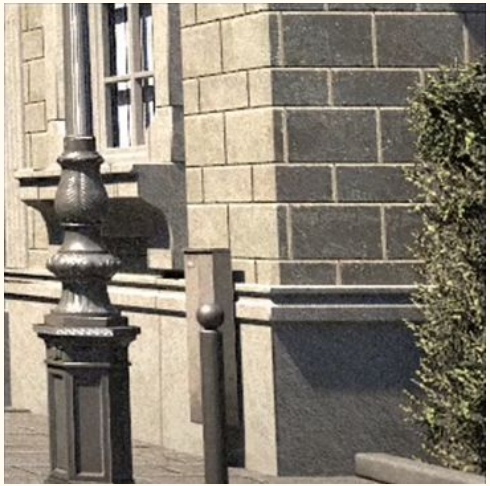
Improving Progressive Denoiser

T7 : Jaehyeon Lee, Chanwoo Cho

Progressive Denoising

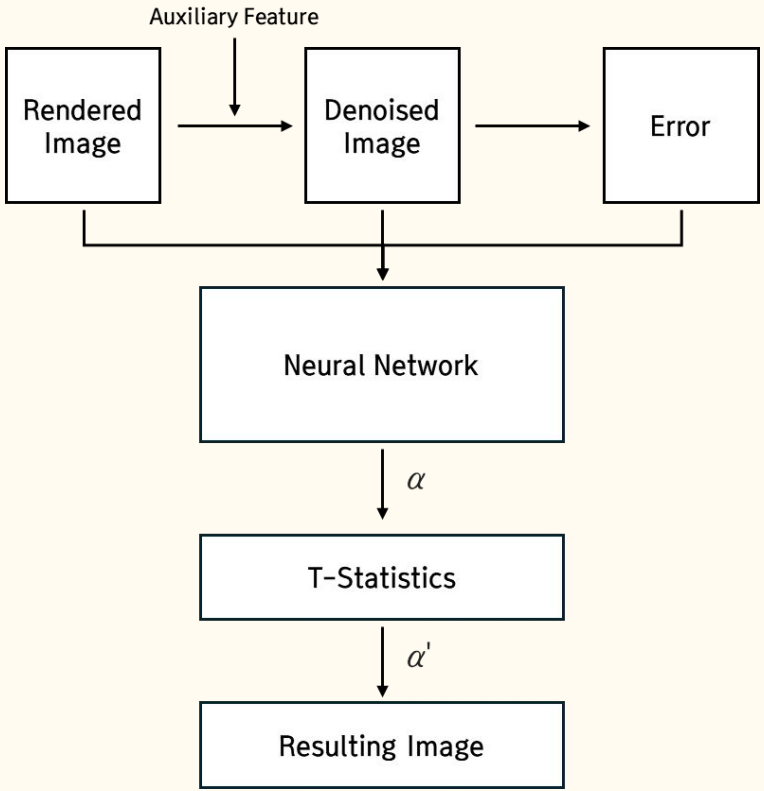
1. Loss of detail
2. Bad performance in high spp

$(1-\alpha)$  $+ \quad \alpha$  $=$ Good Image!

Rendered Image

Denoised Image

Progressive Denoising



1. Generate denoised image from rendered image.

2. Calculate error using **SURE** from denoised image.

$$\mathrm{SURE}(F,x) = \frac{1}{d}\left(\|F(x)-x\|^2 + 2\mathrm{tr}(J_F(x)\cdot\Sigma) - \mathrm{tr}(\Sigma)\right)$$

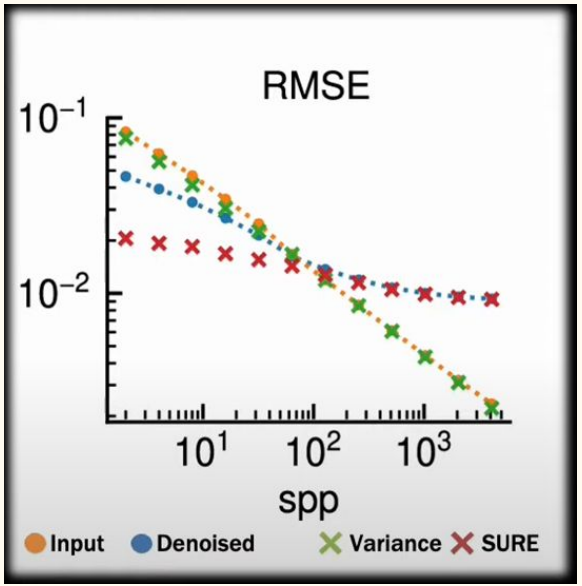
3. Receive α as output from NN.

4. Rescale α with t-statistics.

5. Generate resulting mixed image.

Limitation

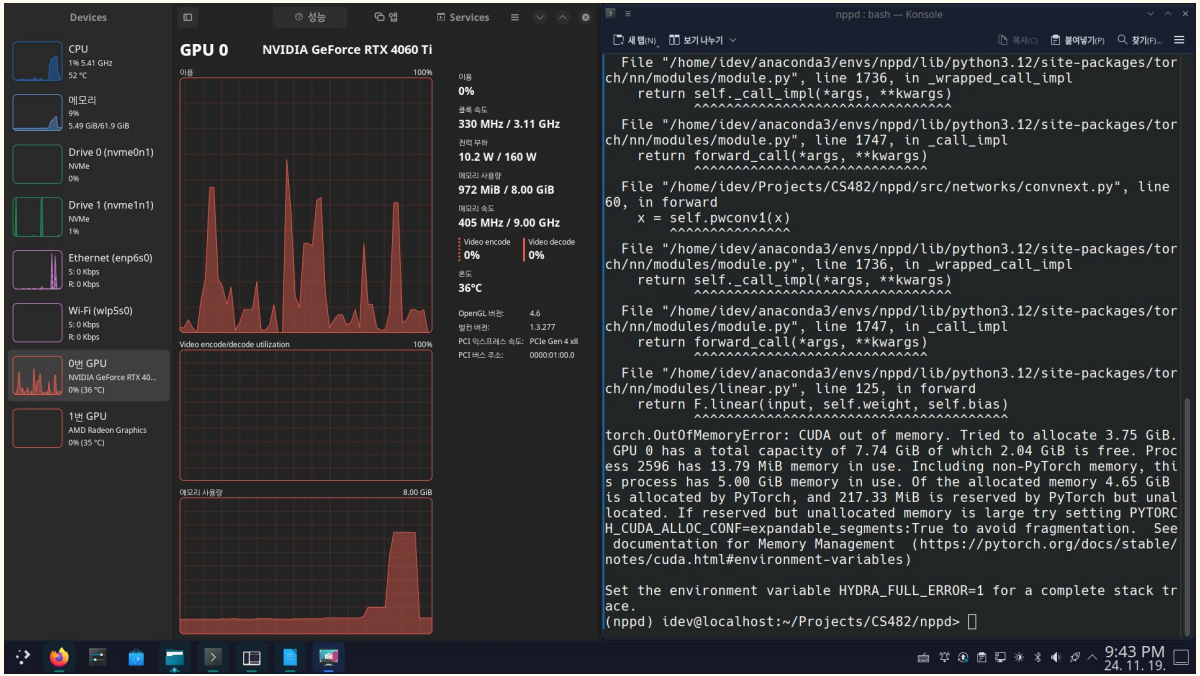
Limitation of method at very low spp. 2 spp in this example.



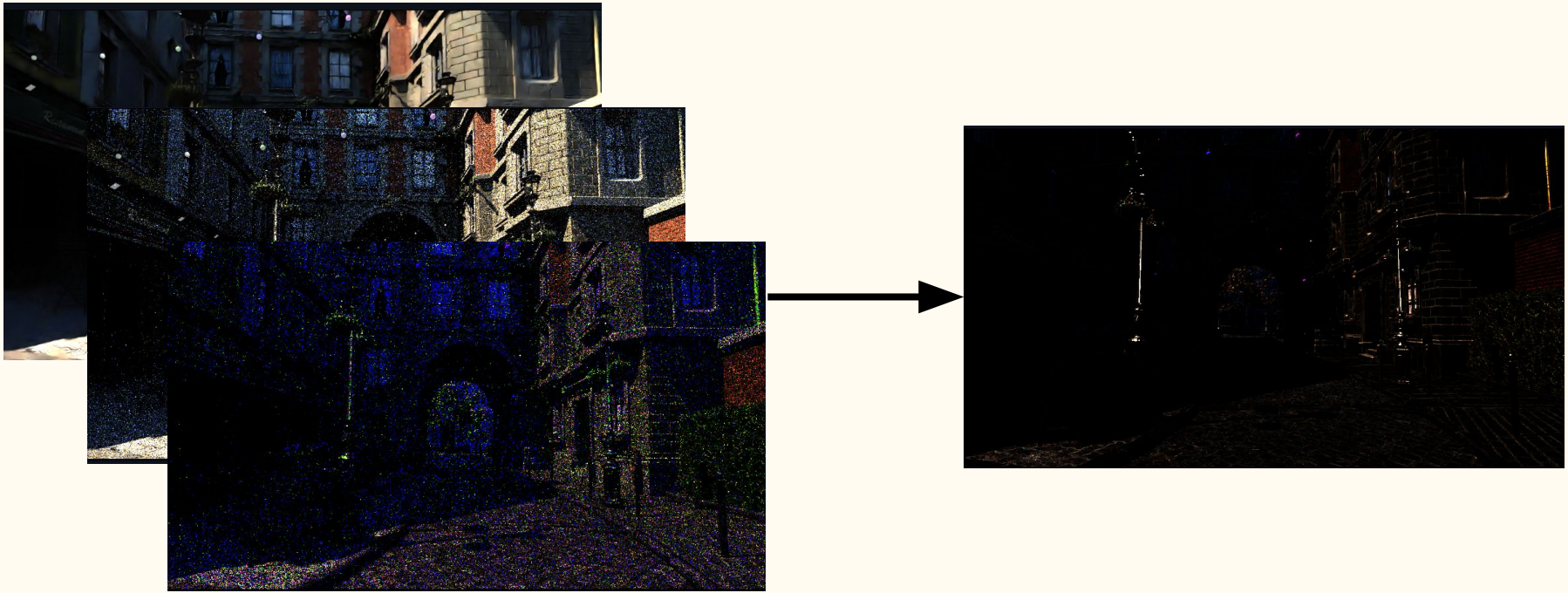
Bad prediction of SURE in low spp.

Improvement - Use better denoiser at low SPP

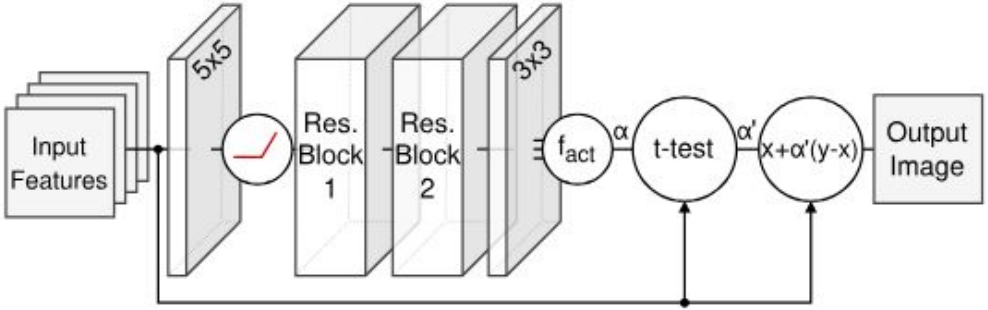
Use NPPD as denoiser → Failed



Improvement - Use better error estimation



Motivation



→ t-statistics in PD

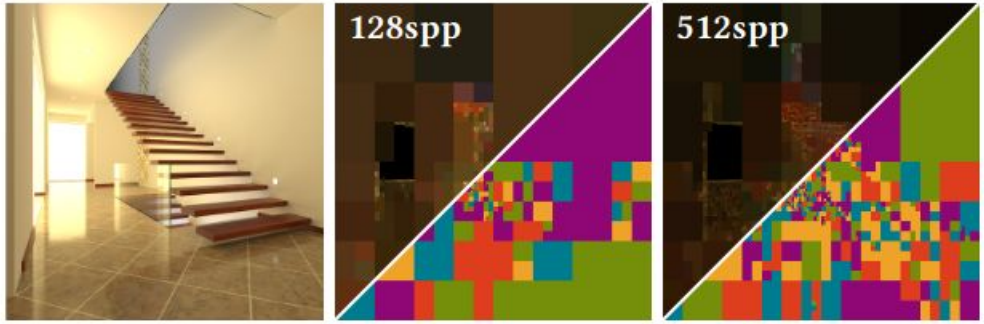
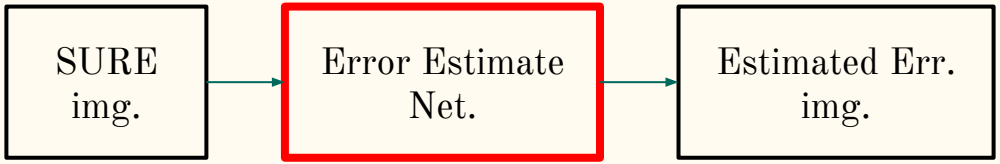


Figure 5: Block averaged SURE and false color visualization

→ Block-average
(Practical Error Estimation for Denoised Monte Carlo Image Synthesis)

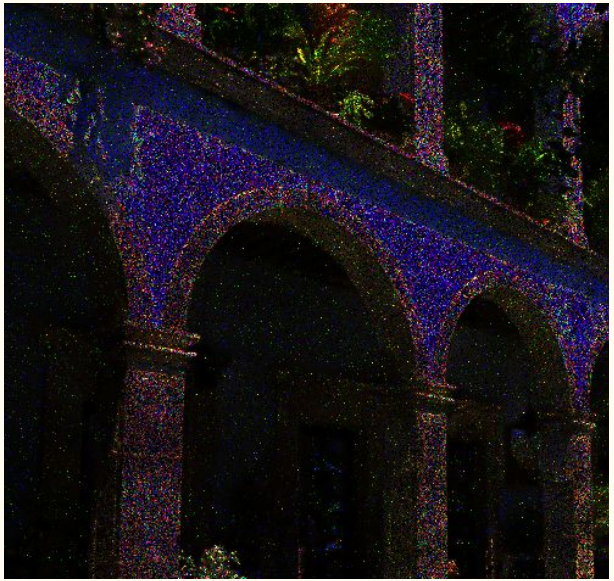
Error estimation - SURE



Less Loss?

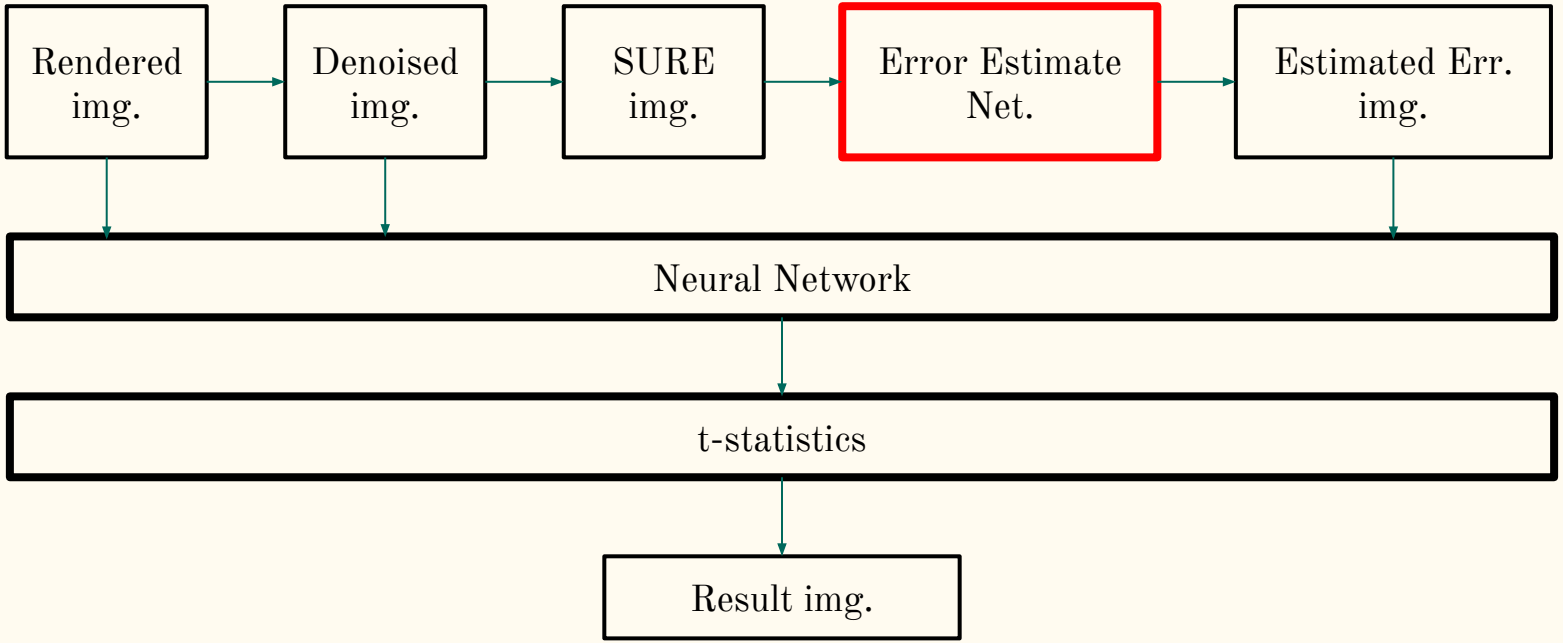
Error estimation - SURE

Small-sized specular regions(may be blurred out)



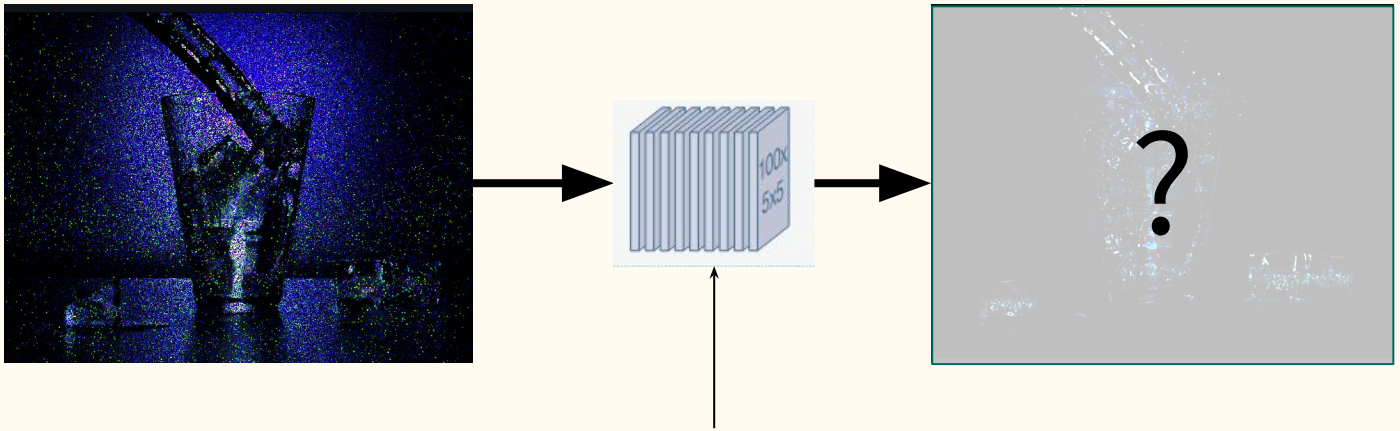
Scattered errors

Implementation



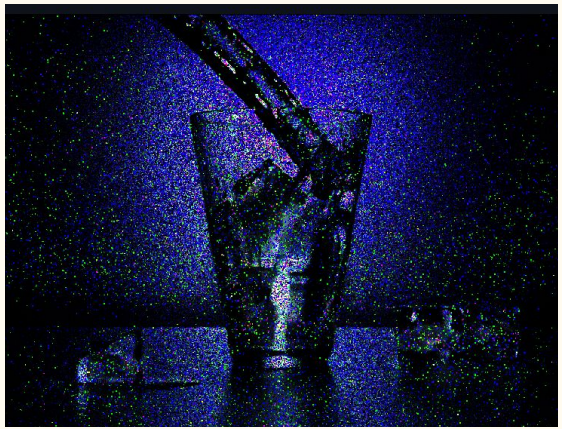
Implementation

Use Kernel-Predicting Network for error estimation

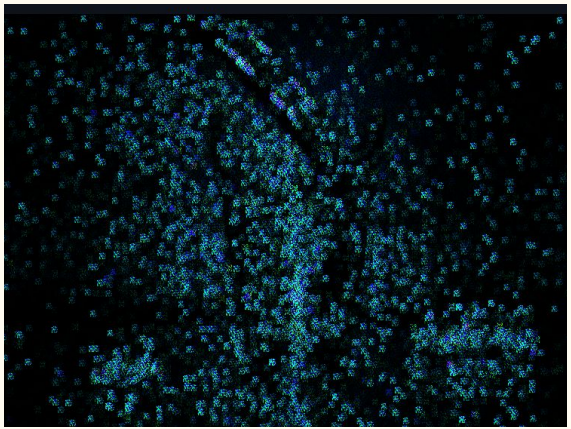


From “Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings”(Bako et al. ACM Transactions on Graphics (TOG), Volume 36, Issue 4)

Result - Error Estimation Result



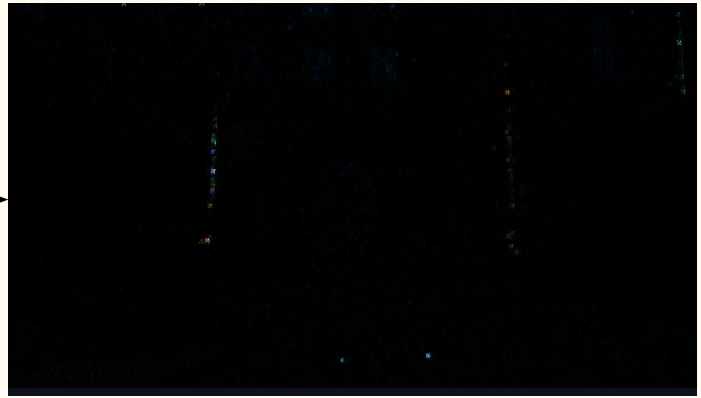
SURE



Estimated Err.

Per-pixel Squared Err.

Result - Error Estimation Result



Result - Final Denoised Result - Good cases

‘bistro-cafe’ scene(64spp)

Ours

PD

Result - Final Denoised Result - Bad cases

‘sanmiguel’ scene, 256spp

PD

Ours

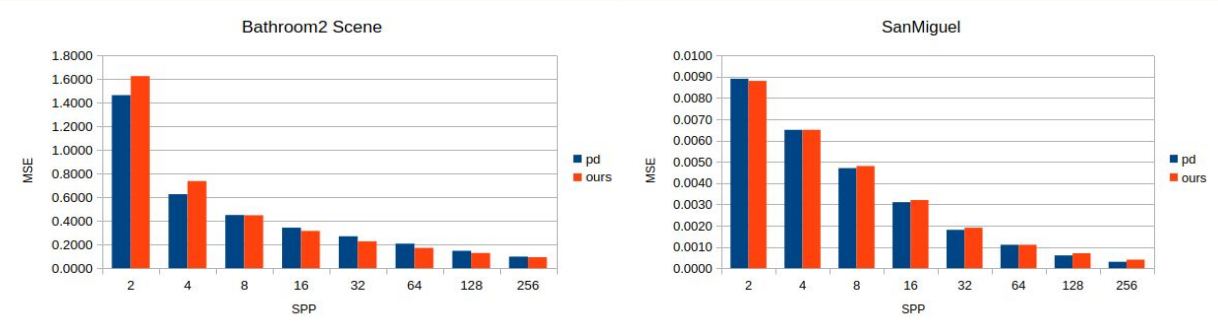
Denoised

Ours

PD

Result - Final Denoised Result

Average MSE(under 256spp) : PD=0.066, Ours=0.070



→ PD is better
(For general scene)

→ Ours is better
(For specular-dominant scene with noisy SURE)

Summary

- Better result in some ‘specular-dominant’ scenes (glass-of-water)
- Worse result in ‘diffuse-dominant’ scenes (sanmiguel)

To get meaningful improvement with this structure, modifying for general cases is inevitable

Further development?

- Estimates RelSE, rather than Squared Error

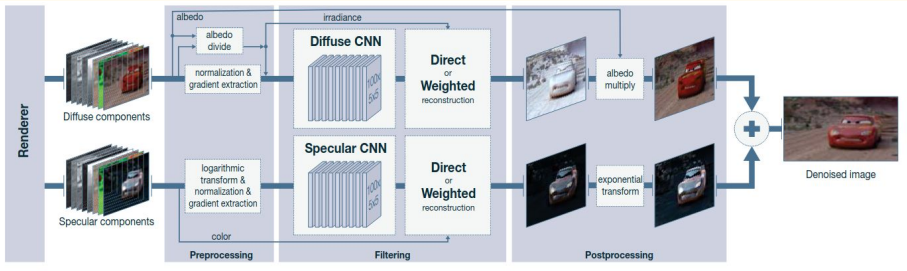
Per-Pixel Squared Error

Per-Pixel Relative Squared Error

Further development?

- More inputs on error estimator

Error estimation for landscape-2spp, ~~dark as my future~~



A denoiser using multiple inputs for denoising

Experiment - Details

- Environment
  - R7 7700X(Inference), RTX 4060Ti(8GB, Training)

- Model
  - Error estimator :
    - 3 Channel input, 5*5 kernel, 100 features, output kernel 11*11, 8 hidden layers
    - Converged less than 200 epoch(4 hours)
    - AdaGrad Optimizer with lr=0.000002
  - Progressive Denoiser:
    - Converged after 14 hours (2500 epoch(pretrained for SURE) + 2500 epoch)

Roles

- Jaehyeon Lee : Implementation, Presentation
- Chanwoo Cho : Theoretical Backgrounds, Presentation

Thank you