Eurographics Symposium on Rendering (DL-only Track) (2020)
C. Dachsbacher and M. Pharr (Editors)

Real-time Monte Carlo Denoising with the Neural Bilateral Grid: Supplementary Document

Xiaoxu Meng[1], Quan Zheng[1,2], Amitabh Varshney[1], Gurprit Singh[2], Matthias Zwicker[1]

[1]University of Maryland, College Park, USA
[2]Max Planck Institute for Informatics, Germany

Contents

1. Architecture of Multi-resolution KPCN

The original kernel prediction denoiser (KPCN [BVM*17] is designed for offline denoising, which takes several seconds to denoise a 720p frame. Instead of comparing with KPCN, we have included a multi-resolution variant of KPCN in our comparisons and we refer to it as MR-KP. Its architecture is presented in Figure 1, where the core component is a convolutional neural network which computes kernels for three levels. The noisy input radiance is filtered at three different resolutions and the filtered results are gradually combined to obtain the final result. Similar to the composition module of [VRM*18], we blend images of two adjacent levels with

$$b_k = S_{up}(d_{k-1}) \cdot w_k + d_k - S_{up}\big(S_{down}(d_k)\big) \cdot w_k. \tag{1}$$

Here, S_{down} is a downsampling function (2×2 average pooling), S_{up} is a nearest-neighbor upsampling function, and d_k is the denoised image from level k. Note that d_4 is a downsampled copy of the original noisy input radiance. w_k is a scalar weight map for level k, which is produced along with kernels.

The validation PSNR and validation loss of ours (2-layer 3-grid) and MR-KP (5-layer and 1-layer) for scene Sponza in the BMFR dataset is shown in Figure 3 and Figure 2. Although MR-KP 5-layer shows lower error in the validation stage, it overfits the training dataset easily. In the test stage, ours shows better generalization performance with PSNR (OURS) = 32.787 while PSNR (MR-KP 5-layer) = 29.476 and PSNR (MR-KP 1-layer) = 29.912.

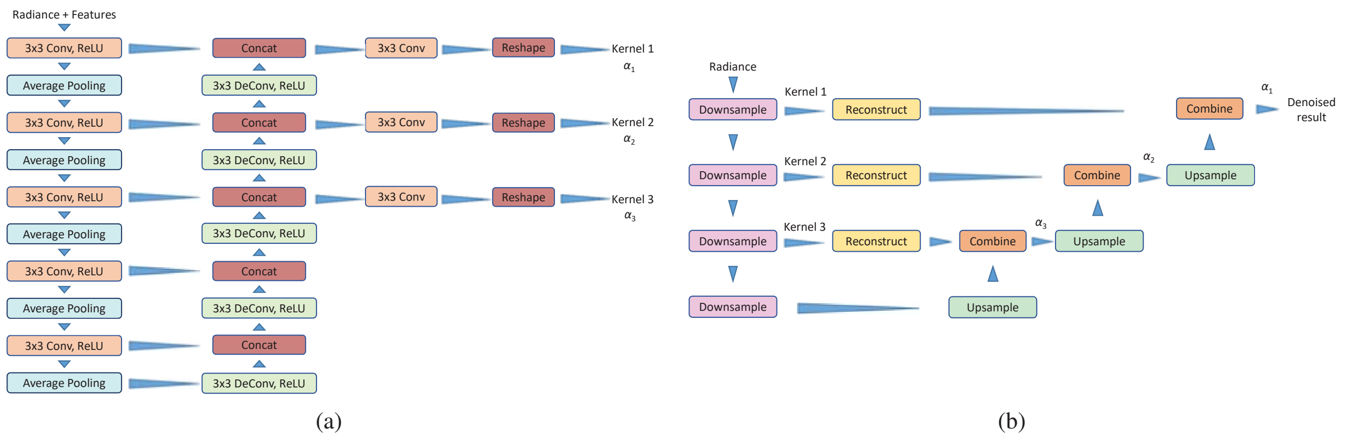

Figure 1: *Architecture of the MR-KP. (a) The kernel prediction network is designed as a convolutional neural network. With the predicted kernels, the noisy input are downsampled and filtered at three levels. (b) Denoised results of two adjacent levels are gradually blended to get the final output*

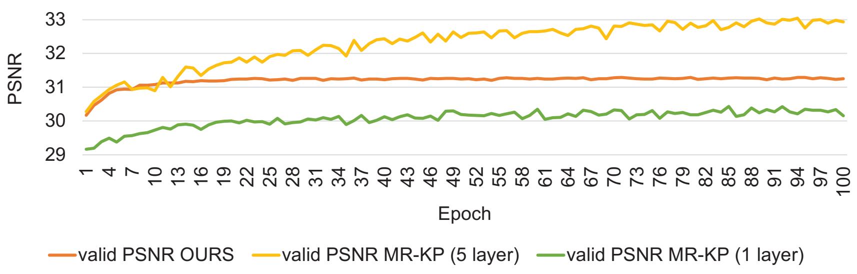

Figure 2: *The validation loss of ours (2-layer 3-grid), MR-KP (1-layer) and MR-KP (5-layer) for scene Sponza in the BMFR dataset.*

2. Additional Comparisons on the Tungsten Dataset

On our Tungsten dataset, we apply our denoiser to denoise 64-*spp* noisy frames. In Figure 4, we show comparisons results on the Classroom, Country Kitchen, and White Room scenes. Additionally, we present average errors over 100 frames in Table 1. Again, we report two versions of error measurements of our denoiser: with outlier removal and without outlier removal. Note that outlier removal will slightly increase the numerical errors.

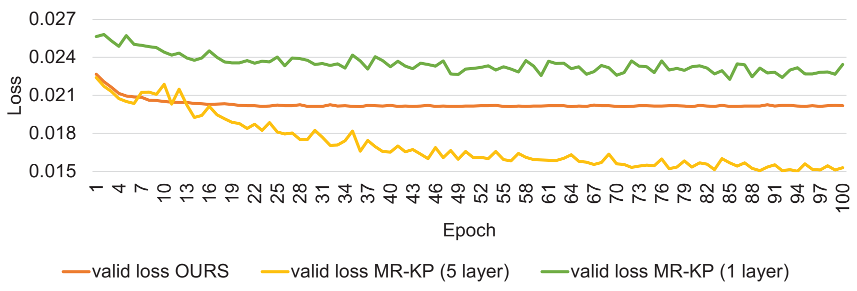

Figure 3: *The validation PSNR of ours (2-layer 3-grid), MR-KP (1-layer), and MR-KP (5-layer) for scene Sponza in the BMFR dataset.*

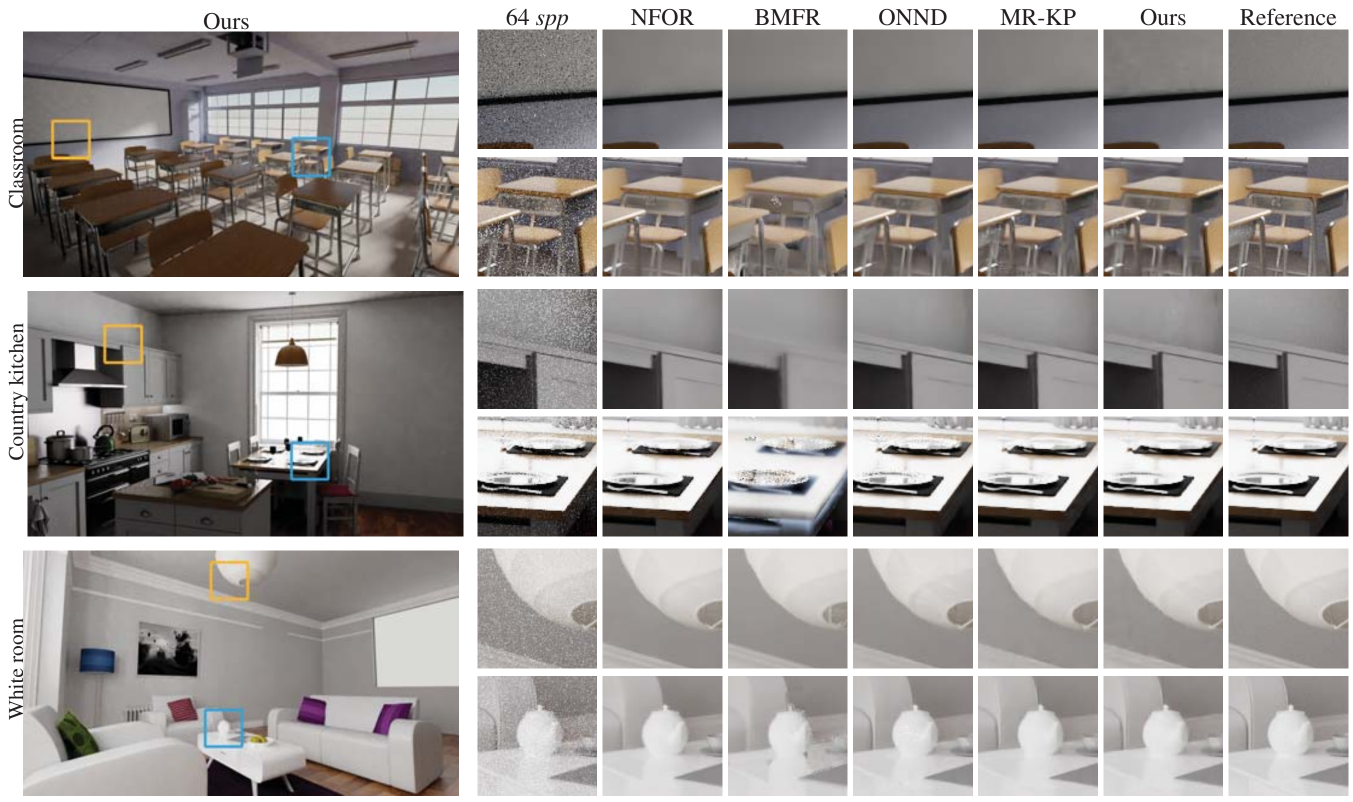

Figure 4: *Visual quality comparisons between our method and compared methods on the Classroom, Kitchen and White Room scenes. We show a single frame from animated sequences of our Tungsten dataset rendered at 64* spp *and not using temporal accumulation. For each scene, closeups of orange frames are shown on the top row and closeups of blue frames are on the bottom row. The reference images are rendered with 4096* spp*. We use the 7-layer 3-grid architecture.*

Table 1: *Numerical errors for our trained denoisers on the Classroom, Country Kitchen and White Room scenes of the Tungsten dataset. Our denoiser uses the 7-layer 3-grid architecture. Input data are rendered at 64* spp *and pre-stage temporal accumulation is not applied. 'Ours' denotes our denoiser with outlier removal preprocessing, whereas 'Ours wo' is without outlier removal.*

Scene	PSNR						SSIM					
	NFOR	BMFR	ONND	MR-KP	Ours wo	Ours	NFOR	BMFR	ONND	MR-KP	Ours wo	Ours
Classroom	31.6664	24.7305	**32.8743**	32.5346	32.1186	31.4955	0.9400	0.8523	**0.9490**	0.9450	0.9415	0.9423
Kitchen	34.6762	24.4801	34.7969	**35.7341**	35.5309	34.8202	0.9728	0.9078	0.9731	0.9740	**0.9741**	0.9734
White room	37.6258	26.3966	36.5971	37.5122	**38.0810**	37.1518	**0.9774**	0.9453	0.9733	0.9769	0.9773	0.9766

3. Additional Evaluation Methods

Besides PSNR and SSIM, we calculate the average relative mean square error (relative-MSE), root mean square error (root-MSE), and symmetric mean absolute percentage error (SMAPE) for denoised images. The results of the 1-*spp* dataset are presented in Table 3, Table 4, and Table 2, respectively. The results of the 64 *spp* dataset are presented in Table 7, Table 8, and Table 6, respectively. To facilitate easy assessment, we have included the full SSIM images (brighter is better) and the relative-MSE images (darker is better) in the supplemental data package. We recommend readers to view these error maps in the interactive viewer for pixel-wise error comparisons.

To evaluate temporal stability of the different denoising approaches quantitatively, we adopt standard metric Video Multi-Method Assessment Fusion (VMAF) [ALM*15]. Table 5 presents the average VMAF over 60 frames on five scenes of the BMFR dataset. It can be observed from Table 5 that our method generally provides high VMAF scores. SVGF performs well in VMAF scores because the predicted frames are temporally smooth. However, SVGF doesn't preserve the correct highlights and glossy reflection, leading to low SSIM and PSNR. Table 9

Table 2: *A comparison of average SMAPE values (lower is better) for evaluating our trained denoisers on 1-*spp *BMFR test data.*

Scene	SMAPE							
	NFOR	BMFR	ONND	SVGF	MR-KP(5-layer)	MR-KP(1-layer)	Ours(3-grid)	Ours(1-grid)
Classroom	5.944	5.306	10.953	8.252	**4.058**	5.484	4.351	4.482
Living room	4.448	3.792	8.834	4.535	3.134	4.212	**2.768**	4.404
San Miguel	24.969	24.438	30.005	27.401	25.278	30.995	21.576	**20.936**
Sponza	7.874	6.518	14.830	10.927	6.685	6.170	**3.944**	4.172
Sponza (glossy)	16.573	15.479	20.619	16.002	11.606	11.991	**9.168**	9.841
Sponza (mov. light)	21.245	30.793	18.367	29.276	12.830	18.619	12.302	**12.225**

Table 3: *A comparison of average relative-MSE values (lower is better) for evaluating our trained denoisers on 1-*spp *BMFR test data.*

Scene	relative-MSE							
	NFOR	BMFR	ONND	SVGF	MR-KP(5-layer)	MR-KP(1-layer)	Ours(3-grid)	Ours(1-grid)
Classroom	2.621	0.406	1.857	0.366	0.188	0.174	0.175	**0.168**
Living room	13.168	13.096	8.759	33.197	7.519	18.654	**3.159**	24.496
San Miguel	202.431	250.519	195.152	93.898	117.102	63.554	**72.004**	110.855
Sponza	1.443	1.188	1.581	0.933	1.142	1.426	**1.013**	0.716
Sponza (glossy)	24.909	41.307	45.183	8.521	29.907	28.771	**21.044**	77.427
Sponza (mov. light)	10.448	10.193	9.015	**4.512**	6.064	5.564	6.229	7.266

presents the average VMAF over 100 frames on five scenes of our Tungsten dataset. In the Tungsten dataset our VMAF results are still generally comparable to the other real-time methods.

4. Additional Ablation Studies

4.1. Architecture comparisons

The core of our neural bilateral grid denoiser is a scalable *GuideNet*. We evaluate three choices of architecture design. The first is a simplified design which has the same shallow convolutional neural network as ours but uses only one bilateral grid, and we refer to it as Arch-1. The second (Arch-2) builds a 3-scale pyramid of bilateral grids but uses a deeper neural network. Besides, we tested a 7-layer architecture (Arch-3) similar to DenseNet [HLVDMW17]. The comparison of visual quality of the three architectures on the Classroom scene is displayed in Figure 6. This shows how our approach scales to higher quality by using more complex networks. Input data are 64 *spp* noisy images and the quantitative errors are also given below denoised results. In addition, the per frame PSNR values are plotted in Figure 5, where Arch-3 provides the highest PSNR on all frame.

4.2. Auxiliary features

Our rendering system provides auxiliary features including depth, albedo and shading normals as by-products, which are readily used for our denoiser. Previous research work [BVM*17,CKS*17] verified that auxiliary features are critical assistance to improve the quality of denoised images. We further investigate the effect of noisy radiance data on the neural bilateral grid denoiser by training it with and without noisy radiance data. As presented in the closeup images of Figure 7, training including noisy radiance as input to the network effectively preserves

Table 4: *A comparison of average root-MSE values (lower is better) for evaluating our trained denoisers on 1-*spp *BMFR test data.*

Scene	root-MSE							
	NFOR	BMFR	ONND	SVGF	MR-KP(5-layer)	MR-KP(1-layer)	Ours(3-grid)	Ours(1-grid)
Classroom	0.032	0.036	0.043	0.056	**0.025**	0.027	0.027	0.027
Living room	0.027	0.032	0.053	0.043	0.025	0.031	**0.024**	0.038
San Miguel	0.081	0.090	0.098	0.116	0.072	0.076	0.066	**0.064**
Sponza	0.031	0.028	0.059	0.066	0.035	0.033	**0.022**	0.023
Sponza (glossy)	0.050	0.056	0.067	0.090	0.048	0.044	**0.033**	0.034
Sponza (mov. light)	0.081	0.145	0.077	0.142	**0.057**	0.068	0.059	0.058

Table 5: *A comparison of average VMAF values (higher is better) for evaluating our trained denoisers on 1-*spp *BMFR test data.*

Scene	VMAF							
	NFOR	BMFR	ONND	SVGF	MR-KP(5-layer)	MR-KP(1-layer)	Ours(3-grid)	Ours(1-grid)
Classroom	79.931	85.478	70.179	**96.130**	88.010	82.192	86.405	87.087
Living room	81.400	81.899	70.888	80.046	78.022	71.756	84.070	**84.706**
San Miguel	45.148	43.667	49.878	49.994	56.944	54.098	**59.539**	59.023
Sponza	84.427	**94.009**	61.838	91.614	85.090	78.51	88.404	89.888
Sponza (glossy)	61.529	69.867	73.418	**94.797**	67.746	64.967	73.851	75.899
Sponza (mov. light)	47.522	55.325	56.674	66.634	**69.142**	57.532	66.983	68.032

Table 6: *A comparison of average SMAPE values (lower is better) for evaluating our trained denoisers on 64* spp *Tungsten test data. 'Ours' denotes our denoiser with outlier removal preprocessing, whereas 'Ours wo' is without outlier.*

Scene	SMAPE					
	NFOR	BMFR	ONND	MR-KP	Ours wo	Ours
Bedroom	3.703	8.429	4.595	3.707	**3.622**	3.878
Classroom	6.484	11.344	7.449	6.350	**6.415**	6.973
Dining room	**6.868**	15.325	13.795	7.964	7.341	8.774
Kitchen	4.981	11.078	6.472	**4.849**	5.269	5.853
White Room	2.860	6.946	3.675	2.919	**2.848**	3.262

Table 7: *A comparison of average relative-MSE values (lower is better) for evaluating our trained denoisers on 64* spp *Tungsten test data. 'Ours' denotes our denoiser with outlier removal preprocessing, whereas 'Ours wo' is without outlier.*

Scene	relative-MSE					
	NFOR	BMFR	ONND	MR-KP	Ours wo	Ours
Bedroom	13.927	940.755	28.056	14.899	12.286	**10.595**
Classroom	18.194	512.290	34.399	35.166	8.683	**5.736**
Dining room	**24.679**	1703.995	122.461	534.431	26.440	32.878
Kitchen	32.837	2565.441	25.115	16.593	11.766	**9.389**
White Room	13.104	446.367	6.767	2.881	3.110	**2.174**

Table 8: *A comparison of average root-MSE values (lower is better) for evaluating our trained denoisers on 64* spp *Tungsten test data. 'Ours' denotes our denoiser with outlier removal preprocessing, whereas 'Ours wo' is without outlier.*

Scene	root-MSE					
	NFOR	BMFR	ONND	MR-KP	Ours wo	Ours
Bedroom	0.0179	0.0530	0.0190	0.0164	**0.0159**	0.0168
Classroom	0.0261	0.0583	**0.0227**	0.0236	0.0248	0.0266
Dining room	0.0155	0.0510	**0.0128**	0.0154	0.0137	0.0142
Kitchen	0.0185	0.0612	0.0183	**0.0164**	0.0168	0.0182
White Room	0.0132	0.0514	0.0149	0.0133	**0.0125**	0.0139

Table 9: *A comparison of average VMAF values (higher is better) for evaluating our trained denoisers on 64* spp *Tungsten test data. 'Ours' denotes our denoiser with outlier removal preprocessing, whereas 'Ours wo' is without outlier.*

Scene	VMAF					
	NFOR	BMFR	ONND	MR-KP	Ours wo	Ours
Bedroom	96.301	86.413	96.344	**96.765**	96.045	94.644
Classroom	93.079	57.656	**99.834**	97.751	96.022	96.530
Dining room	98.063	68.861	**99.875**	99.639	98.755	98.801
Kitchen	95.744	66.687	98.215	**98.486**	96.056	96.735
White Room	97.906	82.036	**98.653**	98.565	97.659	97.131

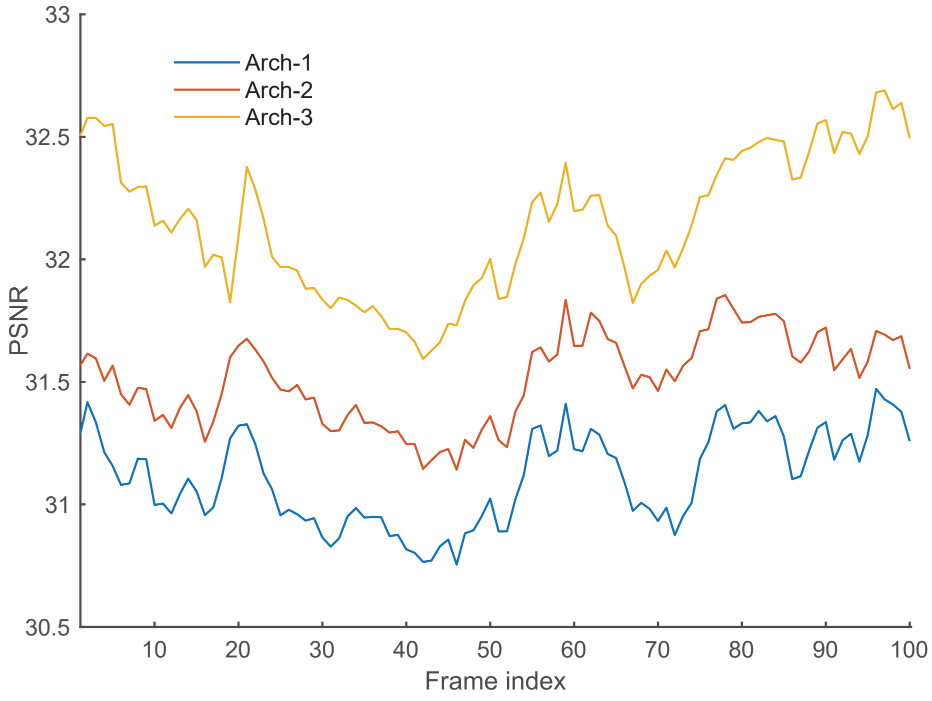

Figure 5: *Per-frame PSNR comparisons for three architectures Arch-1, Arch-2 and Arch-3 on the Classroom scene. This scene has an animated sequence of* 100 *frames.*

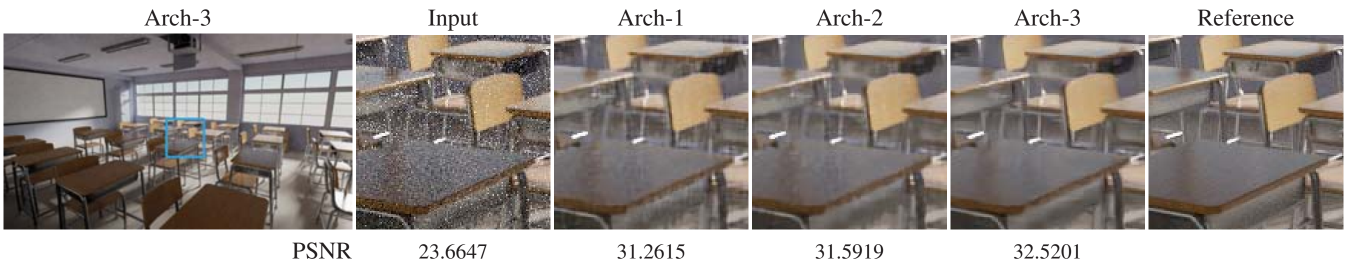

Figure 6: *Visual comparison of three architectures, including Arch-1 (2-layer 1-grid), Arch-2 (2-layer 3-grid), and Arch-3 (7-layer 3-grid) on the Classroom scene from the Tungsten dataset.*

the highlights and glossy reflection. Without the noisy radiance data, lighting effects with respect to surface materials will disappear. This is reasonable because noisy radiance data includes the interaction between lighting and materials.

4.3. Albedo removal

Our method firstly removes albedo from noisy input frames. Note that meta features including depth, normal and albedo are send to the denoisers at the same time. Finally, we multiplies albedo back to the denoised result. In Figure 8, we investigated the effect of removing albedo from the noisy radiance input. As shown in the insets, our denoised result successfully preserves the details from albedo.

4.4. Interactive viewer and multimedia material

In addition to this document, we also upload a compressed package of results with an interactive viewer and a video along with this submission.

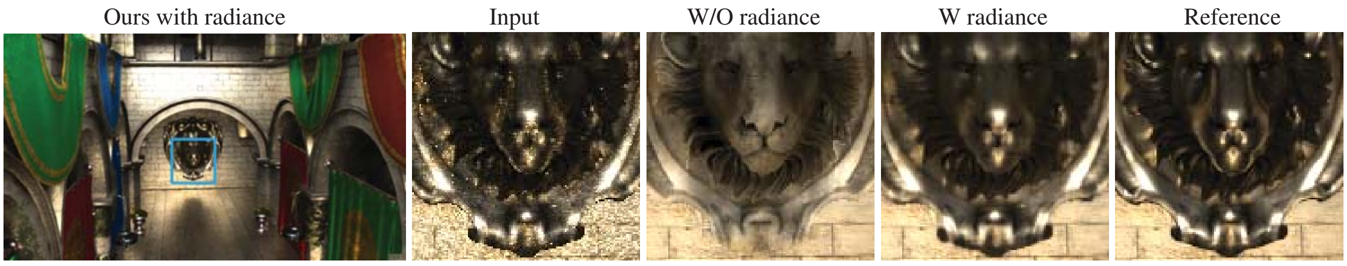

Figure 7: *Training our denoiser with and without radiance as an input channel to the network. The test data is a 1-*spp *rendered image from the Sponza scene. Including radiance as an input channel to the neural network is important to preserve illumination effects that are not captured by the other features.*

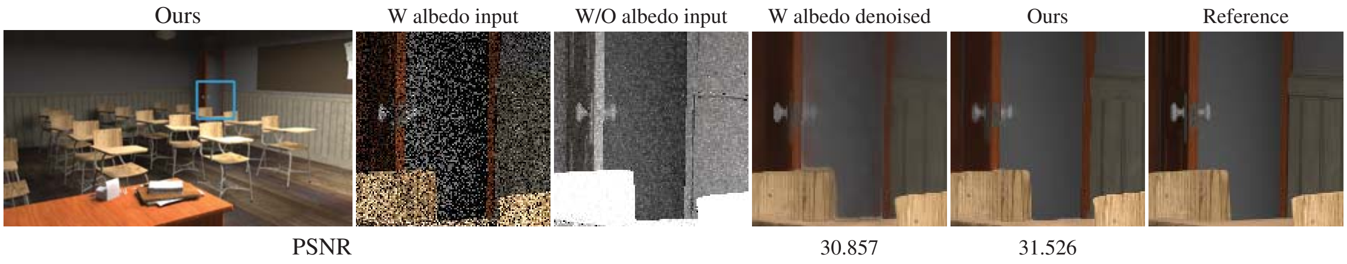

Figure 8: *Comparisons of with and without albedo removal from nooisy input radiance. Training is conducted on the 1-*spp*BMFR dataset and we show results of the Classroom scene.*

5. Derivatives of Neural Bilateral Grid

The bilateral grid construction and slicing modules have been implemented in CUDA as plug-in operators to integrate with TensorFlow. In this section, we introduce the grid construction, slicing and corresponding derivatives of the neural bilateral grid.
Table 10 lists the definitions of symbols which we will use in the following derivation.

Table 10: *Definitions of the symbols which we will use in the following derivation.*

δ_i	a scalar, the compression ratio of the W and Hdimensions. $\delta_h = \delta_w = \delta_i$.
δ_d	a scalar, the compression ratio of D dimension
ImageIn	The input image with dimension of $(W, H, 3)$
Guide	The guide image with dimension of (W, H)
Grid	The bilateral grid with dimension of $(\frac{W}{\delta_i}, \frac{H}{\delta_i}, D, 3)$
ImageOut	The output image with dimension of $(W, H, 3)$

5.1. Grid Construction

For an element in *Grid* with the coordinate of (u, v, w, c), the value of the element $Grid(u, v, w, c)$ is the weighted-sum of the pixels with a cluster of coordinates (x', y', c) in *ImageIn*.

To move the pixels or grid elements to the center between two pixels or grid elements, we use biased coordinates for both the coordinates of pixels and the coordinates of the bilateral grid (Figure 9). That is to say, we use $(u+0.5, v+0.5, w+0.5, c)$ and $(x'+0.5, y'+0.5, c)$ in the calculation of **tent** $(*,*)$. Note that the color channel c is not biased. To reduce clutter, we omit the "+0.5" in the following equations.

$Grid(u, v, w, c)$ can be computed as

$$Grid(u,v,w,c) = \frac{\sum_{x',y'} \mathbf{tent}\left(\frac{x'}{\delta_i}, u\right) \cdot \mathbf{tent}\left(\frac{y'}{\delta_i}, v\right) \cdot \mathbf{tent}\left(\frac{Guide(x',y')}{\delta_d}, w\right) \cdot ImageIn(x,y,c)}{\sum_{x',y'} \mathbf{tent}\left(\frac{x'}{\delta_i}, u\right) \cdot \mathbf{tent}\left(\frac{y'}{\delta_i}, v\right) \cdot \mathbf{tent}\left(\frac{Guide(x',y')}{\delta_d}, w\right)}, \tag{2}$$

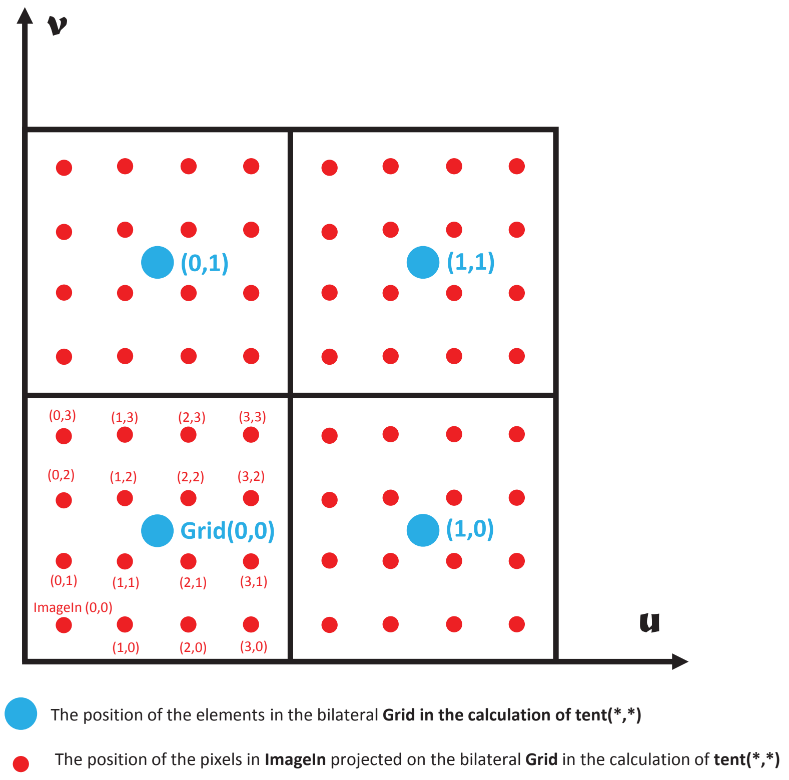
v
u
(0,1)
(1,1)
Grid(0,0)
(1,0)
(0,3) (1,3) (2,3) (3,3)
(0,2) (1,2) (2,2) (3,2)
(0,1) (1,1) (2,1) (3,1)
ImageIn (0,0) (1,0) (2,0) (3,0)
The position of the elements in the bilateral Grid in the calculation of tent(*,*)
The position of the pixels in ImageIn projected on the bilateral Grid in the calculation of tent(*,*)

Figure 9: *The example of the bilateral grid. To move the pixels / grid elements to the center between two pixels / grids, we add an offset of 0.5 for* (u,v,w) *and* (x,y).

where

$$\begin{aligned} x_{left} &\le x' < x_{left} + 2 * \delta_i \\ y_{up} &\le y' < y_{up} + 2 * \delta_i \end{aligned} . \tag{3}$$

We define **tent** (m,n) as

$$\textbf{tent}\,(m,n) = max(1.0 - |m-n|^*, 0.0) \tag{4}$$

To make the **tent** $(*,,)*$ differentiable, we define the absolute value function as:

$$|a|^* = \sqrt{a^2+\varepsilon} \tag{5}$$

where $\varepsilon = 1e-7$.

For convenience, we write:

$$\alpha = \sum_{x',y'} \textbf{tent}\left(\frac{x'}{\delta_i}, u\right) \cdot \textbf{tent}\left(\frac{y'}{\delta_i}, v\right) \cdot \textbf{tent}\left(\frac{Guide(x',y')}{\delta_d}, w\right) \cdot ImageIn(x,y,c) \tag{6}$$

$$\beta = \sum_{x',y'} \textbf{tent}\left(\frac{x'}{\delta_i}, u\right) \cdot \textbf{tent}\left(\frac{y'}{\delta_i}, v\right) \cdot \textbf{tent}\left(\frac{Guide(x',y')}{\delta_d}, w\right) \tag{7}$$

Then,

$$Grid(u,v,w,c) = \frac{\alpha}{\beta} \tag{8}$$

5.1.1. The gradient of *Grid* w.r.t. *ImageIn*

The gradient of $Grid$ $w.r.t.$ $ImageIn$ $\frac{\partial Grid}{\partial ImageIn}$ has a dimension which is the same with the $ImageIn_{W\times H\times 3}$. The back-propagated gradient $backprop$ is a matrix which has the same dimension with the $Grid$.

For each element in $\frac{\partial Grid}{\partial ImageIn}_{W\times H\times 3}$ with coordinate of (x,y,c),

$$\frac{\partial Grid}{\partial ImageIn(x,y,c)} = \sum_{u',v',w'} \frac{\partial Grid(u',v',w',c)}{\partial ImageIn(x,y,c)} \cdot backprop(u',v',w',c), \tag{9}$$

where

$$\begin{aligned} u_{left} \le u' \le u_{left}+1 \\ v_{up} \le v' \le v_{up}+1 \\ w_{front} \le w' \le w_{front}+1 \end{aligned} \tag{10}$$

with

$$u_{left} = \left\lfloor \frac{x}{\delta_i} - 0.5 \right\rfloor$$

$$v_{up} = \left\lfloor \frac{y}{\delta_i} - 0.5 \right\rfloor$$

$$w_{front} = \left\lfloor \frac{Guide(x,y)}{\delta_d} - 0.5 \right\rfloor .$$

We have

$$\frac{\partial Grid(u,v,w,c)}{\partial ImageIn(x,y,c)} = \frac{\frac{\partial \alpha}{\partial ImageIn(x,y,c)} \cdot \beta - \frac{\partial \beta}{\partial ImageIn(x,y,c)} \cdot \alpha}{\beta^2} \quad , \tag{11}$$

where

$$\begin{aligned} &\frac{\partial \alpha}{\partial ImageIn(x,y,c)} \\ &= \sum_{x',y'} \mathbf{tent}\left(\frac{x'}{\delta_i},u\right) \cdot \mathbf{tent}\left(\frac{y'}{\delta_i},v\right) \cdot \mathbf{tent}\left(\frac{Guide(x',y')}{\delta_d},w\right) \cdot \frac{\partial ImageIn(x',y',c)}{\partial ImageIn(x,y,c)} \\ &= \mathbf{tent}\left(\frac{x}{\delta_i},u\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i},v\right) \cdot \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d},w\right) \cdot \frac{\partial ImageIn(x,y,c)}{\partial ImageIn(x,y,c)} \\ &= \mathbf{tent}\left(\frac{x}{\delta_i},u\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i},v\right) \cdot \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d},w\right) \end{aligned} \tag{12}$$

$$\begin{aligned} &\frac{\partial \beta}{\partial ImageIn(x,y,c)} \\ &= \frac{\partial \left(\sum_{x',y'} \mathbf{tent}\left(\frac{x'}{\delta_i},u\right) \cdot \mathbf{tent}\left(\frac{y'}{\delta_i},v\right) \cdot \mathbf{tent}\left(\frac{Guide(x',y')}{\delta_d},w\right)\right)}{\partial ImageIn(x,y,c)} \\ &= 0 \end{aligned} \tag{13}$$

Because $\frac{\partial \beta}{\partial ImageIn(x,y,c)} = 0$, we can rewrite Equation 11 as

$$\begin{aligned} \frac{\partial Grid(u,v,w,c)}{\partial ImageIn(x,y,c)} &= \frac{\frac{\partial \alpha}{\partial ImageIn(x,y,c)} \cdot \beta - \mathbf{0} \cdot \alpha}{\beta^2} \\ &= \frac{\frac{\partial \alpha}{\partial ImageIn(x,y,c)}}{\beta} \end{aligned} \tag{14}$$

Then we can calculate each element in the gradient function $\frac{\partial Grid}{\partial ImageIn}$ (Equation 9) using Equation 7, 12, 14.

5.1.2. The gradient of *Grid* w.r.t. *Guide*

The gradient of *Grid w.r.t. Guide* $\frac{\partial Grid}{\partial Guide}$ has a size which is the same as the *Guide*. Also, the back-propagated gradient *backprop* is a matrix which has the same size as the *Grid*.

For each element in $\frac{\partial Grid}{\partial Guide}_{W\times H}$ with coordinate of (x,y),

$$\frac{\partial Grid}{\partial Guide(x,y)} = \sum_c \left(\sum_{u',v',w'} \frac{\partial Grid(u',v',w',c)}{\partial Guide(x,y)} \cdot backprop(u',v',w',c) \right), \tag{15}$$

where the range of u',v',w' is the same as that of Equation 10.

We have

$$\frac{\partial Grid(u,v,w,c)}{\partial Guide(x,y)} = \frac{\frac{\partial \alpha}{\partial Guide(x,y)} \cdot \beta - \frac{\partial \beta}{\partial Guide(x,y)} \cdot \alpha}{\beta^2} \tag{16}$$

where,

$$\begin{aligned} &\frac{\partial \alpha}{\partial Guide(x,y)} \\ &= \sum_{x',y'} \mathbf{tent}\left(\frac{x'}{\delta_i},u\right) \cdot \mathbf{tent}\left(\frac{y'}{\delta_i},v\right) \cdot \frac{\partial \mathbf{tent}\left(\frac{Guide(x',y')}{\delta_d},w\right)}{\partial Guide(x,y)} \cdot ImageIn(x',y',c) \\ &= \mathbf{tent}\left(\frac{x}{\delta_i},u\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i},v\right) \cdot \frac{\partial \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d},w\right)}{\partial Guide(x,y)} \cdot ImageIn(x,y,c) \\ &= \mathbf{tent}\left(\frac{x}{\delta_i},u\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i},v\right) \cdot \frac{\partial \mathbf{tent}(\gamma,w)\,\frac{1}{\delta_d}}{\partial \gamma} \cdot ImageIn(x,y,c) \end{aligned} \tag{17}$$

where

$$\gamma = \frac{Guide(x,y)}{\delta_d}. \tag{18}$$

$$\begin{aligned} &\frac{\partial \beta}{\partial Guide(x,y)} \\ &= \sum_{x',y'} \mathbf{tent}\left(\frac{x'}{\delta_i},u\right) \cdot \mathbf{tent}\left(\frac{y'}{\delta_i},v\right) \cdot \frac{\partial \mathbf{tent}\left(\frac{Guide(x',y')}{\delta_d},w\right)}{\partial Guide(x,y)} \\ &= \mathbf{tent}\left(\frac{x}{\delta_i},u\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i},v\right) \cdot \frac{\partial \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d},w\right)}{\partial Guide(x,y)} \\ &= \mathbf{tent}\left(\frac{x}{\delta_i},u\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i},v\right) \cdot \frac{\partial \mathbf{tent}(\gamma,w)\,\frac{1}{\delta_d}}{\partial \gamma} \end{aligned} \tag{19}$$

Then we can calculate the gradient $\frac{\partial Grid}{\partial Guide}$ (Equation 15) using Equation 6, 7, 16, 17, 19.

5.2. Grid Slicing

For a pixel in *ImageOut* with the coordinate of (x,y,c), the pixel value $ImageOut(x,y,c)$ is the weighted-sum of the non-zero elements in the bilateral grid with coordinates of (u',v',w',c) as shown in Equation 20.

$$ImageOut(x,y,c) = \frac{\sum_{u',v',w'} \mathbf{tent}\left(\frac{x}{\delta_i},u'\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i},v'\right) \cdot \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d},w'\right) \cdot Grid(u',v',w',c) \cdot \mathbf{1}\left[Grid(u',v',w',c) > 0\right]}{\sum_{u',v',w'} \mathbf{tent}\left(\frac{x}{\delta_i},u'\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i},v'\right) \cdot \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d},w'\right) \cdot \mathbf{1}\left[Grid(u',v',w',c) > 0\right]}, \tag{20}$$

where the range of (u', v', w') is the same as that of Equation 10.

In a similar way to Section 5.1, we use biased coordinate of $(u'+0.5, v'+0.5, w'+0.5, c)$ and $(x+0.5, y+0.5, c)$ in the calculation of $\mathbf{tent}(*,*)$ to move the pixel/grid elements to the center between two pixel/grid elements. Note that, we have omitted "+0.5" in the $\mathbf{tent}(*,*)$ to reduce clutter.

For convenience, we write:

$$\mu = \sum_{u',v',w'} \mathbf{tent}\left(\frac{x}{\delta_i}, u'\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i}, v'\right) \cdot \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d}, w'\right) \cdot Grid(u',v',w',c) \cdot \mathbf{1}\left[Grid(u',v',w',c) > 0\right] \tag{21}$$

$$\eta = \sum_{u',v',w'} \mathbf{tent}\left(\frac{x}{\delta_i}, u'\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i}, v'\right) \cdot \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d}, w'\right) \cdot \mathbf{1}\left[Grid(u',v',w',c) > 0\right] \tag{22}$$

Then,

$$ImageOut(x,y,c) = \frac{\mu}{\eta} \tag{23}$$

5.2.1. The gradient of *ImageOut* w.r.t. *Grid*

The gradient of *ImageOut w.r.t.* Grid $\frac{\partial ImageOut}{\partial Grid}$ has a dimension which is the same as the Grid. The back-propagated gradient *backprop* is a matrix which has the same dimension with *ImageOut*.

For each element in $\frac{\partial ImageOut}{\partial Guide}{}_{\frac{W}{\delta_i} \times \frac{H}{\delta_i} \times \frac{256}{\delta_d} \times c}$ with coordinate of (u, v, w, c),

$$\frac{\partial ImageOut}{\partial Grid(u,v,w,c)} = \sum_{x,y} \frac{\partial ImageOut(x,y,c)}{\partial Grid(u,v,w,c)} \cdot backprop(x,y,c) \tag{24}$$

The range of (x, y) is the same as that of Equation 3.

We have

$$\frac{\partial ImageOut(x,y,c)}{\partial Grid(u,v,w,c)} = \frac{\frac{\partial \mu}{\partial Grid(u,v,w,c)} \cdot \eta - \frac{\partial \eta}{\partial Grid(u,v,w,c)} \cdot \mu}{\eta^2}, \tag{25}$$

where

$$\begin{aligned}
&\frac{\partial \mu}{\partial Grid(u,v,w,c)} \\
&= \sum_{u',v',w'} \mathbf{tent}\left(\frac{x}{\delta_i}, u'\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i}, v'\right) \cdot \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d}, w'\right) \cdot \left(\frac{\partial Grid(u',v',w',c)}{\partial Grid(u,v,w,c)} \cdot \mathbf{1}\left[Grid(u',v',w',c) > 0\right] + \right. \\
&\left. Grid(u',v',w',c) \cdot \frac{\mathbf{1}\left[Grid(u',v',w',c) > 0\right]}{\partial Grid(u,v,w,c)}\right) \\
&= \mathbf{tent}\left(\frac{x}{\delta_i}, u\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i}, v\right) \cdot \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d}, w\right) \cdot \left(\frac{\partial Grid(u,v,w,c)}{\partial Grid(u,v,w,c)} \cdot \mathbf{1}\left[Grid(u,v,w,c) > 0\right] + Grid(u,v,w,c) \cdot \frac{\mathbf{1}\left[Grid(u,v,w,c) > 0\right]}{\partial Grid(u,v,w,c)}\right) \\
&= \mathbf{tent}\left(\frac{x}{\delta_i}, u\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i}, v\right) \cdot \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d}, w\right) \cdot \left(1 \cdot \mathbf{1}\left[Grid(u,v,w,c) > 0\right] + Grid(u,v,w,c) \cdot 0\right) \\
&= \mathbf{tent}\left(\frac{x}{\delta_i}, u\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i}, v\right) \cdot \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d}, w\right) \cdot \mathbf{1}\left[Grid(u,v,w,c) > 0\right],
\end{aligned} \tag{26}$$

and γ is defined in Equation 18.

$$\begin{aligned}
&\frac{\partial \eta}{\partial Grid(u,v,w,c)} \\
&= \sum_{u',v',w'} \mathbf{tent}\left(\frac{x}{\delta_i}, u'\right) \cdot \mathbf{tent}\left(\frac{y}{\delta_i}, v'\right) \cdot \mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d}, w\right) \cdot \frac{\partial \mathbf{1}\left[Grid(u',v',w',c) > 0\right]}{\partial Grid(u,v,w,c)} \\
&= 0
\end{aligned} \tag{27}$$

Because $\frac{\partial\eta}{\partial Grid(u,v,w,c)} = 0$, we can rewrite Equation 25 as

$$\begin{aligned}\frac{\partial ImageOut(x,y,c)}{\partial Grid(u,v,w,c)} &= \frac{\frac{\partial\mu}{\partial Grid(u,v,w,c)}\cdot\eta - \frac{\partial\eta}{\partial Grid(u,v,w,c)}\cdot\mu}{\eta^2} \\ &= \frac{\frac{\partial\mu}{\partial Grid(u,v,w,c)}\cdot\eta - 0\cdot\mu}{\eta^2} \\ &= \frac{\frac{\partial\mu}{\partial Grid(u,v,w,c)}}{\eta}\end{aligned} \tag{28}$$

Then we can calculate the gradient $\frac{\partial ImageOut}{\partial Grid}$ (Equation 24) using Equation 28, 26, 22.

5.2.2. The gradient of *ImageOut* w.r.t. *Guide*

The gradient of *ImageOut w.r.t.* Guide $\frac{\partial ImageOut}{\partial Guide}$ has a size which is the same as the Guide. The back-propagated gradient *backprop* is a matrix which has the same size as *ImageOut*.

For each element in $\frac{\partial ImageOut}{\partial Guide}_{W\times H}$ with coordinate of (x,y)

$$\frac{\partial ImageOut}{\partial Guide(x,y)} = \sum_c \frac{\partial ImageOut(x,y,c)}{\partial Guide(x,y)}\cdot backprop(x,y,c), \tag{29}$$

We have

$$\frac{\partial ImageOut(x,y,c)}{\partial Guide(x,y)} = \frac{\frac{d\mu}{dGuide(x,y)}\cdot\eta - \frac{d\eta}{dGuide(x,y)}\cdot\mu}{\eta^2} \tag{30}$$

where,

$$\begin{aligned}&\frac{\partial\mu}{\partial Guide(x,y)} \\ &= \sum_{u',v',w'} \mathbf{tent}\left(\frac{x}{\delta_i},u'\right)\cdot\mathbf{tent}\left(\frac{y}{\delta_i},v'\right)\cdot\frac{\partial\mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d},w'\right)}{\partial Guide(x,y)}\cdot Grid(u',v',w',c)\cdot\mathbf{1}\left[Grid(u',v',w',c)>0\right] \\ &= \sum_{u',v',w'} \mathbf{tent}\left(\frac{x}{\delta_i},u'\right)\cdot\mathbf{tent}\left(\frac{y}{\delta_i},v'\right)\cdot\frac{\partial\mathbf{tent}(\gamma,w')}{\partial\gamma}\cdot\frac{1}{\delta_d}\cdot Grid(u',v',w',c)\cdot\mathbf{1}\left[Grid(u',v',w',c)>0\right]\end{aligned} \tag{31}$$

where γ is defined in Equation 18, the range of (u',v',w') is the same with that of Equation 10.

$$\begin{aligned}&\frac{\partial\eta}{\partial Guide(x,y)} \\ &= \sum_{u',v',w'} \mathbf{tent}\left(\frac{x}{\delta_i},u'\right)\cdot\mathbf{tent}\left(\frac{y}{\delta_i},v'\right)\cdot\frac{\partial\mathbf{tent}\left(\frac{Guide(x,y)}{\delta_d},w'\right)}{\partial Guide(x,y)}\cdot\mathbf{1}\left[Grid(u',v',w',c)>0\right] \\ &= \sum_{u',v',w'} \mathbf{tent}\left(\frac{x}{\delta_i},u'\right)\cdot\mathbf{tent}\left(\frac{y}{\delta_i},v'\right)\cdot\frac{\partial\mathbf{tent}(\gamma,w')}{\partial\gamma}\cdot\frac{1}{\delta_d}\cdot\mathbf{1}\left[Grid(u',v',w',c)>0\right]\end{aligned} \tag{32}$$

Then we can calculate the gradient $\frac{\partial ImageOut}{\partial Guide}$ (Equation 29) using Equation 21, 22, 30, 31, 32.

References

[ALM*15] Aaron A., Li Z., Manohara M., Lin J. Y., Wu E. C., Kuo C. . J.: Challenges in cloud based ingest and encoding for high quality streaming media. In *2015 IEEE International Conference on Image Processing (ICIP)* (Sep. 2015), pp. 1732–1736. doi:10.1109/ICIP.2015.7351097. 3

[BVM*17] Bako S., Vogels T., McWilliams B., Meyer M., Novák J., Harvill A., Sen P., Derose T., Rousselle F.: Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 97. 1, 4

[CKS*17] CHAITANYA C. R. A., KAPLANYAN A. S., SCHIED C., SALVI M., LEFOHN A., NOWROUZEZAHRAI D., AILA T.: Interactive reconstruction of monte carlo image sequences using a recurrent denoising autoencoder. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 98. 4

[HLVDMW17] HUANG G., LIU Z., VAN DER MAATEN L., WEINBERGER K. Q.: Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 4700–4708. 4

[VRM*18] VOGELS T., ROUSSELLE F., MCWILLIAMS B., RÖTHLIN G., HARVILL A., ADLER D., MEYER M., NOVÁK J.: Denoising with kernel prediction and asymmetric loss functions. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 124. 1