Adaptive Incident Radiance Field Sampling and Reconstruction Using Deep Reinforcement Learning Supplementary Report

ACM Reference format:
. 2020. Adaptive Incident Radiance Field Sampling and Reconstruction Using Deep Reinforcement Learning Supplementary Report. 1, 1, Article 0329 (February 2020), 3 pages.
https://doi.org/0000001.0000001_2

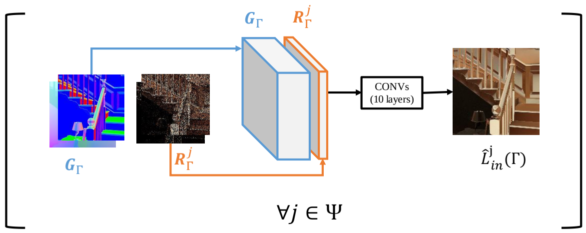

Fig. 1. Illustration of the image network. w and h are image space axises. Subscript i and Γ represent pixel $\boldsymbol{x}_i$ and its nearby pixel set, respectively; superscript j represents the radiance field block B^j out of its set Φ. G_Γ is image space auxiliary feature map; $\boldsymbol{R}_\Gamma^j$ is radiance feature map of B^j; the final output is the predicted incident radiance. The CNN works on the image space for each radiance field block B^j.

1 ALTERNATIVE RECONSTRUCTION NETWORK

1.1 Image Network

Our image network is inspired by an existing image space filtering and denoising method, DPCN [Bako et al. 2017]. This image network receives the image space auxiliary feature maps, e.g., positions, and then convolves them with radiance feature maps in the image space for each separate radiance field block. As illustrated in Fig. 1, the input is image space auxiliary features G_Γ and radiance features $\boldsymbol{R}_\Gamma^j$ of one radiance field block B^j, and the output is the predicted radiance of the radiance field block of all pixels. We use nine hidden layers with 100 kernels of 5×5 per layer and one output layer without activation functions. Compared to the image-direction network, the image network processes each radiance field block individually, and thus takes much longer inference time. In addition, without the information of nearby radiance field blocks, the results are worse when there are not enough samples. More comparisons in the main paper.

Besides directly predicting the radiance as output, we also tried predicting the filtering kernel, named KPCN, as proposed by Bako et al. [2017]. We found that KPCN has a faster convergence speed in the image space, but has a larger output, e.g., 21×21 channels per pixel, which results in more inferencing time and memory consumption. The overhead is negligible in the image space, but becomes significant for handling hundreds of radiance field blocks. As a result, we did not adopt the filtering kernel approaches.

2020. XXXX-XXXX/2020/2-ART0329 $15.00
https://doi.org/0000001.0000001_2

1.2 Direction-image Network

In addition to the image space, features in the direction space (i.e., the space of incident radiance field parameterized by the spherical coordinates) can help reconstruct the incident radiance field. The direction-image network is based on the opposite idea of the image-direction network. It contains two parts of direction and image sub-networks. The direction part first explores directional features in the direction space. The second part (i.e., image part) takes the learned feature maps from the direction part, and then convolves them with geometry feature maps to predict the final output.

The input of the first part is the direction space features $\boldsymbol{D}_\Gamma^j$. We arrange radiance field blocks in the direction space according to their directional coordinates and then convolve these blocks to explore data coherence within nearby radiance field blocks. The basic directional features include the average incident radiance (3 channel) and second hits' average distance (1 channel). Similar to the image-direction network, we also calculate their variances. However, only the samples of one pixel itself is too sparse at most cases. To include more valid information, we treble input features by gathering them in all of $1 \times 1, 3 \times 3$ and 5×5 pixel windows, i.e., putting all samples into the window to the centered pixel's radiance field blocks. The idea is similar to reusing nearby pixels' directional samples to reconstruct the radiance field [Lehtinen et al. 2012]. This part contains 3 hidden convolutional hidden layers with with 50 kernels of 3×3. The output is $4 \times 4 \times 6$ image feature maps $\boldsymbol{Fi}_i^\Phi$ (i.e., one $1 \times 1 \times 6$ slice for one radiance field block) each pixel.

The second part first rearranges the feature $\boldsymbol{Fi}_i^\Phi$ into the image space feature maps. However, we found that geometry information is still very helpful for the image part. Therefore we append the image space auxiliary feature map G_Γ to the output of the first part. The second part of convolution is in the image space for each radiance field block. The sub-network contains 6 standard convolutional hidden layers with 100 kernels of 5×5 and one output layer without the activation function.

1.3 Direction Network

The direction network purely explores data coherence in the direction space. As showed in Fig. 3, it takes as input the directional features $\boldsymbol{D}_\Gamma^j$ of a pixel and outputs the radiance field block for the pixel. Because it only takes directional information into consideration, the lost of geometry features leads to blurred geometry boundary. In addition, the results are not very smooth in the image space. Please refer to the main paper for more comparisons.

The direction network has 8 standard convolutional hidden layers (i.e., with convolutional kernel, bias and activation function) and one

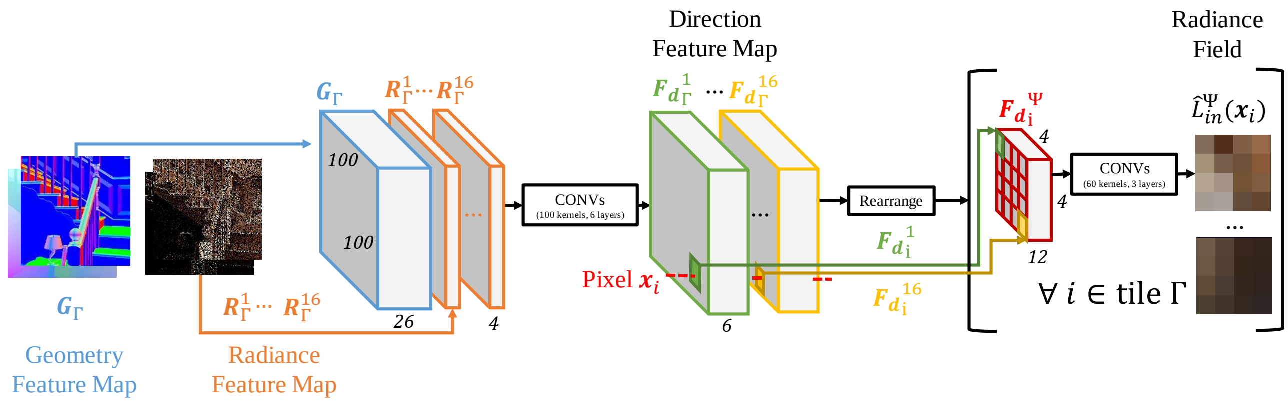

Fig. 2. Illustration of the direction-image network. w and h are image space axes; ϕ and θ are direction space axes. Subscript i and Γ represent pixel $\boldsymbol{x}_i$ and its pixel set, respectively; superscript j and Φ represent radiance field block B^j and its radiance field block set, respectively; $\boldsymbol{D}$ are direction features; $\boldsymbol{F}\, \boldsymbol{i}$ represents the image space feature map extracted by the network; G_Γ is image space auxiliary feature map of Γ; the final output is the predicted incident radiance. The first CNNs work on the direction space for each separate pixel $\boldsymbol{x}_i$, and the second CNNs work on the image space for each separate radiance field block B^j. The rearrange operation breaks down the output of first part and then arranges them in the image space.

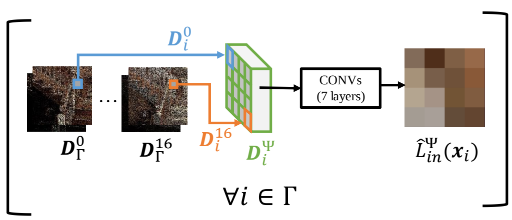

Fig. 3. Illustration of the direction network. ϕ and θ are axes of the direction. Subscript i and Γ represent pixel $\boldsymbol{x}_i$ and pixel set, respectively; superscript j and Φ represent radiance field block B^j and radiance field block set, respectively; $\boldsymbol{D}_i^{\Phi}$ are direction features; the final output is the predicted incident radiance. The CNN works on the direction space for each pixel, separately.

output layer without the activation function. Each layer contains 50 kernels of 3×3.

1.4 Convergence

Fig. 4 shows the convergence statistics of four R-networks. Even though the direction-image network performs consistently better than others, the image-direction is chosen as our solution for its inference efficiency.

Unlike the R-network, the Q-network is supposed to be involved multiple times. Therefore, we want to keep its layers as less as possible to decrease the overhead. Fig. 5 plots the loss of Q-network with different layers. We choose 3-layers since it has a similar loss to 4- and 5-layers, but has a less computation overhead.

REFERENCES

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 Transactions on Graphics (TOG)* 36, 4 (2017), 97.

Jaakko Lehtinen, Timo Aila, Samuli Laine, and Frédo Durand. 2012. Reconstructing the indirect light field for global illumination. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 51.

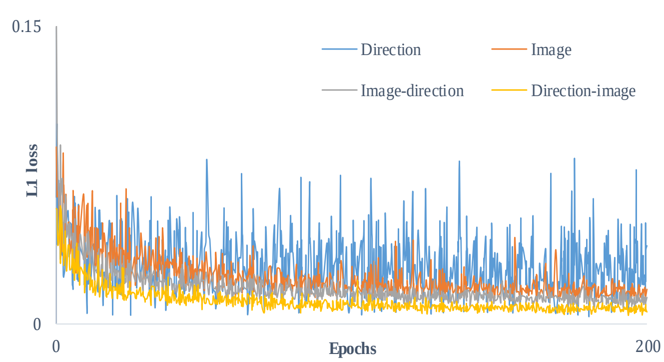

Fig. 4. Convergence of the R-networks. The direction-image shows the best quality with equal spp, but the image-direction has better quality with the equal computational time.

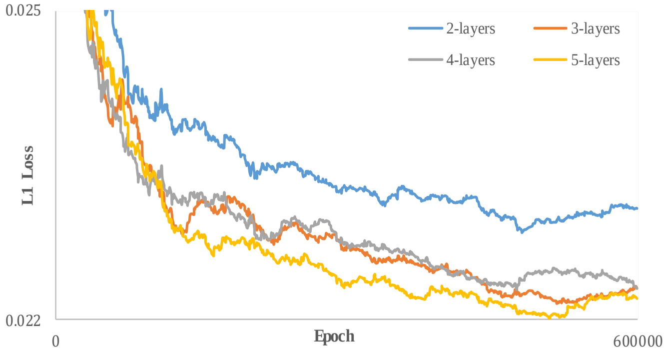

Fig. 5. Convergences of the Q-network with varying number of layers.

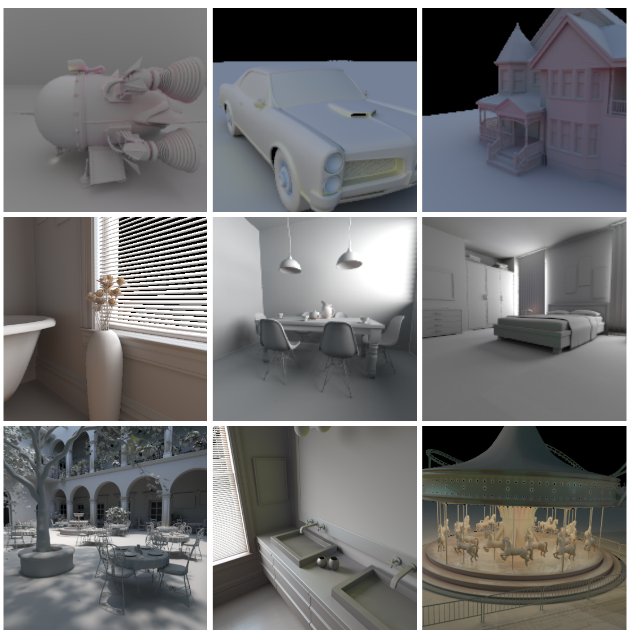

Fig. 6. Nine scenes from the training set.