Denoising Deep Pixels for Deep Compositing (Pixar Technical Memo 24-01)

SHILIN ZHU, Pixar Animation Studios, USA
MARK MEYER, Pixar Animation Studios, USA

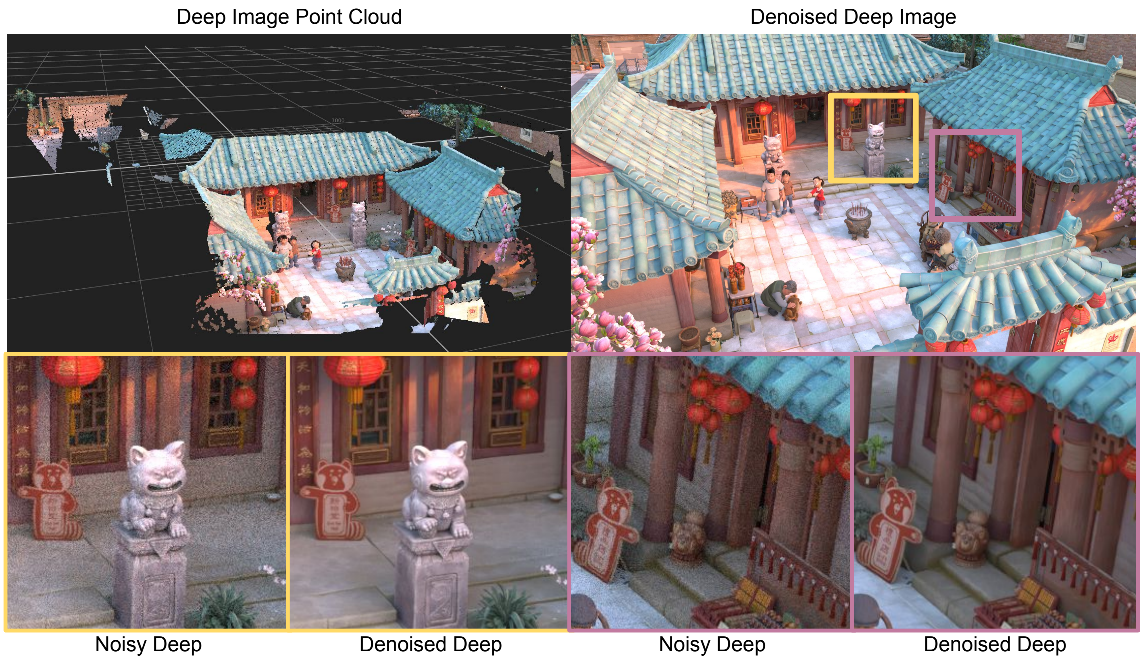

Fig. 1. An example deepColor denoising result on a selected test shot (similar performance on deepAlpha). The proposed denoiser applies directly on point cloud which is a 3D re-projection of the deep image and filters noise at all depths. The denoised output is still a deep image which keeps the original depth layout thus the result is compatible with deep compositing workflow. Applying the standard flat denoiser to the flattened image can achieve similar quality on screen space but the resulted image is always flat which cannot be used in deep compositing due to the loss of deep structure. ©Pixar

Denoising path traced images is essential in the production rendering pipeline. Existing denoisers only apply to 2D flat images, which introduces challenges in the compositing stage where multiple rendered components are combined together to produce the final look. In recent years, deep image has become more popular and preferable in the industry because it can store values at different depths, making deep compositing possible. Despite the benefit of using a deep image, the lack of a proper denoising algorithm introduces issues in production, and the standard flat image denoiser is not directly applicable by design because of its inability to process depth bins. In this project, we develop a compatible 3D denoiser that can process deep pixels effectively, which opens up more possibilities in post processing such as compositing and image editing.

1 INTRODUCTION

Ever since path tracing was born in the domain of rendering, denoising has been a constant topic of graphics research. From traditional filtering to state-of-the-art machine learning based techniques, there have been numerous denoisers proposed by researchers, which can be essentially categorized into pixel-based and sample-based solutions depending on the type of images rendered out. The former stores a single value on each pixel and the latter stores every ray-traced sample. Neither of these is ideal in practice because pixel aggregation is oversimplified whereas keeping all samples is a burden to the storage. Deep image is a better balanced representation in between, where each deep pixel contains a few depth bins in 3D and each bin aggregates multiple adjacent samples. Since deep image is capable of gathering and storing values from different depths, it supports deep compositing operations such as object insertion and removal, depth slicing, cropping, transformation, color correction, among others [2] and becomes increasingly popular.

In production pipeline, we have been using flat image denoisers (KPCN [1, 3]) and relying on extra channels such as alpha or IDs for compositing images from multiple render passes. However, the lack of depth information often causes edge artifacts and restrictions on what compositing can be performed, which leads to re-rendering in many cases. To solve this problem, DeepEXR image is developed and is able to create a binning structure for storing samples at different depths. But without a proper denoiser, deep images (e.g., deepAlpha) normally need to be rendered to near convergence by tracing a lot of samples.

A straightforward solution is to flatten the deep image in advance and then apply the standard KPCN denoiser. However, we quickly abandon the direction because the output of KPCN is always a flat image which is incompatible with deep compositing workflow, and there is no robust approach to recover the lost depth information after flattening the noisy deep image. We tackle this long-standing problem which has been demanded for years by borrowing a machine learning solution that can preserve the deep image structure after denoising, similar to traditional deep image filters [2] but ours performs better with a neural network. The key concept is to predict denoising kernels based on 3D point clouds as opposed to 2D regular convolutions on flat pixels performed by existing KPCN denoiser.

2 METHOD AND RESULTS

Deep image is essentially a 3D representation associated with a camera. Therefore, we can re-project deep pixels back to 3D space and construct a point cloud with each point carrying the data at designated depth and pixel locations. Points are connected through a radius-based neighboring search based on a mixed criterion of 3D distance and pixel adjacency, and fine to coarse graph layouts are created at different scale levels of the neural architecture through graph-based downsampling and upsampling to leverage both local and global information. Denoising on such irregular 3D structure is clearly a lot more nontrivial than on regular flat images, as a result we utilize 3D deep learning to accomplish the task.

We experiment with a variety of point cloud and Graph Neural Network (GNN) operators listed in PyTorch Geometric, a geometric deep learning library, to encode inputs into feature embedding without changing the depth layout. We select the graph attention network as the main backbone architecture. Following our existing KPCN denoiser, we encode and reconstruct 3D denoising kernels at multiple scales, where kernels are applied back to noisy graphs to produce the final denoised output. The presented workflow can be naturally extended to any data stored in the point cloud, although our main use cases are deepColor and deepAlpha images.

Since there is no readily available deep image dataset, we manually collect 200 unique shots from Disney and Pixar's *Turning Red* and *Elemental* including a diverse set of scenes and depth ranges. Each shot is rendered with multiple sample count and variance levels, and reference with 8K-16K samples with adaptive sampling. The same set of input features (AOVs) as our KPCN denoiser is also rendered along with the main color. We use RenderMan's DeepEXR driver to generate all training data which took months.

As presented in Figure.1, the proposed deep image denoiser can perform the noise removal task effectively. In fact, the denoising is carried out at all depth levels when slicing from near to far distance to the camera, therefore the denoised output always retains being a deep image. It is fundamentally different compared to the flat image case where denoising is performed only at image plane. Filtering every location in 3D space allows deep compositing to be achieved by artists without constantly worrying about the noise to appear.

3 DISCUSSION AND CONCLUSION

Deep image is becoming increasingly popular in production recently, and denoising is an essential piece to achieve deep compositing in practice. We have demonstrated a working solution that performs denoising effectively in 3D space, which is only the first step towards a complete deep workflow in our pipeline. Besides depth(Z)-based deep image, ID-based deep image also needs to be considered since it is widely adopted in production. The binning and compression algorithm on deep pixels affects the depth layout as well as denoising, which requires future exploration. Volumetric effects often have special treatment by the renderer, thus additional work is needed to address all rendering cases we might encounter in production.

REFERENCES

[1] Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. 2017. Kernel-predicting convolutional networks for denoising Monte Carlo renderings. *ACM Trans. Graph.* 36, 4 (2017), 97–1.

[2] Delio Vicini, David Adler, Jan Novák, Fabrice Rousselle, and Brent Burley. 2019. Denoising deep monte carlo renderings. In *Computer Graphics Forum*, Vol. 38. Wiley Online Library, 316–327.

[3] Thijs Vogels, Fabrice Rousselle, Brian McWilliams, Gerhard Röthlin, Alex Harvill, David Adler, Mark Meyer, and Jan Novák. 2018. Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–15.