IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 January 2024, accepted 20 February 2024, date of publication 23 February 2024, date of current version 29 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3369588

RESEARCH ARTICLE

Real-Time Monte Carlo Denoising With Adaptive Fusion Network

JUNMIN LEE, SEUNGHYUN LEE, MIN YOON, (Associate Member, IEEE), AND BYUNG CHEOL SONG, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Inha University, Incheon 22212, Republic of Korea

Corresponding author: Byung Cheol Song (bcsong@inha.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by Korean Government [Ministry of Science and ICT (MSIT)] under Grant 2022R1A2C2010095 and Grant 2022R1A4A1033549, and in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by Korean Government (MSIT) [Artificial Intelligence (AI) Convergence Innovation Human Resources Development (Inha University)] (Development of High-definition CIS and High-speed DVS enabled AI System on Chip (SoC) for Object and Motion Recognition) under Grant RS-2022-00155915 and Grant 2022-0-00955.

ABSTRACT Real-time Monte Carlo denoising aims to denoise a 1spp-rendered image with a limited time budget. Many latest techniques for real-time Monte Carlo denoising utilize temporal accumulation (TA) as a pre-processing to improve the temporal stability of successive frames and increase the effective spp. However, existing techniques using TA used to suffer from significant performance degradation when TA does not work well. In addition, they have the disadvantage of deteriorating performance in dynamic scenes because pixel information of the current frame cannot be sufficiently utilized due to the pixel averaging effect between temporally adjacent frames. To solve this problem, this paper proposes a framework that utilizes both 1spp images and temporally accumulated 1spp (TA-1spp) images. First, the multi-scale kernel prediction module estimates kernel maps for filtering 1spp images and TA-1spp images, respectively. Then, the filtered images are properly fused so that the two advantages of 1spp and TA-1spp images can create synergy. Also, the remaining noise is removed through the refinement module and fine details are reconstructed to improve the model flexibility, beyond using only the kernel prediction module. As a result, we achieve better quantitative and qualitative performance at 39% faster than state-of-the-art (SOTA) real-time Monte Carlo denoisers.

INDEX TERMS Image processing, rendering, real-time de-noising.

I. INTRODUCTION

Ray tracing [1] is a typical global illumination algorithm for rendering realistic graphic images. In particular, Monte Carlo path tracing is a de facto standard in film and game production [2]. In order to obtain a high-quality image through Monte Carlo path tracing, a significantly large number of samples per pixel (spp) is required, which causes a huge computational cost. In other words, obtaining high-quality images in real time through Monte Carlo path tracing is still considered a difficult goal to achieve. There have been many challenges in terms of hardware and software to achieve this goal. First, with the rapid development of semiconductor technology, a lot of hardware-based techniques for ray tracing on mobile CPUs and GPUs have been developed [3], [4]. However, as the refresh rate and resolution of displays rapidly increase, the resolution of graphic images to be generated also increases. This still remains a problem for real-time rendering [5]. So, the practical spp budget for real-time applications must be very limited.

As a solution to this, an approach that renders noisy images generated with extremely low spp, e.g., 1-4 spp in real time, and then applies denoising afterward, has emerged. Since this approach can be implemented as a post-processing without major changes to existing renderers, it is attracting attention not only from industry but also from academia. If a low spp is applied for real-time rendering, the tracing time can be noticeably reduced. However, the intensity of noise is

The associate editor coordinating the review of this manuscript and approving it for publication was Andrea F. Abate.

rather strong, so it becomes difficult to restore the rendered images.

Convolutional neural networks (CNNs) can be applied to remove noise effectively [5], [6], [7]. But, the application of large-sized CNNs makes real-time rendering impossible again. Therefore, some methods proposed to compensate for insufficient capacity by using small CNNs with strong inductive bias. For example, Meng et al. [2] proposed a so-called neural bilateral grid denoiser that removes noise in bilateral grid space by grafting a differentiable bilateral grid to CNN architecture. Fan et al. [8] proposed a network capable of real-time denoising by predicting the kernel map of a single channel, reducing the overhead caused by kernel prediction. In addition, they compensated for the lack of performance by increasing the effective spp of low spp images and improving temporal stability through temporal accumulation (TA). However, this method still has room for improvement in terms of performance and run-time. Furthermore, in scenes with low temporal consistency, that is, in cases where the light source changes rapidly, e.g., fast animation, geometric changes and etc, TA may not respond well to the rapid changes due to the pixel smoothing between adjacent frames. In other words, TA may adversely affect performance for dynamic scenes.

To solve the above-mentioned problem, we propose a novel denoising framework that utilizes both 1spp images and temporally accumulated images (TA-1spp). The proposed framework uses the advantages of TA-1spp, i.e., the increase in effective spp and the improvement in temporal stability, and at the same time, increases the overall performance by directly utilizing current pixel information even in scenes with low temporal consistency. First, to effectively use 1spp and TA-1spp images simultaneously, we adopt a light-weight kernel prediction method [6]. Here, kernel prediction is to predict the weights of pixels in the $k \times k$ kernel through CNNs for effective restoration. Due to the straightforward arrangement of the projected kernel, which involves a weighted sum of adjacent pixels utilizing an optimized configuration centered around the main pixel, rapid and effective denoising becomes feasible.

Second, we propose a method to effectively filter and fuse the 1spp and TA-1spp images by adopting the multi-scale structure proposed by Vogels et al. [9]. Since the 1spp image is more sparse than the TA-1spp image, a large receptive field is required. So, 1spp images are filtered on a small scale, and TA-1spp images having a high effective spp are filtered on a large scale. The filtered 1spp and TA-1spp images are input to a fusion module like unsharp masking. Then, a post-processing module, i.e., a refinement module, is applied to remove the survived noise in the fused image. Here, the refinement module pursuing residue-based prediction can remove residual noise from fused images even with low capacity, and also restore fine details.

Finally, by cascading the kernel prediction module and the refinement module, the lack of flexibility of the kernel prediction dependent on neighboring pixels is mitigated. As a result, effective denoising is realized. According to the experimental results, the proposed method not only shows better visual quality than SOTA methods, but also quantitatively provides 39% faster speed with PSNR as high as 1.843dB. The contribution points of this paper are summarized as follows.

- We propose the first framework that utilizes both 1spp and TA-1spp images for Monte Carlo denoising. This framework not only increases the effectiveness of TA by improving the visual quality of the early frames, but is also robust for scenes with rapid light source changes.
- The model flexibility is increased by combining the kernel prediction module and the refinement module, so 1spp path-traced images are denoised in real time.
- As a result, it shows better qualitative and quantitative performance with 39% faster speed than SOTA methods for real-time Monte Carlo denoising.

II. RELATED WORKS

This section mainly describes deep learning and real-time techniques, which are closely related to the proposed denoising algorithm. Sections II-A and II-B focus on a kernel-based and real-time approach similar to the proposed method, and Section II-C describes other deep learning-based approaches not covered in the Section II-A. Here, the research trend on deep learning-based Monte Carlo denoising is described by referring to Huo and Yoon [10].

A. DEEP LEARNING-BASED MONTE CARLO DENOISING

Kalantari et al. [11] introduced supervised learning for the first time in Monte Carlo denoising. They observed the complex relationship between the input noisy image and the ideal filter parameters, and based on this observation, predicted the optimal weights of a bilateral filter through a multi-layer perceptron. In addition, various auxiliary features such as world positions, shading normals, and texture values were used as input to obtain high-quality images. Since then, as CNNs have succeeded in solving many computer vision and graphics problems, new Monte Carlo denoising structures using CNNs and noise-free auxiliary feature buffers have been studied more and more.

Bako et al. [6] proposed a CNN model with deep depth to predict the kernel weights per pixel, and succeeded in producing a complex yet generalized kernel. However, estimating a kernel of such a large size takes a lot of computation time and memory consumption. Conversely, using a smaller kernel size results in lower quality. In other words, this trade-off issue is still unsolved. To address this problem, Vogels et al. [9] noted that filtering with a small kernel at reduced resolution is almost equivalent to filtering with a large receptive field at the original resolution, and proposed the kernel prediction of a multi-resolution architecture (MR-KP). However, MR-KP targets an image rendered with at least 16spp, and is not suitable for real-time Monte Carlo denoising as it was initially designed for offline applications. Back et al. [12] initially predicted kernels using path traced images, images denoised with existing denoisers [13], [14], and auxiliary features. Then, they predicted a combination kernel using the two images (path traced image and denoised image) and the predicted kernel, and combined the two images. Zheng et al. [15] introduced an optimization-based technique that combines multiple individual Monte Carlo denoisers. In other words, the output images from various denoisers are weighted and summed on a per-pixel basis. What Back et al. [12], Zheng et al. [15], and the proposed approach have in common is that they convert two or more images into a weighted sum. However, the combination kernel and ensemble-based denoiser leverages existing denoisers for better denoising results. Therefore, they are unsuitable for real-time applications. On the other hand, the proposed approach achieves the fusion of 1ssp and TA-1spp within a single denoiser. Notably, in contrast to conventional methods, the proposed method accomplishes real-time denoising with improved performance at a marginal additional cost.

B. REAL-TIME MONTE CARLO DENOISING

Chaitanya et al. [5] first proposed an autoencoder (Optix Neural Network denoiser, ONND) for denoising an image rendered with 1spp. They directly predicted a pixel instead of a kernel for noise removal, and added a recurrent connection to the autoencoder structure to improve temporal stability. Işik et al. [7] proposed a novel filtering that uses pairwise affinity of per-pixel deep features learned from the raw path-tracing samples to learn iteratively-applied 2D dilated kernels. And they improved the temporal stability by using a temporal aggregation mechanism based on the same pairwise affinity. The previous methods have achieved up to interactive speed, but are rather slow to be called real-time yet. Schied et al. [16] used an extended hierarchical filter (Spatio-temporal Variance-Guided Filtering; SVGF) with a customized edge-stopping function to progressively filter out TA frames. Koskela et al. [17] applied augmented QR factorization and stochastic normalization to image blocks for block-wise feature regression (BMFR). This helps improve speed in GPU implementation. However, the above techniques [16], [17] must depend on reprojected frame accumulation [18] to get a higher effective spp. On the other hand, some techniques utilize TA as a data pre-processing. For instance, Meng et al. [2] proposed a neural bilateral grid denoiser (NBGD) that applied a differential bilateral grid to CNNs. They used a trained mapping function to collect adjacent pixels in 3D bilateral gird space and then removed noise on 3D space in real time. However, bilateral grid-based approach has the disadvantage of higher computational overhead than the kernel-based one [8]. Fan et al. [8] employed a kernel prediction network that is called Weight Sharing Kernel Prediction Network (WSKPN). In order to reduce the overhead incurred in per-pixel kernel prediction, they encoded a single-channel kernel map, that is, predicted a weight sharing kernel map, and then unfolded the predicted single-channel kernel map to construct a kernel. Finally, they realized real-time denoising of 1spp images. In addition, to shorten the run-time during inference, a re-parameterization technique [19] was used.

Since TA of 1spp noisy images has the effect of increasing the effective spp, it can be a solution for real-time denoising of 1spp images. However, as described in Section V, we face two problems: Visual quality deterioration in the early frames when only TA-1spp is used, and color distortion due to pixel averaging phenomenon in dynamic scenes with low temporal consistency. Therefore, we propose a novel solution which jointly utilizes pure 1spp images and TA-1spp images.

C. OTHER DEEP LEARNING-BASED TECHNIQUES

Unlike typical pixel-based Monte Carlo denoisers, Gharbi et al. [20] proposed a sample-based kernel splatting network. By estimating the contribution of Monte Carlo samples through a kernel splatting structure, it showed more natural and robust performance than pixel-based techniques when denoising images with specific visual effects (e.g., motion blur, depth of field). Munkberg and Hasselgren [21] extended Gharbi et al. [20]. They splatted samples into multiple layers to convert sample-based to layer-based, and then denoised the samples-splatted layers. This technique maintains similar quality to previous per-sample techniques even while using a fraction of the computational cost and memory requirements. Hasselgren et al. [22] adopted the multi-scale kernel prediction structure such as MR-KP [9] to denoise adaptively re-sampled Monte Carlo images and achieved interactive speed. However, these sample-based denoising techniques not only increase memory cost linearly in proportion to the number of samples (or samples-splatted layers), but also have a disadvantage that is less accessible than pixel-based denoisers.

In addition, Xu et al. [23] applied adversarial learning, and proposed a novel conditioned auxiliary feature modulation method to better utilize auxiliary features. Firmino et al. [24] proposed a progressive denoising technique with Stein‘s unbiased risk estimate (SURE). This allows denoising to be used only when it is beneficial and to reduce the effect at large sample numbers.

III. PROPOSED METHOD

This section describes the overall structure of the proposed denoiser with a data pre-processing. As in Figure 1, the proposed method consists of three modules: Multi-scale kernel prediction, filtering and fusion, and refinement. The multi-kernel prediction module receives 1spp images, TA-1spp images, and auxiliary features, and then predicts kernel maps for filtering 1spp images and TA-1spp images, respectively. The filtering and fusion module filters the 1spp image and the TA-1spp image by multiplying the predicted kernel map and the unfolded noisy image, and then fuses them to combine the advantages of the two rendered images.

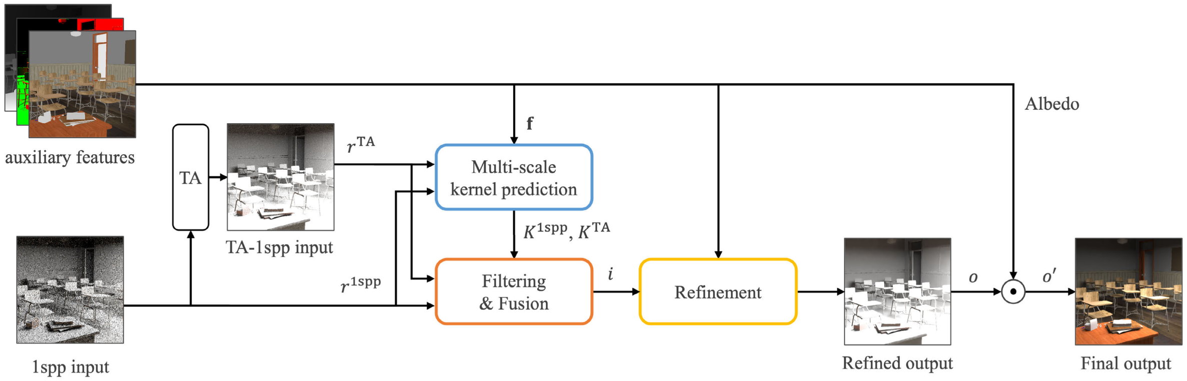

FIGURE 1. The overall framework of the proposed denoiser.

Finally, the refinement module removes noise remaining after fusion, and restores fine details.

A. PRE-PROCESSING

Prior to denoising, we demodulate albedos from noisy images as in Chaitanya et al. [5]. Demodulated images with most complex textures removed enable efficient learning. Note that we can modulate the albedo again to bring out texture details without over-blurring. Next, TA-1spp images are generated by applying TA as in [2], [8], and [17]. At this time, stable learning may be difficult if a rendered image with a high dynamic range is directly input into neural networks [6], [25]. So, finally, we transform the rendered image into a low dynamic range image through tone-mapping prior to denoising. Here, shading normals and depth are scaled linearly in the [0, 1] range.

B. MULTI-SCALE KERNEL PREDICTION

The multi-scale kernel prediction module predicts a per-pixel kernel, i.e., a kernel map to effectively filter 1spp images and TA-1spp images based on multi-scale architecture [9], as in Figure 2 (a). Multi-scale architecture $\mathcal{M}$ has a light-weight autoencoder structure for real-time operation. $\mathcal{M}$ receives 1spp image $r^{1\text{spp}}$, TA-1spp image r^{TA}, and auxiliary features $\mathbf{f}$ such as albedo, shading normal, and depth, and predicts two kernel maps $K^{1\text{spp}}$ and K^{TA} for $r^{1\text{spp}}$ and r^{TA} respectively:

$$K^{1\text{spp}}, K^{\text{TA}} = \mathcal{M}(r^{1\text{spp}}, r^{\text{TA}}, \mathbf{f}) \tag{1}$$

$K^{1\text{spp}}$ and K^{TA} are obtained by slicing feature maps of two different scales of the decoder. Slicing is performed according to the desired kernel size. In this paper, the kernel size k is set to 7 for large receptive fields, in other words, 49 channels are used. Since $r^{1\text{spp}}$ has more sparse pixels than r^{TA}, we can increase the effective spp by defining a smaller kernel map as $K^{1\text{spp}}$ and applying $K^{1\text{spp}}$ to the down-sampled $r^{1\text{spp}}$. On the other hand, since r^{TA} has higher effective spp, K^{TA} is used for filtering at the original scale of r^{TA}.

In summary, the proposed multi-scale kernel prediction efficiently estimates the kernel map for denoising each rendered image, and the estimated kernel maps are applied at a scale with a high effective spp to enable high-performance denoising.

C. FILTERING AND FUSION

The next module filters and fuses 1spp images and TA-1spp images with kernel maps predicted in the multi-scale kernel prediction step. As described above, since the kernel map K is implemented in the form of a $k \times k$ channel feature map, the rendered image needs to be unfolded, which is represented conceptually in Figure 2 (b). Here, k is set to 7. The process of applying K to the unfolded rendered image is implemented in a weighted sum way, and the filtering process for each color channel c is expressed by

$$d_c = \sum_{j}^{k\times k} K_j \cdot \text{Unfold}(r_c) \tag{2}$$

Note that we filter the non-tone-mapped HDR image to retain the original HDR distribution here.

Next, in order to fuse the filtered 1spp image $d^{1\text{spp}}$ and the filtered TA-1spp image d^{TA}, we appropriately transform the scale-compositor module proposed by Vogels et al. [9] and adopt it as a fusion module $\mathcal{F}$. The structure of $\mathcal{F}$ is given in Figure 2 (c), and the fused image i is obtained by

$$i = \mathcal{F}(d^{1\text{spp}}, d^{\text{TA}}) = d^{\text{TA}} - \alpha\mathbf{UD}d^{\text{TA}} + \alpha\mathbf{U}d^{1\text{spp}} \tag{3}$$

where α is the weight map of adaptive fusion estimated by CNN. $\mathbf{D}$ and $\mathbf{U}$ denote the 2× down- and up-sampling operators, which are implemented as average pooling and nearest-neighbor interpolation, respectively.

The proposed fusion framework effectively combines the advantages of two rendered images by injecting the dynamic features of the 1spp image into the TA-1spp image. How much performance is improved by our fusion framework is demonstrated in Section V-A. As a result, high restoration performance is achieved even with a small CNN having minimal capacity. This means that we break through the cost-performance trade-off of existing Monte Carlo denoising algorithms.

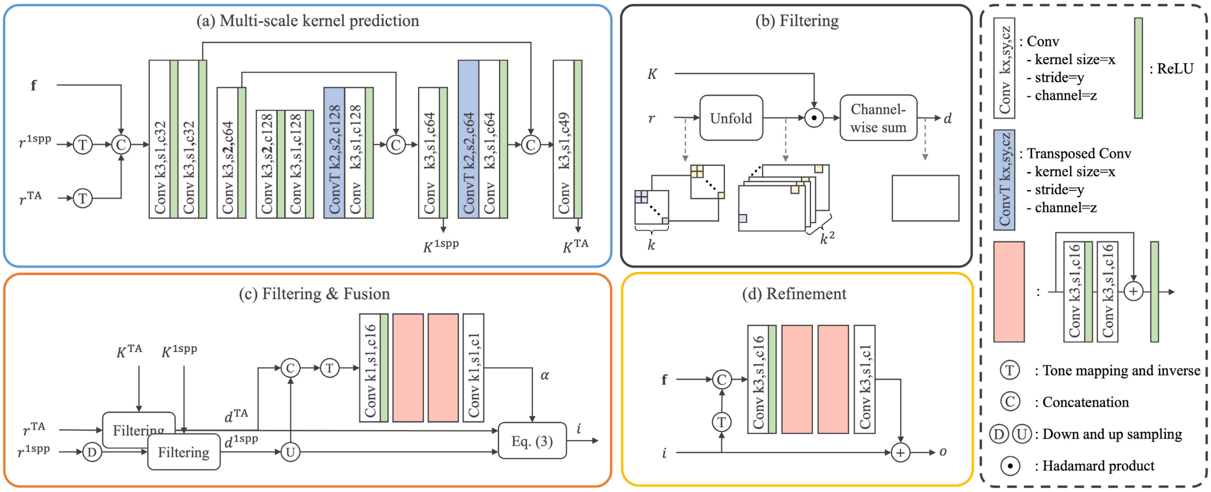

FIGURE 2. Details of each module of the proposed denoiser.

D. REFINEMENT

Although the proposed filtering and fusion module significantly removes the noise of the low-spp rendered image, there is still a limit to effectively restoring even high-frequency components such as fine details. Therefore, we overcome this limitation with the refinement module of Figure 2 (d).

The refinement module $\mathcal{R}$ receives the fused image i and auxiliary features $\mathbf{f}$ as input and outputs the final refined image o:

$$o = i + \mathcal{R}(i, \mathbf{f}) \tag{4}$$

$\mathcal{R}$ enhances fine details by removing noise that still survives after kernel filtering with low capacity through residual learning. In addition, $\mathcal{R}$ shows reliable denoising performance by complementing the low flexibility of the kernel prediction module. The performance improvement by $\mathcal{R}$ is proven in Section V-B.

To sum up, the proposed denoising framework succeeds in taking advantage of both 1spp and TA-1spp images through filtering and fusion based on multi-scale kernel prediction. Furthermore, by attaching a residual learning-based refinement module, the flexibility of the entire framework can be improved even with low capacity. Therefore, we were able to achieve high denoising performance in real time, and its experimental proofs are given in the next section.

IV. EXPERIMENTS

This section first describes the dataset used for learning and inference, as well as the training setup. Next, evaluation metrics are depicted. Finally, the benchmarked techniques are described and the comparison results with the conventional techniques are presented.

A. DATASETS

There are few published examples of datasets for real-time Monte Carlo denoising. So, we adopt the (virtually only) BMFR dataset used in Koskela et al. [17]. The BMFR dataset consists of 6 scenes with various rendering effects (e.g., glossy reflection, soft shadow, diverse illumination). Each scene contains 60 frames with smooth camera movement where the noisy image was rendered at 1spp and the reference image at 4096spp, respectively.

Also, to evaluate the generalization performance in high spp, we additionally employ the high-quality Tungsten dataset released by [2]. The Tungsten dataset consists of a total of 8 publicly available Tungsten scenes [26] including complex geometry information and lighting conditions. Each scene is composed of 100 frames. Here, a noisy image is rendered with 64spp and the reference image with 4096spp, respectively. Since the noisy images are rendered in 64spp for offline applications, the Tungsten dataset does not utilize TA.

Both datasets have a resolution of 1280 × 720. We use albedo, shading normal, and camera-space depth as auxiliary features. This setup is similar to those of Meng et al. [2] and Fan et al. [8].

B. TRAINING

As in Meng et al. [2] and Fan et al. [8], we adopt all other scenes except the test scene as the training dataset. So, the performance on the test data can be an indicator of the generalization ability of the trained denoiser [2]. As suggested by Vogels et al. [9], we define the symmetric mean absolute percentage error (SMAPE) as the loss function for training:

$$\text{SMAPE}(o', t) = \frac{1}{3N} \sum_{p \in N} \sum_{c \in C} \frac{|o'_{p,c} - t_{p,c}|}{|o'_{p,c}| + |t_{p,c}| + \epsilon} \tag{5}$$

where o' is the final output of the proposed denoiser multiplied by albedo, t is the reference image, N is the number of pixels in the image, and c is the color channel. ϵ is set to 0.001 in this paper.

TABLE 1. PSNR comparison for the BMFR dataset.

Scene	SVGF [16]	ONND [5]	BMFR [17]	NBGD [2]	WSKPN [8]	Proposed method
Classroom	25.034	27.312	28.965	31.519	32.827	**33.216**
Living room	27.239	25.586	30.025	32.294	33.245	**33.725**
San Miguel	18.736	20.172	20.969	23.650	24.268	**24.771**
Sponza	24.401	24.698	31.111	33.188	33.561	**35.643**
Sponza (glossy)	20.917	23.460	25.005	29.548	28.686	**31.089**
Sponza (mov. light)	17.260	22.296	17.377	24.818	25.323	**30.525**
Average	22.264	23.920	25.575	29.170	29.652	**31.495**

TABLE 2. SSIM comparison for the BMFR dataset.

Scene	SVGF [16]	ONND [5]	BMFR [17]	NBGD [2]	WSKPN [8]	Proposed method
Classroom	0.952	0.924	0.955	0.968	0.977	**0.978**
Living room	0.950	0.953	0.965	0.968	0.975	**0.978**
San Miguel	0.790	0.744	0.789	0.820	0.849	**0.868**
Sponza	0.927	0.852	0.948	0.973	0.980	**0.986**
Sponza (glossy)	0.913	0.867	0.907	0.941	0.943	**0.963**
Sponza (mov. light)	0.876	0.811	0.858	0.946	0.955	**0.970**
Average	0.901	0.858	0.904	0.936	0.946	**0.957**

The proposed denoiser was implemented in PyTorch [27], and its run-time was measured on Nvidia RTX 2080Ti. In the training phase, input data was randomly cropped into 128 × 128, and random horizontal and vertical flipping were applied for data augmentation. Also, in order to increase robustness for color distortion, bright distortion with [0.8, 1.2] range was applied to each rendered image. The network was trained for 300 epochs using Adam optimizer [28]. The batch size and learning rate were set to 16 and 0.001, respectively.

C. EVALUATION METRICS

We adopt Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM) [29], Root Mean Square Error (RMSE) and Symmetric Mean Absolute Percentage Error (SMAPE) as metrics to evaluate denoising performance.

First, PSNR is defined by Eq. 6, where MSE is the mean squared error between o' multiplied by the albedo of the proposed denoiser and the reference image t, and R indicates the maximum pixel value. The higher the PSNR, the more similar it is to the reference image.

$$\mathrm{PSNR}(o', t) = 10 \log \frac{R^2}{\mathrm{MSE}(o', t)} \tag{6}$$

SSIM is defined by Eq. 7, where $\mu_{o'}$, μ_t, $\sigma_{o'}$, σ_t, and $\sigma_{o't}$ stand for local means, standard deviations, and cross-covariance for images o' and t. And C_1 and C_2 are constants defined by the dynamic range value, and for further details, please refer to [29]. The closer the SSIM is to 1, the more similar it is to the reference image.

$$\mathrm{SSIM}(o', t) = \frac{(2\mu_{o'}\mu_t + C_1)(2\sigma_{o't} + C_2)}{(\mu_{o'}^2 + \mu_t^2 + C_1)(\sigma_{o'}^2\sigma_t^2 + C_2)} \tag{7}$$

RMSE indicates the square root of MSE, and SMAPE is defined by Eq. 5. The smaller RMSE and SMAPE are, the more similar they are to the reference image.

D. ANALYSIS AND COMPARISON RESULTS

This subsection describes the benchmarked techniques, and quantitatively compares our denoiser with the conventional techniques on the BMFR and Tungsten datasets.

TABLE 3. RMSE comparison for the BMFR dataset.

Scene	SVGF [16]	ONND [5]	BMFR [17]	NBGD [2]	WSKPN [8]	Proposed method
Classroom	0.0561	0.0431	0.0356	0.0265	0.0229	**0.0219**
Living room	0.0435	0.0526	0.0316	0.0227	0.0219	**0.0206**
San Miguel	0.1160	0.0982	0.0895	0.0644	0.0614	**0.0578**
Sponza	0.0661	0.0591	0.0282	0.0207	0.0214	**0.0168**
Sponza (glossy)	0.0900	0.0671	0.0564	0.0318	0.0370	**0.0280**
Sponza (mov. light)	0.1418	0.0773	0.1450	0.0572	0.0553	**0.0299**
Average	0.0856	0.0662	0.0644	0.0372	0.0366	**0.0292**

TABLE 4. SMAPE comparison for the BMFR dataset.

Scene	SVGF [16]	ONND [5]	BMFR [17]	NBGD [2]	WSKPN [8]	Proposed method
Classroom	0.0405	0.0528	0.0261	0.0206	**0.0190**	0.0199
Living room	0.0220	0.0418	0.0182	0.0140	0.0137	**0.0136**
San Miguel	0.1278	0.1425	0.1160	0.0982	0.1129	**0.0905**
Sponza	0.0530	0.0715	0.0314	0.0190	0.0210	**0.0163**
Sponza (glossy)	0.0759	0.0966	0.0730	0.0442	0.0488	**0.0386**
Sponza (mov. light)	0.1408	0.0882	0.1492	0.0593	0.0553	**0.0343**
Average	0.0767	0.0822	0.0690	0.0426	0.0456	**0.0355**

For the BMFR dataset of a real-time denoising purpose, we compare the following real-time denoising algorithms with the proposed method: ONND [5], SVGF [16], BMFR [17], 2-layer 3-grid NBGD [2], and 3-layer WSKPN [8].

Then, for the Tungsten dataset of the offline denoising purpose, we designed large models such as 7-layer 3-grid NBGD (NBGD-7) and MR denoiser of WSKPN (WSKPN-MR). Also, since TA images are not used here, we perform filtering and fusion by down-sampling 64 spp to three scales. Refer to the Appendix. VII-A for our large model structure. We compare the proposed method with ONND [5], 5-layer MR-KP [9], NBGD-7, and WSKPN-MR.

To make the comparison as fair as possible, we reused the results shared by Meng et al. [2], where the codes provided by the authors were used for experiments. WSKPN was trained on the BMFR dataset by using the official code released by Fan et al. [8] and we provided the results.

First, let's look at the real-time denoising performance, which is the main target of this paper. Tables 1, 2, 3 and 4 are PSNR, SSIM, RMSE and SMAPE results for the BMFR dataset, respectively. The proposed denoiser shows 1.843dB higher PSNR, 0.011 higher SSIM, 0.074 lower RMSE and 0.010 lower SMAPE than WSKPN, i.e., SOTA algorithm.

Especially, for the Sponza moving-light scene, our PSNR was improved by as much as 5.202dB beyond SOTA. In the Sponza moving-light scene, the camera is fixed and only the light source changes, so the performance degradation caused by TA is most prominent there. For this scene, while other techniques show very low restoration performance, the proposed method provides significant performance improvement because it fully utilizes current pixel information from 1spp images. As a result, the disadvantages of using only TA-1spp can be overcome through the fusion of 1 spp and TA-1spp images.

The following are quantitative results for Tungsten. Like the previous studies, we choose 5 scenes, i.e., Bedroom, Classroom, Dining-room, Kitchen, and White room, among 8 scenes, and then provide the results only for the five scenes. Tables 5, 6, 7 and 8 show that the proposed method shows comparable performance to other methods for the five scenes.

TABLE 5. PSNR comparison for the Tungsten dataset.

Scene	ONND [5]	MR-KP [9]	NBGD-7 [2]	WSKPN-MR [8]	Proposed method (large)
Bedroom	34.438	**36.738**	35.983	36.340	36.566
Dining room	37.953	36.879	37.309	38.030	**38.676**
Kitchen	34.797	35.734	35.531	35.880	**36.021**
Classroom	32.874	32.535	32.119	32.820	**33.298**
White room	36.597	37.512	38.081	38.530	**38.675**

TABLE 6. SSIM comparison for the Tungsten dataset.

Scene	ONND [5]	MR-KP [9]	NBGD-7 [2]	WSKPN-MR [8]	Proposed method (large)
Bedroom	0.971	**0.977**	0.974	**0.977**	0.976
Dining room	0.970	0.981	0.979	0.981	**0.983**
Kitchen	0.973	0.974	0.974	0.976	**0.978**
Classroom	0.949	0.945	0.942	0.949	**0.951**
White room	0.973	0.977	0.977	**0.979**	**0.979**

TABLE 7. RMSE comparison for the Tungsten dataset.

Scene	ONND [5]	MR-KP [9]	NBGD-7 [2]	WSKPN-MR [8]	Proposed method (large)
Bedroom	0.0190	0.0157	0.0159	0.0154	**0.0149**
Classroom	0.0227	0.0230	0.0248	0.0234	**0.0216**
Dining-room	0.0128	0.0133	0.0137	0.0131	**0.0117**
Kitchen	0.0183	**0.0159**	0.0168	0.0167	**0.0159**
White room	0.0149	0.0129	0.0125	0.0122	**0.1168**

TABLE 8. SMAPE comparison for the Tungsten dataset.

Scene	ONND [5]	MR-KP [9]	NBGD-7 [2]	WSKPN-MR [8]	Proposed method (large)
Bedroom	0.0194	**0.0146**	0.0158	0.0150	**0.0146**
Classroom	0.0321	0.0280	0.0299	0.0284	**0.0265**
Dining-room	0.0467	0.0294	0.0268	0.0274	**0.0234**
Kitchen	0.0257	0.0202	0.0223	0.0214	**0.0195**
White room	0.0149	0.0116	0.0124	0.0117	**0.111**

This demonstrates the generalization ability of our denoiser even for high spp input.

Table 9 shows the analysis of run-time. Here, run-time means the time from video input to result output, and each value in the table is an average of run-time per frame of all scenes. With SOTA performance, the proposed model guarantees real-time performance by achieving a whopping 239 FPS at HD resolution. Even though the proposed method has a run-time similar to SVGF [16], it shows a higher PSNR by 9.231dB. Also, note that ours shows better performance with only 71% and 39% lower run-times than NBGD and WSKPN, respectively. On the other hand, even in the model for the Tungsten dataset, our run-time is 63% less than that of WSKPN-MR, i.e., SOTA in this dataset.

Finally, we analyze the time required for each module of our denoiser (see Table 10). Note that this paper aims at real-time denoising. So, we designed a kernel prediction module that takes only 1.46 ms using a light-weight autoencoder. Then, after unfolding the input image, we presented a filtering step in the form of simply multiplying the predicted kernel map and summing it in the channel direction. We also utilized Vogels et al. [9]'s scale-compositor, which can quickly and effectively fuse two filtered images, namely filtered 1spp and TA-1spp. As a result, the time required for filtering and fusion was only 1.64 ms. In the end, residual noise removal and fine details enhancement were achieved with an additional time of 0.89 ms through a small residual-based refinement module.

In summary, the proposed method shows comparable or higher denoising performance than existing methods with significantly lower run-time. In other words, the proposed method breaks through the performance-cost trade-off.

E. QUALITATIVE RESULTS

Figure 3 illustrates the results for the BMFR dataset. First, examining the results of the Classroom and Living room, we can observe that our denoiser successfully restores the structural features compared to other denoisers, while also achieving a smoother and more natural restoration of shadows. Looking at the early frames (e.g., 6th frame) of San Miguel, the proposed method has a much better denoising performance than other methods. Specifically, in the first row, NBGD and WSKPN do not properly remove the noise of early frames that have not been temporally accumulated yet. And, in SVGF, ONND and BMFR, the chair shape and the water bottle appear blurred. In the second row, NBGD and WSKPN still do not completely remove noise, and SVGF handles the bottom area regardless of the reference image. Also, ONND and BMFR look blurry overall, and they do not properly restore shadows. Observing the first row of Sponza, our denoiser preserves edges with greater clarity compared to NBGD and WSKPN. On the contrary, SVGF, ONND, and BMFR exhibit an overall blur, resulting in indistinct edges and inadequate texture restoration. Turning to the second row, our denoiser best retains the wrinkles in the fabric and the details of the patterns. In the case of the first row of Sponza glossy, SVGF, ONND, and BMFR introduce color distortion to the glossy regions. NBGD falls short in achieving sufficient denoising, while WSKPN exhibits inadequate softness at the boundary between glossy and non-glossy areas. Conversely, our denoiser effectively restores glossy areas with softness and without color distortion. Furthermore, it's worth noting the resemblance between the second row of Sponza glossy and the second row of Sponza. In essence, our denoiser accurately restores the detail of the wrinkles, showing a similarity to the reference. Finally, in the first row of the Sponza moving-light, our denoiser represents the light most closely to the reference. Since NBGD and WSKPN use

TABLE 9. Run-time comparison at HD resolution.

	Method	Run-time (ms)	Device
Real-time	SVGF [16]	4.40	Nvidia Titan X
	BMFR [17]	1.60	Nvidia RTX 2080
	NBGD [2]	16.46 / 13.97	Nvidia RTX 2080 / 2080 Ti
	WSKPN [8]	6.58	Nvidia RTX 2080 Ti
	Proposed method	3.99	Nvidia RTX 2080 Ti
Offline	ONND [5]	55.00	Nvidia Titan X
	MR-KP [9]	39.53	Nvidia RTX 2080
	NBGD-7 [2]	84.99 / 50.28	Nvidia RTX 2080 / 2080 Ti
	WSKPN-MR [8]	22.70	Nvidia RTX 2080 Ti
	Proposed method (large)	8.34	Nvidia RTX 2080 Ti

TABLE 10. Run-time analysis of proposed denoiser at HD resolution.

	Module	Run-time (ms)	Device
Proposed method	Kernel Prediction	1.46	Nvidia RTX 2080 Ti
	Filtering and Fusion	1.64	
	Refinement	0.89	

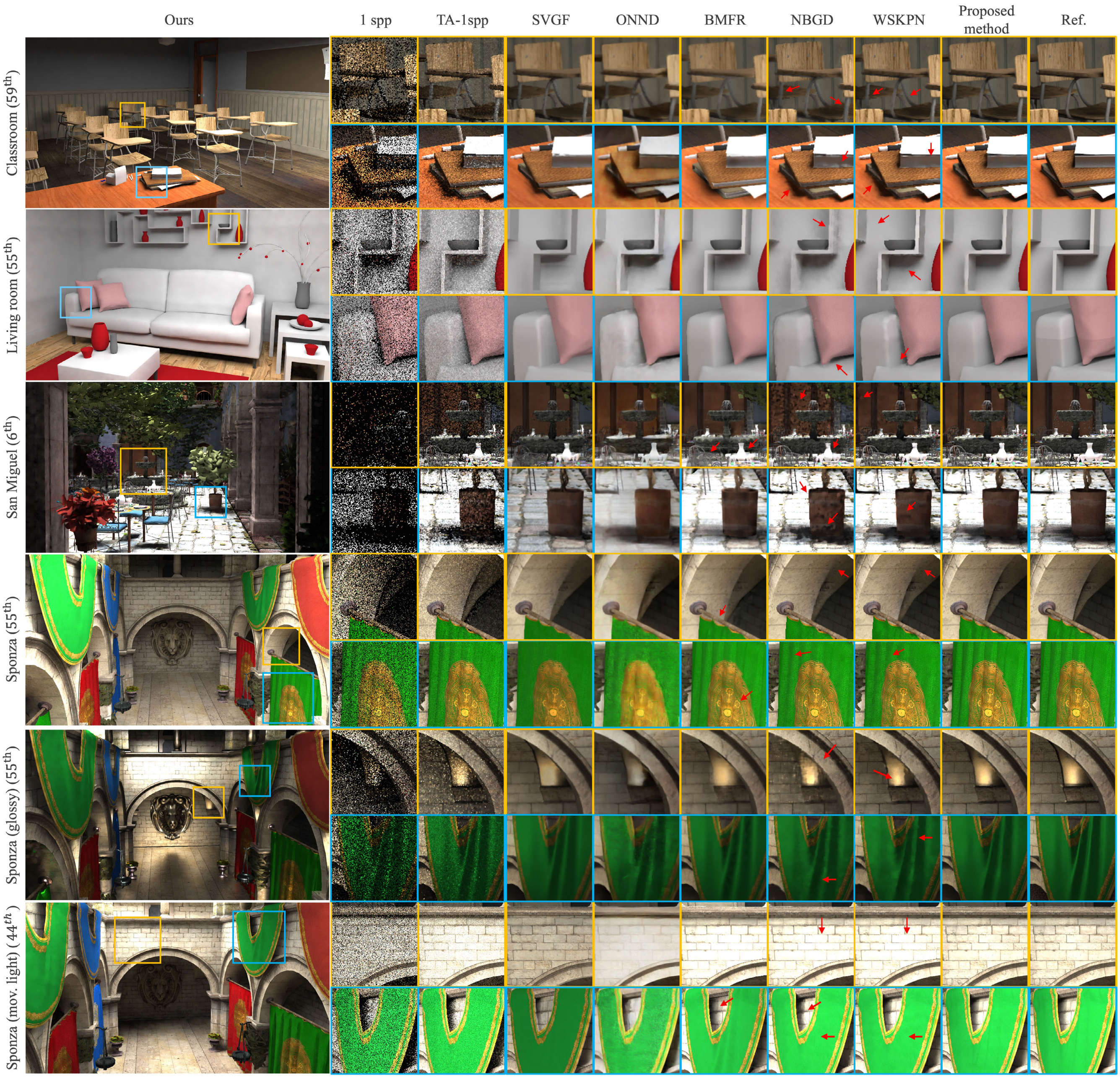

FIGURE 3. **Qualitative results on the BMFR dataset. Here, 'Ref.' indicate the reference image.**

only TA-1spp images, they do not respond well to rapidly changing pixels. In the second row, all techniques except BMFR and ours do not properly restore the wrinkles of the fabric. Although BMFR restores the wrinkles to some extent, it does not properly restore the shadows and light behind the fabric.

Figure 4 illustrates the results for the Tungsten dataset. Beginning with the Bedroom scene, our denoiser excels at restoring the shape of transparent objects compared to the others. Furthermore, in the first row of the Dining room scene, the proposed method effectively removes white noise without introducing artifacts, surpassing the performance of the other denoisers. Moving on to the second row of the Dining room and the first row of the Kitchen, our denoiser distinctly restores structural features with remarkable clarity. Lastly, focusing on the second row of the Kitchen scene, our denoiser successfully restores the details of the microwave without any noticeable lumpiness

Thus, we claim that our denoiser succeeds in improving performance from the early frames and is also robust against rapid pixel changes because it utilizes both 1spp and TA-1spp together. Furthermore, the results on the Tungsten dataset

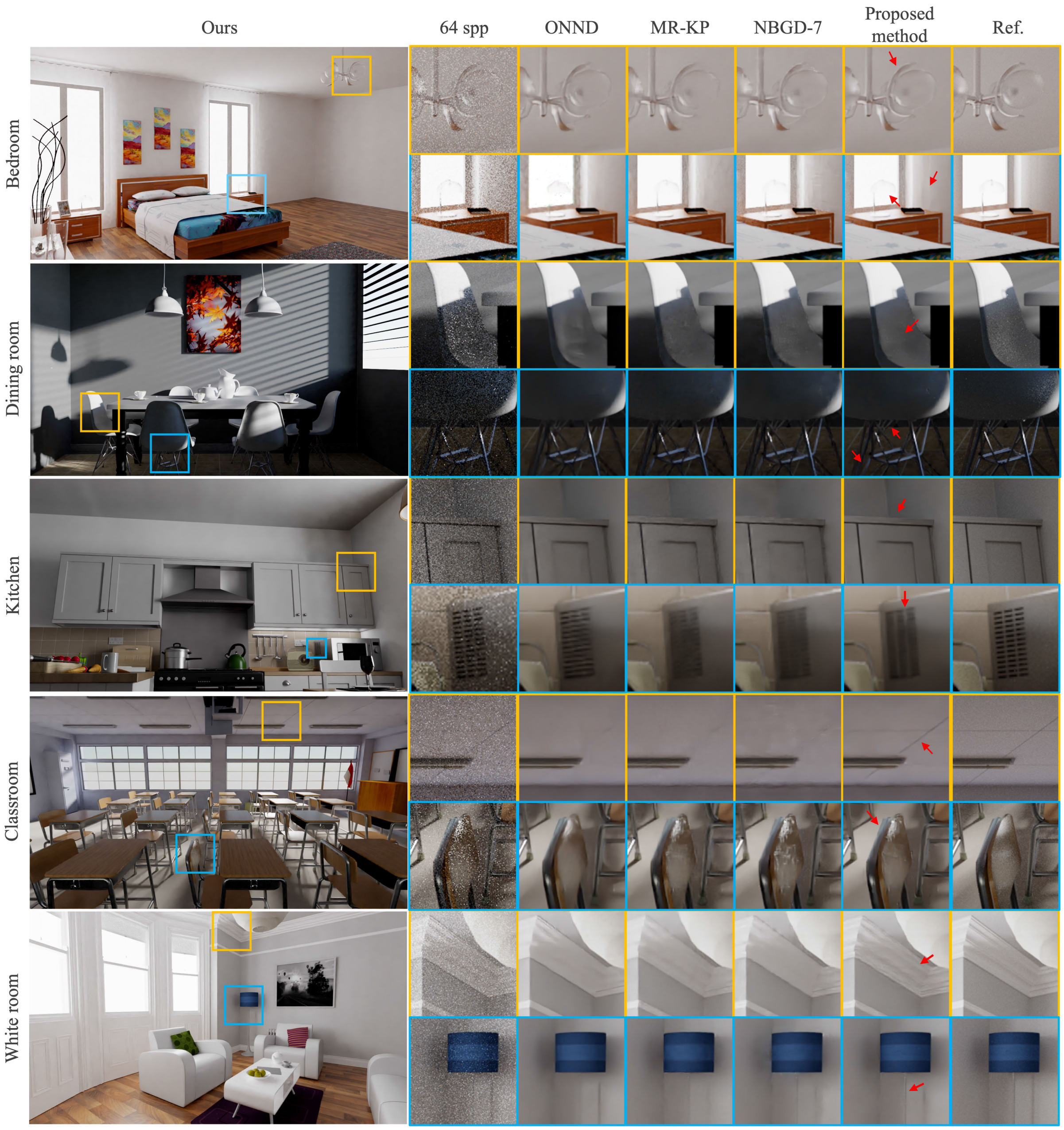

FIGURE 4. Qualitative results on the Tungsten dataset.

demonstrate that our denoiser effectively handles high spp input.

V. ABLATION STUDY

This ablation study is to verify the effect of the key module of our denoiser, i.e., the fusion and refinement. This experiment is conducted only on the BMFR dataset, which is the main target of this paper.

A. FUSION

The first experiment is designed to verify the fusion effect of two rendered images with different effective spp, that is, 1spp and TA-1spp. Here, the proposed method is compared with the case of using only one rendered image. The first, second, and third columns of Table 11 and Figure 5 show quantitative and qualitative results.

TABLE 11. Effects of fusion and refinement on the BMFR dataset.

Scene	PSNR			
	1spp only	TA-1spp only	Fusion only	Fusion w/ refine
Classroom	30.641	32.275	33.153	**33.216**
Living room	30.589	32.982	33.414	**33.725**
San Miguel	22.436	24.360	24.723	**24.771**
Sponza	33.840	33.668	35.580	**35.643**
Sponza (glossy)	27.548	30.153	30.819	**31.089**
Sponza (mov. light)	29.560	25.231	30.062	**30.525**
Average	29.102	29.778	31.292	**31.495**

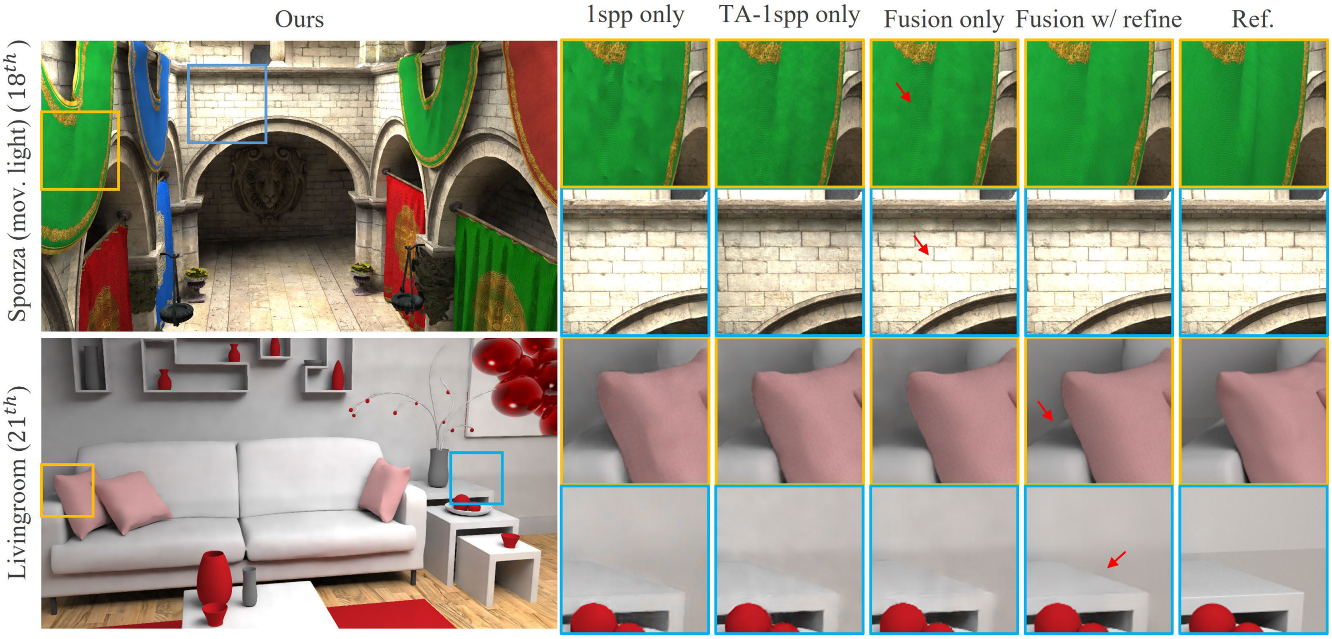

FIGURE 5. Qualitative results for ablation studies.

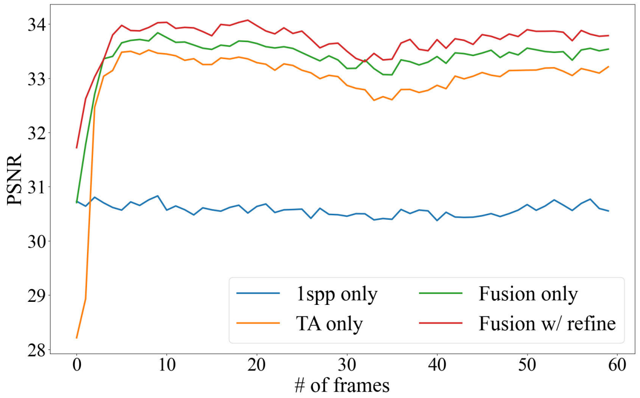

FIGURE 6. PSNR change over time for the living room scene.

Since most BMFR scenes have only smooth camera motion without light source change, using only TA-1spp images with high effective spp shows 0.676dB higher PSNR than using only 1spp images. On the other hand, in the case of Sponza moving-light scene where the light source changes rapidly, using only 1spp that utilizes pixel information of the current frame provides better performance. From the experimental result that the proposed fusion provides SOTA performance in all scenes, it is proved that the proposed method successfully takes the advantages of 1spp and TA-1spp.

To further analyze this trend, Figure 6 shows the restoration performance for Living room scene on a time axis. As described above, 1spp shows good performance in the early frames, and TA-1spp is better in the latter frames when TA is sufficiently progressed. Note that the proposed method, i.e., 'Fusion w/ refine', which has the advantage of both rendered images, always shows the best performance.

B. REFINEMENT

Table 11 show that 'Fusion w/ refine' raises the PSNR by 0.203dB from 'Fusion only'. At this time, the run-time increased by about 17%. As a result, the proposed refinement module improves performance with low capacity. Consequently, each component of the proposed approach plays a role in effectively reducing the noise in Monte Carlo rendered images in accordance with our objectives. Additional qualitative results can be found in the third and fourth columns of Figure 5.

VI. DISCUSSION

This paper presents a software solution to mitigate the phenomenon that the displayed image is perceived as darker than the source image due to our human visual system when viewing the displayed image in an ambient light environment. However, various distortions (e.g., noise) other than ambient lighting issue can occur. Unfortunately, until now, we have not been able to find a model that simulates those distortions. If such a degradation model is available in the future, a method for improving image quality adaptive to various distortions can be devised.

Next, the degradation model we used here considers the brightness (lux) of ambient light and the display specification to simulate how the displayed image can be perceived at a specific lux. Accordingly, we designed a model with only the lux of the ambient light as a parameter, and used a lighting box for experiments according to lux. That is, only artificial light sources are considered in this paper. Therefore, the proposed method has limitations in handling various cases (e.g., non-uniform lighting, backlit scenarios, etc) that can occur in natural light sources.

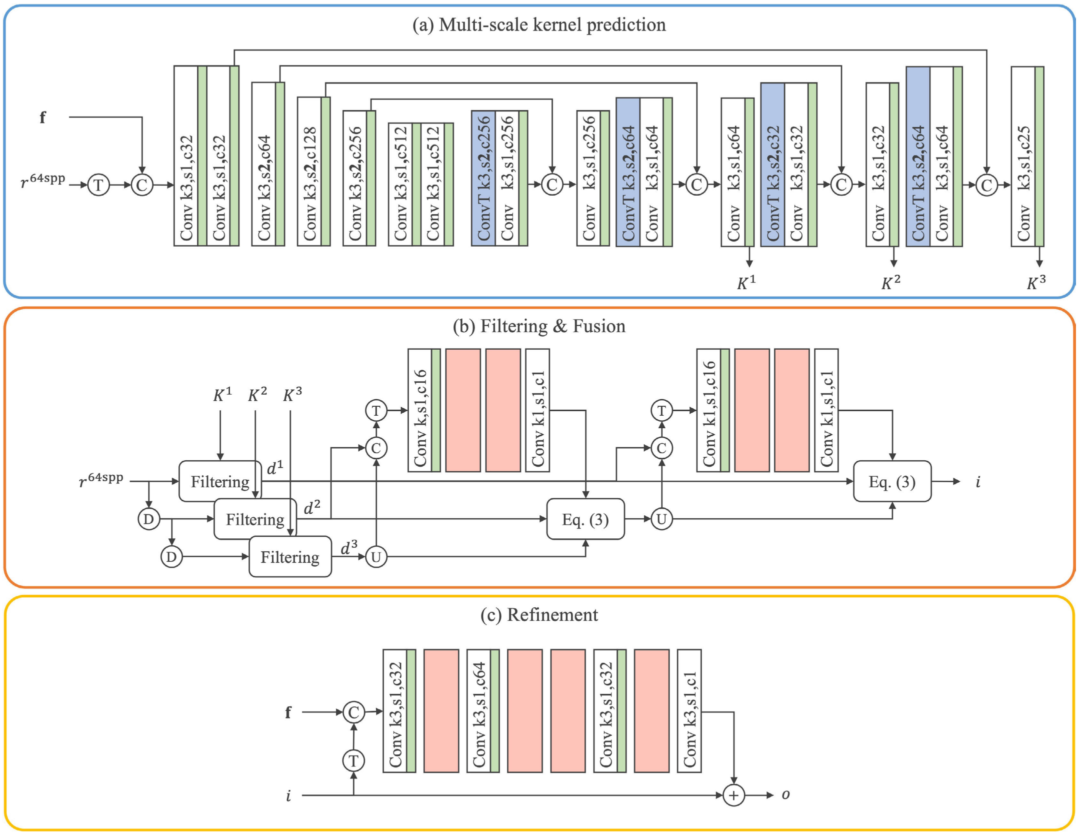

FIGURE 7. Details of each module of the proposed denoiser (large).

Since the variability between natural light source and artificial light source definitely exists, research on this will be needed in the future.

VII. CONCLUSION

This paper proposes a denoiser to obtain high-quality Monte Carlo path traced images in real time. We observed the pros and cons of each of the 1spp image and the TA-1spp image, and devised a novel fusion and refinement framework that could combine only the strengths of the two images. First, kernel maps for filtering 1spp image and TA-1spp image are estimated through multi-scale kernel prediction based on a light-weight autoencoder. Then, the two separately filtered images are fused. At this time, the fusion weight is adaptively calculated depending on the input data. Finally, the residual learning-based refinement module provides better model flexibility than using only kernel prediction, and successfully refines the fused images. Extensive experiments prove that our denoiser not only shows better restoration performance at 39% faster than the conventional real-time technique, but also has good generalization performance even for high spp.

Nevertheless, the proposed method also has some limitations. First, in scenes with high temporal consistency (e.g., Living room), our denoiser encountered an issue where features from unnecessary 1spp were incorporated, leading to a degradation of temporal stability. Consequently, a trade-off between the swift restoration of reference image pixels without color distortion and maintaining temporal stability will require further discussion Second, the proposed method basically requires TA images, so the additional cost for obtaining TA images is unavoidable. Third, the performance of the proposed method may be affected by the TA technique adopted. To overcome these limitations, we need to develop a new TA that can be used within our denoising pipeline or a temporal filter for better temporal stability. This will be our future works.

APPENDIX

A. NETWORK ARCHITECTURE FOR TUNGSTEN DATASET

This section describes the large architecture of the proposed denoiser, which is represented in Fig. 7. The proposed denoiser (large) predict 3 scales of 5 × 5 kernel maps, i.e., K^1, K^2, K^3. Next, each kernel is applied to the original and down-scaled 64spp images, and filtered images are fused progressively. Finally, large refinement modules are applied to remove residual noise and improve fine details.

REFERENCES

[1] T. Whitted, "An improved illumination model for shaded display," in *Proc. ACM SIGGRAPH Courses (SIGGRAPH)*. New York, NY, USA: ACM, 2005, p. 4.

[2] X. Meng, Q. Zheng, A. Varshney, G. Singh, and M. Zwicker, "Real-time Monte Carlo denoising with the neural bilateral grid," in *Proc. EGSR*. London, U.K.: The Eurographics Association, Jun. 2020, pp. 13–24.

[3] W.-J. Lee, Y. Shin, J. Lee, J.-W. Kim, J.-H. Nah, S. Jung, S. Lee, H.-S. Park, and T.-D. Han, "SGRT: A mobile GPU architecture for real-time ray tracing," in *Proc. 5th High-Perform. Graph. Conf.* New York, NY, USA: ACM, Jul. 2013, pp. 109–119.

[4] J.-H. Nah, H.-J. Kwon, D.-S. Kim, C.-H. Jeong, J. Park, T.-D. Han, D. Manocha, and W.-C. Park, "RayCore: A ray-tracing hardware architecture for mobile devices," *ACM Trans. Graph.*, vol. 33, no. 5, pp. 1–15, Sep. 2014.

[5] C. R. A. Chaitanya, A. S. Kaplanyan, C. Schied, M. Salvi, A. Lefohn, D. Nowrouzezahrai, and T. Aila, "Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, Aug. 2017.

[6] S. Bako, T. Vogels, B. Mcwilliams, M. Meyer, J. NováK, A. Harvill, P. Sen, T. Derose, and F. Rousselle, "Kernel-predicting convolutional networks for denoising Monte Carlo renderings," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–14, Aug. 2017.

[7] M. Işik, K. Mullia, M. Fisher, J. Eisenmann, and M. Gharbi, "Interactive Monte Carlo denoising using affinity of neural features," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–13, Aug. 2021.

[8] H. Fan, R. Wang, Y. Huo, and H. Bao, "Real-time Monte Carlo denoising with weight sharing kernel prediction network," *Comput. Graph. Forum*, vol. 40, no. 4, pp. 15–27, Jul. 2021.

[9] T. Vogels, F. Rousselle, B. Mcwilliams, G. Röthlin, A. Harvill, D. Adler, M. Meyer, and J. Novák, "Denoising with kernel prediction and asymmetric loss functions," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–15, Aug. 2018.

[10] Y. Huo and S.-E. Yoon, "A survey on deep learning-based Monte Carlo denoising," *Comput. Vis. Media*, vol. 7, no. 2, pp. 169–185, Jun. 2021.

[11] N. K. Kalantari, S. Bako, and P. Sen, "A machine learning approach for filtering Monte Carlo noise," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–12, Jul. 2015.

[12] J. Back, B.-S. Hua, T. Hachisuka, and B. Moon, "Deep combiner for independent and correlated pixel estimates," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–12, Dec. 2020.

[13] M. Kettunen, M. Manzi, M. Aittala, J. Lehtinen, F. Durand, and M. Zwicker, "Gradient-domain path tracing," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–13, Jul. 2015.

[14] B. Bitterli, F. Rousselle, B. Moon, J. A. Iglesias-Guitián, D. Adler, K. Mitchell, W. Jarosz, and J. Novák, "Nonlinearly weighted first-order regression for denoising Monte Carlo renderings," *Comput. Graph. Forum*, vol. 35, no. 4, pp. 107–117, Jul. 2016.

[15] S. Zheng, F. Zheng, K. Xu, and L.-Q. Yan, "Ensemble denoising for Monte Carlo renderings," *ACM Trans. Graph.*, vol. 40, no. 6, pp. 1–17, Dec. 2021.

[16] C. Schied, A. Kaplanyan, C. Wyman, A. Patney, C. R. A. Chaitanya, J. Burgess, S. Liu, C. Dachsbacher, A. Lefohn, and M. Salvi, "Spatiotemporal variance-guided filtering: Real-time reconstruction for path-traced global illumination," in *Proc. High Perform. Graph.* New York, NY, USA: ACM, Jul. 2017, pp. 1–12.

[17] M. Koskela, K. Immonen, M. Mäkitalo, A. Foi, T. Viitanen, P. Jääskeläinen, H. Kultala, and J. Takala, "Blockwise multi-order feature regression for real-time path-tracing reconstruction," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–14, Oct. 2019.

[18] L. Yang, D. Nehab, P. V. Sander, P. Sitthi-amorn, J. Lawrence, and H. Hoppe, "Amortized supersampling," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 1–12, Dec. 2009.

[19] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "RepVGG: Making VGG-style ConvNets great again," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13728–13737.

[20] M. Gharbi, T.-M. Li, M. Aittala, J. Lehtinen, and F. Durand, "Sample-based Monte Carlo denoising using a kernel-splatting network," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Aug. 2019.

[21] J. Munkberg and J. Hasselgren, "Neural denoising with layer embeddings," *Comput. Graph. Forum*, vol. 39, no. 4, pp. 1–12, Jul. 2020.

[22] J. Hasselgren, J. Munkberg, M. Salvi, A. Patney, and A. Lefohn, "Neural temporal adaptive sampling and denoising," *Comput. Graph. Forum*, vol. 39, no. 2, pp. 147–155, May 2020.

[23] B. Xu, J. Zhang, R. Wang, K. Xu, Y.-L. Yang, C. Li, and R. Tang, "Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–12, Dec. 2019.

[24] A. Firmino, J. R. Frisvad, and H. W. Jensen, "Progressive denoising of Monte Carlo rendered images," *Comput. Graph. Forum*, vol. 41, no. 2, pp. 1–11, May 2022.

[25] J. Guo, M. Li, Q. Li, Y. Qiang, B. Hu, Y. Guo, and L.-Q. Yan, "GradNet: Unsupervised deep screened Poisson reconstruction for gradient-domain rendering," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–13, Dec. 2019.

[26] B. Bitterli. (Sep. 2016). *Rendering Resources*. [Online]. Available: https://benedikt-bitterli.me/resources

[27] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. NIPS*, 2017.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[29] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput., 2003*, vol. 2, Nov. 2003, pp. 1398–1402.

JUNMIN LEE received the B.S. and M.S. degrees in electronic engineering from Inha University, Incheon, South Korea, in 2021 and 2023, respectively. Her research interests include computer vision and image processing.

SEUNGHYUN LEE received the B.S. and Ph.D. degrees in electronic engineering from Inha University, Incheon, South Korea, in 2017 and 2023, respectively. His research interests include computer vision and machine learning.

MIN YOON (Associate Member, IEEE) received the B.S. degree in electronic engineering from Inha University, Incheon, South Korea, in 2022, where he is currently pursuing the M.S. degree in electrical and computer engineering. His research interests include computer vision and deep learning.

BYUNG CHEOL SONG (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1994, 1996, and 2001, respectively. From 2001 to 2008, he was a Senior Engineer with the Digital Media Research and Development Center, Samsung Electronics Company Ltd., Suwon, South Korea. In 2008, he joined the Department of Electronic Engineering, Inha University, Incheon, South Korea, and he is currently a Professor. His research interests include the general areas of image processing and computer vision.