Eurographics Symposium on Rendering 2021
A. Bousseau and M. McGuire
(Guest Editors)

Volume 40 (2021), Number 4

Real-time Monte Carlo Denoising with Weight Sharing Kernel Prediction Network: Supplementary Document

Hangming Fan, Rui Wang†, Yuchi Huo†, Hujun Bao

State Key Lab of CAD&CG, Zhejiang University
†Corresponding author: rwang@cad.zju.edu.cn, huo.yuchi.sc@gmail.com

1. Details of the Neural Network Structure

1.1. RepVGG Block Structure

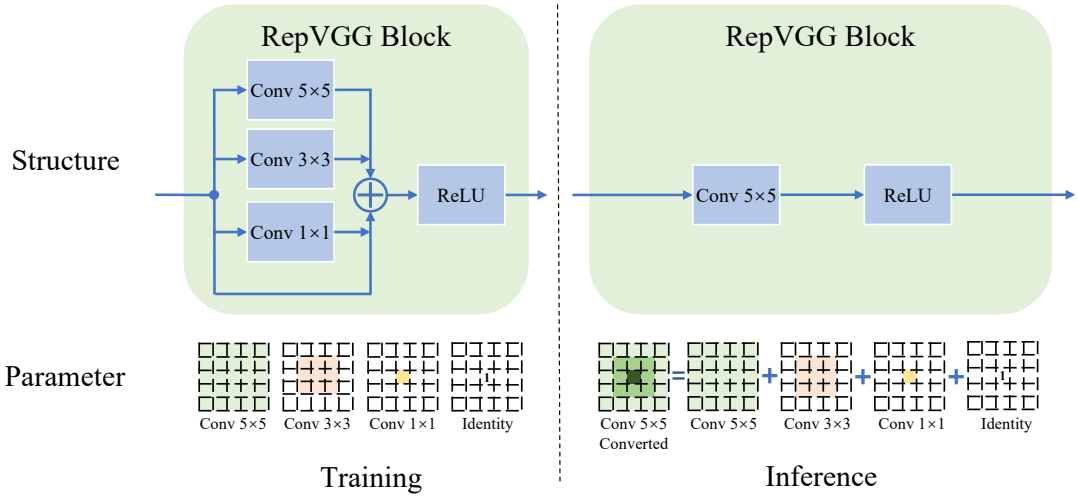

Figure 1: *Structural re-parameterization between training-time and inference-time. For the parameter-level illustration, we use single-channel input and single-channel output convolutional layer for example case, where the parameters of the* 5×5 *kernel size Conv layer can be represented as a* 5×5 *matrix (bottom left), and it is similar for* 3×3 *and* 1×1 *Conv layers after padding with zero values (presented with empty grid cell). At the inference-time, we construct one single* 5×5 *convolutional parameters with an element-wise addition (bottom right).*

For the 1-spp BMFR dataset, we build our *ImportanceNet* with the efficient *RepVGG Block* [DZM*21]. The *RepVGG Block* has different structures in training-time and inference-time. At the training-time, it has multiple branches: 1×1 Conv layer, 3×3 Conv layer, 5×5 Conv layer, and identity branch (Figure 1, top left). Since all the identity branch, 1×1 and 3×3 convolution kernel parameters can be padded with zero values to be presented as a 5×5 convolution kernel (Figure 1, bottom left), we construct a single 5×5 convolution kernel with an element-wise addition to the trained and zero-padded parameters of the branches for inference (Figure 1, bottom right), which is called the structural re-parameterization technique [DZM*21]. Consequently, the converted *RepVGG Block* structure has only one single branch compositing with a 5×5 Conv layer and a ReLU layer (Figure 1, top right), so the network architecture for inference is an efficient fully convolutional network. Note that we only add the identity branch in Conv layer where the output channel count equals the input.

The conversion of *RepVGG Block* needs only to be done once, which can be treated as an offline post-process step right after the training. Besides, *RepVGG Block* executes completely the same computations before and after the conversion, so it will not reduce the network precision.

The original *RepVGG* architecture achieves the best performance with the batch-normalization (BN) layer's nonlinear behavior in structural

Table 1: *Average relative-MSE comparison (lower is better) on 1-*spp *BMFR test data. Ours 6-layer represents the 6-layer convolutional neural network architecture and Ours 3-layer represents the 3-layer convolutional neural network architecture.*

Scene	relative-MSE								
	NFOR	BMFR	ONND	SVGF	MR-KP	KP	NBGD	Ours(6-layer)	Ours(3-layer)
Classroom	0.0568	0.0784	0.0326	0.0725	0.0117	**0.0103**	0.0180	0.0109	0.0130
Living room	0.0651	0.0736	0.0463	0.1188	**0.0068**	0.0102	0.0256	0.0093	0.0113
San Miguel	1.2393	1.7860	0.7842	0.6968	0.2406	**0.2251**	0.4723	0.2393	0.0245
Sponza	0.0565	0.0423	0.1455	0.0512	0.0100	**0.0091**	0.0128	0.0097	0.0120
Sponza (glossy)	0.2538	0.3049	0.1812	0.0954	0.0689	**0.0425**	0.0971	0.0455	0.0551
Sponza (mov. light)	0.1383	0.1983	0.1681	0.1111	0.0373	**0.0360**	0.0529	0.0433	0.0421

re-parameterization [DZM*21]. Although the variant of *RepVGG Block* we used loses this property when not including the BN layer, it still retains the over-parameterization property, which is important for our real-time application because the multi-branches structure is practically beneficial to training and this structure will not introduce additional costs to network inference.

1.2. Multi-resolution Kernel Prediction Structure

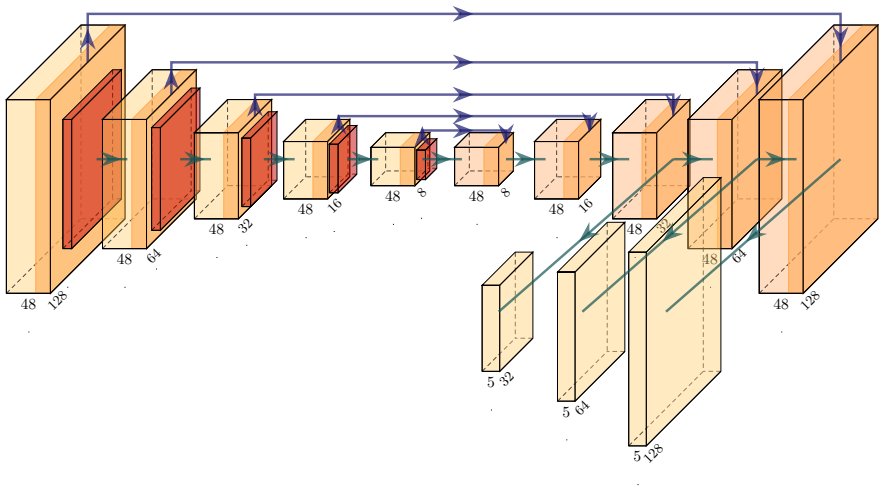

Figure 2: *Network architecture for denoising Tungsten dataset.*

For the 64-spp Tungsten dataset, we build our *ImportanceNet* with a more complex multi-resolution network architecture as shown in Figure 2. This network has a U-Net architecture, and we add three additional convolutional layers to predict our importance map at the last three resolutions.

2. Additional Evaluation Metrics on the 1-spp BMFR Dataset

2.1. Training Details

Note that temporal accumulation operation with geometry rejection strategy acts up for moving light effects [SKW*17, KIM*19], which mismatches the temporal information and introduces bias to the input image. This bias would mislead the training process for a supervised-learning method, so we remove the BMFR dataset's *sponza-moving-light* scene (static camera, changing light position) from the training data. This setting experimentally improved the reconstruction quality of ours and the previous denoisers MR-KP and NBGD [MZV*20] by giving better quantitative metrics and visual quality. Besides, we take the *sponza-moving-light* scene as test data to check our method's generalization ability to moving light effects.

Table 2: *Average RMSE comparison (lower is better) on 1-*spp *BMFR test data. Ours 6-layer represents the 6-layer convolutional neural network architecture and Ours 3-layer represents the 3-layer convolutional neural network architecture.*

Scene	RMSE								
	NFOR	BMFR	ONND	SVGF	MR-KP	KP	NBGD	Ours(6-layer)	Ours(3-layer)
Classroom	0.0321	0.0356	0.0431	0.0561	**0.0223**	**0.0223**	0.0265	0.0229	0.0245
Living room	0.0272	0.0316	0.0526	0.0435	0.0204	**0.0198**	0.0227	0.0199	0.0227
San Miguel	0.0813	0.0895	0.0982	0.1160	0.0629	0.0617	0.0644	**0.0614**	0.0644
Sponza	0.0307	0.0282	0.0591	0.0661	**0.0186**	0.0189	0.0207	0.0189	0.0207
Sponza (glossy)	0.0504	0.0564	0.0671	0.0900	**0.0285**	0.0292	0.0318	0.0289	0.0318
Sponza (mov. light)	0.0811	0.1450	0.0773	0.1418	0.0556	0.0556	0.0572	**0.0552**	0.0572

Table 3: *Average SMAPE comparison (lower is better) on 1-*spp *BMFR test data. Ours 6-layer represents the 6-layer convolutional neural network architecture and Ours 3-layer represents the 3-layer convolutional neural network architecture.*

Scene	SMAPE								
	NFOR	BMFR	ONND	SVGF	MR-KP	KP	NBGD	Ours(6-layer)	Ours(3-layer)
Classroom	0.0289	0.0261	0.0528	0.0405	0.0177	**0.0176**	0.0206	0.0185	0.0203
Living room	0.0201	0.0182	0.0418	0.0220	0.0124	0.0124	0.0140	**0.0118**	0.0137
San Miguel	0.1172	0.1160	0.1425	0.1278	0.1037	0.1086	**0.0982**	0.1052	0.1106
Sponza	0.0377	0.0314	0.0715	0.0530	**0.0180**	0.0183	0.0190	0.0183	0.0194
Sponza (glossy)	0.0770	0.0730	0.0966	0.0759	**0.0374**	0.0393	0.0442	0.0387	0.0425
Sponza (mov. light)	0.1012	0.1492	0.0882	0.1408	**0.0556**	0.0578	0.0593	0.0563	0.0580

Table 4: *Average VMAF comparison (higher is better) on 1-*spp *BMFR test data. Ours 6-layer represents the 6-layer convolutional neural network architecture and Ours 3-layer represents the 3-layer convolutional neural network architecture.*

Scene	VMAF								
	NFOR	BMFR	ONND	SVGF	MR-KP	KP	NBGD	Ours(6-layer)	Ours(3-layer)
Classroom	79.815	85.333	70.147	**96.095**	90.953	89.280	85.412	88.105	85.879
Living room	81.285	81.735	70.842	79.989	83.268	85.050	**85.866**	83.726	78.283
San Miguel	45.080	43.596	49.803	49.840	59.909	**60.295**	60.212	60.021	58.689
Sponza	84.373	93.934	61.799	91.555	**94.302**	92.875	90.410	91.759	88.536
Sponza (glossy)	61.477	69.807	73.334	**94.793**	84.422	79.906	76.839	82.292	77.363
Sponza (mov. light)	47.552	55.348	56.693	66.840	**75.414**	70.732	69.131	70.385	65.630

Table 5: *Error metrics comparisons on 1-spp BMFR test scenes to evaluate the effectiveness of our kernel fusion module. KP refers to the basic kernel prediction method, and KP-fusion refers to the KP extended with our kernel fusion module. Ours refers to the complete architecture described in our paper, and Ours-same-size refers to our architecture fusing* 6 *kernels with the same filtering size* $k_i = 13$.

Scene	PSNR				SSIM			
	KP	KP-fusion	Ours-same-size	Ours	KP	KP-fusion	Ours-same-size	Ours
Classroom	33.047	**33.308**	32.572	32.827	0.978	**0.979**	0.976	0.977
Living room	34.090	**34.506**	33.548	34.063	0.978	**0.980**	0.977	0.979
San Miguel	24.215	**24.348**	23.946	24.269	0.851	**0.857**	0.846	0.849
Sponza	34.595	**35.068**	34.319	34.600	0.982	0.983	0.983	0.983
Sponza (glossy)	30.719	**30.739**	30.385	30.805	0.960	0.960	0.959	**0.961**
Sponza (mov. light)	25.324	**25.424**	35.259	25.374	0.958	**0.960**	0.958	0.958

Table 6: *PSNR and SSIM comparison of KP variants for each test scene. KP-1 represents a KPCN variant with two layers CNN and filtering kernel size 13. KP-2 represents a KPCN variant with five layers CNN and filtering kernel size 7.*

Scene	PSNR			SSIM		
	KP1	KP2	Ours	KP1	KP2	Ours
Classroom	33.047	32.522	**32.827**	0.972	0.974	**0.977**
Living room	34.090	33.032	**34.063**	0.968	0.972	**0.979**
San Miguel	23.871	24.160	**24.269**	0.837	0.841	**0.849**
Sponza	33.776	33.780	**34.600**	0.976	0.980	**0.983**
Sponza (glossy)	29.653	30.202	**30.805**	0.948	0.957	**0.961**
Sponza (mov. light)	24.998	25.156	**25.374**	0.945	0.950	**0.958**

Table 7: *Error metrics comparisons on 64-spp Tungsten test scenes of Bedroom, Classroom, and Living room. We use the multi-resolution neural network architecture with 2 fused kernels each level. Compared with these methods, ours achieves comparable quality.*

Scene	PSNR						SSIM					
	NFOR	ONND	MR-KP	KP	NBGD-7	Ours(MR)	NFOR	ONND	MR-KP	KP	NBGD-7	Ours (MR)
Bedroom	35.05	34.44	36.32	**36.36**	35.98	36.34	0.973	0.971	0.975	0.976	0.974	**0.977**
Classroom	31.67	32.87	**32.99**	32.89	32.12	32.82	0.940	0.949	0.950	**0.951**	0.942	0.949
Living room	37.63	36.60	38.02	38.46	38.08	**38.53**	0.977	0.973	0.978	**0.979**	0.977	**0.979**

2.2. Evaluation Metrics Comparison

We also compute additional evaluation metrics, including relative mean square error (relative-MSE), root mean square error (RMSE), symmetric mean absolute percentage error (SMAPE), and Video Multi-Method Assessment Fusion (VMAF) [ALM*15]. As shown in Table 1, Table 2, and Table 3, our method is superior in most of the pixel-wise error metrics. The VMAF scores in Table 4 show that we achieve similar temporal stability to the state-of-the-art real-time neural denoiser.

2.3. Metrics comparison of kernel fusion module evaluation

In theory, the network can use multiple kernels of equal sizes in the fusion module. We experimentally checked this by fusing 6 kernels with the same size $k_i = 13$. While the result in Table 5 shows this variant performs worse than the configuration of fusing with different kernel sizes, and our analysis is that fusing with different sizes is an explicit and helpful constrain about the noise frequency for the training, which is similar to why the layer-based denoiser [MH20] performs better with an ordered alpha-blending than a direct weighted average. Besides, we also extended the basic kernel prediction method with our kernel fusion module to see how much our importance map affects the result and further check the effectiveness of our kernel fusion module, which is presented in Table 5.

2.4. Additional metrics comparison with Kernel Prediction Variants

To further compare our method and the basic kernel prediction method, we design another two architecture variants of KPCN: a 2-layer network with filtering kernel size 13 (KP-1) and a 5-layer network with filtering kernel size 7 (KP-2). We show the metrics comparison in Table 6. The results show that our method achieves the best quantitative quality because it maintains both a deep network and a large filtering size.

3. Additional Comparison on the 64-spp Tungsten Dataset

For the 64-spp Tungsten dataset, we use a three-resolution architecture, and for each resolution we construct and fuse two filtering kernels with sizes 3 and 5. The visual comparisons in Figure 3 show that our method can generate more smooth glossy reflections and soft shadows and produce fewer artifacts than NBGD and MR-KP. The PSNR and SSIM of these three scenes are presented in Table 7. We compute the average numerical error metrics over 100 consecutive frames and present the results in Table 8, Table 9, Table 10, Table 11. The comparison results show that our method has a comparable denoising ability for the high spp input in both pixel-wise error metrics and temporal stability.

References

[ALM*15] AARON A., LI Z., MANOHARA M., LIN J. Y., WU E. C.-H., KUO C.-C. J.: Challenges in cloud based ingest and encoding for high quality streaming media. In *2015 IEEE International Conference on Image Processing (ICIP)* (2015), IEEE Press, p. 1732–1736. URL: https://doi.org/10.1109/ICIP.2015.7351097, doi:10.1109/ICIP.2015.7351097. 4

Table 8: *Average relative-MSE comparison (lower is better) on 64-spp Tungsten test data. We use the multi-resolution neural network architecture with 2 fused kernels each level.*

Scene	relative-MSE					
	NFOR	ONND	MR-KP	KP	NBGD-7	Ours
bedroom	0.0335	0.0423	0.0076	**0.0072**	0.0258	0.0092
classroom	0.0450	0.0609	0.0151	0.0139	0.0290	**0.0137**
dining-room	0.0537	0.1847	0.0420	0.0439	0.0483	**0.0260**
kitchen	0.0636	0.0455	0.0148	**0.0133**	0.0293	0.0193
living-room	0.0262	0.0133	0.0041	**0.0039**	0.0089	0.0043

Table 9: *Average RMSE comparison (lower is better) on 64-spp Tungsten test data. We use the multi-resolution neural network architecture with 2 fused kernels each level.*

Scene	RMSE					
	NFOR	ONND	MR-KP	KP	NBGD-7	Ours
bedroom	0.0179	0.0190	0.0157	0.0155	0.0159	**0.0154**
classroom	0.0261	**0.0227**	0.0230	0.0233	0.0248	0.0234
dining-room	0.0155	**0.0128**	0.0133	**0.0128**	0.0137	0.0131
kitchen	0.0185	0.0183	**0.0159**	0.0165	0.0168	0.0167
living-room	0.0132	0.0149	0.0129	0.0126	0.0125	**0.0122**

Table 10: *Average SMAPE comparison (lower is better) on 64-spp Tungsten test data. We use the multi-resolution neural network architecture with 2 fused kernels each level.*

Scene	SMAPE					
	NFOR	ONND	MR-KP	KP	NBGD-7	Ours
bedroom	0.0162	0.0194	**0.0146**	0.0150	0.0158	0.0150
classroom	0.0301	0.0321	**0.0280**	0.0281	0.0299	0.0284
dining-room	**0.0252**	0.0467	0.0294	0.0287	0.0268	0.0274
kitchen	0.0215	0.0257	**0.0202**	0.0209	0.0223	0.0214
living-room	0.0126	0.0149	**0.0116**	0.0123	0.0124	0.0117

Table 11: *Average VMAF comparison (higher is better) on 64-spp Tungsten test data. We use the multi-resolution neural network architecture with 2 fused kernels each level.*

Scene	VFAM					
	NFOR	ONND	MR-KP	KP	NBGD-7	Ours
bedroom	96.31	96.34	98.32	95.29	96.06	**98.47**
classroom	93.09	**99.84**	98.87	97.94	96.05	97.66
dining-room	98.04	**99.87**	99.86	**99.87**	98.74	99.17
kitchen	95.70	98.19	**99.04**	97.35	96.06	98.91
living-room	97.90	98.64	**99.04**	98.21	97.63	98.86

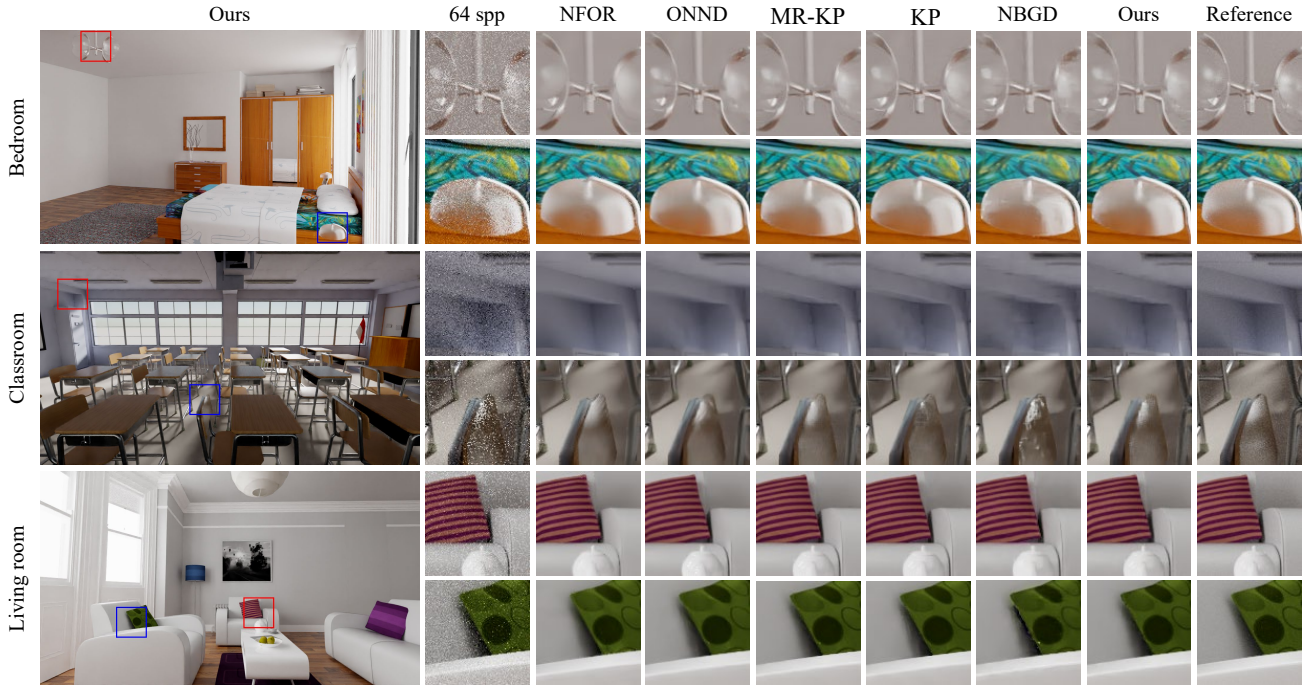

Figure 3: *Visual comparisons of denoising quality on the 64-*spp *Tungsten test scenes of Bedroom, Classroom, and Living room. We use the multi-resolution neural network architecture with 2 fused kernels each resolution.*

[DZM*21] DING X.-H., ZHANG X.-Y., MA N.-N., HAN J.-G., DING G., SUN J.: Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2021). 1, 2

[KIM*19] KOSKELA M., IMMONEN K., MÄKITALO M., FOI A., VIITANEN T., JÄÄSKELÄINEN P., KULTALA H., TAKALA J.: Blockwise multi-order feature regression for real-time path-tracing reconstruction. *ACM Trans. Graph. 38*, 5 (June 2019). URL: https://doi.org/10.1145/3269978, doi:10.1145/3269978. 2

[MH20] MUNKBERG J., HASSELGREN J.: Neural denoising with layer embeddings. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 1–12. 4

[MZV*20] MENG X., ZHENG Q., VARSHNEY A., SINGH G., ZWICKER M.: Real-time Monte Carlo Denoising with the Neural Bilateral Grid. In *Eurographics Symposium on Rendering - DL-only Track* (2020), Dachsbacher C., Pharr M., (Eds.), The Eurographics Association. doi:10.2312/sr.20201133. 2

[SKW*17] SCHIED C., KAPLANYAN A., WYMAN C., PATNEY A., CHAITANYA C. R. A., BURGESS J., LIU S., DACHSBACHER C., LEFOHN A., SALVI M.: Spatiotemporal variance-guided filtering: real-time reconstruction for path-traced global illumination. In *Proceedings of High Performance Graphics*. 2017, pp. 1–12. 2