PBR-Net: Imitating Physically Based Rendering Using Deep Neural Network

Peng Dai, *Student Member, IEEE*, Zhuwen Li, *Member, IEEE*, Yinda Zhang, *Member, IEEE*, Shuaicheng Liu, *Member, IEEE*, and Bing Zeng, *Fellow, IEEE*

***Abstract*— Physically based rendering has been widely used to generate photo-realistic images, which greatly impacts industry by providing appealing rendering, such as for entertainment and augmented reality, and academia by serving large scale high-fidelity synthetic training data for data hungry methods like deep learning. However, physically based rendering heavily relies on ray-tracing, which can be computational expensive in complicated environment and hard to parallelize. In this paper, we propose an end-to-end deep learning based approach to generate physically based rendering efficiently. Our system consists of two stacked neural networks, which effectively simulates the physical behavior of the rendering process and produces photo-realistic images. The first network, namely shading network, is designed to predict the optimal shading image from surface normal, depth and illumination; the second network, namely composition network, learns to combine the predicted shading image with the reflectance to generate the final result. Our approach is inspired by intrinsic image decomposition, and thus it is more physically reasonable to have shading as intermediate supervision. Extensive experiments show that our approach is robust to noise thanks to a modified perceptual loss and even outperforms the physically based rendering systems in complex scenes given a reasonable time budget.**

***Index Terms*— Physically based rendering, intrinsic image, stacked neural network, shading, modified perceptual loss.**

I. Introduction

PHYSICALLY based rendering (PBR) has been widely used to generate photo-realistic color images, which are in high demand for entertainment industry. While deep learning has been demonstrated to be very successful for many vision problems, PBR also facilitates the neural network training by contributing natural looking synthetic color images [1]. However, PBR can be prohibitively computational expensive, and the rendering procedure could take up to hours to converge especially for indoor environment with complicated illumination and geometry which is super hard for ray tracing. Even at a reasonable trade-off between rendering time and quality, generating a large-scale synthetic dataset using PBR took up to 1 month on a cluster with hundreds machines holding 56-core CPU [1]. In this paper, we propose a deep learning based framework that learns to produce high-quality PBR images in nearly real-time.

We speed up PBR by simulating the majority expensive component efficiently with deep learning. Inspired by intrinsic image decomposition, a common photo can be decomposed into reflectance and shading components, where the reflectance can be rendered fast, usually called albedo, but the shading requires considerable amount of computations through ray tracing. Therefore, we propose to train a neural network to predict the shading and then combine it with the efficiently rendered albedo to produce photo-realistic color images. Comparing to a naive end-to-end black box, our network focuses more on the illumination without the distraction from color variance. More particularly, we address the following challenges.

The first challenge is how to estimate shading efficiently. Essentially shading is a 2D map and can be calculated through a integral operation [2], [3], which reflects the interplay between illumination and objects in traditional rendering. However, the traditional method is computational expensive and time-consuming because of the ray bounce, inter-reflection, etc. Given ground truth, i.e. physically based rendered shading using unlimited running time, estimating shading can be formulated into a fully convolutional neural network taking the required information, such as the surface normal, depth, and illumination as input, and trained with a ℓ_1 loss with regard to the ground truth. However we find the model trained in this way produces blurry results as shown in the experimental section. Inspired by Chen and Koltun [4], we utilize the perceptual loss from a pre-trained network such as VGG-19 [5] on ImageNet [6], which successfully removes the artifacts but produces results with strong grid patterns. We suspect a possible reason could be that the losses defined on high-level perceptual features are in overly small resolution, which causes mosaic artifacts up to the size of receptive field in the output image. To address this issue, we remove the high-level perceptual layers from the loss, and in practice we found a combination of the first 3 layers that gives satisfying performance.

Manuscript received August 25, 2018; revised July 22, 2019; accepted March 24, 2020. Date of publication April 16, 2020; date of current version April 29, 2020. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61872067, Grant 61872068, and Grant 61720106004, in part by the "111" Projects under Grant B17008, and in part by the Sichuan Science and Technology Program under Grant 2019YFH0016 and Grant 2018GZ0071. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Vishal Monga. *(Corresponding author: Shuaicheng Liu.)*

Peng Dai, Shuaicheng Liu, and Bing Zeng are with the Institute of Image Processing, School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: liushuaicheng@uestc.edu.cn).

Zhuwen Li is with Nuro Inc., Mountain View, CA 94043 USA.

Yinda Zhang is with Google Research, Mountain View, CA 94043 USA.

Digital Object Identifier 10.1109/TIP.2020.2987169

Given shading predicted for the camera viewpoint, how to efficiently combine it with the reflectance is yet another challenge. A straightforward way is to multiply them directly and performs a tone-mapping or linear projection [7]–[9]. However, this approach is sensitive to the noise in the estimated shading, and only create up to an approximation [2], [10]. In contrast, we train a composition network, which takes the reflectance and shading as input and directly produce the color image. We found that this composition network works significantly better than manually designed visualization algorithms and is more robust against noise.

The contributions of this paper are mainly in four aspects. First, we propose a deep learning framework that efficiently generate high quality PBR up to nearly real time. Second, we empirically find a combination of different layers in perceptual loss to help avoiding artifacts in the result. Third, our network estimates shading rather than the rendered image directly, which allows the network to tackle the most computational expensive component of the rendering process and focus on generating shading without distraction. Last, we train a network to combine shading and reflectance for the final color image, which generates higher-quality results comparing to traditional methods.

II. Related Work

A. Physically Based Rendering

PBR is a rendering algorithm based on the physical properties of light in the real world. The theory of PBR was elaborated by Pharr *et al.* [11]. Recently, due to the revolution of data hungry methods, such as deep learning, the demand for efficient rendering goes up for preparing large scale synthetic training set [1].

Recently, deep learning has been used to speed up PBR. Chaitanya *et al.* [12] synthesis images with strong noise quickly by using a small sample rates; after that, a network is designed to realize operation of denoising. Different from our work, their work relies on traditional rendering engine and pays more attention on image post-processing (denoising), while we design a physics-driven system that imitates the whole rendering process.

B. Photo-Realistic Image Generation

Our work is also related and inspired by previous work for photo-realistic image generation. Photo-realistic images are widely used and desired in many fields (e.g., film production, super-resolution, interior design). Beers *et al.* [13] generated photo-realistic and super-resolution facial images from random vectors through a progressively grown generative adversarial network. Luan *et al.* [14] and Mechrez *et al.* [15] transfer photo-realistic style into content image to generate another photo-realistic image in deep learning. Chen and Koltun [4] generate photo-realistic images from semantic layouts through a cascade refine network (CRN). And their work is further promoted by Wang *et al.* [16] and Qi *et al.* [17]. Besides, some other networks (e.g. pix2pix [18], CycleGAN [19]) are also used to produce photo-realistic images, for example translating sketch to photo-realistic images. Compared to these image generation work, our model takes necessary data for rendering as input and shows more respect to the physical procedure of the rendering in network architecture design.

C. Intrinsic Image Decomposition

Given an observed image (**I**), it can be decomposed into a reflectance (**R**) image multiplying a shading (**S**) image pixel-by-pixel. Without further constraints, the decomposition is a highly ill-posed problem in that the number of unknown variables (**R,S**) are twice the known values (**I**). The classic retinex algorithm is first introduced by Land and McCann [20], which analyzes local image deviations in shading and reflectance, following which different assumptions and priors were proposed [21]–[23]. In particular, some methods [24], [25] tried to learn the priors to judge the image derivatives, while others proposed additional constraints to reduce the number of unknowns [22], [26]. With the success of deep learning, high quality decomposition results were reported by various deep approaches. Baslamisli *et al.* [27] combines the physics-based reflection model, reflectance and shading gradients in deep learning capacities for improved performances. Janner *et al.* [9] proposed a Rendered Intrinsics Network (RIN) which can predict reflectance, shape, and lighting conditions given a single image. Lettry *et al.* proposed an end-to-end learning solution that can be trained without any ground truth supervision [28]. To deal with the lack of training data, Han *et al.* [29] synthesized training pairs with physical based renders. They feed the dataset to train a deep neural network for the decomposition and further fine-tune it for real-world images. In this work, we do composition rather than intrinsic decomposition. The intrinsic images – predicted optimal shading image for PBR and reflectance image are combined through a composition network.

III. Method

A. Overview

Given appropriate rendering resources, such as geometry, lights, albedo, etc., many off-the-shelve physically based renderers, such as Blender [30], Mitsuba [31] and Maya [32], can produce photo-realistic images that are not distinguishable from real-world photos if there is no running time limit. Our goal is to design a neural network architecture which takes these rendering resources as input and efficiently produces photo-realistic images of similar quality with that from PBR. Fortunately, most of the rendering input source data can be represented in 2D images, which allows us to use the well-known convolutional neural network (CNN) architecture. Specifically in our work, scene geometry is encoded in 2D depth and normal maps, denoted as $D \in \mathbb{R}^{w\times h\times 1}$ and $N \in \mathbb{R}^{w\times h\times 3}$ respectively; illumination is encoded in two 1-channel panoramic illumination images $L_d, L_i \in \mathbb{R}^{W\times H\times 1}$ with distance and intensity values; albedo is encoded in a reflectance map $R \in \mathbb{R}^{w\times h\times 3}$. Note that all of these sources can be rendered extremely fast through typical rasterization, and the time consumption is neglectable compared to the PBR itself.

(a) Generated shading (b) Reflectance (c) Generated image (d) Reference

Fig. 1. One example that demonstrates the reverse process of image intrinsic decomposition to generate rendering results. First, our network generate shading (a) from scene information. Then, it combines the shading and reflectance (b) with a composition network to automatically get approximate physically based rendering outputs (c). (d) Reference images that rendered by render engine (Mitsuba).

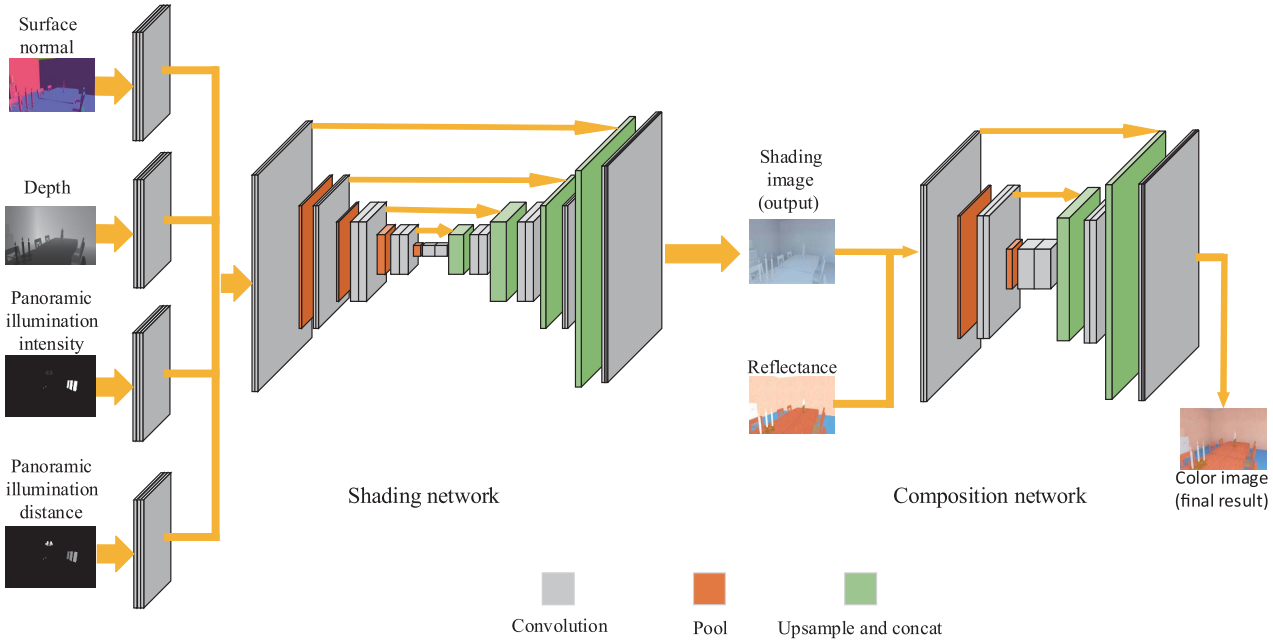


Fig. 2. Network architecture and workflow. Our network mainly consists of two sub-networks, i.e. shading network and composition network. Shading network synthesizes shading image using surface normal, depth, panoramic illumination (distance and intensity) as inputs. Composition network combines the generated shading image with the reflectance to produce a color image.

Providing the above information, we train an end-to-end neural network in a supervised way taking the PBR rendering, e.g. from Zhang *et al.* [1], as the ground truth. The overview of our approach and network architecture are illustrated in Figure 2. Our network consists of two major components. The first network receives a concatenation of D, N, L_d and L_i (e.g., all except R) and predicts the shading of the scene $S \in \mathbb{R}^{w \times h \times 3}$ (Figure 1 (a)), and the second network receives S (Figure 1 (a)) and R (Figure 1 (b)) to predict the final photo-realistic image (Figure 1 (c)), which is expected to be similar to the ground truth (Figure 1 (d)).

B. Shading Network and Composition Network

As mentioned before, both shading network and composition network can be formulated into fully convolutional neural network. We adopt U-Net with short-cut connection [34] as the backbone of our network and modify it to serve our purpose. As shown in Figure 2, the shading network starts from several independent convolution layers from each of the inputs to extract features, which are then concatenated and feed as input to the U-Net. For the composition network, it directly concatenates the output shading from the shading network with the reflectance map as the input to produce the final rendering. The composition network is shorter than the shading network as it requires less long range context, and fewer parameters are easier to optimize.

C. Rendering Input

The inputs of our network consist of essential information for rendering, including geometry, reflectance, illumination, such that the shading network has the potential to simulate the ray-tracing and produce proper shading.

More specifically, the 3D virtual geometry is represented as 2D depth map and surface normal map in camera view. The reflectance, i.e. the albedo, encodes the color reflected from an object and is constant to illumination, which is also represented as a 2D map. All of these information can be rendered efficiently from Mitsuba.

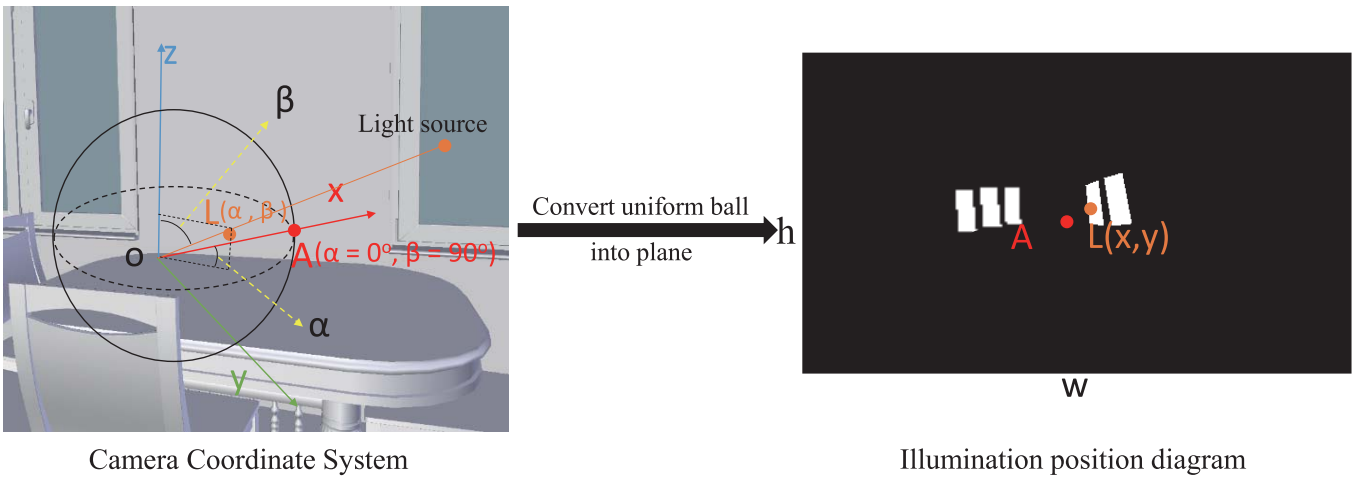
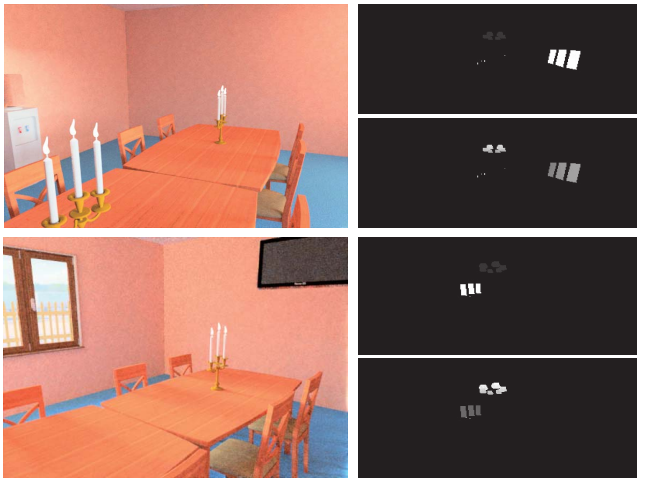


Fig. 3. A schematic diagram for looking for panoramic illumination. In camera coordinate system, point o is the position of camera, x axis represents where camera looks at, z axis represents camera's top orientation and the orange point is a light source. It is easy to calculate the value of the yaw angel $\alpha \in (-180^o, 180^o)$ and the pitch angle $\beta \in (0^o, 180^o)$ geometrically. To produce panoramic illumination image, we establish linear mapping relationship between (α, β) and position (x, y), $x \in (0, 360)$, $y \in (0, 180)$. In illumination position diagram, point L corresponds to light source in camera coordinate system, point A (center) represents where camera look at.



(a) Images from camera view (b) Illumination (Intensity and Distance)

Fig. 4. Two examples of illumination map. (a) Two images rendered from different camera views in the same room. (b) The corresponding illumination maps, encoding intensity and distance information of direct light sources. We can discover that the illumination maps contains invisible light sources in images (a).

Intuitively, the illumination can be also represented into a 2D map, with values on each pixel encoding the strength and the distance of the light source, if any, from the corresponding inbound direction. However, we find it cannot capture light sources outside the camera frustum, which have huge impact on the rendering. To handle this, we propose to use a panoramic illumination image, which encodes all light sources that are visible from the camera to render. The panoramic illumination image is an equirectangular reprojection of a unit sphere to a 2D regular image, yet still with each pixel encoding the strength and distance for light sources.

Figure 3 gives the pipeline for generating panoramic illumination image, and Figure 4 displays two examples. In Figure 4 (a), there are two images rendered from different camera views in the same room. And Figure 4 (b) is their corresponding panoramic illumination images, encoding intensity and distance information of direct light sources. When compare Figure 4 (a) and Figure 4 (b), we find that illumination images contain invisible light source in Figure 4 (a).

(a) Color image (b) Shading image

Fig. 5. Example of ground truth. (a) Physically based rendered color image. (b) Shading image, the texture is removed.

D. Ground Truth

We require ground truth for physically based rendered shadings and color images as the supervision for the shading network and composition network respectively. Following Zhang *et al.* [1], we use Mitsuba to generate our ground truth PBR color image. For the ground truth shading, we remove the texture from the virtual scene and re-render the PBR image. Figure 5 shows examples of our color image (Figure 5 (a)) and shading (Figure 5 (b)) ground truth.

E. Losses

To generate photo-realistic images, we adopt the perceptual loss [4], [35] based on a pre-trained visual perception network (we use VGG-19 network [5]). Unlike ℓ_1 or ℓ_2 loss, the perceptual loss helps to learn both local details and global structures since different layers in the network represent an image at different levels of abstractions. Mathematically, let

(a) Network (shading) (b) Network (image) (c) OpenGL (d) Mitsuba

Fig. 6. Rendering results using neural network and software. (a) Shading generated by network. (b) Color images generated by network. (c) Color images rendered by $OpenGL$. (d) Color images rendered by $Mitsuba$ (Ground truth).

$\{\Phi_l\}$ be a collection of layers in the visual perception network. For a training sample (X, R, Y, S), the perceptual loss of shading network is defined as

$$\mathcal{L}_1(\theta) = \sum_{l}^{N} \lambda_l \|\Phi_l(S) - \Phi_l(f(X;\theta))\|_1, \tag{1}$$

where $l = 0, \ldots, N$, N is the number of layers in the pre-trained visual perception network, $\{\lambda_l\}$ are the hyperparameters which balance the contributions of each layer l to the loss. Note that for $l = 0$, the loss is the ℓ_1 distance between the network output and ground truth.

Similarly, the perceptual loss of composition network is defined as

$$\mathcal{L}_2(\rho) = \sum_{l}^{N} \lambda_l \|\Phi_l(Y) - \Phi_l(g(f(X;\theta), R;\rho))\|_1, \tag{2}$$

Empirically, we find that the hyperparameters used to balance different layers of VGG-19 in perceptual loss are important, because of the various properties of different layers. Specifically, the higher layer of VGG-19 represents higher-level features which is robust to noise in protecting structure but may introduce strong grid patterns; the lower layers of VGG-19 represents lower-level features which leads to precise outputs but easily produces blurry results with some unexpected effects. These situations will be further detailed and exhibited in our experiments later.

IV. Implementations

A. Datasets

The data that we use for experiments includes surface normal, depth, reflectance, shading, panoramic illumination and PBR (ground truth). We use SUNCG [36] as it provide a large number of indoor scene with realistic furniture layout for rendering purpose. We follow Zhang *et al.* [1] to sample the camera viewpoints and use the physically based rendering from the PBRS dataset they provided. We generate the other inputs, e.g. reflectance map and panoramic illumination image, and ground truth, e.g. physically based rendered shading ground truth from the raw models in SUNCG. To make sure the shading is consistent with the color rendering from Zhang *et al.* [1], we use the same indoor and outdoor illumination with them.

B. Training Details

We adopt a two-stage training schema to make the training stable. The shading network is firstly trained and fixed once it converges reasonable well. Then, we train the composition network using the products of well-trained shading network to approximate the final outputs.

Specifically, we randomly choose 20, 000 training instances in each epoch for the training of composition network. The shading and composition networks are trained with 14 epochs and 12 epochs respectively with Adam [37]. During training, we set the batch size to 1, and the learning rate to 0.0001. To complete the training procedure, it costs about 104 hours on one Nvidia 1080Ti GPU.

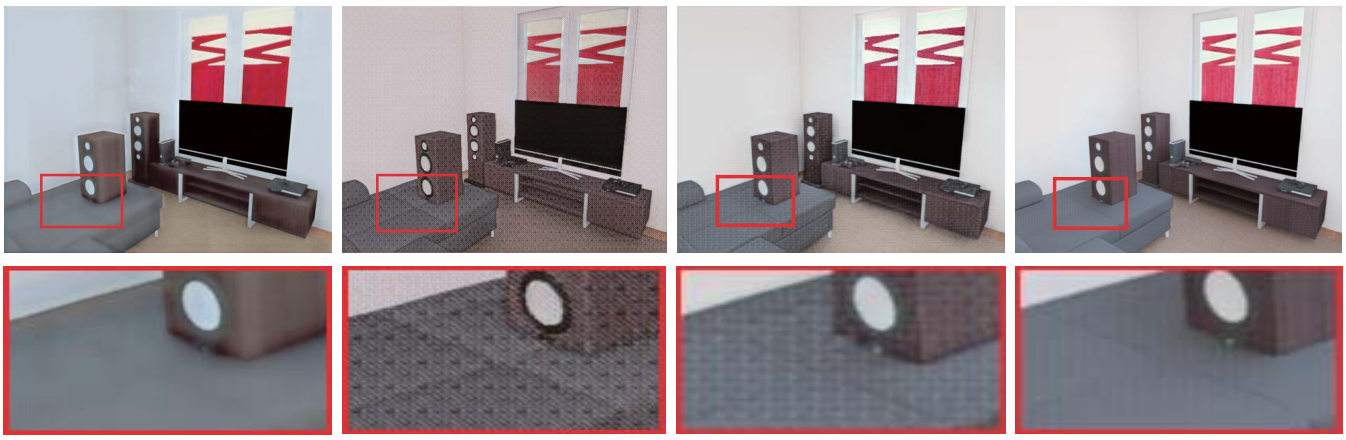
V. Experiments

In order to certify our work's efficiency and validity, we conduct several valuable experiments. In section A, we compare our network's products with software's products in quality and time consumption to certify our superiority. In section B, we prove our loss function, composition network, shading network and approach to realize physically based rendering using intrinsic images are valid. In section C, we compare our results with baselines (pix2pix [18], CAN [33], CycleGAN [19], U-Net [34]) in qualitative and quantitative ways. In section D, we test our network on more complex scenarios.

A. Compare With Software Rendering Results

In order to illustrate the effectiveness of our work, we compare our results generated by neural network with results rendered by software, such as $OpenGL$ and $Mitsuba$.

Figure 6 show the results generated in different ways. Figure 6 (a) is the shading image, an intermediate product of our network. By comparing the result images visually, we notice that the images rendered by our network



(a) L_1 loss (b) L_1 loss (fifth layer) (c) Full-perceptual loss (d) Part-perceptual loss

Fig. 7. Results of different loss functions. (a) Network output using $L1$ loss function, image is blurry. (b) Network output only use the fifth layer of VGG-19 to calculate $L1$ loss, has strong pattern but protects structure well. (c) Network output using standard perceptual loss function (5 CONV layers from the VGG-19 network), patterns can be observed in the image. (d) Network output using part-perceptual loss function (3 CONV layers from the VGG-19 network), image is clean and sharp.

(Figure 6 (b)) are superior than the ones rendered by $OpenGL$ (Figure 6 (c)), in terms of both illumination variation and colors. Moreover, our results are comparable with $Mitsuba$ rendering results (Figure 6 (d)). However, the time costs have wide variations. We can get a rendering image (360×480) about 40 milliseconds through $OpenGL$ and about 10.2 seconds through our network while about 3 minutes by using $Mitsuba$ (a render system, computes ray tracing [38] relying on CPU) with an i7, 4 cores CPU. Besides, our network can benefit from GPU, which takes 0.145 seconds to generate the output on a GPU 1080Ti in this work. In conclusion, using our network can get satisfactory results and save time by a large margin.

B. Validation

1) Does Loss Function Matter?: In this part, we use different loss functions for comparison and try to recombine the VGG-19 [5] layers which are used to calculate perceptual loss. And the purpose of this experiment is to provide a guidance for a better choice of loss function and help us better determine hyperparameters in Eq. 1 and Eq. 2.

We apply these loss functions above into our rendering task, and the results are displayed in Figure 7. When we use $L1$ loss function (i.e. $\lambda_i = 0, i = 1 \dots 5$), the generated image is blurry, e.g., Figure 7 (a). The reason for this phenomenon is because the $L1$ loss function calculates differences of two images in a low-level abstraction, and it is sensitive and not robust to unexpected effects, such as noise caused by low sample rate, etc. [12]; note that such noise is hard to avoid in indoor physically based rendering, which often happens in the dataset [1] adopted in our experiments. In the contrast, the perceptual loss compares images in high-level abstraction and evaluates structures and sematics at the same time. Acquiescently, the standard perceptual loss function (full-perceptual loss) uses 5 CONV layers of the VGG-19 network to calculate loss value [4], [39], [40]. And the products of full-perceptual loss function are shown in Figure 7 (c). Please notice the obvious pattern on the surface of sofa in Figure 7 (c). What's more, we also adopt the fifth layer of VGG-19 alone to calculate the $L1$ loss value and guide the training process (i.e. $\lambda_i = 0, i = 0 \dots 4$),the generate result shows well-protected structure but is filled with strong patterns, as we can see in Figure 7 (b). Thus, in order to avoid blur and pattern effects, we use high-level features but discard or decrease higher layers' effects of the VGG-19 network.

As a result, we combine the first 3 layers to get a new loss function (part-perceptual loss). Specifically, the hyperparameters (i.e. $\lambda_i, i = 0 \dots 5$) in Eq. 1 and Eq. 2 used to combine loss value of different VGG-19 layers in this paper are on the basis of Chen *et al.* [4] and empirically multiply 1.0, 1.5, 1.5, 0.5, 0.0, 0.0 respectively. The results of part-perceptual loss function are given in Figure 7 (d).

2) Does Intrinsic Decomposition Matter?: We have divided our end-to-end physically based rendering network into two parts, a shading generating network (with inputs surface normal, depth, illumination map) which are used to generate shading and a composition network (with inputs reflectance and generated shading) which combines generated shading and reflectance automatically. An alternative way is to train one fully convolutional network (with all inputs combined) to directly produce the rendering image. In order to verify the effectiveness of our network architecture, we combine all inputs together and feed-forward them into one neural network (U-Net [34]). During the process, no shading image will be generated and the final outputs will be produced directly. The corresponding results are exhibited in Figure 8. Figure 8 (a) are shading images generated by shading network, Figure 8 (b) are color images generated by composition network, Figure 8 (c) are color images generated directly by U-Net with all inputs combined.

The illumination variation is stronger in Figure 8 (a) than in color images Figure 8 (b) and Figure 8 (c), where illumination variation is distracted by color. Thus, providing shading images as an intermediate supervision contributes to capture

(a) Shading (shading+composition) (b) Color image (shading+composition) (c) Color image (tranform directly)

Fig. 8. Verify the superiority of shading network plus composition network. (a) Shading images generated by shading network. (b) Color images generated by composition network. (c) Color images generated directly by U-Net with all inputs combined. Notice that illumination variation in shading images (a) is stronger than in color images (b), (c). With the help of shading (a), color images (b) easily capture more significant illumination information than color images (c). Such as shadows behind the computer, lighted up lamp on the wall, etc. On the other hand, direct transformation produces non-uniform color which impairs the visual quality a lot. Please note the wallpaper at bottom in images (c).

more accurate illumination information. As demonstrated in Figure 8 (a) and Figure 8 (b), where illumination variation is better than illumination variation in Figure 8 (c). Such as shallow shadows behind computer, the lighted up lamp on the wall. On the other hand, direct transformation produces non-uniform color which impairs the visual quality a lot, as shown in Figure 8 (c) the third row. Specifically, non-uniform color appears at the bottom of wallpaper. One possible reason is that simultaneously calculating light transport and interaction between light and reflectance (constant to illumination transform) with all inputs simply combined is too complex for network to regress without excellent capability and huge training datasets. On the contrary, our shading plus composition network appropriately splits the task into two parts, one for light transport (i.e. shading network), and another for interaction between light and reflectance (i.e. composition network), which alleviates the requirements on network capability and training datasets. We will have more analysis on the advantages of the stacked shading and composition networks in section D under more complex scenarios.

3) Does Composition Network Matter?: We can recover an image (**I**) by utilizing Figure 9 (a) shading image (**S**) and Figure 9 (b) reflectance image (**R**). The traditional way to get **I** is multiplying **S** and **R** directly

$$\mathbf{I} \approx \mathbf{S} \odot \mathbf{R}. \tag{3}$$

like an inverse process of image intrinsic decomposition [9], [41]. Usually, the recovered image produced by traditional method can not be displayed on screen directly, because the pixels' value are in high dynamic range (HDR), while the visual range is [0,255]. Thus, some other operations, such as truncation, linear projection or tone mapping [7]–[9], are required to realize visualization. Figure 9 (c) and Figure 9 (d) show the visualized images with method used by Janner *et al.* [9] and tone mapping respectively. In Figure 9 (a), we can see objects closer to light sources (windows) are brighter, specifically, the bed's top left corner in Figure 9 (a) top. However, the illumination variation at bed's top left corner is desalinated in Figure 9 (c), (d) top. Inferring visualization will introduce deviation and visual differences.

Our composition network skips the step of visualization and combines **S** and **R** automatically to get a color image. From the results (Figure 9 (e)) produced by composition network, we notice that illumination variation is more reasonable, concretely, the illumination variation at bed's top left corner is well preserved in Figure 9 (e) top. What is more, color in Figure 9 (e) is more natural, and is closer to Figure 9 (b) when compared with color in Figure 9 (c), (d). Besides, the other reason for adopting network to combine shading and reflectance instead of multiplying them directly is that Eq. 3 is an approximate equation. This formulation simplifies the model, which has influence in photo-realistic images generating [2], [3], [10]. Thus, we hope neural network, to some extend, can automatically compensate this deviation and further improve the final performance, which is proved effective in Figure 9.

4) How Well Does Shading Prediction Network Work?: A good shading network should be sensible to inputs transform and generates reasonable shading images. Thus, we try to use different panoramic illumination to evaluate our shading network on it's outputs. Figure 10 shows generated shading and corresponding color images under different panoramic illumination. In Figure 10 (a), there is a light source at the left in panoramic illumination image which indicates a window exists at left rear of camera in real 3D scene. Physically, the object closer to light source should be brighter, and Figure 10 (b) shows that the lower left corner of shading image is brighter, which proves the reasonableness of our shading network. What is more, we also exchange the top and bottom panoramic illumination in Figure 10 (a), the exchanged panoramic illumination images are shown in Figure 10 (f). Then, We use these modified panoramic illumination images to generate new results, notice that the corresponding shading Figure 10 (e) and color images Figure 10 (d) are brighter while using larger light sources, please notice the shadow in cabinet (Figure 10 (d), (e) bottom), and are darker while using smaller light sources (Figure 10 (d), (e) top). This phenomenon is physically reasonable and proves our shading network performs well again.

C. Perceptual Experiment

In perceptual experiment, we compare our approach to other four baselines, CAN [33], pix2pix [18], U-Net [34] and CycleGAN [19] qualitatively and quantitatively. These baselines are representative in image-to-image translation and image generation. For fair comparison, all baselines use the same inputs and supervision ground truths (except for shading images) as in our approach. In addition, we replace the $L1$ and $L2$ loss function used in baselines with our loss

Fig. 9. Combine shading and reflectance. (a) Shading images. (b) Reflectance images. (c), (d) Recovered images, multiplying shading and reflectance directly; and visualized through linear projection and tone mapping respectively. The illumination variation in them is desalinated. (e) Recovered images, using composition network. The illumination variation is similar to illumination variation in shading images and have better performance in image quality.

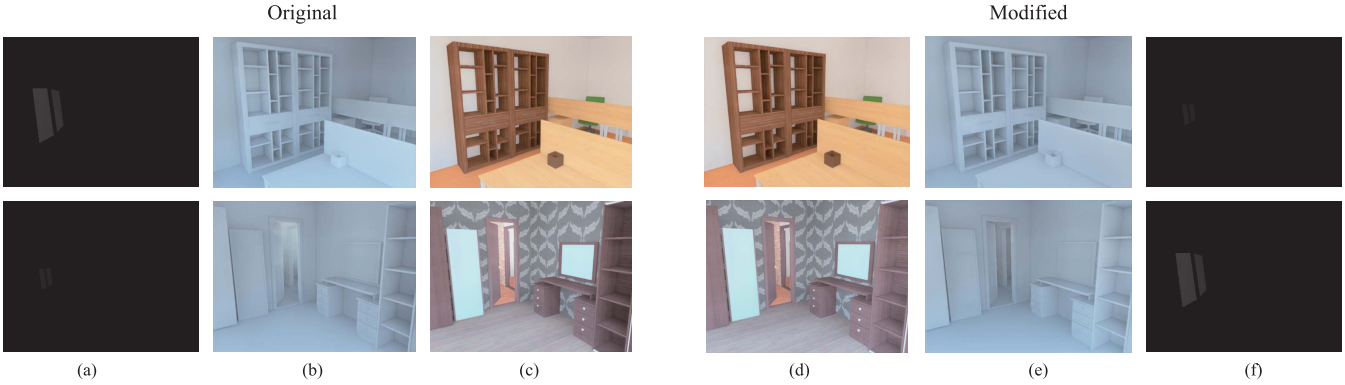


Fig. 10. Shading and corresponding color images generated under different panoramic illumination. (a) Panoramic illumination image. (b), (c) Shading and color images generated under panoramic illumination image (a). (f) Modified panoramic illumination. (d), (e) Color and shading images generated under panoramic illumination image (f). We notice that lower left corner of shading image (b) are brighter when light sources exist at left of panoramic illumination image. What is more, comparing results generated by using original and modified panoramic illumination, we can find that scene become brighter while using larger light sources and darker while using smaller light source.

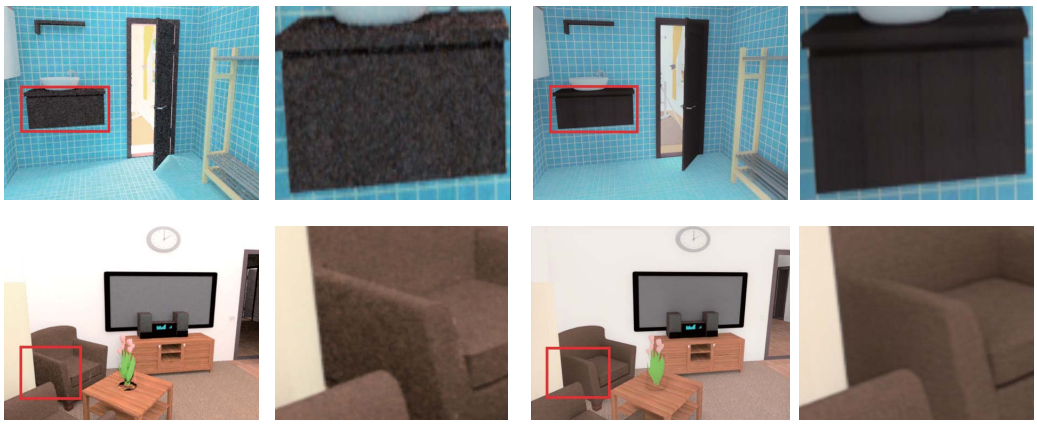


Fig. 11. Compare our results with ground truth in some difficult situation. (a) Ground truth rendered by Mitsuba, (b) Our results. We can get similar visual effect globally when compare ground truth (a) with our result (b). However, there is strong noise existing in ground truth due to limited rendering time. Nevertheless, our results are clean and can be produced faster through network.

function in this paper. In all the baselines, we combine all the source inputs and directly produce the color rendering as the output.

1) Qualitative Comparison to Baselines: In Figure 12, we analyze visual effects qualitatively on images synthesized by different approaches. In global, ground truth and images

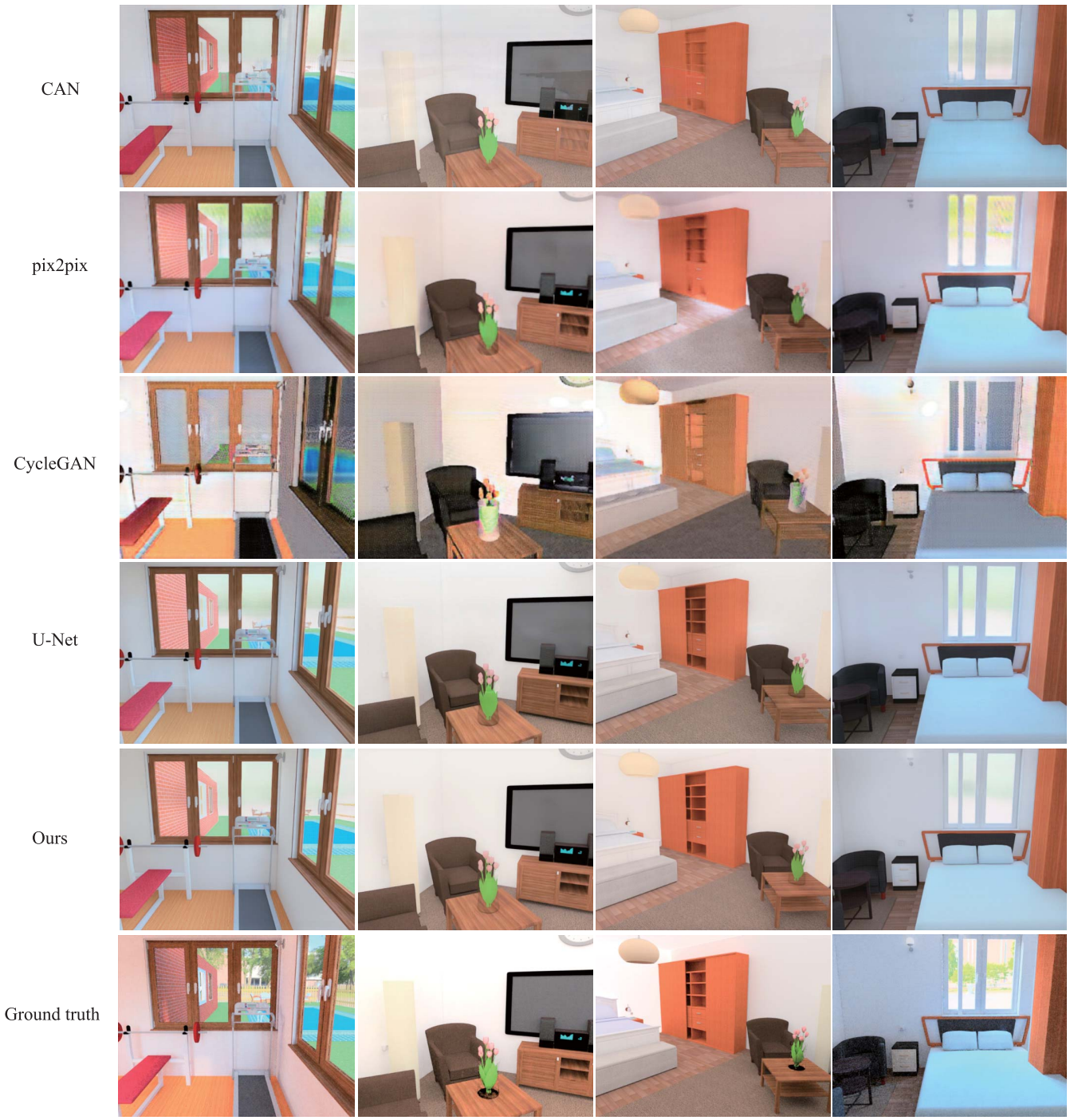


Fig. 12. Ground truths, our results and images synthesized by CAN [33], pix2pix [18], CycleGAN [19], U-Net [34]. Images synthesized by our approach and U-Net are sharp and ground truth alike. While images synthesized by CAN, pix2pix, CycleGAN have problems in blur and shadows, please notice the open cabinets. Emphatically, lamp on the wall is lighted up (right column) only in our approach.

synthesized by our approach and U-Net are sharp, and they are approximate in visual. While the products of CAN, pix2pix and CycleGAN have problems in blur and artifacts. In detail, we discover that the open cabinets have obvious shadows in ground truth and images produced by U-Net, pix2pix and our approach. However, the shadow does not exist in CAN and CycleGAN. What is more, using our approach (shading network + composition network) is helpful in illumination variation learning. For example, lamp on the wall (right column) is lighted up only in our approach and ground truth. Further comparison with U-Net will be discussed in section D under more complex scenarios.

Ground truth is in general good, but bad in certain difficult situation (e.g. some regions lack illumination and the limited rendering time). In these hard case, we perform even better. For instance, Figure 11 gives a comparison between our results and ground truth in noise standard. Figure 11 (a) are ground truths with close observation, Figure 11 (b) are our results with

TABLE I

THE AVERAGE LPIPS VALUE AND SSIM VALUE BASED ON DIFFERENT METHODS. LPIPS OF OUR METHOD IS THE SMALLEST AND SSIM OF OUR METHOD IS THE LARGEST, BOTH METRICS INDICATE THAT OUR RESULTS ARE CLOSEST TO GROUND TRUTHS

	CycleGAN	CAN	pix2pix	U-Net	Ours
LPIPS ↓	0.089	0.064	0.054	0.049	**0.048**
SSIM ↑	0.551	0.681	0.690	0.729	**0.742**

close observation. Our results can get similar visual effect with ground truths globally, while there is strong noise existing in ground truths due to the limited rendering time. However, our network don't have this issue and is equipped with function of automatically denoising. We owe this phenomenon (denosing) to a well-combined perceptual loss function.

2) Perceptual Metric: Recently, Zhang *et al.* [42] proposed a new perceptual metric to compare the similarity of two images, named learned perceptual image patch similarity (LPIPS) metric. LPIPS has advantages over traditional metrics, such as mean squared error (MSE), structural similarity index (SSIM [43]), Peak Signalto Noise Ratio (PSNR), and is closer to human choices. Thus, we choose LPIPS as the main metric to measure the similarity between ground truths and results produced by networks (CAN, pix2pix, CycleGAN, U-Net and Ours). The second line of Table I stores the average value of LPIPS. Note that the network we use for collecting LPIPS is AlexNet [44], and images from test set of each method are used to calculate the average value of LPIPS. Besides, smaller LPIPS value means closer to ground truth. In Table I, the methods on the right side indicate better performance. Specifically, LPIPS value of ours is 0.048, which is the smallest one. It indicates that our results are closest to ground truths. Further, we also provides traditional metric (SSIM) in the third line of Table I. Notice that, SSIM value of our approach is 0.742, the largest one, verifying that images generated by our approach are closest to ground truth again.

3) User Study: In Chen and Koltun [4] experiment, randomized pairwise images are displayed to users for judgement. Similarly, we imitate their experimental protocol to randomly combine an image synthesized by our approach with another image synthesized by other methods (CAN, pix2pix, CycleGAN, U-Net, Ground truth), and let participants judge which one has higher quality (less blur, artifacts and noise; reasonable illumination variation, e.g., the objects closer to light source should be brighter) and is more realistic (mostly depends on participants' first impression) in seconds with a total of 50 pairs. In order to get precise metrics, all of our participants are graduate students who major in image processing and about thirty-three percent of them are female. Each participant is given at most 3 minutes to make judgements from 50 randomly combined image pairs (each pair about 3 seconds), totally 24 participants. Table II reports the results of comparison, and the percentage represents the rate of our approach is better than others. Across the statistical results, our approach outperforms CAN, pix2pix, CycleGAN, U-Net and Ground truth in 73.16%, 91.52%, 97.83%, 63.64% and 75.95%, respectively. It is interesting that our results have advantage over ground truth. Inferring the size of images (360×480) we used for user study were larger, which made noise obvious in some regions due to limited rendering time in ground truth.

(a) Scene_0 (b) Scene_1 (c) Scene_2 (d) Scene_3

Fig. 13. We adopt four public available scenes [45], which are more complex, for finetune and test. (a) Bedroom by SlykDrako. (b) The Grey & White Room by Wig42. (c) The White Room by Jay-Artist. (d) The Modern Living Room by Wig42. All images are rendered by Mitsuba, with sample 1024.

D. Generalization

The color rendering in PBRS [1] are not optimized for quality but speed, so that the renderings are noisy and of low physics related effects such as specular highlight and soft shadow. To fully verify the capability of our model capturing physically based rendering effect, we finetune our model with a small amount but high quality physically based rendering. This is a reasonable training schema in practice since high quality physically based rendering can be computational expensive and prohibitive to produce in large scale. To get 3D scenes with better material set up, we pick four scenes from public community [45], in which three are used for generating training data, and the other one for testing. We made necessary but minimal modification to make physically based rendering possible on these scenes, such as adding a HDR environment map, etc. In each scene, we randomly sample 128 cameras following Zhang et.al [1]. Examples of rendering from the four scenes are shown in Figure 13.

1) Test on Complex Scenarios: Figure 14 shows our results. Please note that the testing scene is precluded from the training data. As can be see, the network not only converts the color tone to more realistic, but also produces certain amount of specular reflection on the floor and soft shadows cast by objects. These physically based rendering related effects in our results may not be as strong as in the ground truth, but indicates that the network learns to compose the light reflections using the geometry and illumination from the input,

TABLE II

RESULTS OF BLIND SELECTION OF RANDOMLY COMBINED PAIRS. THE PERCENTAGE REPRESENTS IMAGES SYNTHESIZED BY OUR APPROACH ARE JUDGED HIGHER QUALITY AND MORE REALISTIC THAN CORRESPONDING IMAGES SYNTHESIZED BY OTHER METHODS (CAN, PIX2PIX, CYCLEGAN, U-NET) AND GROUND TRUTH

	Ours > CAN	Ours > pix2pix	Ours > CycleGAN	Ours > U-Net	Ours > Ground truth
Percentage	73.16%	91.52%	97.83%	63.64%	75.95%

(a) Test in scene_1 (b) Test in scene_2

Fig. 14. Test in complex scenes. (a) Test in scene_1. (b) Test in scene_2. The first and second rows are groundtruth images. The third and fourth rows are fintuned PBR-Net predictions. Such predictions have similar illumination variation and specular reflections as groundtruth.

(a) U-Net (b) PBR-Net (shading) (c) PBR-Net (color) (d) GT

Fig. 15. Test in scene_2 with different models. (a) Transform directly using U-Net. (b) Shading images from PBR-Net. (c) Color images from PBR-Net. (d) Ground truth. Both networks are fine-tuned on scene_0 and scene_3. PBR-Net prediction (b), (c) has stronger specular reflections than U-net prediction (a). Please note the sofa armrests.

instead of barely a pixel-wise color or style transfer. Moreover, our model generates a rendering in 145 ms while the GT would require 300 sec. Potentially with more high quality training data, the result quality may be further improved.

2) The Effectiveness of PBR-Net: We also verify the necessity of supervising shading in the middle under complex scenarios. Like in section B (b) and section C (a), we compare to the model (i.e. U-Net) that directly produce rendering without

constraint on shading images, and the results are shown in Figure 15. Consistent with what we found in section B (b) and section C (a), the direct transformation model has weakness in illumination variation capture. As shown in Figure 15, there are stronger specular reflections in PBR-Net prediction Figure 15 (b), (c) than in U-Net prediction Figure 15 (a). Specifically, the specular reflections on sofa armrests.

VI. Conclusion

In this paper, we propose a network architecture for speeding up physically based rendering given necessary rendering information. This architecture uses shading image as intermediate supervision, which is inspired by intrinsic decomposition. At last, a composition network is designed to improve performance. By analyzing the character of loss function, we make our approach robust to noise.

In the future, our goal is to generate more photo-realistic images, even videos through network by solving following obstacles. Firstly, the existing datasets have margin with photo-realistic images because of material and illumination, etc. At present, the free high-quality 3D models are very few and synthesize such photo-realistical images are time-consuming. Maybe a feasible approach to acquire these information (e.g., photo-realistical image, geometry, material, illumination) is from the real word, and most of these information are accessible. Nonetheless, there are obstacles in material estimation from a photo-realistical image, and this will be our future work. Secondly, only a HDR environment map and some indoor light sources are used in synthesizing training datasets. Thus, diversifying the illumination seems beneficial in generating various images and facilitating generalization. Lastly, only a single image is rendered in this paper, however, it is not enough for video generation. Because there is a common problem (e.g., flicker) in video synthesis due to temporal inconsistency. A solution to this problem is to explicitly adopt temporal constraints. For example, taking optical flow, between two consecutive frames, as a constraint. What is more, some specifically-designed network structures also seem helpful, such as LSTM [46].

References

[1] Y. Zhang *et al.*, "Physically-based rendering for indoor scene understanding using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5057–5065.

[2] A. Meka *et al.*, "LIME: Live intrinsic material estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6315–6324.

[3] G. Liu, D. Ceylan, E. Yumer, J. Yang, and J.-M. Lien, "Material editing using a physically based rendering network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2280–2288.

[4] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1520–1529.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[7] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Trans. Graph. (TOG)*, vol. 21, no. 3, pp. 267–276, Jul. 2002.

[8] P. Debevec and S. Gibson, "A tone mapping algorithm for high contrast images," in *Proc. 13th Eurographics Workshop Rendering*, 2002, pp. 145–156.

[9] M. Janner, J. Wu, T. D. Kulkarni, I. Yildirim, and J. Tenenbaum, "Self-supervised intrinsic image decomposition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5938–5948.

[10] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf, "Revisiting deep intrinsic image decompositions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8944–8952.

[11] M. Pharr, W. Jakob, and G. Humphreys, *Physically Based Rendering: From Theory to Implementation*. San Mateo, CA, USA: Morgan Kaufmann, 2016.

[12] C. R. A. Chaitanya *et al.*, "Interactive reconstruction of Monte Carlo image sequences using a recurrent denoising autoencoder," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, Jul. 2017.

[13] A. Beers *et al.*, "High-resolution medical image synthesis using progressively grown generative adversarial networks," 2018, *arXiv:1805.03144*. [Online]. Available: http://arxiv.org/abs/1805.03144

[14] F. Luan, S. Paris, E. Shechtman, and K. Bala, "Deep photo style transfer," vol. 2, 2017, *arXiv:1703.07511*. [Online]. Available: https://arxiv.org/abs/1703.07511

[15] R. Mechrez, E. Shechtman, and L. Zelnik-Manor, "Photorealistic style transfer with screened Poisson equation," 2017, *arXiv:1709.09828*. [Online]. Available: http://arxiv.org/abs/1709.09828

[16] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8798–8807.

[17] X. Qi, Q. Chen, J. Jia, and V. Koltun, "Semi-parametric image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8808–8816.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[19] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

[20] E. H. Land and J. J. McCann, "Lightness and retinex theory," *J. Opt. Soc. Amer.*, vol. 61, no. 1, p. 1, Jan. 1971.

[21] A. Bousseau, S. Paris, and F. Durand, "User-assisted intrinsic images," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 1–10, Dec. 2009.

[22] L. Shen, P. Tan, and S. Lin, "Intrinsic image decomposition with non-local texture cues," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–7.

[23] Y. Weiss, "Deriving intrinsic images from image sequences," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, 2001, pp. 68–75.

[24] J. T. Barron and J. Malik, "Shape, illumination, and reflectance from shading," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1670–1687, Aug. 2015.

[25] T. Narihira, M. Maire, and S. X. Yu, "Learning lightness from human judgement on relative reflectance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2965–2973.

[26] Q. Zhao, P. Tan, Q. Dai, L. Shen, E. Wu, and S. Lin, "A closed-form solution to retinex with nonlocal texture constraints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1437–1444, Jul. 2012.

[27] A. S. Baslamisli, H.-A. Le, and T. Gevers, "CNN based learning using reflection and retinex models for intrinsic image decomposition," 2017, *arXiv:1712.01056*. [Online]. Available: http://arxiv.org/abs/1712.01056

[28] L. Lettry, K. Vanhoey, and L. van Gool, "Unsupervised deep single-image intrinsic decomposition using illumination-varying image sequences," 2018, *arXiv:1803.00805*. [Online]. Available: http://arxiv.org/abs/1803.00805

[29] G. Han, X. Xie, J. Lai, and W.-S. Zheng, "Learning an intrinsic image decomposer using synthesized RGB-D dataset," *IEEE Signal Process. Lett.*, vol. 25, no. 6, pp. 753–757, Jun. 2018.

[30] *Blender*. Accessed: 1994. [Online]. Available: http://www.blender.org/

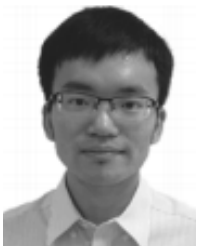
[31] *Maya*. Accessed: 1998. [Online]. Available: https://www.autodesk.com.sg/products/maya/overview

[32] *Mitsuba Physically Based Render*. Accessed: 2010. [Online]. Available: http://www.mitsuba-renderer.org/

[33] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2516–2525.

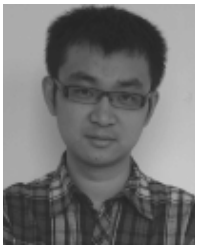
[34] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2015, pp. 234–241.
[35] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 694–711.
[36] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser, "Semantic scene completion from a single depth image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 190–198.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[38] A. S. Glassner, *An Introduction to Ray Tracing*. Amsterdam, The Netherlands: Elsevier, 1989.
[39] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2414–2423.
[40] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 658–666.
[41] H. Barrow and J. Tenenbaum, "Recovering intrinsic scene characteristics," *Comput. Vis. Syst*, vol. 2, pp. 3–26, Apr. 1978.
[42] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," 2018, *arXiv:1801.03924*. [Online]. Available: http://arxiv.org/abs/1801.03924
[43] 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.
[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.
[45] B. Bitterli. (2016). *Rendering Resources*. [Online]. Available: https://benedikt-bitterli.me/resources/
[46] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, 2014, pp. 1–5.

Peng Dai (Student Member, IEEE) received the B.E. degree in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2017, where he is currently pursuing the master's degree with the Institute of Image Processing, School of information and communications engineering. His research interests include computer vision and computer graphics.



Zhuwen Li (Member, IEEE) received the B.E. degree in computer science from Tianjin University, in 2008, the master's degree in computer science from Zhejiang University, in 2011, and the Ph.D. degree from the Department of Electrical and Computer Engineering, National University of Singapore, in 2014. He is currently a Research Scientist at Nuro, Inc. He is working on perception in autonomous driving, specifically in 3D structure recovery, motion analysis, point cloud analysis, and object detection.

Yinda Zhang (Member, IEEE) received the bachelor's degree from Tsinghua University, Beijing, China, and the master's degree from the National University of Singapore, and the Ph.D. degree in computer science from Princeton University. He is currently a Research Scientist at Google. His research interests lie at the intersection of computer vision, computer graphics, and machine learning. He is actively working on empowering 3D vision and perception via machine learning, including dense depth estimation, 3D shape analysis, and 3D scene understanding.



Shuaicheng Liu (Member, IEEE) received the B.E. degree from Sichuan University, Chengdu, China, in 2008, and the M.S. and Ph.D. degrees from the National University of Singapore, Singapore, in 2010 and 2014, respectively. In 2014, he joined the University of Electronic Science and Technology of China, where he is currently an Associate Professor with the School of Information and Communication Engineering, Institute of Image Processing. His current research interests include computer vision and computer graphics.

Bing Zeng (Fellow, IEEE) received the B.Eng. and M.Eng. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and the Ph.D. degree in electrical engineering from the Tampere University of Technology, Tampere, Finland, in 1991. He was a Postdoctoral Fellow with the University of Toronto from 1991 to 1992, and as a Researcher with Concordia University from 1992 to 1993. He then joined The Hong Kong University of Science and Technology (HKUST). After 20 years of service at HKUST, he returned to UESTC in 2013, through China's 1000-Talent-Scheme. At UESTC, he leads the Institute of Image Processing to focus on image and video processing, 3D and multi-view video technology, and visual big data. During his tenure at HKUST and UESTC, he has graduated over 30 master's and Ph.D. students, received over 20 research grants, filed eight international patents, and authored or coauthored over 250 articles. He was elected as a Fellow of the IEEE in 2016, for contributions to image and video coding and received the Second Class Natural Science Award (the first recipient) from the Chinese Ministry of Education in 2014. He was the General Co-Chair of the IEEE VCIP-2016, Chengdu, in 2016, and serves as the General Co-Chair of PCM-2017. He served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY for eight years and received the Best Associate Editor Award in 2011. He is currently on the Editorial Board of the *Journal of Visual Communication and Image Representation*.