Received January 22, 2021, accepted February 5, 2021, date of publication February 16, 2021, date of current version March 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3059887

Accelerated Path Tracing With GAN and Matrix Completion

QIWEI XING, CHUNYI CHEN, AND ZHIHUA LI
School of Computer Science and Technology, Changchun University of Science and Technology, Changchun 130022, China

Corresponding author: Chunyi Chen (chenchunyi@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant U19A2063, in part by the Jilin Provincial Science and Technology Development Program of China under Grant 20190302113GX, and in part by the 13th Five-year Science and Technology Research Program of Jilin Provincial Department of Education under Grant JJKH20200792KJ.

ABSTRACT Denoising-based techniques have recently been shown to be effective for accelerating path tracing rendering methods. However, there remains a problem which is input images need the minimum necessary samples number in order to ensure the quality of the output. In this paper, we propose a new accelerated path tracing approach with generative adversarial networks(GAN) and matrix completion. Unlike the methods based on denoising with neural network, we randomly render part of pixels of input image, which are much less than other methods. Next, we utilize the trained GAN to pre-complete the initializing missing pixels. Because of the accuracy and fast-convergence of GAN, our pre-completion results are more accurate than other methods. Then, according to the results of pre-completion, we present the pre-completed images as a low-rank matrix and make use of the matrix completion to recovers missing values accurately even in high details. To improve the efficiency of solving matrix completion, we modified the original weighted nuclear norm minimization with a parameter adjustment(PAWNNM) strategy. The result shows better visual quality, texture details and convergence efficiency than the state-of-the-art acceleration methods, especially the methods based on denoising with neural network.

INDEX TERMS Accelerated path tracing, modified GAN, matrix completion, improved PAWNNM.

I. INTRODUCTION

The last few years have seen a decisive move of the movies making industry towards rendering utilizing physically-based approaches, mostly implemented in terms of path tracing algorithm [1]–[3]. Besides, because of their generality, fast start-up, and progressive nature, path tracing has been an important method in many applications in scientific visualization, such as video, games *et al.* [4]–[5]. Unfortunately, such method takes a prohibitive a lot of time to obtain the images with better quality because of the large number of samples required per pixel [6]. So, accelerating path tracing rendering has been one of the most popular ways in the research filed [7].

While Monte Carlo techniques [8] can accelerate the integration process, images without noise remain expensive computational cost. To address the shortcomings, the accelerated rending methods which reduce the samples per pixel have been proposed [7], [9]–[11]. Although reducing samples number could accelerate rendering process, there lots of noises in images. To solve this problem, image-based denoising techniques have matured quickly in recent years [12]. They take less processing time and are often easy to integrate into existing rendering pipelines. Recently, machine learning-based denoising approaches [6], [13]–[15] have been demonstrated to provide more effective means to denoise the images. However, according the Nyquist sampling theorem, these methods are limited by the minimum number of samples per pixel [16]. That is to say, if the samples number is less than the minimum value, they could not obtain the denoising images with better images.

To overcome the limitation, other researchers proposed the CS(compressive sensing)-based approaches, which called CR(compressive rendering) [17]. They render only a fraction of the pixels in the image then try to estimate the values of the missing pixels to complete the rest of the images. They have no requirement for minimum samples number. However, the CR approaches rely on the samples distribution and the solution of CS equation with Newton iterative method is not accuracy. Later, Liu *et al.* [18] improved the CR algorithm with CNN(convolution neural network) and low-rank matrix completion. But the quality of final results

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

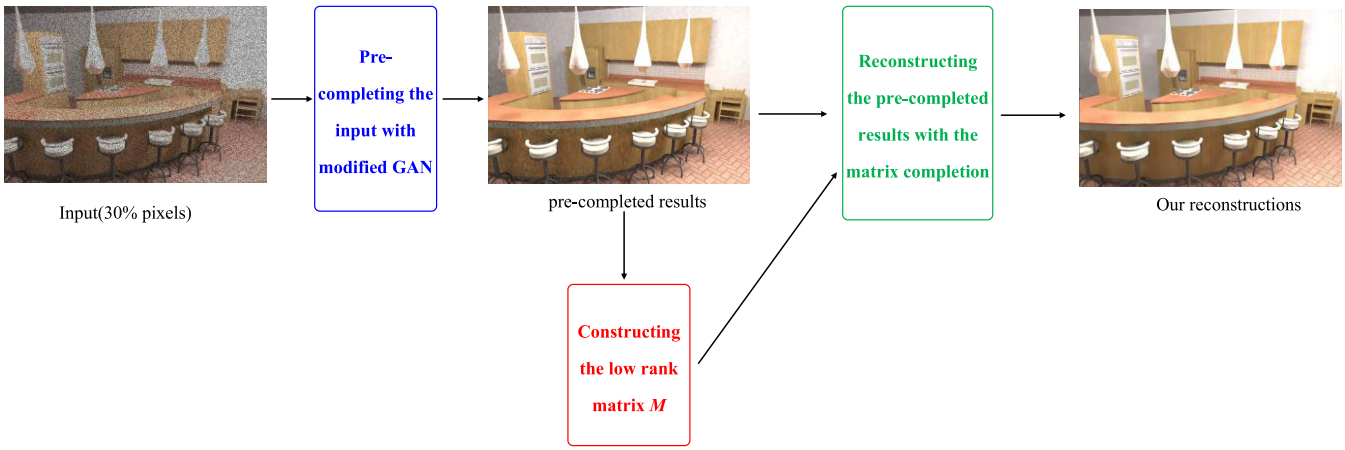


FIGURE 1. The framework of our major algorithm. As shown in Fig, there two major parts. The first one is Pre-completion based on GAN. And the other one is Reconstruction based on matrix completion. Input is the incomplete images rendered by path tracing just including part pixels. We make use the pre-trained modified GAN with hybrid loss function to pre-complete the input. At the same time, we construct a low-rank matrix according to the pre-completed results. Then, we utilize the matrix completion theory and the constructed matrix to reconstruct the final images.

and algorithm efficiency are not good as the methods based on denoising. Because utilizing CNN to pre-complete the inputs is not accuracy and the convergent speed of CNN is too slow. Besides they only took consider into the features based on image space when solving the matrix completion equation, which results in the approximate solution problem. Therefore, we present a novel reconstruction method which efficiently recovers missing data in incompletely rendered images with the modified GAN and matrix completion. Compared with Liu's method, we make use of GAN to replace CNN to pre-complete the missing data and we improved the original parameter adjustment strategy used to solve the matrix completion equation with the additional features, such as shading normal, visibility and so on. So, our results are of better quality and faster speed. Compared with the state-of-the-art accelerate path tracing algorithms, especially the methods based on denoising framework, our approach could get the images with same quality in less time. The whole experiment results could be seen in Part **IV**. Generally, our major framework is shown in **FIGURE 1**.

As shown in **FIGURE 1**, firstly we render an incomplete scene image including part pixels. And then modified GAN is used to pre-complete the missing samples. According to the pre-completed results, we construct a matrix next. Finally, we use matrix completion to estimate the missing samples. In summary, our contributions are:

- We combine the GAN and matrix completion technique to reconstruct the incomplete path tracing.
- We modify the GAN with multi-loss function and critic net to improve the results of pre-completion. Our modified GAN contributes to reduce the reconstruction error and spatial noise in images.
- In order to obtain the images with better results in less time, we improve the original multi-weighted nuclear norm minimization with a parameter adjustment(PAWNNM) strategy, which includes not only features in image space, but the features of shading process.
- to improve the results of pre-completion and obtain the better images with less time.
- Our method has been evaluated against state-of-the-art methods and shows better results in terms of visual quality and statistics. In particular our method outperforms with limited samples ($\leq$35%).

II. RELATED WORK

A. ACCELERATED PATH TRACING WITHOUT DEEP LEARNING

Path tracing was first proposed by Veach [19] expressed as the rendering equation. But evaluating rendering equation took too much time and the images rendered were over blurred. Therefore, several methods have been proposed for improving the origin path tracing [7], [9]–[11]. However, although these methods improved the quality of images, their efficiency were too slow to be used in real time applications. So accelerating path tracing has been the main research point. In recent years, many successful accelerated techniques used for path tracing. In 2015, Zwicker *et al.* [12] summarized non-deep learning approaches, and divided them into two general categories: the priori methods and the posteriori methods. The priori methods leverage information obtained from the analysis of the rendering equation to enhance path tracing samples and then generate adaptive reconstruction filters based on the information. For example, Ramamoorthi *et al.* [20] applied derivative analysis to enhance adaptive sampling and conduct a more comprehensive and thorough first-order analysis of lighting, shading, and shadows in direct illumination. Later, Jarosz *et al.* [21] improved the Ramamoorthi's work with a second-order analysis of indirect illumination. The other category of methods utilized a family of reconstruction filters and develop error estimation to reconstruct images. These methods mainly migrated from the algorithms of denoising nature images and treated the renderer as a black box [22]. For instance, Bauszat *et al.* [23] proposed the guided filter methods for denoising path tracing rendering. Rousselle *et al.* [24]

leveraged non-local mean filter for denoising. Moon *et al.* [25] leveraged a linear model to approximate the ground truth. Bauszat *et al.* [26] proposed a robust error estimation method for path tracing. Bitterli *et al.* [27] designed collaborative non-linear regression to reconstruct noise-free images. Unfortunately, these methods generally need to select the optimal filter models and parameters manually, and the input noisy images require the necessary samples.

B. ACCELERATED PATH TRACING BASED ON DENOISING WITH NEURAL NETWORK

To improve the previous non-deep learning approaches, the machine learning was introduced to accelerate path tracing. In particular, most of the approaches based on denoising path tracing presented the neural network. It is worth noting that Kalantari *et al.* [13] firstly introduced a novel denoising framework to denoise path tracing. Although it can avoid limitations caused by manually selecting parameters, it still inherits limits from a fixed filter. Recently, Bako *et al.* [14] presented a kernel-predict convolutional network(KPCN), which mainly utilizes deep convolutional network(CNN), dividing noisy images into two components and leveraging feature buffers as the input of networks, to predict filter kernels for each pixel. Yang *et al.* [22] presented a deep CNN for reconstructing path tracing rendering. They designed an end-to-end network and feed feature buffers and noisy images to the network directly. Chaitanya *et al.* [15] proposed a recurrent autoencoder to reconstruct path tracing image sequence. Although these methods have obtained good performance in the quality of images and efficiency of path tracing, if we reduced the original samples number of input images, they cannot generate satisfactory results. This is because there are few features in the images with less samples. That is to say, these methods are still limited by the necessary samples number. Inspired by the compressive sensing theory [18], we attempt to utilize the missing data images, whose samples number is less than necessary one.

C. ACCELERATED PATH TRACING BASED ON COMPRESSIVE RECONSTRUCTION

To overcome the shortcomings of previous work, compressive sensing has been introduced in computer graphics for reconstructing images with missing several pixels [28], [29]. CR(compressive rendering) methods evaluated a subset of the pixel samples in the spatial domain and transformed the image into the wavelet domain for reconstructing missing values [18]. However, the CS approach required computations in a transform domain and the transformed images should be sparse in the transformed domain basis. Later, Miandji *et al.* [30] proposed dictionary learning to provide an optimal basis for specific visual input in compressed rendering. But the computational cost of dictionary learning is too big.

Recently, to matrix completion and blind source separation methods have been introduced in signal processing to recover missing data in a low rank matrix [28], [29]. Although solving the rank minimization problem is NP hard, it can be approximated, and exactly solved in several cases, as a nuclear norm minimization(NNM) problem. Cai *et al.* [31] provided a singular value thresholding algorithm(SVT) for solving NNM. They demonstrated that we could solve the low rank matrix with a soft thresholding operation on the singular values of the observed matrix. But, matrix completion and separation schemes are seldom adopted in computer graphics [19]. Huo *et al.* [32] took advantage of the low rank property of a many lights matrix for solving many lights rendering problems with a matrix separation scheme. Li *et al.* [33] adapted matrix completion for inpainting while exploring a low rank matrix by grouping similar patches within an image. Although they tested their inpainting scheme to recover missing pixels of rendered images, they mainly focused on recover the missing pixels of photos, so their method were not useful in recovering path tracing renderings. To address the problem, Liu *et al.* [19] improved Li's work. They attempted to combine CNN techniques and PAWNNM strategy used for solving NNM problem. However, because of the inaccuracy of CNN and the limited used features, their results were not good as the ones of the methods based on denoising with deep learning. Therefore, according to the limitations of Liu's method, we proposed a novel accelerated path tracing framework with improved GAN and matrix completion to recovery the incomplete path-tracing images. Especially, when solving the matrix completion equation, we improve the origin PAWNNM strategy.

III. ACCELERATED PATH TRACING BASED ON GAN AND MATRIX COMPLETION

In order to make use of the theory of matrix completion to estimate the missing pixels of images rendered by path tracing, we need to provide a proper low rank matrix that contains sparse random samples. In this paper, we form a pixel patch using neighboring pixels, and then group similar pixel patches from elsewhere in the image to form the low-rank matrix [33]. However, identifying similar pixel patches across an image has extra challenges due to missing pixel values. Therefore, a sophisticated pre-completion step is required to estimate missing pixel values in each patch [33]. To reduce the estimated error, we use a pre-trained GAN to precompute the missing values efficiently in a short time. Our low rank reconstruction is shown in **FIGURE 2**.

A. PRE-COMPLETION BASED ON GAN

Of course, there are other pre-completion methods proposed. Li *et al.* [19] proposed a pre-completion step that initializes missing pixels. They address pre-completion for low frequency and high frequency images differently. For a low frequency image, matrix completion is directly applied for pre-completion, and a total variation(TV) regularized reconstruction [34] is used for high frequency images that do not meet the low rank condition. Although it provides promising initialization, the process requires long computation time that is a clear limitation to adapting in the method for previsualization.

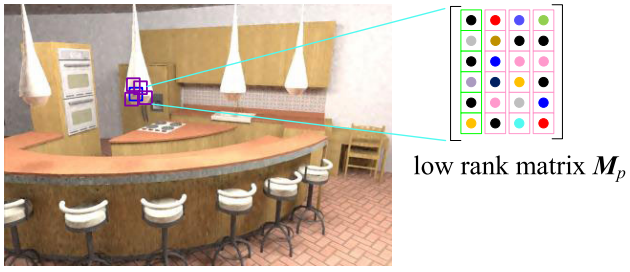


FIGURE 2. Low rank matrix construction for pixel patch. From the pre-completed results, similar pixel patches(red squares) for a current pixel patch(green square) are selected and stored into columns of the matrix M_p.

We present a novel pre-completion scheme using a modified GAN. Our GAN is trained by pairs of a reference image I_r, and an incomplete image I_i that contains randomly removed pixel values in I_i. Then, the pre-trained GAN is used to initialize missing values for pre-completion. Based on the GAN presented by Ref [35], we modify the structure of their network. Our GAN is seen in **FIGURE 3**.

1) LOSS FUNCTION

As described in existing works [36], [37], using pixel-wise content loss like L_1 or L_2 loss tends to produce blurry results, since most image reconstruction tasks are essentially ill-posed problems. Selecting L_2 as loss function will maximize PSNR(peak signal-to-noise ratio) value, but this is not enough to guarantee perceptual quality. Hence, we utilize the hybrid loss function $L(\Theta)$ to measure the difference between the estimated values $F(\boldsymbol{c}_{out};\Theta)$ and the reference values $\boldsymbol{c}_{gt}$ [36]:

$$L(\Theta) = \frac{1}{n}\sum_{i=1}^{n} h\left(F\left(\boldsymbol{c}^{i}_{out};\Theta\right),\boldsymbol{c}_{gt}\right), \tag{1}$$

where n is number of training samples, is the parameters learned by network and h is the hybrid function which

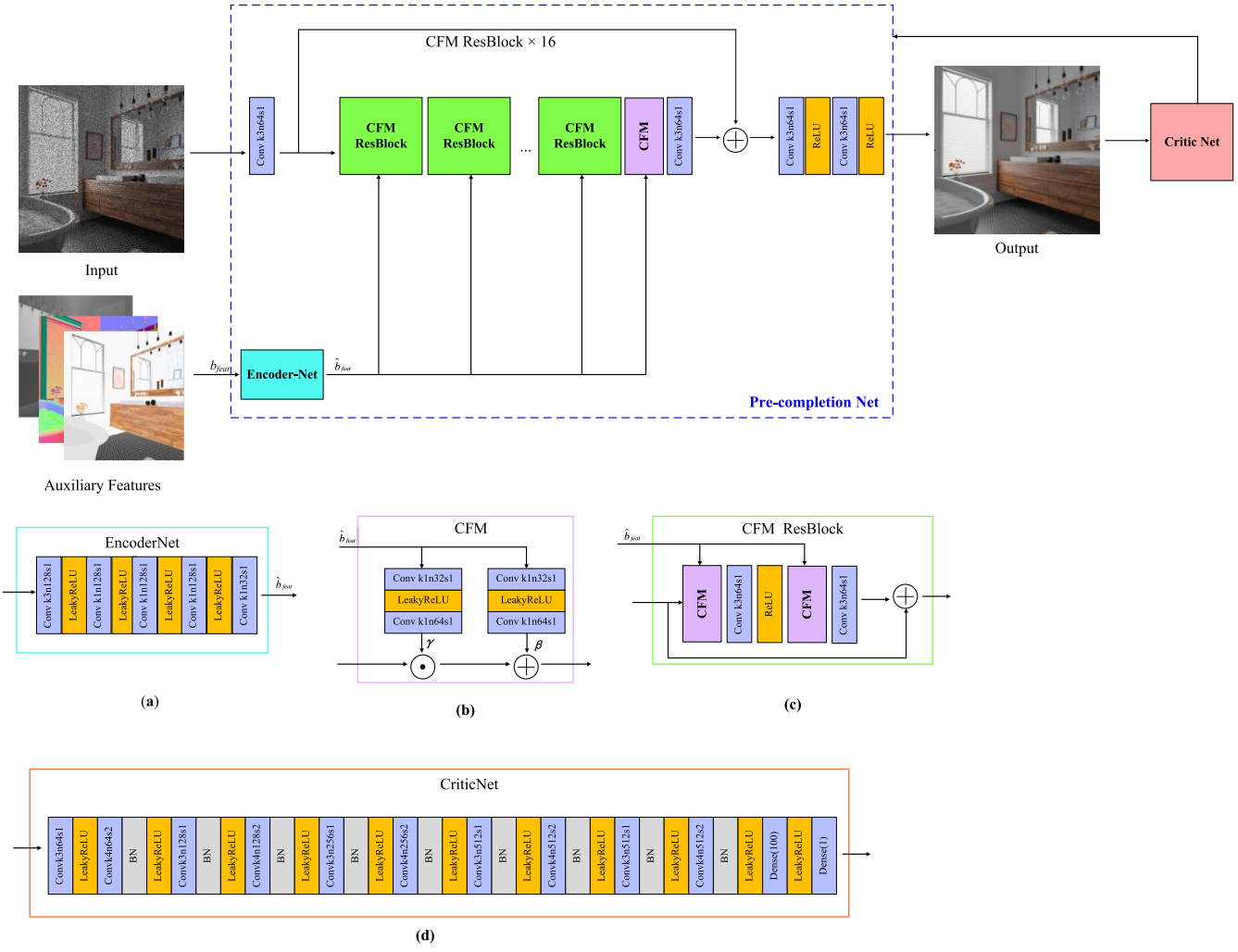


FIGURE 3. The framework of our GAN. (a) Encoder Net; (b) Illustration of the conditioned feature modulation; (c) Illustration of the residual block(ResBlock) for conditioned feature modulation; (d) Critic Net. Interpretation of network layer annotations: e.g., k3n128s1 indicates that kernel size is 3, number of feature channels is 128 and step is 1.

contains 4 terms, texture loss term(seen as Eq. 4), perceptual loss term(seen as Eq. 5), sharpness loss term(seen as Eq. 6) and similarity loss term (seen as Eq. 7) respectively. F($\boldsymbol{c}_{out}$) is the result of pre-completion and it can be computed as follows [37]:

$$F_l(\boldsymbol{c}_{out}) = G\left(\boldsymbol{c}_{in}, \boldsymbol{b}_{feat}\right). \tag{2}$$

Here $\boldsymbol{c}_{out}$ is the corresponding output images of the incomplete input $\boldsymbol{c}_{in}$. $\boldsymbol{b}_{feat}$ is the additional features, including the shading normal, depth, texture and visibility. We optimize G's parameters Θ using generative adversarial training [37]:

$$\min_{\Theta_G} \max_{\Theta_D} D\left(G\left(\boldsymbol{c}_{in}, \boldsymbol{b}_{feat}\right), \boldsymbol{c}_{gt}\right), \tag{3}$$

where D is the critic network with parameters. G and D are jointly trained to minimize the loss function.

Next, we would describe each term of our loss function:

The first sub-term of our hybrid loss function is the texture loss and it can be described as follows [38]:

$$L_{Tex}(G, D) = -E_{\boldsymbol{c}_{in}}\left[D\left(\boldsymbol{c}_{in}\right)\right] + E_{\boldsymbol{c}_{gt}}\left[D\left(G\left(\boldsymbol{c}_{gt}\right)\right)\right] + \lambda E_{\boldsymbol{c}_{out}}\left[\left(\left\|\nabla_{\boldsymbol{c}_{out}} D\left(\boldsymbol{c}_{out}\right)\right\|_2 - 1\right)^2\right], \tag{4}$$

where, the first two items represent the Wasserstein distance, the final one represents the gradient vanish item used for network normalization. G and D are the networks described above. E is a set of data samples with specific distribution. And λ is the penalty coefficient. Minimize texture loss can keep more texture details [38].

The second term of our loss function is the perceptual loss. It can be described as follows [39]:

$$L_{Per}(G) = E_{(\boldsymbol{c}_{in}, \boldsymbol{c}_{out})}\left[\frac{1}{whd}\left\|\phi\left(G\left(\boldsymbol{c}_{out}\right)\right) - \phi\left(\boldsymbol{c}_{in}\right)\right\|_F^2\right], \tag{5}$$

where ϕ is the feature extractor and $||\cdot||$ is the *Frobenius* norm. w, h, d represent the width, height and depth, respectively.

The third one is the sharpness loss. It is described in mathematical form as [40]:

$$L_{Sharp}(G) = E_{(\boldsymbol{c}_{in}, \boldsymbol{c}_{out})}\left[\left\|S\left(G\left(\boldsymbol{c}_{out}\right)\right) - S\left(\boldsymbol{c}_{in}\right)\right\|_2\right], \tag{6}$$

Here, $||\cdot||2$ is the L2 distance.

The final one is the similarity loss term. We set the loss function with the SSIM(structural similarity index) as follows [39]:

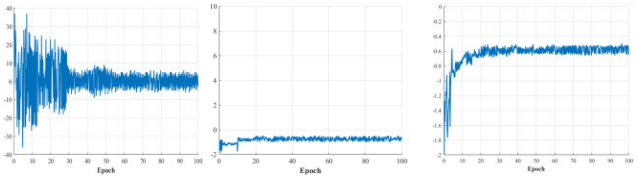
$$L_{SSIM}(G) = 1 - SSIM\left(\boldsymbol{c}_{out}, \boldsymbol{c}_{in}\right). \tag{7}$$

In summary, Eq. 1 can be rewritten as follows [40]:

$$L(\Theta) = \frac{1}{n}\sum_{i=1}^{n}\left[\alpha L_{Tex} + \beta L_{Per} + \gamma L_{Sharp} + \omega L_{SSIM}\right]. \tag{8}$$

where α, β, γ, ω are weight coefficients of the above four terms. During our experiments, we set them as $\alpha = 0.005$, $\beta = 0.0095$, $\gamma = 0.95$, $\omega = 0.095$, $\lambda = 10$(in Eq. 4).

Our comment is based on Wasserstein [40] with a gradient penalty, which enables stable training of a wide variety of GAN architectures with almost hyper-parameter tuning. Furthermore, Wasserstein distance has less restriction on balancing the training process of the generator and the critic, making it possible to pre-train the latter on large-scale datasets first, and then, fine tuning it on the small render dataset. We compared the Wasserstein distance, L_1 distance and L_2 distance when training. The results can be seen in **FIGURE 4**.



a. L_2 Distance *b*. L_1 Distance *c*. Wasserstain Distance

FIGURE 4. The training curves for L_2 distance, L_1 distance and Wasserstein distance show the pre-completion GAN converges.

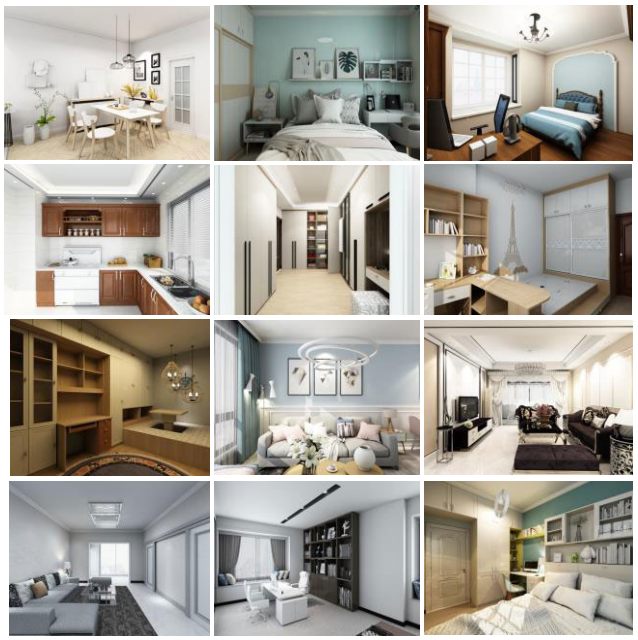


FIGURE 5. Example reference images rendered from 1000 indoor scenes.

From the **FIGURE 4**, we can find that the Wasserstein distance could be convergence faster and with less up-and-down-motion.

2) TRAINING DATASET

Large-scale datasets are necessary to avoid over-fitting for deep neural networks. In order to train our generating critic network, we commercial render, including 900 for training and 100 for validation. These scene frames were selected from diverse room designs with abundant illumination conditions as well as various materials and geometries [35], which span different generating circumstances (see example scenes in **FIGURE 5**). The reference images for training were rendered with 16k *spp*(samples per pixel). Since the state-of-the-art methods use public scenes as testing or part of training data, we also download datasets released by Tungsten [27]

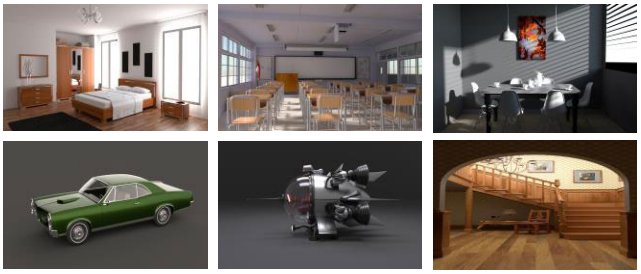


FIGURE 6. Example reference images rendered by Tungsten. The whole set includes approximately 200 images with modified environment maps, camera parameters, and/or materials from each of 6 scenes.

for evaluation. Example scenes of this dataset are seen in **FIGURE 6**.

B. CONSTRUCTION THE LOW-RANK MATRIX BASED ON SIMILAR PIXEL PATCH

After pre-completion, we create a pixel patch $\boldsymbol{P}_p(i,j)$ in pixel(i, j) having $n_p \times m_p$ pixels (e.g. green square in FIGURE 2), and the similar pixel patches $\boldsymbol{P}_p s$(e.g. red squares in FIGURE 2) are computed by weighted L_2 distance. In the weighted scheme, we assign more weights for rendered pixels than initialized pixels from the pre-completion. The selected similar patches are stored in each column of a low rank matrix $\boldsymbol{M}_p$ as in FIGURE 2, and the missing values in the matrix is estimated by our low rank matrix recovery step(see Part III). After completing missing values in patch $\boldsymbol{P}_p(i,j)$, the next patch $\boldsymbol{P}_p(i+\delta_p, j)$ will be iteratively processed where $1 \leq \delta_p < \min(n_p, m_p)/2$.

C. LOW RANK MATRIX RECOVERY

1) PAWNNM STRATEGY

As described in Ref [31], missing values in the low rank matrix can be recovered using the SVT algorithm. We mark the incomplete low rank matrix as M and the original matrix as $\boldsymbol{X}$ to be approximated. The recover problem can be defined as a low rank matrix minimization [41]:

$$\begin{aligned} &\min_{X} \ rank\ (\boldsymbol{X}) \\ &\text{s.t. } \boldsymbol{X}_{ij} = \boldsymbol{M}_{ij} \quad (i,j) \in \Omega, \end{aligned} \tag{9}$$

where the Ω is the collection containing the positions of the observed samples. Although solving the rank minimization is NP-hard, it is indicated that this problem had a convex relaxation as a nuclear norm minimization(NNM) problem [41]:

$$\begin{aligned} &\min_{X} \|\boldsymbol{X}\|_* \\ &\text{s.t. } \boldsymbol{X}_{ij} = \boldsymbol{M}_{ij} \quad (i,j) \in \Omega, \end{aligned} \tag{10}$$

where the nuclear norm is defined as [42]:

$$\|\boldsymbol{X}\|_* = \sum_{j=1}^{n} \sum\nolimits_j (\boldsymbol{X}). \tag{11}$$

Here the $\sum_j (\boldsymbol{X})$ are singular values of matrix $\boldsymbol{X}$. Next we use the SVT algorithm to rewritten the Eq. 9 as follows [31]:

$$\begin{aligned} &\min_{X} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{M}\|_F^2 + \xi \|\boldsymbol{X}\|_* \\ &\text{s.t. } P_\Omega (\boldsymbol{X}) = P_\Omega (\boldsymbol{M}), \end{aligned} \tag{12}$$

where the $\|\|_F^2$ is the Frobenius norm and P_Ω is the orthogonal projector onto the span of matrix vanishing outside of collection, so that the pixel(i,j) of $P_\Omega (\boldsymbol{X})$ is equal to the one of $\boldsymbol{X}$ and zero otherwise. ξ is a positive weight. We assume the solution of the Eq. 12 is as follows [31]:

$$\boldsymbol{X} = \arg\min_{X} \frac{1}{2} \|\boldsymbol{M} - \boldsymbol{X}\|_F^2 + \lambda \|\boldsymbol{X}\|_*. \tag{13}$$

Based on matrix completion theory, the solution can be obtained by [42]

$$\boldsymbol{X} = \boldsymbol{U} \boldsymbol{S}_\lambda (\Sigma) \boldsymbol{V}^T, \tag{14}$$

where $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the SVD of matrix $\boldsymbol{M}$, and $\mathrm{S}_\lambda(\ \boldsymbol{\Sigma})$ is the soft-thresholding function on the diagonal matrix with the constant weight ξ. For each element Σ_{ii} in diagonal matrix $\boldsymbol{\Sigma}$, we set [18]

$$S_\lambda (\boldsymbol{\Sigma})_{ii} = \max (\boldsymbol{\Sigma}_{ii} - \xi, 0). \tag{15}$$

From the above, we can find that each singular value is treated equally, and the soft thresholding operator shrinks each singular value with a constant. In order to optimize high frequency details efficiently, we modify the constant weight ξ in Eq. 15 as multiple weights [42]:

$$\xi_i = {\omega_i}\Big/{\left(\sum\nolimits_{ii} (X) + \varepsilon\right)}. \tag{16}$$

where $\Sigma_{\mathrm{ii}}(\boldsymbol{X})$ is the i^{th} singular value of $\boldsymbol{X}$, is the weight for the i^{th} singular value, and ε is a small regularizer to avoid dividing by zero. But when implementing, there is improving space for the strategy. Ref [42] showed that the recovery quality can be increased with a reducing weight during every optimization loop. In other words, we utilize a parameter adjustment strategy. In math, the weight can be progressively reduced as follows [42]:

$$\omega_{i+1} = c\omega_i. \tag{17}$$

Here c is the constant.

2) RECOVERY BASED ON MATRIX COMPLETION

Algorithm 1 shows the whole recovery process, based on utilized PAWNNM strategy. The inputs are the incomplete rendered data $\boldsymbol{M}$, and the corresponding sampling mask $\boldsymbol{\Phi}$, that the renderer has used to determine which samples to take.

Note that the first step in each iteration is to construct a low rank matrix X, using the PAWNNM strategy to recover the missing data and we iterate this process to generate a good, reconstructed result.

Algorithm 1 Recovery Based on Matrix Completion

Input: incomplete rendered data $\boldsymbol{M}$, sampling mask $\boldsymbol{\Phi}$

1: **Begin**
2: Initialization: $x^0 = \boldsymbol{M}$, set threshold: $\omega = \omega_0$;
3: **for** $i = 1$: k **do** ***step*** 4-9:
4: Form a low rank matrix $\boldsymbol{X}_i$ based on x^i;
5: Singular value decomposition for each matrix $\boldsymbol{X}_i$: $[\boldsymbol{U}_i, \boldsymbol{\Sigma}_i, \boldsymbol{V}_i] = SVD(\boldsymbol{X}_i)$;
6: Utilize weighted singular value thresholding: $\Sigma_i = S_{\lambda_i}(\Sigma_i)$;
7: Obtain the results x^{i+1} by computing $\boldsymbol{X}_{i+1} = \boldsymbol{U}_i \boldsymbol{\Sigma}_i \boldsymbol{V}_i^T$;
8: Iterative updating: $x^{i+1} = \boldsymbol{\Phi M} + x^i - \boldsymbol{\Phi} x^i$;
9: Parameter shrinkage: $\omega_{i+1} = c\omega_i$
10: **end**

Output: The reconstructed result x^k

IV. RESULTS AND DISCUSSION

A. EXPERIMENTAL SETUP

Our method has been tested on various scenes including detailed scenes with high frequency content for which it is challenging to recover missing information without introducing blurry artifacts. In order to create partially sampled rendering images as well as the reference images, we implement our algorithm in C++ and integrated it into the open source render PBRT2 [43]. The rendered static images are 1280×960 pixels. All of our experiments were performed on a workstation with an Intel(R) Core(TM) i7-9700K 3.60GHz CPU with 16G RAM and NVDIA GeForce RTX 2080 GPU with 8G RAM. We set initial values by experiment. The initial values in Eq.(17) are $\omega_0 = 80.5$, and $c = 0.9$. The number of iterations in Algorithm 1 is set at $k = 45$.

B. RESULTS

Note that, we describe the advantages of our algorithm framework in three ways: (1) compared with the widely used denoising methods based on adaptive sampling and filtering; (2) compared with the state-of-the-art denoising methods based on convolutional neural network; (3) compared with the widely used reconstruction methods, especially with the similar method present by Liu *et al.* [18]. To describe the quality of the image, we utilize the MSE [44], SSIM [45], and PSNR [45].

1) COMPARED WITH THE PATH TRACING DENOISING METHODS BASED ON ADAPTIVE SAMPLING AND FILTERING

FIGURE 7 shows the comparison between our method and state-of-the-art path tracing denoising approaches based on adaptive sampling and filtering, including RHF [46], SBF [47] and RPF [48], on three scenes with different distributed effects.

First, we examine the path traced Kitchen scene with global illumination and many texture details [shown as **FIGURE 7(a)**]. The samples number of all images are 8spp. The image rendered by RHF is more blur than others and it does not preserve the essential details, like the texture on the wall(red rectangle), the metallic luster on the chandelier(green rectangle). Although the result of SBF could preserve more details than RHF, the image is seriously aliased, like at the edge region of the chandelier(green rectangle). The image rendered by RPF is as good as ours, but our result is of better quality(indicated under the images as PSNR and SSIM). The most important of our methods is the least time cost(indicated under the images).

The Gold scene is a challenging, path-traced scene containing one gold and a floor with texture [shown as **FIGURE 7(b)**]. RHF method cannot preserve the geometry and texture details in the scene. Moreover, the approaches do not preserve the glossy details on the gold and their time cost is too large. Our approach is not only the fastest, but it also preserves the textures while having fewer artifacts than the other techniques.

The Cabin scene is a path-traced scene that includes global illumination and depth of field [shown as **FIGURE 7(c)**]. RPF can smooth the image, but over blur the geometry of the steps and texture in the water. The result of SBF contain residual noise in the depth of field regions. Although both our approach and RPF handle the depth of field regions well, RPF produces visible artifacts in the smooth regions due to the low sampling rate. Meanwhile, we are able to generate a smooth result that is closer to ground truth.

2) COMPARED WITH THE PATH TRACING DENOISING METHODS BASED ON CONVOLUTIONAL NEURAL NETWORK

FIGURE 8 shows the comparison between our approach and state-of-the-art denoising methods based on convolutional neural network. For completeness, we perform the same sampling rate to ensure fairness.

The Toasters scene [shown in **FIGURE 8(a)**] is a path-traced one which includes soft shadow and depth of the field. The RAE and KPCN remove the noise in shadows, but over blurs the depth of field regions(green rectangle). Moreover, although both NFOR and our approach can generate the images of better quality, our sampling rate is the least of all methods. The Pool balls scene[shown in **FIGURE 8(b)**] is a challenging, path-traced containing five pool balls with motion blur and soft shadow. From the statistics, the results of all methods are of better quality. Despite having a lowest MSE and a highest SSIM, we produce a noise-free result with quarter samples compared with other methods. The Dragon scene [shown in **FIGURE 8(c)**] is a path-traced scene including a red dragon with global illumination. Unfortunately, none of the other methods can effectively avoid the edge aliasing(red rectangle) at low rate. Note that, the reference image still has visible light leak at the dragon claw region in red rectangle at 16K spp, while we produce a relatively noise-free result. Seen from the SSIM and MSE, our image is of better quality than the other methods.

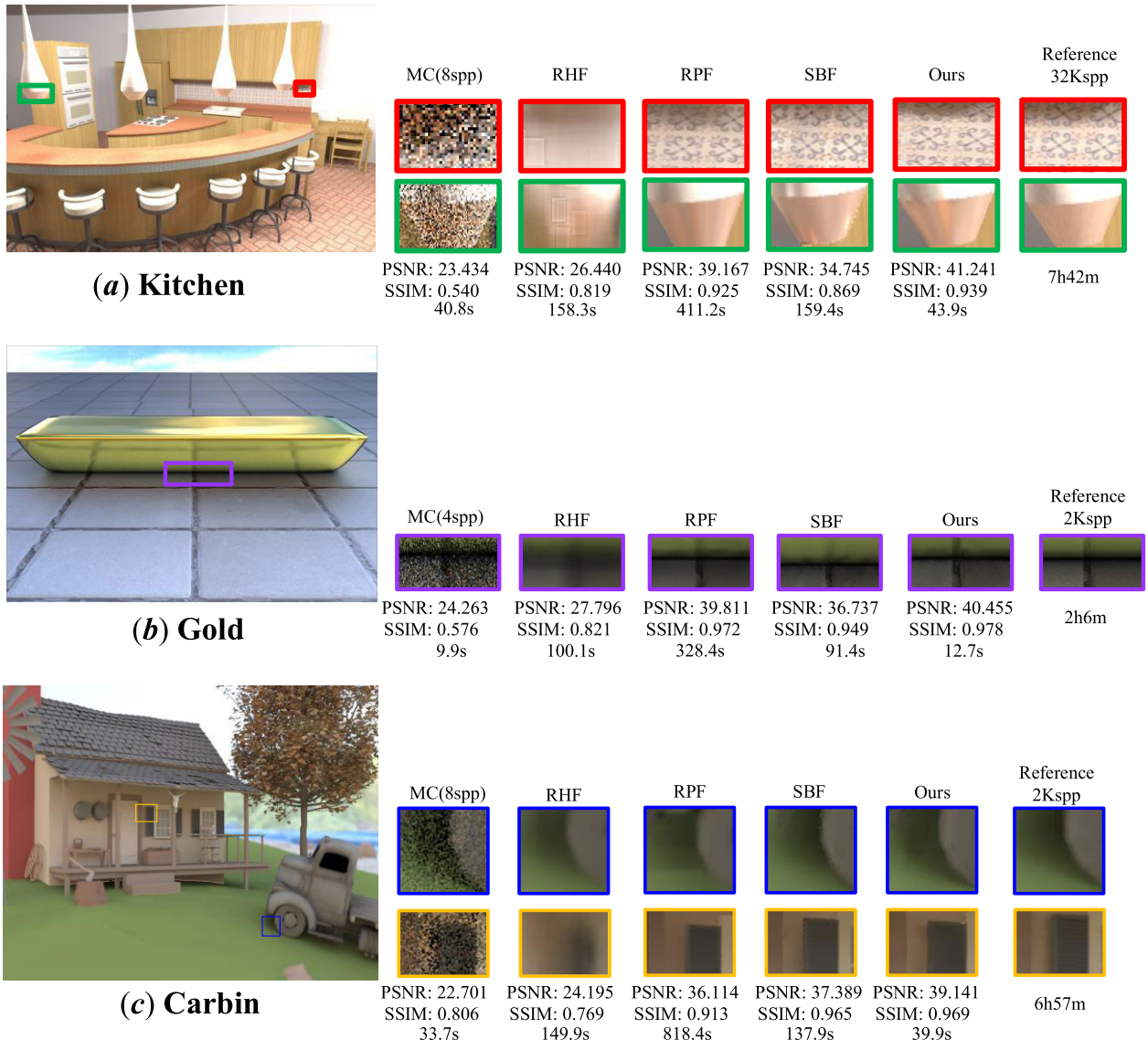


FIGURE 7. We compare our reconstruction results with the accelerated path tracing method based on denoising with adaptive sampling and filtering: RHF [46], RPF [48] and SBF [47] on a test set rendered scenes in 8spp or 4spp. Below the image, we calculate the value of PSNR and SSIM for measuring image quality.

3) COMPARED WITH OTHER PATH TRACING RECONSTRUCTION METHODS

Since the reconstruct technique based on compressive sensing produced by Liu *et al.* [18] is computationally expensive, we omit its comparison here.

And the reconstruction approach presented by Veach [19] could not generate the images of as good quality as other denoising methods, like KPCN [14], NFOR [27], RAE [49]. So, we improve its method. As can be seen in Figure 9, our MSE consistently decreases even at the low sampling rates. We choose three test scenes for the comparison between our approach and the Liu's method in the same low sampling rate. The results are shown in **FIGURE 9**.

From the results, the Conference Room scene [shown in **FIGURE 9(a)**] is a path-traced scene that includes direct illumination and indirect illumination. From the details of the chairs(yellow rectangle), the reconstruction image of Liu's approach is noisier and more aliasing than ours. The Villa scene [shown in **FIGURE 9(b)**] is a path-traced scene that contains global illumination, high light and glossy. Seen from the details of the bookshelf(red rectangle), although Liu's approach can preserve the details, there is noise in their image. Our result is smoother and clearer. Plants[shown in **FIGURE 9(c)**] is a path-traced scene with depth of field and much fog. The detail contrast among the Liu's approach and ours shows that their result is much over blurred and lower quality than ours. Besides, we compute the aggregate numerical performance of different input samples. Result can be seen as TABLE 1. Since the sampling randomness of path tracing and the influence of different effects, the MSE and SSIM metric is fluctuated. We calculated the average metric.

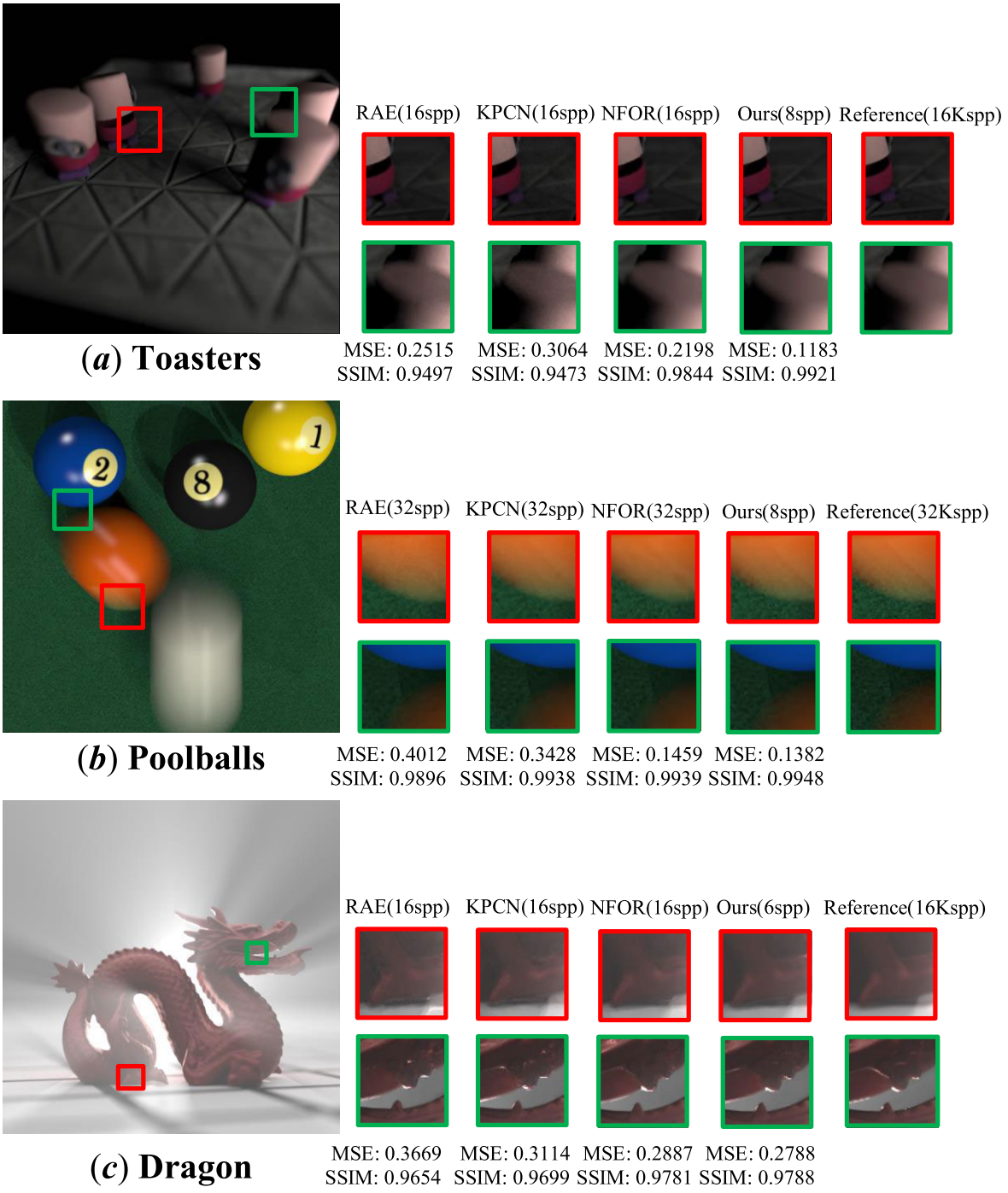


FIGURE 8. We compare our reconstruction results with the accelerated path tracing based on denoising methods with the convolutional neural network: NFOR [27], RAE [49] and KPCN [14] on a test set rendered scenes in 16spp or 32spp(our sampling rate is less than the others). Below the image, we calculate the values of MSE and SSIM to measure image quality.

From the result, perceptually and quantitatively, our method outperforms Liu's approach.

C. DISCUSSIONS AND ANALYSIS

In our method, we utilize the improved GAN to pre-complete the incomplete images rendered by path tracing and according to the results of pre-completed, construct the low rank matrix. Finally, we use the matrix completion theory to reconstruct the images. There are two differences between ours and other methods. First, as discussed in Part **III-*A***), the pre-completion based on GAN, which can reduce the error caused by noise and incomplete information and in some cases, a small error might result in obvious noisy or blurred pixels. Here we compared the reconstruction images with pre-completion and without pre-completion. The results can be seen in **FIGURE 10**.

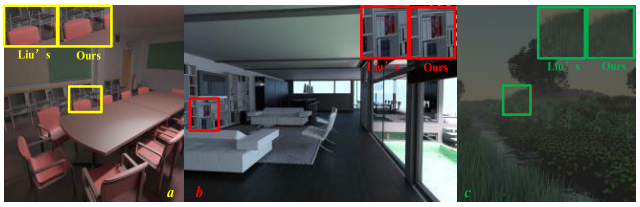


FIGURE 9. We compare our reconstruction results with the path tracing reconstruction method presented by Liu *et al.* [19] on a test set rendered scenes in 16spp or 32spp. Among these scenes, from left to right, scene a is the *Conference Room* rendered in 16spp. Scene b is the *Villa Room* rendered in 32spp and scene c is *Plants with Fog* rendered in 16spp. Our sampling rate is the same as Liu's method.

TABLE 1. Comparison of our results and LIU's with different input samples number.

Input samples rate	Approaches	SSIM($\times 10^{-2}$)	MSE($\times 10^{-3}$)
20%	ours	**89.48**	**0.313**
	Liu's	82.29	0.477
30%	ours	**95.31**	**0.094**
	Liu's	90.70	0.190
40%	ours	**98.28**	**0.027**
	Liu's	94.24	0.059

Note that, here we take the *Conference Room* scene[shown in Fig 10(a)] as the example scene. Aggregate numerical performance comparison between ours and Liu's on the example scene. We compute the SSIM and MSE values.

FIGURE 10. The reconstruction results comparison of our method with the pre-completion and without pre-completion. Take the San Miguel scene as an example.

As shown in **FIGURE 10**, with the pre-completion, we can obtain the better-quality images with enhanced texture. Because of the low rank matrix completion, with incomplete rendering information, we could not successfully construct the matrix and hardly to reconstruct the final images. Besides, our pre-completion process not only use the color information of pixels, but the auxiliary features, which can help us improve the accuracy of pre-completion and easily find the pixels of similar structure.

Since artifacts in an intermediate image of path tracing can be caused by insufficient samples, we claim that matrix completion is an efficient choice as a screen space reconstruction for previsualization because this process has no limitation of necessary samples number. Seen as **FIGURE 8**, our method provides an efficient image space solution to previsualize early-stage intermediate results with the lowest samples, which are minimal blurry artifacts in high

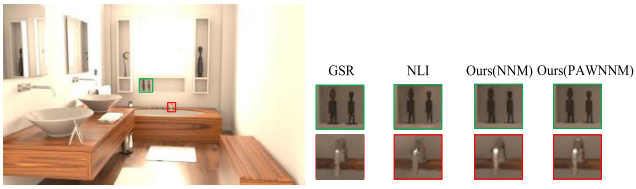


FIGURE 11. The reconstruction results comparison of our method with the pre-completion and without pre-completion. Take the San Miguel scene as an example.

TABLE 2. PSNR/SSIM results of comparison between the NLI, GSR, Our1(with NNM), Ours2(with PAWNNM) (the scene shown in Figure 12).

Sampling rate	NLI	GSR	Ours1	Ours2
20%	28.23/0.807	29.39/0.884	30.02/0.882	30.29/0.907
30%	31.48/0.905	32.02/0.916	33.15/0.944	33.74/0.958
40%	33.83/0.953	34.17/0.961	36.06/0.968	37.23/0.979

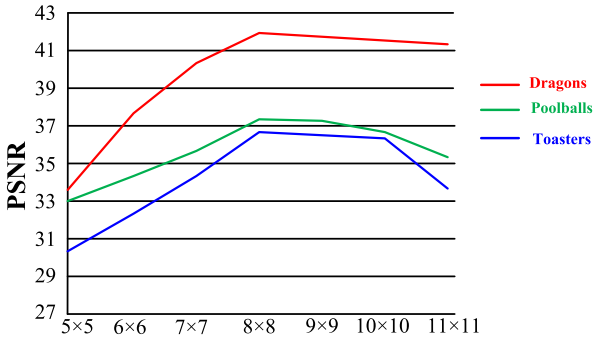


FIGURE 12. The corresponding PSNR of our generation images on three test scenes shown in Figure 9 with different patch size choices. Notice that, the bigger the value of PSNR, the better the images.

frequency details and to reduce spatial noise. Furthermore, our utilized PAWNNM strategy shows promising results for inpainting providing better recovery. Here, we compared ours with recent methods with NLI strategy [33] and GSR strategy [50]. The results can be seen in **FIGURE 11** and we calculated the values of PSNR and SSIM for each generated image, whose results are shown in TABLE 2.

From the above experiments results, our strategy outperforms other strategy. Furthermore, the size of pixel patch also results in the quality of the final images. The relationship between the patch size and PSNR of images is shown as **FIGURE 12**. So, after reconstructing values in a pixel patch, we process the next pixel patch while overlapping the patches by two pixels to avoid block artifacts. The size of pixel patches are chosen by experiment to balance computation time and quality.

Besides, to show the advantages of our methods at low sampling rate and without the limitation of minimum samples number, we compared our approach with the Liu's method [19], RAE [49], KPCN [14] and NFOR [27] to reconstruct the images in our dataset. We plot the convergence curve of PSNR metric on our validation set at different sampling rate. Particularly, the PSNR values and reconstruction error values represent the average values on our datasets. The

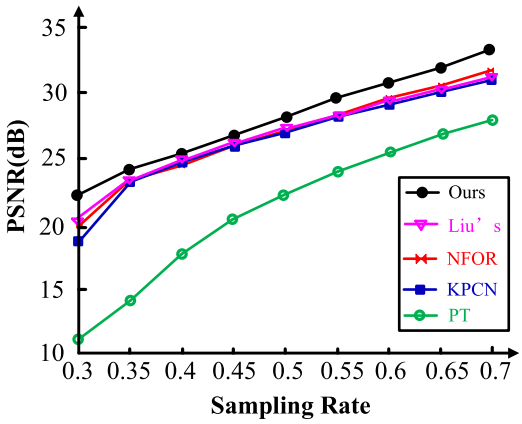


FIGURE 13. The average PSNR of our results compared with Liu's method [19], NFOR [27], KPCN [14] and original path tracing(PT) on our validation datasets. Notice that, the bigger the value of PSNR, the better the images. Seen from the figure, we can find that at low sampling, especially at 30% sampling rate, our results outperform in images quality. And with samples number increases, our images are better than other methods.

results are shown in **FIGURE 13**. With the comparison, it is easy to tell that, at low sampling rate, our method outperforms than other methods.

To show the advantages of our network, especially the convergence speed and convergence efficiency, we applied our network, the Liu's network [19], the network of NFOR [27] and RAE [49] to pre-complete our training datasets. Because KPCN [14] utilize the CNN to predict the filter kernel parameters instead of pixel results, we did not compare our network with them. We evaluated the average normalized mean squared error(MSE) values of our validation datasets when the network was convergent and plot the convergence curve of MSE metric during the training process. The results are shown in **FIGURE 14**. Seen from the Figure, although the RAE's network converges faster a little than ours, our network outperforms better than other networks in convergence efficiency. And difference of convergence speed is little.

D. LIMITATION AND FUTURE WORK

Although we show promising results, our method has some limitations. The performance of matrix completion relies on the size and number of low-rank matrix. Over a few experiments, we found the appropriate patch sizes and an appropriate number of similar patches in an image, but careful analysis of scene information may provide better optimal choices. However, it is challenging to do this, because in general, detailed scene information cannot be known before a significant number of samples have been taken.

On the other hand, the current implementation of our recovery is not efficient compared with the denoising methods based on convolutional neural network in terms of the total time, although our samples number is less than other methods. In the future, there are three major ways in which we could feasibly improve our performance: (1) our current implementation is based on Python without GPU parallel optimization, (2) our code is without necessary code optimization and (3) we can introduce the optimal patch size chosen process.

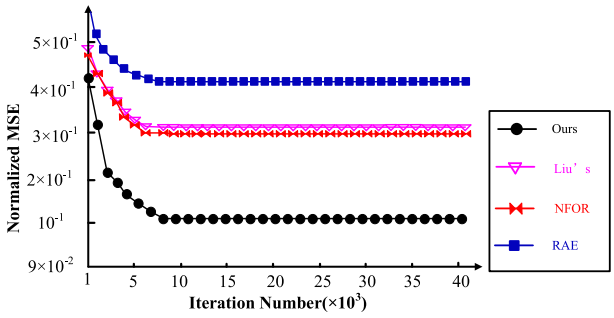


FIGURE 14. The average Normalized MSE of our results compared with Liu's method [19], NFOR [27] and RAE [49] on our training datasets. Notice that, the smaller the value of normalized MSE, the better the images. Seen from the figure, we can find that when networks were convergent, our MSE value is the smallest of all methods. Although RAE's network(indicated with blue lines and rectangle) converges faster than ours, their MSE values are biggest of all methods. That is to say, their images are of least quality. On the contrary, our images are of best quality and the difference of convergence speed between ours and RAE is little. On the whole, our network outperforms in convergence efficiency and convergence speed.

V. CONCLUSION

To overcome the low efficiency shortcomings of original path tracing and avoid the limitation of the minimum necessary samples number, we present a novel accelerated path tracing method with the GAN and the matrix completion. First, we randomly render part of pixels of input images with path tracing. Then a modified GAN provides fast pre-completion for initializing missing values and represent the pre-completed images as the low-rank matrix. Next, subsequent weighted nuclear norm minimization with an improved parameter adjustment strategy(PAWNNM) efficiently recovers missing values even in high frequency details. From the results, we can find that our method performs better than the state-of-the-art accelerated methods in visual quality and image details. Our proposed approach offers a new way of thinking about the path tracing acceleration problem and might provide fruitful avenues for exploration in the future.

REFERENCES

[1] B. Burley, D. Adler, M. J.-Y. Chiang, H. Driskill, R. Habel, P. Kelly, P. Kutz, Y. K. Li, and D. Teece, "The design and evolution of disney's hyperion renderer," *ACM Trans. Graph.*, vol. 37, no. 3, Jul. 2018, Art. no. 33.

[2] H. Dahlberg, D. Adler, and J. Newlin, "Machine-learning denoising in feature film production," in *Proc. ACM SIGGRAPH Talks (SIGGRAPH)*, Los Angeles, CA, USA, Jul. 2019, Art. no. 21.

[3] L. Fascione, J. Hanika, M. Fajardo, P. Christensen, B. Burley, and B. Green, "Path tracing in production—Part 1: Production renderers," in *Proc. ACM SIGGRAPH Courses (SIGGRAPH)*, Los Angeles, CA, USA, Jul. 2017, Art. no. 13.

[4] S. McAuley, S. Hill, N. Hoffman, Y. Gotanda, B. Smits, B. Burley, and A. Martinez, "Practical physically-based shading in film and game production," in *Proc. ACM SIGGRAPH Courses (SIGGRAPH)*, Los Angeles, CA, USA, Aug. 2012, Art. no. 10.

[5] P. H. Christensen, G. Harker, J. Shade, B. Schubert, and D. Batali, "Multiresolution radiosity caching for global illumination in movies," in *Proc. ACM SIGGRAPH Talks (SIGGRAPH)*, Los Angeles, CA, USA, Aug. 2012, Art. no. 47.

[6] 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, Aug. 2018, Art. no. 124.

[7] M. Kettunen, M. Manzi, M. Aittala, J. Lehtinen, F. Durand, and M. Zwicker, "Gradient-domain path tracing," *ACM Trans. Graph.*, vol. 34, no. 4, Jul. 2015, Art. no. 123.
[8] A. Pilleboue, G. Singh, D. Coeurjolly, M. Kazhdan, and V. Ostromoukhov, "Variance analysis for Monte Carlo integration," *ACM Trans. Graph.*, vol. 34, no. 4, Jul. 2015, Art. no. 124.
[9] W. Jakob and S. Marschner, "Manifold exploration: A Markov chain Monte Carlo technique for rendering scenes with difficult specular transport," *ACM Trans. Graph.*, vol. 31, no. 4, Aug. 2012, Art. no. 58.
[10] A. S. Kaplanyan, J. Hanika, and C. Dachsbacher, "The natural-constraint representation of the path space for efficient light transport simulation," *ACM Trans. Graph.*, vol. 33, no. 4, Jul. 2014, Art. no. 102.
[11] J. Lehtinen, T. Karras, S. Laine, M. Aittala, F. Durand, and T. Aila, "Gradient-domain metropolis light transport," *ACM Trans. Graph.*, vol. 32, no. 4, Jul. 2013, Art. no. 95.
[12] M. Zwicker, W. Jarosz, J. Lehtinen, B. Moon, R. Ramamoorthi, F. Rousselle, P. Sen, C. Soler, and S.-E. Yoon, "Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering," *Comput. Graph. Forum*, vol. 34, no. 2, pp. 667–681, May 2015.
[13] N. K. Kalantari, S. Bako, and P. Sen, "A machine learning approach for filtering Monte Carlo noise," *ACM Trans. Graph.*, vol. 34, no. 4, Jul. 2015, Art. no. 122.
[14] 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, Jul. 2017, Art. no. 97.
[15] 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, Jul. 2017, Art. no. 98.
[16] Q. Xing and C. Chen, "Path tracing denoising based on SURE adaptive sampling and neural network," *IEEE Access*, vol. 8, pp. 116336–116349, May 2020.
[17] P. Sen and S. Darabi, "Compressive rendering: A rendering application of compressed sensing," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 4, pp. 487–499, Apr. 2011.
[18] P. Liu, J. Lewis, and T. Rhee, "Low-rank matrix completion to reconstruct incomplete rendering images," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 8, pp. 2353–2365, Aug. 2018.
[19] E. Veach, "Robust Monte Carlo methods for light transport simulation," Ph.D. dissertation, Dept. Elect. Eng., Stanford Univ., Palo Alto, CA, USA, 1997.
[20] R. Ramamoorthi, D. Mahajan, and P. Belhumeur, "A first-order analysis of lighting, shading, and shadows," *ACM Trans. Graph.*, vol. 26, no. 1, p. 2, Jan. 2007.
[21] W. Jarosz, V. Schönefeld, L. Kobbelt, and H. W. Jensen, "Theory, analysis and applications of 2D global illumination," *ACM Trans. Graph.*, vol. 31, no. 5, pp. 1–21, Aug. 2012.
[22] X. Yang, D. Wang, W. Hu, L.-J. Zhao, B.-C. Yin, Q. Zhang, X.-P. Wei, and H. Fu, "DEMC: A deep dual-encoder network for denoising Monte Carlo rendering," *J. Comput. Sci. Technol.*, vol. 34, no. 5, pp. 1123–1135, Sep. 2019.
[23] P. Bauszat, M. Eisemann, and M. Magnor, "Guided image filtering for interactive high-quality global illumination," *Comput. Graph. Forum*, vol. 30, no. 4, pp. 1361–1368, Jun. 2011.
[24] F. Rousselle, M. Manzi, and M. Zwicker, "Robust denoising using feature and color information," *Comput. Graph. Forum*, vol. 32, no. 7, pp. 121–130, Oct. 2013.
[25] B. Moon, N. Carr, and S.-E. Yoon, "Adaptive rendering based on weighted local regression," *ACM Trans. Graph.*, vol. 33, no. 5, Sep. 2014, Art. no. 170.
[26] P. Bauszat, M. Eisemann, E. Eisemann, and M. Magnor, "General and robust error estimation and reconstruction for Monte Carlo rendering," *Comput. Graph. Forum*, vol. 34, no. 2, pp. 597–608, May 2015.
[27] 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.
[28] P. Peers, D. K. Mahajan, B. Lamond, A. Ghosh, W. Matusik, R. Ramamoorthi, and P. Debevec, "Compressive light transport sensing," *ACM Trans. Graph.*, vol. 28, no. 1, Jan. 2009, Art. no. 3.
[29] J. Gu, S. K. Nayar, E. Grinspun, P. N. Belhumeur, and R. Ramamoorthi, "Compressive structured light for recovering inhomogeneous participating media," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, p. 1, Mar. 2013.
[30] E. Miandji, J. Kronander, and J. Unger, "Compressive image reconstruction in reduced union of subspaces," *Comput. Graph. Forum*, vol. 34, no. 2, pp. 33–44, May 2015.
[31] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.
[32] Y. Huo, R. Wang, S. Jin, X. Liu, and H. Bao, "A matrix sampling-and-recovery approach for many-lights rendering," *ACM Trans. Graph.*, vol. 34, no. 6, Nov. 2015, Art. no. 210.
[33] W. Li, L. Zhao, Z. Lin, D. Xu, and D. Lu, "Non-local image inpainting using low-rank matrix completion," *Comput. Graph. Forum*, vol. 34, no. 6, pp. 111–122, Sep. 2015.
[34] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imag. Sci.*, vol. 1, no. 3, pp. 248–272, Jan. 2008.
[35] 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, Nov. 2019, Art. no. 224.
[36] I. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," 2017, *arXiv:1701.00160*. [Online]. Available: https://arxiv.org/abs/1701.00160
[37] W. Lotter, G. Kreiman, and D. Cox, "Unsupervised learning of visual structure using predictive generative networks," 2015, *arXiv:1511.06380*. [Online]. Available: http://arxiv.org/abs/1511.06380
[38] Q. Yang, P. Yan, Y. Zhang, H. Yu, Y. Shi, X. Mou, M. K. Kalra, Y. Zhang, L. Sun, and G. Wang, "Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1348–1357, Jun. 2018.
[39] Z. Shi, J. Li, Q. Cao, H. Li, and Q. Hu, "Low-dose spectral CT denoising method via a generative adversarial network," *J. Jilin Univ., Eng. Technol. Ed.*, vol. 49, no. 5, pp. 1–10. Sep. 2019.
[40] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: https://arxiv.org/abs/1701.07875
[41] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, Apr. 2009.
[42] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *Int. J. Comput. Vis.*, vol. 121, no. 2, pp. 183–208, Jul. 2016.
[43] M. Pharr and G. Humphreys, *Physically Based Rendering: From Theory to Implementation*. San Mateo, CA, USA: Morgan Kaufmann, 2010.
[44] F. Rousselle, C. Knaus, and M. Zwicker, "Adaptive sampling and reconstruction using greedy error minimization," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 1–12, Dec. 2011.
[45] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[46] M. Delbracio, P. Musé, A. Buades, J. Chauvier, N. Phelps, and J.-M. Morel, "Boosting Monte Carlo rendering by ray histogram fusion," *ACM Trans. Graph.*, vol. 33, no. 1, Jan. 2014, Art. no. 8.
[47] T.-M. Li, Y.-T. Wu, and Y.-Y. Chuang, "SURE-based optimization for adaptive sampling and reconstruction," *ACM Trans. Graph.*, vol. 31, no. 6, Nov. 2012, Art. no. 194.
[48] P. Sen and S. Darabi, "On filtering the noise from the random parameters in Monte Carlo rendering," *ACM Trans. Graph.*, vol. 31, no. 3, May 2012, Art. no. 18.
[49] N. Divakar and R. V. Babu, "Image denoising via CNNs: An adversarial approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 1076–1083.
[50] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3336–3351, Aug. 2014.