Kernel Refinement for Monte Carlo Denoising

Mid-term Proposal Presentation

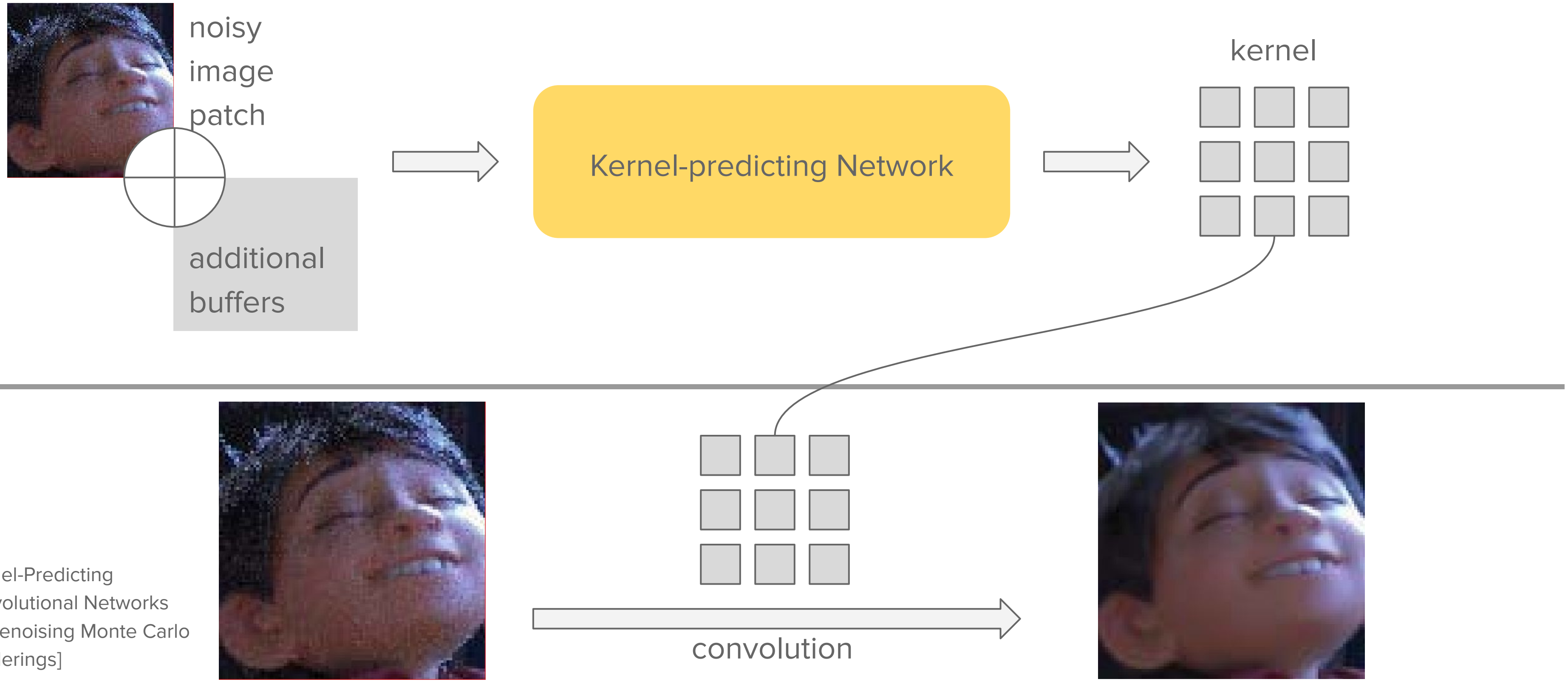
Team 4: Olivia Gerdis Odenthal, Kyubeom Han

Monte Carlo Noise

32000 samples per pixel
1083.58 secs

8 samples per pixel
2.54831 secs

Kernel-based Denoising



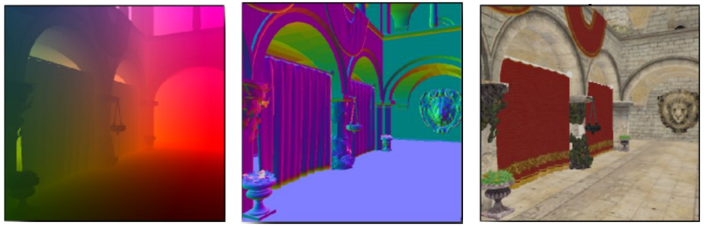
[Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings]

Different Denoising Strategies

Only Geometry-Buffers (G-Buffers)

Info about the first hit

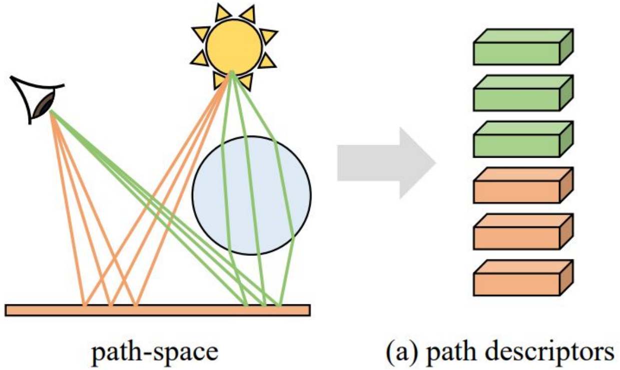
- depth / position
- normal
- albedo



[https://jo.dreggn.org/home/2010_atrous.pdf]

Additional Path features (“P-buffers”)

Info about the whole ray path

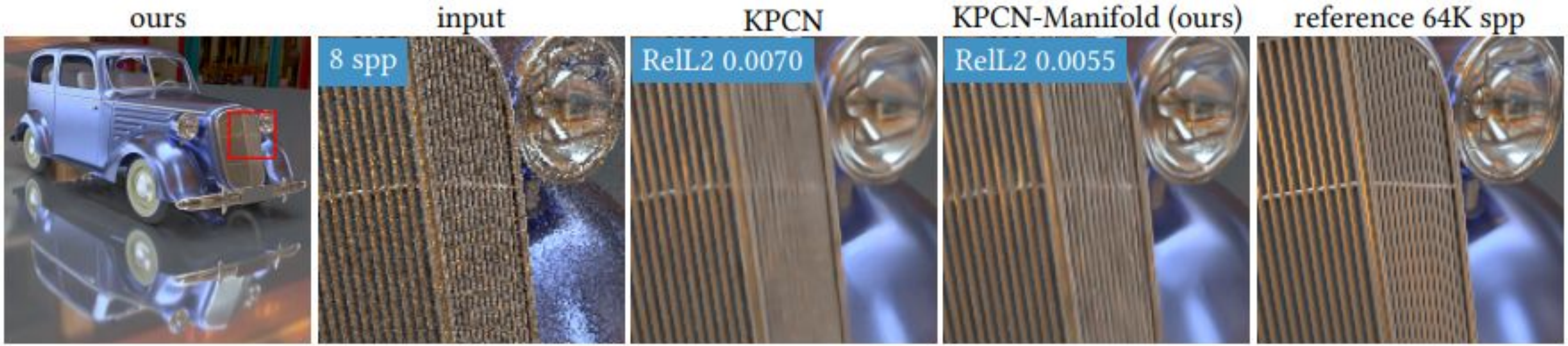


[Weakly-Supervised Contrastive Learning in Path Manifold for Monte Carlo Image Reconstruction]

Only Geometric Feature

- faster
- requires less memory

- Limitations: complex lighting conditions

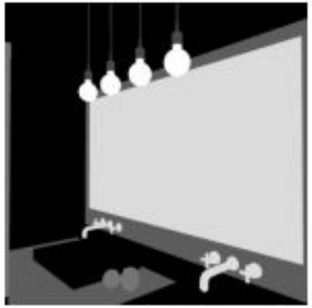


[Weakly-Supervised Contrastive Learning in Path Manifold for Monte Carlo Image Reconstruction]

Path descriptors

smoothness of bounce ( )

1



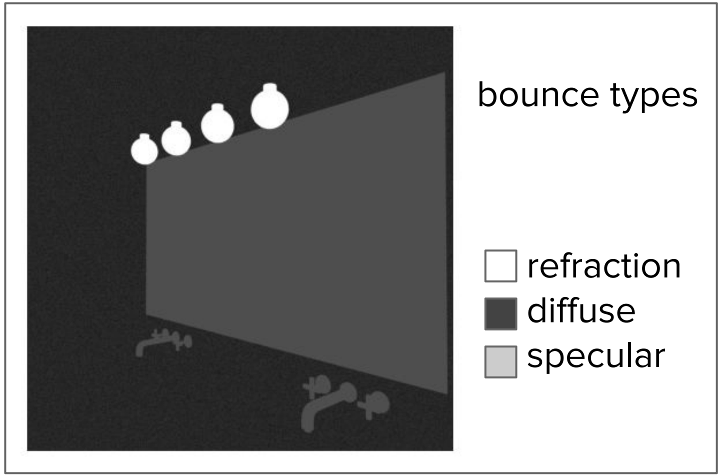
2

3

4

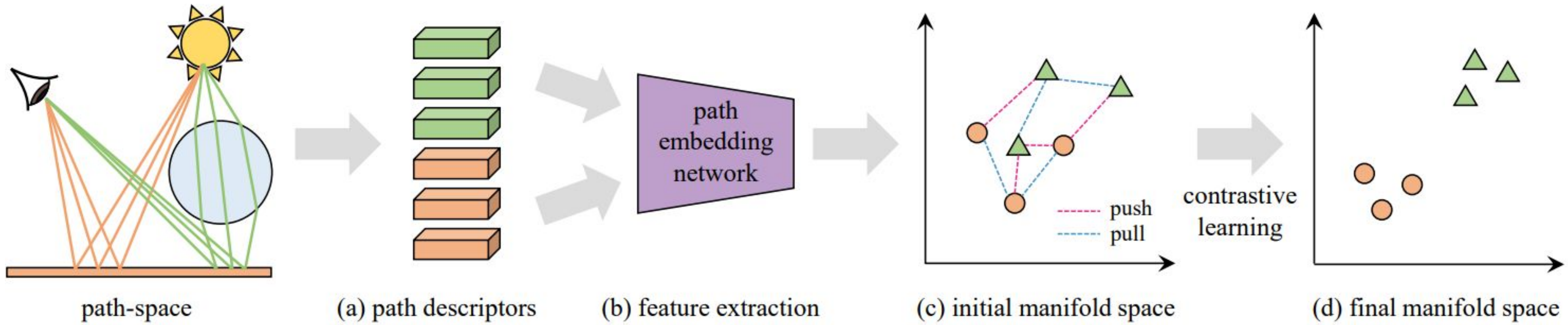
5

6



Embedding the path features

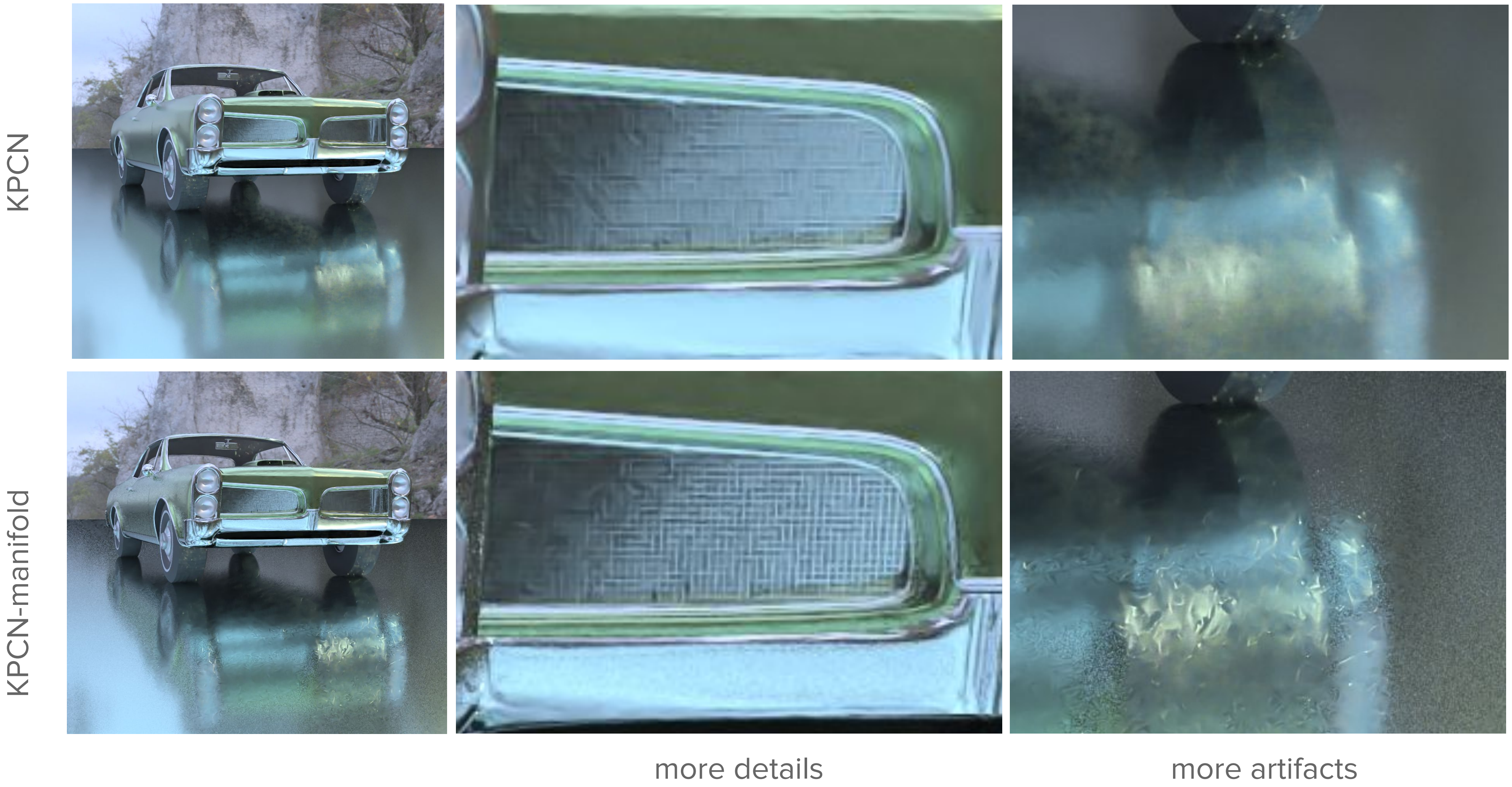
Weakly-Supervised Contrastive Learning in Path Manifold for Monte Carlo Image Reconstruction



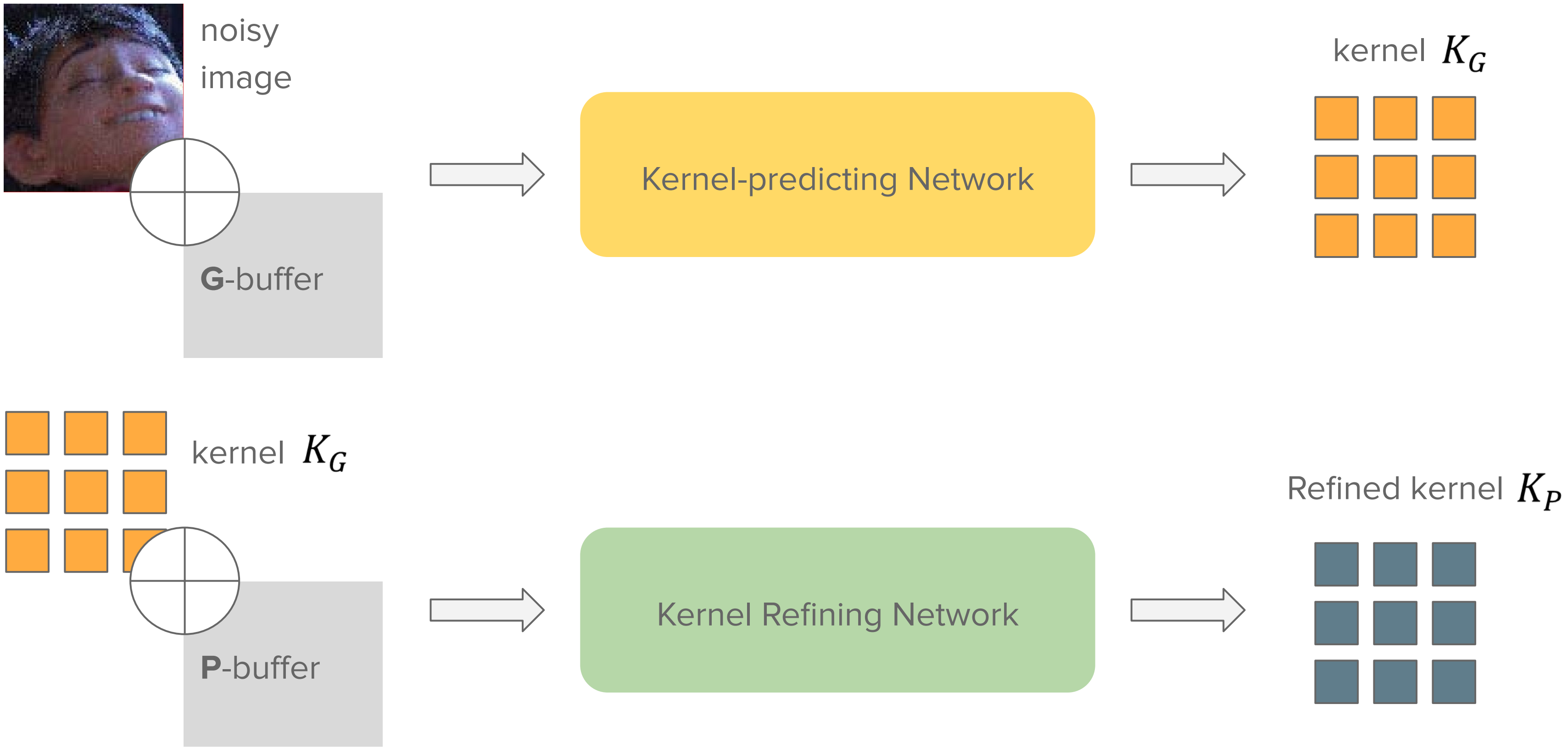
Problem with current P-buffer methods

input KPCN KPCN-manifold target

no P-Buffer with P-buffer


KPCN
KPCN-manifold
more details
more artifacts

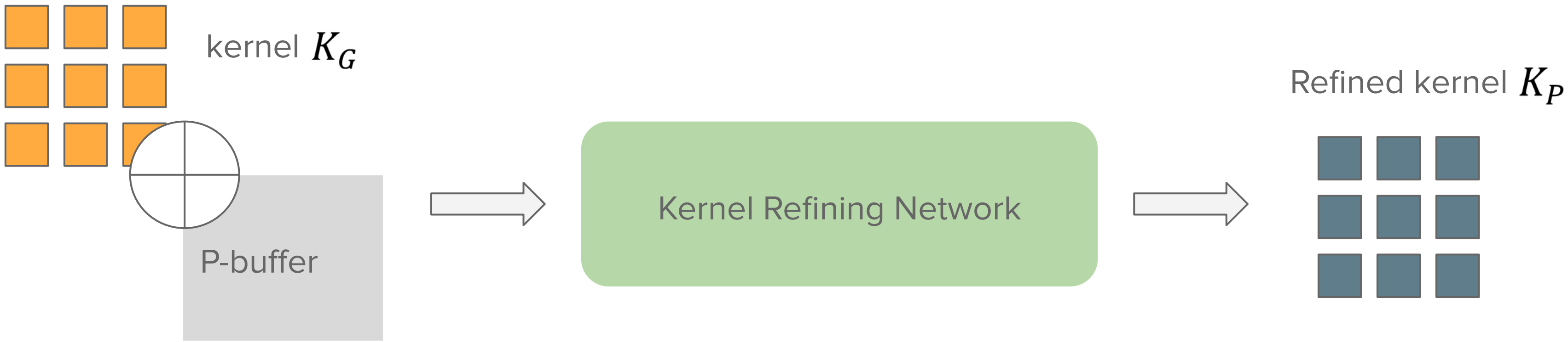
Our Plan - Kernel Refinement



Our Plan - Kernel Refinement

Instead of directly estimating the kernel using p-buffers, we **let the p-buffer refine the kernel estimated with only g-buffers**

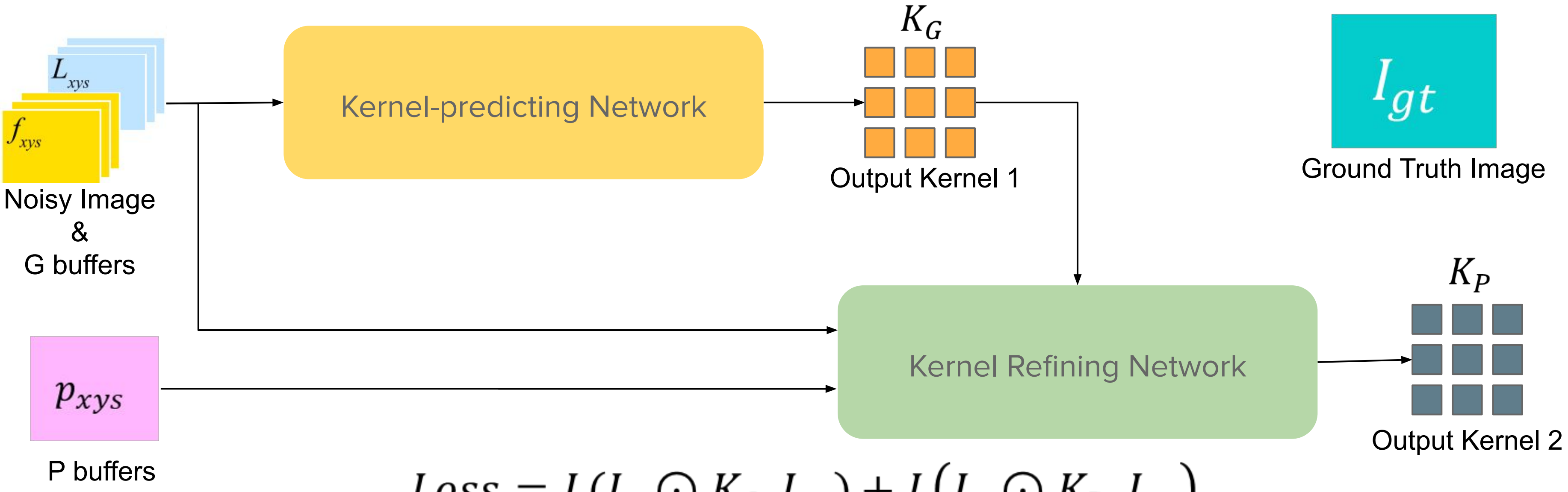
Aim to balance the effects of g-buffers & p-buffers for kernel prediction



Kernel Refinement Strategy #0

Directly feeding the kernel into the neural network might lead to worse training

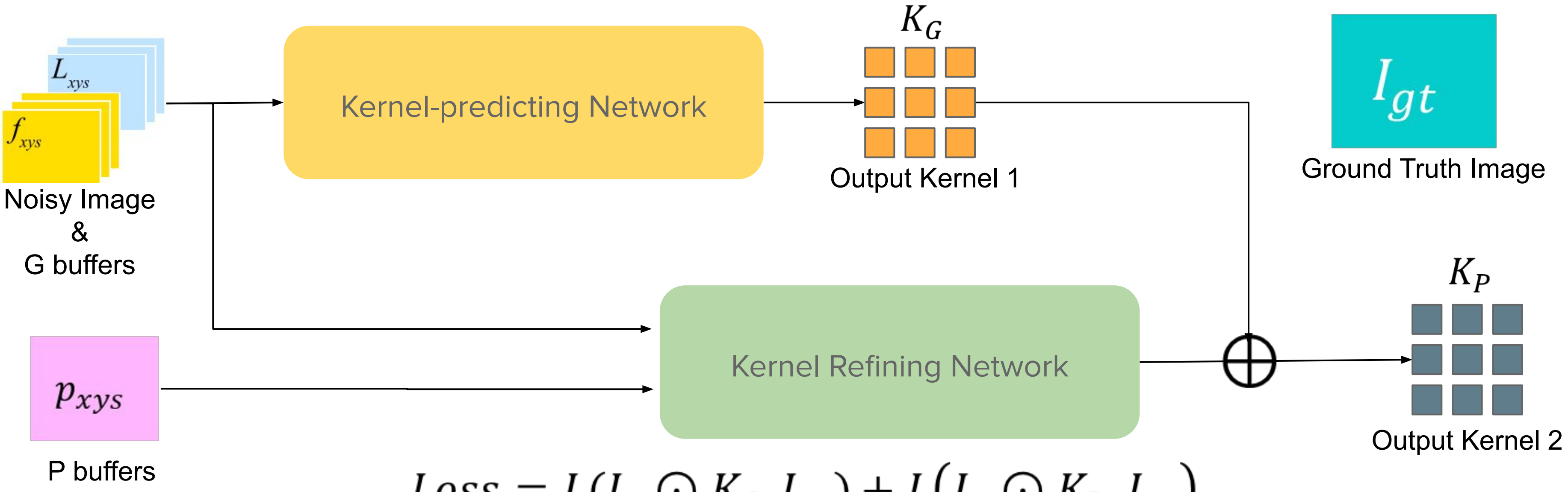
Curse of dimensionality! (Kernel: HxWx21x21, input features: HxWx34)



$$Loss = L(I_n \odot K_G, I_{gt}) + L\big(I_n \odot K_P, I_{gt}\big)$$

Kernel Refinement Strategy #1

Add skip connection in order to reduce overhead!

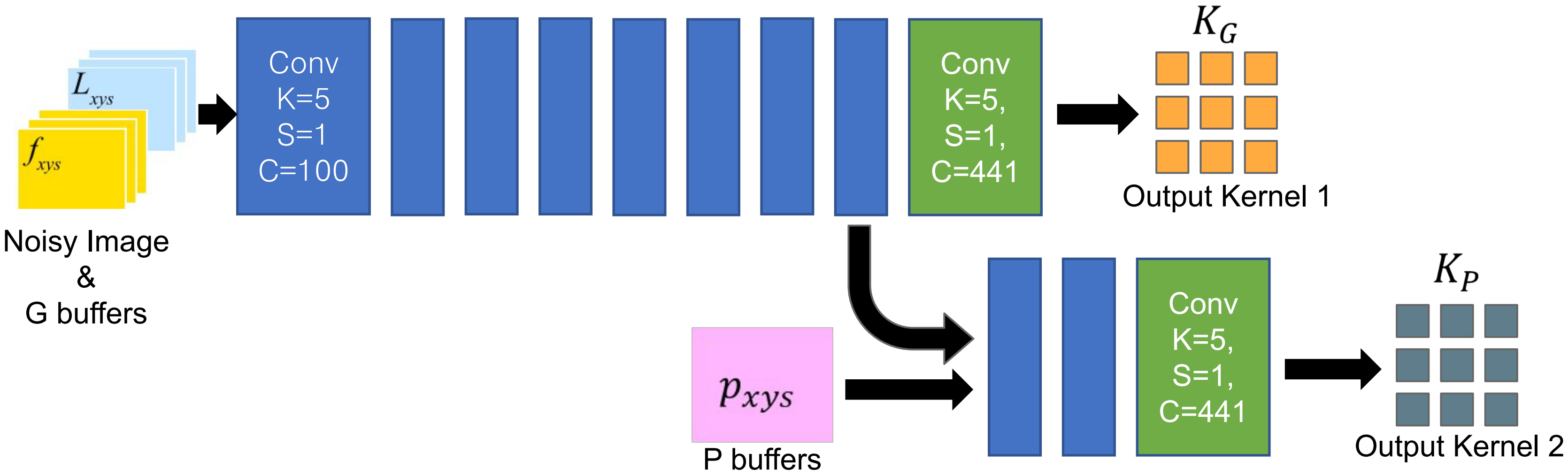


$$Loss = L(I_n \odot K_G, I_{gt}) + L\left(I_n \odot K_P, I_{gt}\right)$$

Kernel Refinement Strategy #2

Work on hidden embedding of kernel predicting networks

Detailed architecture might vary depending on type of denoisers



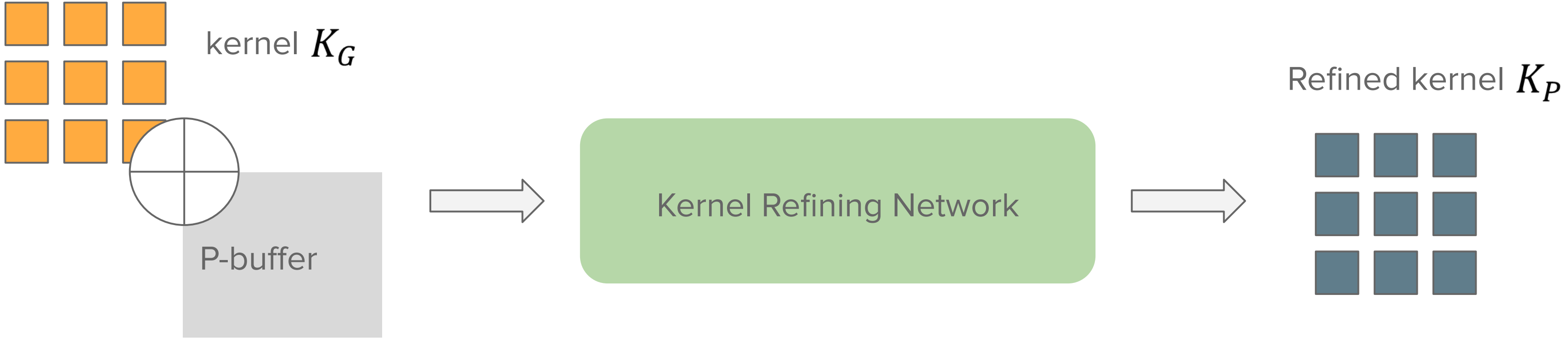
$$Loss = L(I_n \odot K_G, I_{gt}) + L\left(I_n \odot K_P, I_{gt}\right)$$

Training with Both Kernels

Refinement works on previously predicting kernel

For effective kernel construction, we want both kernels to show good performance

We train with denoised images with both kernels to **jointly train kernel-predicting network and kernel-refining network**

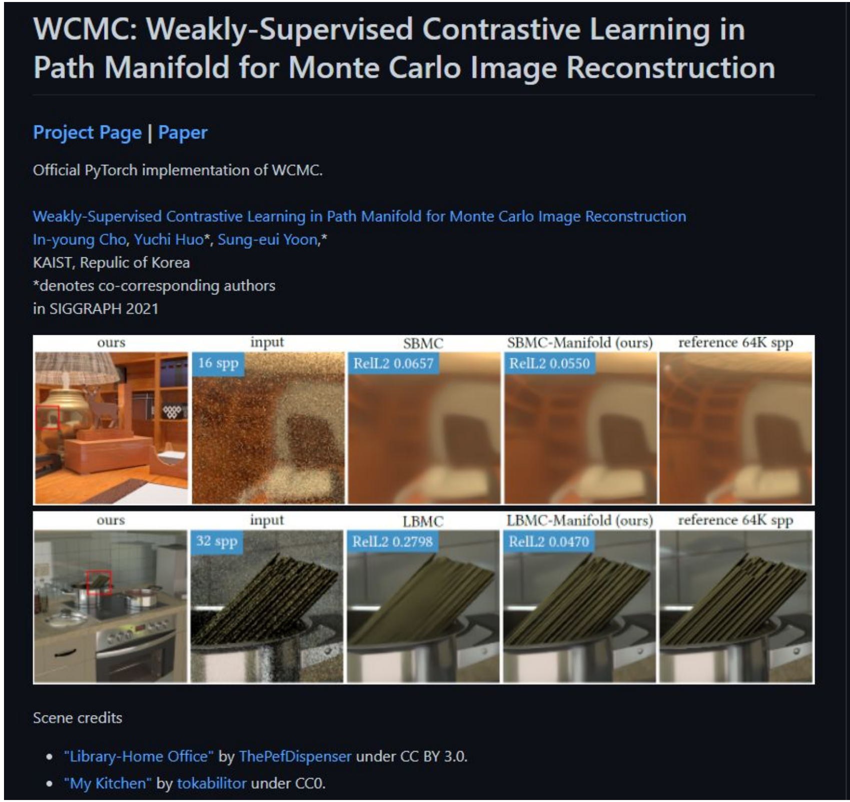


$$Loss = L(I_n \odot K_G, I_{gt}) + L\left(I_n \odot K_P, I_{gt}\right)$$

Current Status - Reproduction

Reproduced with code from WCMC

- Contains path manifold module for p-buffers
- Supports three denoisers
    - KPCN(Bako et al. 2017)
    - SBMC(Gharbi et al. 2019) to be presented...
    - NDLE(Bako et al. 2020)


WCMC: Weakly-Supervised Contrastive Learning in Path Manifold for Monte Carlo Image Reconstruction

Project Page | Paper

Official PyTorch implementation of WCMC.

Weakly-Supervised Contrastive Learning in Path Manifold for Monte Carlo Image Reconstruction
In-young Cho, Yuchi Huo*, Sung-eui Yoon,*
KAIST, Repulic of Korea
*denotes co-corresponding authors
in SIGGRAPH 2021

Scene credits

- "Library-Home Office" by ThePefDispenser under CC BY 3.0.
- "My Kitchen" by tokabilitor under CC0.

Current Status - Dataset

Using OptiX path tracer for dataset generation

To utilize limited scenes(18), we apply various augmentations

- Random camera parameters & positions
- Random material properties
- Random flip & rotation

450 images for training (1280x1280)

- Noisy: 8spp, GT: 8000spp

Training dataset for KPCN takes about 1.2TB

Dataset for SBMC, LBMC will be rendered later

Current Status - KPCN & KPCN-Manifold

Input KPCN KPCN-Manifold GT

relMSE 0.319082 **0.270142**

relMSE 0.017008 **0.009552**

Further Plans

~~Reproduce the original work~~

~~Create dataset for training~~

~~Train the original work with dataset~~

Test our kernel refinement methods on KPCN & KPCN-Manifold

If successful, we apply our idea to other denoisers SBMC and LBMC

Roles

Olivia Gerdis Odenthal

- Survey MC Denoising papers
- Design & Implement Neural Network
- Train our method & Discussion

Kyubeom Han

- Reproduce previous works
- Generating Dataset
- Design Neural Network
- Discussion

Overall Flow of DL-based MC Denoising