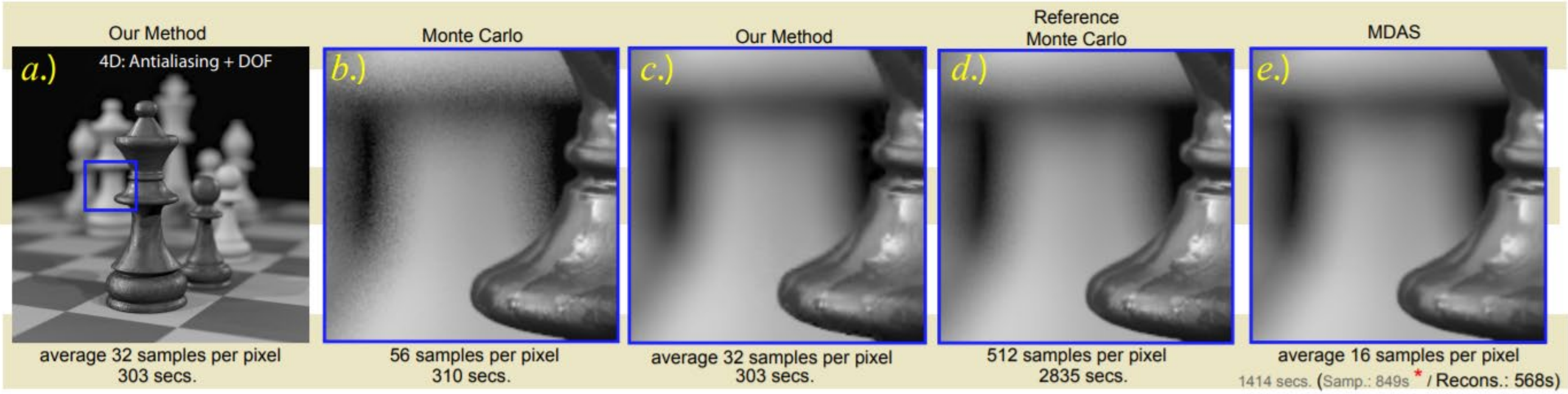


Image from [Overbeck et al. 09]

Recent Image Space Adaptive Rendering

- Controlling sample density using sample variances is not optimal
- Suppose a typical combination:
  - Path tracing + image denoising
  - The variance of the final image (reconstruction variance) can be very different from the sample variance
  - Adaptive sampling should allocate more samples to the regions with high reconstruction errors

Recent Image Space Adaptive Rendering

- Adaptive Sampling and Reconstruction using Greedy Error Minimization
  - [Rousselle et al. 11]
- Control two parameters
  - Sampling rate per pixel
  - Reconstruction parameter per pixel
- Target combination
  - Path tracing + Gaussian filter

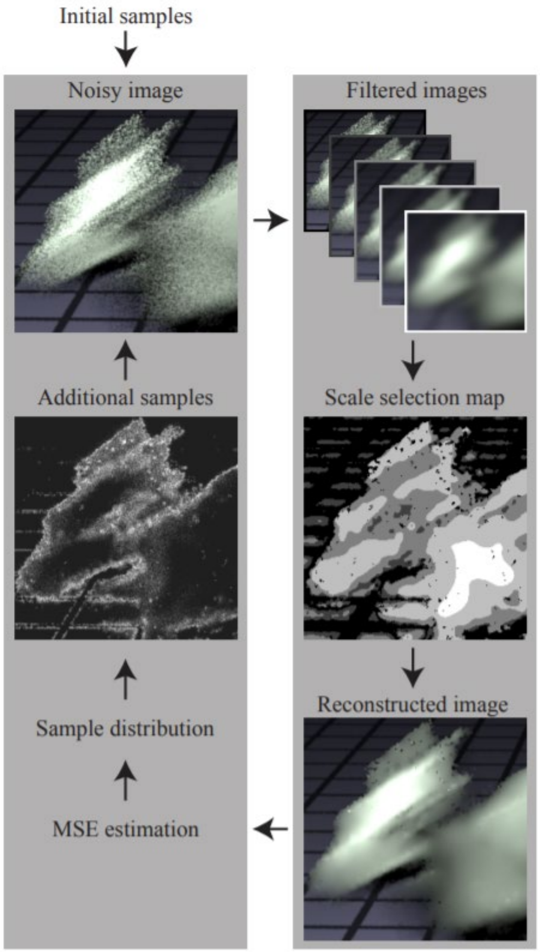


Image from [Rousselle et al. 11]

Recent Image Space Adaptive Rendering

- Adaptive Sampling and Reconstruction using Greedy Error Minimization
  - [Rousselle et al. 11]

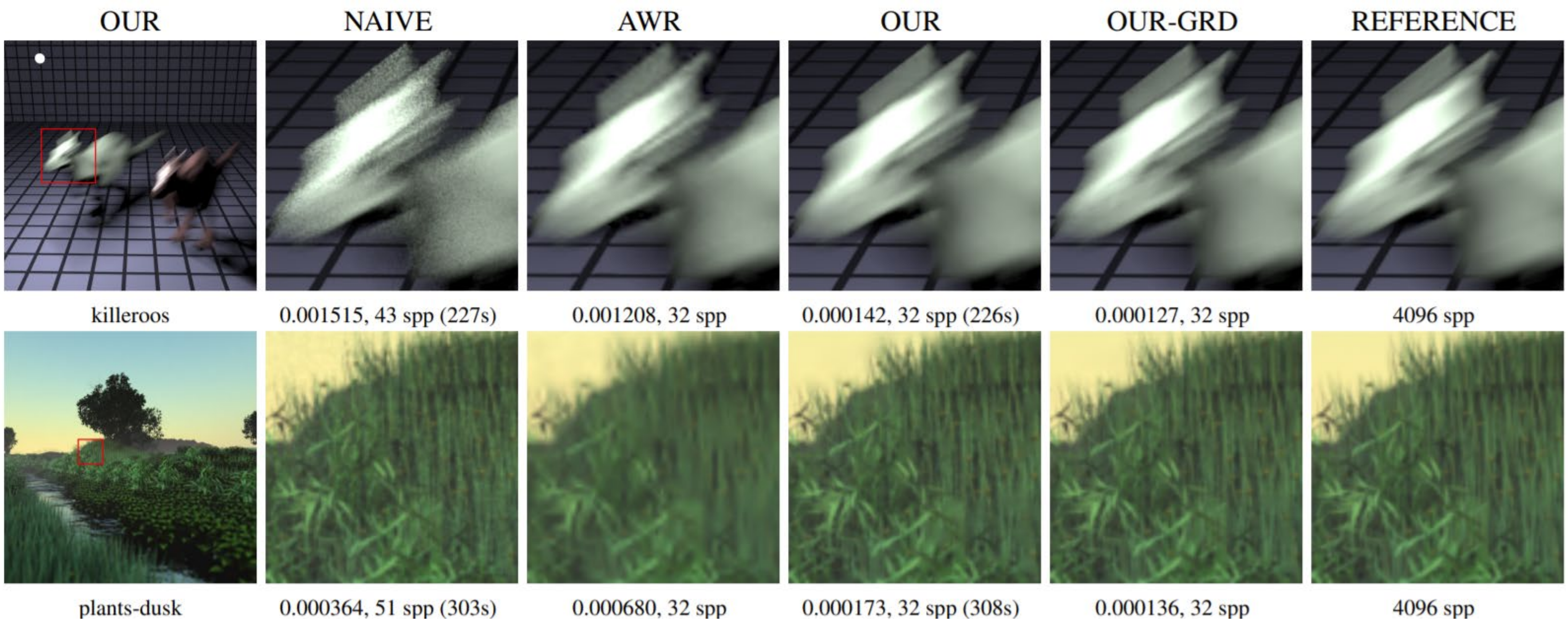


Image from [Rousselle et al. 11]

Recent Image Space Adaptive Rendering

- Adaptive Sampling and Reconstruction using Greedy Error Minimization
  - [Rousselle et al. 11]
- Pros.
  - Very simple and fast
  - Work reasonably well
- Cons.
  - Isotropic filter (Gaussian) is not effective for non-linear functions (image edges)

Recent Image Space Adaptive Rendering

- Nonlinear reconstructions
  - Adaptive Rendering with Non-Local Means Filtering [Rousselle et al. 12]
    - Non-local means
  - On Filtering the Noise from the Random Parameters in Monte Carlo Rendering [Sen and Darabi 11]
    - Cross bilateral filtering
  - Adaptive Rendering based on Weighted Local Regression [Moon et al. 14]
    - Local regression
  - Nonlinearly Weighted First-order Regression for Denoising Monte Carlo Renderings [Bitterli et al. 16]
    - Local regression + non-local means

Recent Image Space Adaptive Rendering

- General procedure
  - Allocate sparse samples initially
  - Reconstruct the image using a non-linear filtering
    - Estimate optimal parameters for the reconstruction
    - Estimate MSE (bias^2 + variance) of the image
  - Allocate more samples to the image regions with high MSEs

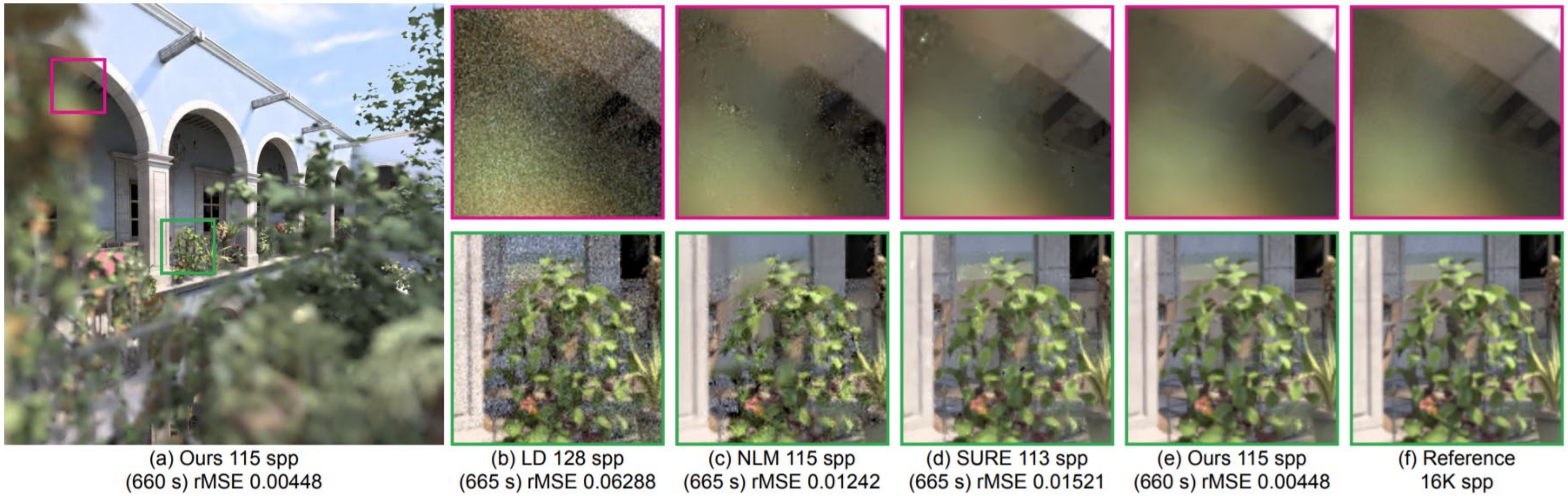


Image from [Moon et al. 14]

Recent Image Space Adaptive Rendering

- Reconstruction performance depends on:
  - Nonlinearity of the chosen filter
  - Robust estimation of errors
  - Rendering-specific features to identify high-frequency image information (edges)
- Adaptive sampling is closely related to a chosen reconstruction method

More Recent Image Space Adaptive Rendering

- So far:
  - Reconstruction filters are hand-crafted ones
  - Estimating MSE is often a tedious task (and also quite difficult to be robust)

- Deep learning based approaches
  - Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings [Bako et al. 17]
  - A general weighting can be represented as
    - $\hat{y}_c = \Sigma_i w_i y_i$
    - $w_i > 0$

  - Key advantage is that we don't need to assume a specific functional form for w_i

More Recent Image Space Adaptive Rendering

- Deep learning based approaches
  - Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings [Bako et al. 17]
  - A general weighting can be represented as
    - $\hat{y}_c = \Sigma_i w_i y_i$
    - $w_i > 0$

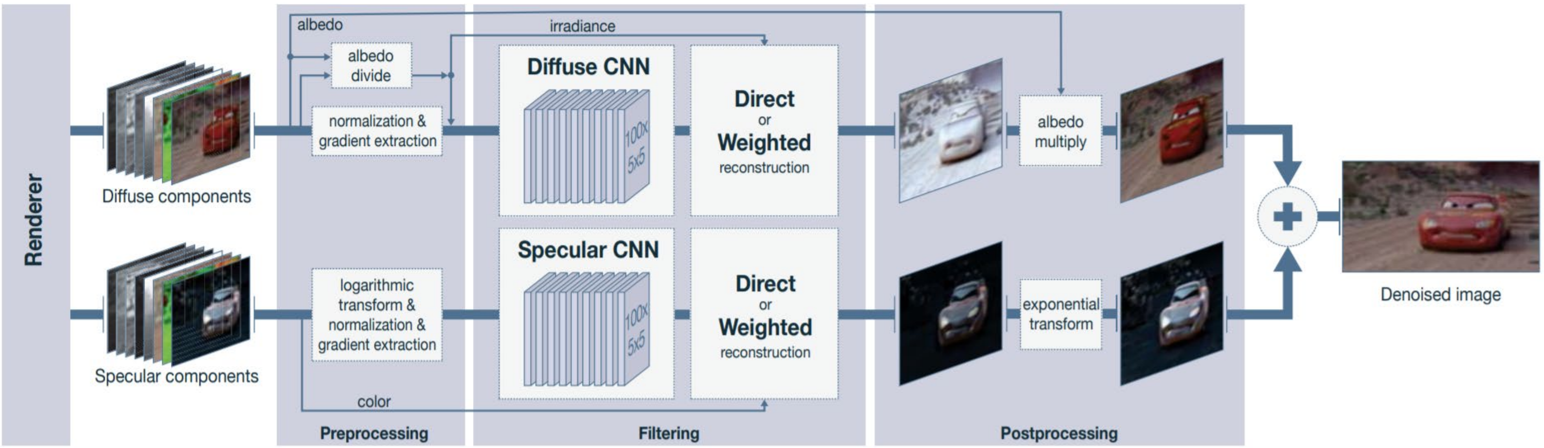


Image from [Bako et al. 17]

More Recent Image Space Adaptive Rendering

- Deep learning based approaches
  - Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings [Bako et al. 17]

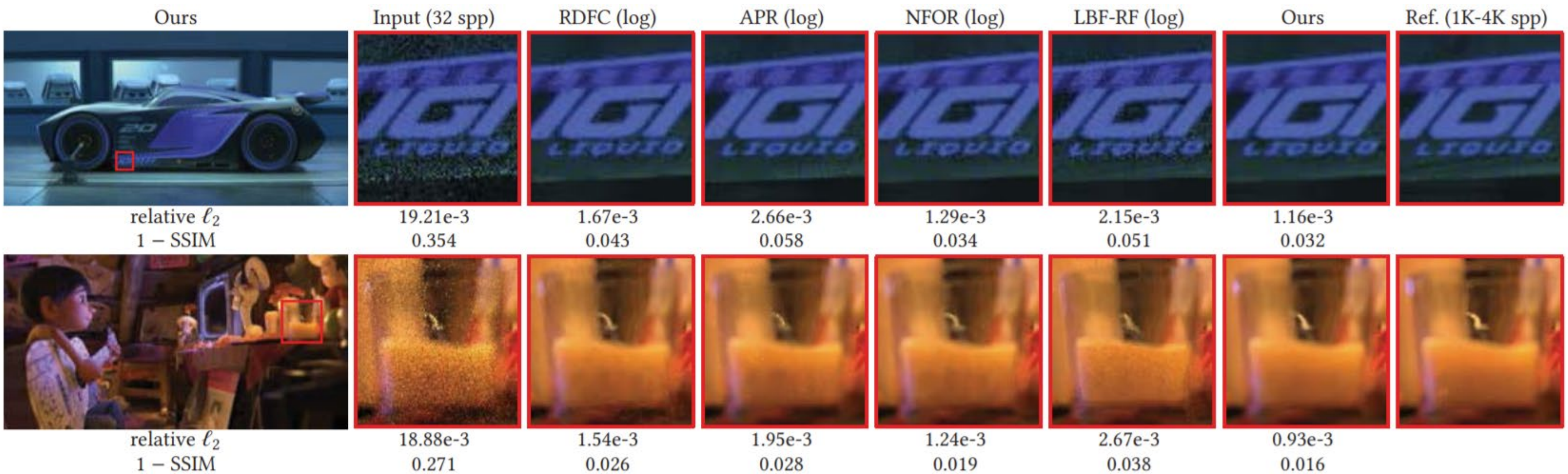


Image from [Bako et al. 17]

More Recent Image Space Adaptive Rendering

- Deep learning based approaches
  - Sample-based Monte Carlo Denoising using a Kernel-Splatting Network [Gharbi et al. 19]
  - A general weighting in a sample space can be represented as
    - $\hat{y}_{c,j} = \Sigma_i \Sigma_j w_{i,j} y_{i,j}$

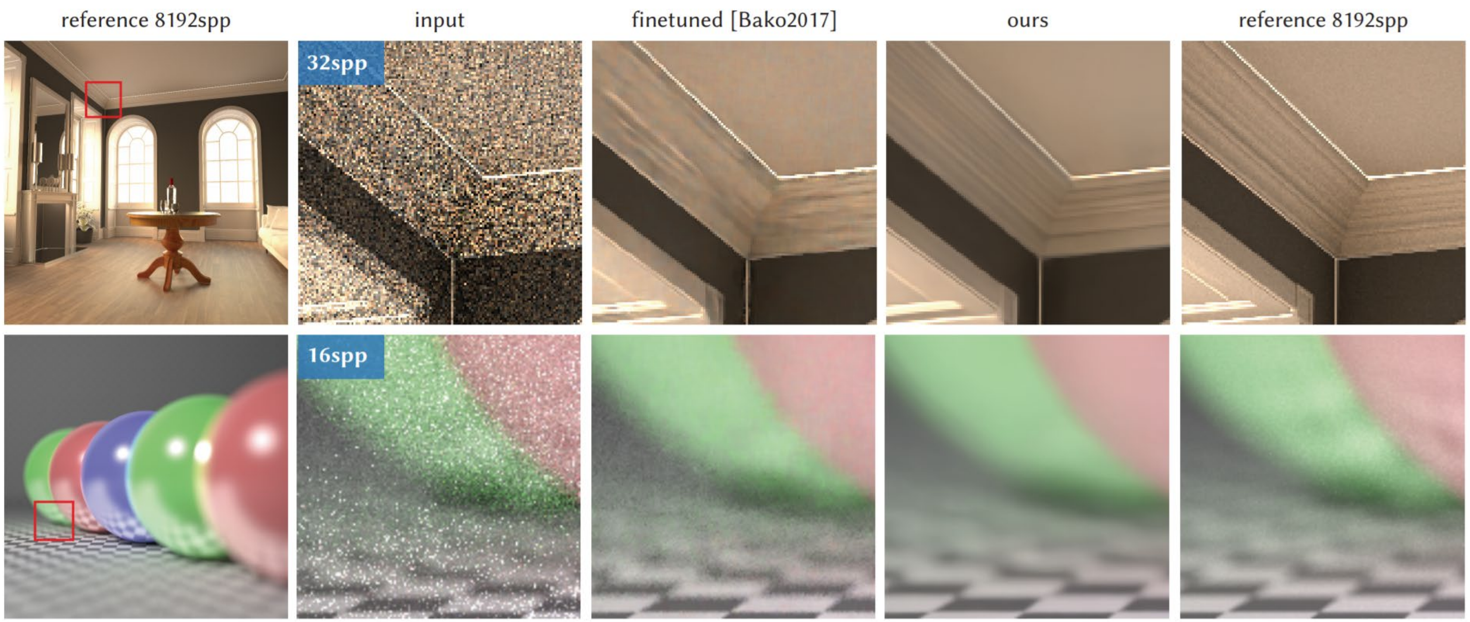


Image from [Gharbi et al. 19]

More Recent Image Space Adaptive Rendering

- What would be the unresolved issues for the adaptive rendering?