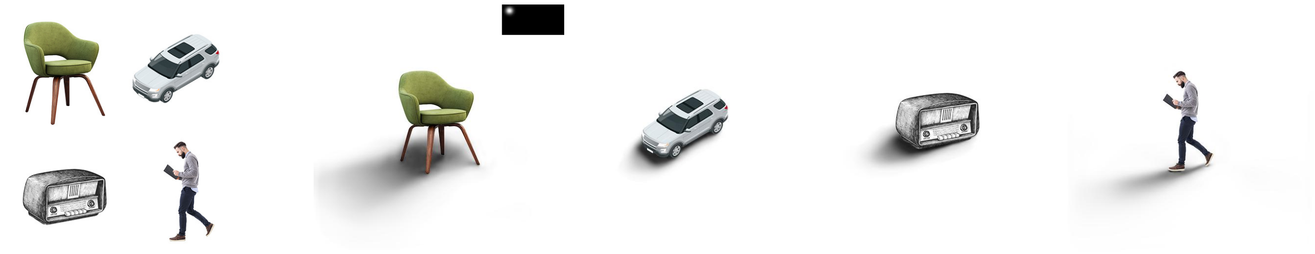
SSN: Soft Shadow Network for Image Compositing

Yichen Sheng
Purdue University

Jianming Zhang
Adobe Research

Bedrich Benes
Purdue University



(a) object cutouts (2D masks)

(b) soft shadows generated by our SNN from the image-based light map above

Figure 1: Our Soft Shadow Network (SSN) produces convincing soft shadows given an object cutout mask and a user-specified environment lighting map. (a) shows the object cutouts for this demo, including different object categories and image types, *e.g.* sketch, picture, vector arts. In (b), we show the soft shadow effects generated by our SSN. The changing lighting map used for these examples is shown at the corner of (b). The generated shadows have realistic shade details near the object-ground contact points and enhance image compositing 3D effect. More animated results can be found on our project page (https://shengcn.github.io/SSN).

Abstract

We introduce an interactive Soft Shadow Network (SSN) to generates controllable soft shadows for image compositing. SSN takes a 2D object mask as input and thus is agnostic to image types such as painting and vector art. An environment light map is used to control the shadow's characteristics, such as angle and softness. SSN employs an Ambient Occlusion Prediction module to predict an intermediate ambient occlusion map, which can be further refined by the user to provides geometric cues to modulate the shadow generation. To train our model, we design an efficient pipeline to produce diverse soft shadow training data using 3D object models. In addition, we propose an inverse shadow map representation to improve model training. We demonstrate that our model produces realistic soft shadows in real-time. Our user studies show that the generated shadows are often indistinguishable from shadows calculated by a physics-based renderer and users can easily use SSN through an interactive application to generate specific shadow effects in minutes.

1. Introduction

Image compositing is an essential and powerful means for image creation, where elements from different sources are put together to create a new image. One of the challenging tasks for image compositing is shadow synthesis. Manually creating a convincing shadow for a 2D object cutout requires a significant amount of expertise and effort, because the shadow generation process involves a complex interaction between the object geometry and light sources, especially for area lights and soft shadows.

Our work eases the creation of soft shadows for 2D object cutouts and provides full controllability to modify the shadow's characteristics. The soft shadow generation requires 3D shape information of the object, which is not available for 2D image compositing. However, the strong 3D shape and pose priors of common objects may provide the essential 3D information for soft shadow generation.

We introduce the Soft Shadow Network (SSN), a deep neural network framework that generates a soft shadow for a 2D object cutout and an input image-based environmental light map. The input of SSN is an object mask. It is agnostic to image types such as painting, cartoons, or vector arts. User control is provided through the image-based environment light map, which can capture complex light configurations. Fig. 1 show animated shadows predicted by SSN for objects of various shapes in different image types. SSN produces smooth transitions with the changing light maps and realistic shade details on the shadow map, especially near the object-ground contact points.

SSN is composed of an Ambient Occlusion Prediction (AOP) module and a Shadow Rendering (SR) module. Given the object mask, the AOP module predicts an ambient occlusion (AO) map on the ground shadow receiver,

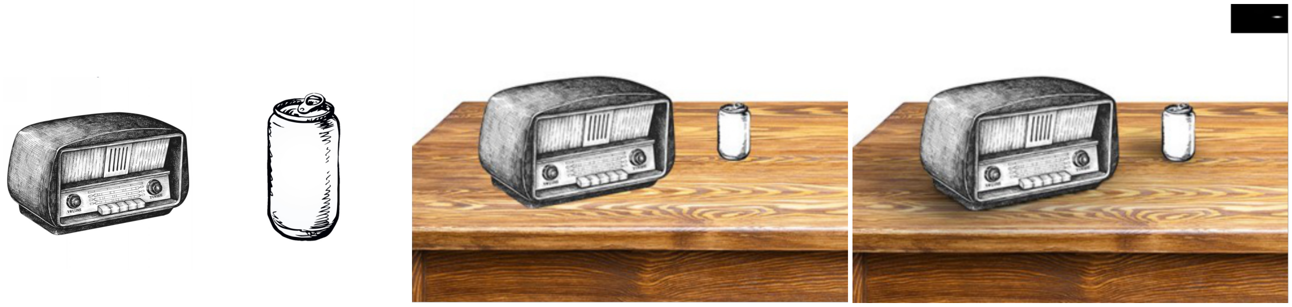


Figure 2: Image compositing using the Soft Shadow Network (SSN). The user adds two object sketches (left) on a background photo (middle) and uses our SSN to generate realistic soft shadows using a light map shown on the top of the right image. It only takes a couple of minutes for the user to achieve a satisfactory shadow effect. A video records this process can be found in the supplementary material.

which is light map-independent and captures relevant 3D information of the object for soft shadow generation. The SR module then takes the AO map, the object mask, and the light map to generate the soft shadow. Users can refine the predicted AO map when needed to provide extra guidance about the object's shape and region with the ground.

We generate training data for SSN using 3D object models of various shapes and randomly sampled complex light patterns. Rendering soft shadows for complex lighting patterns is time-consuming, which throttles the training process. Therefore, we propose an efficient data pipeline to render complex soft shadows on the fly during the training. In addition, we observe that the shadow map has a high dynamic range, which makes the model training very hard. An inverse shadow map representation is proposed to fix this issue.

A perceptual user study shows that the soft shadows generated by SSN are visually indistinguishable from the soft shadows generated by a physics-based renderer. Moreover, we demonstrate our approach as an interactive tool that allows for real-time shadow manipulation with the system's response to about 5ms in our implementation. As confirmed by a second user study, photo editors can effortlessly incorporate a cutout with desirable soft shadows into an existing image in a couple of minutes by using our tool (see Fig. 2). Our main contributions are:

1. A novel interactive soft shadow generation framework for generic image compositing.
2. A method to generate diverse training data of soft shadows and environment light maps on the fly.
3. An inverse map representation to improve the training on HDR shadow maps.

2. Related Work

Soft Shadow Rendering We review soft shadow rendering methods in computer graphics. All these methods require 3D object models, but they are related to our data generation pipeline.

A common method for soft shadow generation from a single area light is its approximation by summing multiple hard shadows [11]. Various methods were proposed to speed up the soft shadow rendering based on efficient geometrical representations [1, 3, 5, 8, 18, 19, 38, 39, 14, 10] or image filtering [2, 12, 42, 21]. However, these methods mostly target shadow rendering for a single area light source and are thus less efficient in rendering shadows for complex light settings.

Global illumination algorithms render soft shadows implicitly. Spherical harmonics [7, 41, 49] based methods render global illumination effects, including soft shadows in real-time by precomputing coefficients in spherical harmonics bases. Instead of projecting visibility function into spherical harmonics bases via expensive Monte-Carlo integration, we use different shadow bases, which are cheaper to compute.

Image Relighting and shadow Synthesis Our method belongs to deep generative models [16, 27] performing image synthesis and manipulation via semantic control [4, 6, 30] or user guidance such as sketches and painting [25, 31].

Deep image harmonization and relighting methods [40, 43, 48, 51] learn to adapt the subject's appearance to match the target lighting space. This line of works focuses mainly on the harmonization of the subject's appearance, such as color, texture, or lighting style [43, 45, 46].

Shadow generation and harmonization can be achieved by estimating the environment lighting and the scene geometry from multiple views [33]. Given a single image, Hold-Geoffroy *et al.* [22] and Gardner *et al.* [15] estimated light maps for 3D object compositing. However, neither the multi-view information nor the object 3D models are available in 2D image compositing. 3D reconstruction methods from a single image [13, 20, 28, 35] can close this gap. But, they require a complex model architecture design for 3D representation and may not be suitable for time-critical applications such as interactive image editing. Also, the

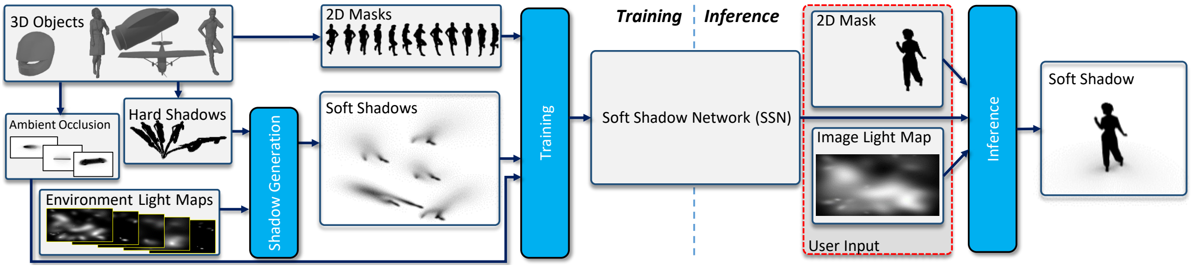


Figure 3: System Overview: During the **training** phase we train the SSN on a wide variety of 3D objects under different lighting conditions. Each 3D object is viewed from multiple common views, and its 2D mask and hard shadows are computed based on a sampling grid. Hard shadows are processed to become a set of shadow bases for efficient soft shadow computation during training. During the **inference** step, the user inputs a 2D mask (for example, a cutout from an existing image) and an image light map (either interactively or from a predefined set). The SSN then estimates a soft shadow.

oversimplified camera model in these methods brings artifacts in the touching area between the object and the shadow catcher. Recent works [23, 24, 29] explored rendering hard shadows or ambient occlusion using the neural network-based image synthesis methods, but they cannot render soft shadows and lack controls during image editing.

SSN provides a real-time and highly controllable way for interactive soft shadow generation for 2D object cutouts. Our method is trained to infer an object's 3D information for shadow generation implicitly and can be applied in general image compositing tasks for different image types.

3. Overview

The Soft Shadow Network (SSN) is designed to quickly generate visually plausible soft shadows given a 2D binary mask of a 3D object. The targeted application is image compositing, and the pipeline of our method is shown in Fig. 3.

The system works in two phases: the first phase trains a deep neural network system to generate soft shadows given 2D binary masks generated from 3D objects and complex image-based light maps. The second phase is the inference step that produces soft shadows for an input 2D binary mask, obtained, for example, as a cutout from an input image. The soft shadow is generated from a user-defined or existing image-based light represented as a 2D image.

The **training phase** (Fig. 3 left) takes as an input a set of 3D objects: we used 186 objects including human and common objects. Each object is viewed from 15 iconic angles, and the generated 2D binary masks are used for training (see Sec. 4.1).

We need to generate soft shadow data for each 3D object. Although we could use a physics-based renderer to generate images of soft shadows, it would be time-consuming. It would require a vast number of soft shadow samples to cover all possible soft shadows combinations with low noise. Therefore, we propose a dynamic soft shadow generation method (Sec. 4.3) that only needs to precompute the ”cheap” hard shadows once before training. The soft shadow is approximated on-the-fly based on the shadow bases, and the environment light maps (ELMs) randomly generated during the training. To cover a large space of possible lighting conditions, we use Environment Light Maps (ELMs) for lighting. The ELMs are generated procedurally as a combination of 2D Gaussians [17, 47] (Gaussian mixture) with the varying position, kernel size, and intensity. We randomly sample the ELMs and generate the corresponding soft shadow ground truth in memory on-the-fly during training.

The 2D masks and the soft shadows are then used as input to train the SSN as described in Sec. 5. We use a variant of U-Net [34] encoder/decoder network with some additional data injected in the bottleneck part of the network.

The **inference phase** (Fig. 3 right) is aimed at a fast soft shadow generation for image compositing. In a typical scenario, the user selects a part of an image and wants to paste it into an image with soft shadows. The ELM can be either provided or can be painted by a simple GUI. The resulting ELM and the extracted silhouette are then parsed to the SNN that predicts the corresponding soft shadow.

4. Dataset Generation

The input to this step is a set of 3D objects. The output is a set of triplets: a binary masks of the 3D object, an approximated but high-quality soft shadow map of the object cast on a planar surface (floor) from an environment light map (ELM), and an ambient occlusion map for the planar surface.

4.1. 3D Objects, Masks, and AO Map

Let's denote the 3D geometries by G_i, where $i = 1, \ldots, |G| = 102$. In our dataset, we used 43 human characters sampled from Daz3D, 59 general objects such as airplanes, bags, bottles, and cars from ShapeNet [9] and ModelNet [50]. Note that the shadow generation requires only

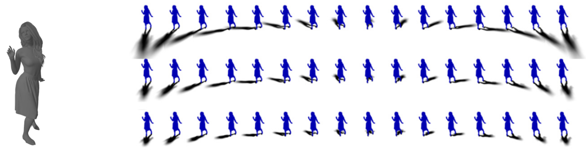


Figure 4: Shadow base example: for each view of each 3D object, we generate 8 × 32 shadow bases (3 × 16 shown here). We reduce the soft shadow sampling problem during training to environment light map generation problem, because we use the shadow bases to approximate soft shadows.

the 3D geometries without textures. Each G_i is normalized, and its min-max box is put in a canonical position with the center of the min-max box in the origin of the coordinate system. Its projection is aligned with the image's top to fully utilize the image space to receive long shadows.

Each G_i is used to generate fifteen masks denoted by M_i^j, where the lower index i denotes the corresponding object G_i and the upper index j is the corresponding view in form $[y, \alpha]$. Each object is rotated five times around the y axis $y = [0^o, 45^o, -45^o, 90^o, -90^o]$ and is displayed from three common view angles $\alpha = [0^o, 15^o, 30^o]$. This gives the total of $|M_i^j| = 1,530$ unique masks (see Fig. 3).

Ambient occlusion (AO) map describes how a point is exposed to ambient light (zero: occluded, one: exposed). We calculate the AO map for the shadow receiver (floor) and store it as an image.

$$A(x,n) = \frac{1}{\pi}\int_{\Omega} V(x,\omega)\cdot max(n,\omega)d\omega, \tag{1}$$

where $V(x,\omega)$ is the visibility term (value is either zero or one) of point x in the solid angle ω. The AO map approximates the proximity of the geometry to the receiver; entirely black pixels are touching the floor. We apply an exponent of one-third on the $A(x,n)$ for a high contrast effect to keep "most" occluded regions strong while weakening those slightly occluded regions.

4.2. Environment Light Maps (ELMs)

The second input to the SSN training phase is the soft shadows (see Fig. 3) that are generated from the 3D geometry of G_i by using environment light maps with HDR image maps in resolution 512 × 256. We use a single light source represented as a 2D Gaussian function:

$$L_k = Gauss(r, I, \sigma^2), \tag{2}$$

where $Gauss$ is a 2D Gaussian function with a radius r, maximum intensity (scaling factor) I, and softness corresponding to σ^2. Each ELM is a Gaussian mixture [17, 47]:

$$ELM = \sum_{k=1}^{K} L_k([x,y]), \tag{3}$$

Table 1: Ranges of the ELM parameters. We use random samples from this space during SSN training.

meaning	parameter	values
number of lights	K	$1,\ldots,50$
light location	$[x,y]$	$[0,1]^2$
light intensity	I	$[0,3]$
light softness	σ^2	$[0,0.1]$

where $[x, y]$ is the position of the light source. The coordinates are represented in a normalized range $[0, 1]^2$.

We provide a wide variety of ELMs that mimic complex natural or human-made lighting configurations so that the SSN can generalize well for arbitrary ELMs. We generate ELMs by random sampling each variable from Eqns (2) and (3) on-the-fly during training. The ranges of each parameter are shown in Table 1. The overall number of possible lights is vast. Note that the ELMs composed of even a small number of lights provide a very high dynamic range of soft shadows. Please refer to our supplementary materials for samples from the generated data and the comparison to physically-based rendered soft shadows.

4.3. Shadow Bases and Soft Shadows

Although we could use a physics-based renderer to generate physically-correct soft shadows, the rendering time for the vast amount of images would be infeasible. Instead, we use a simple method of summing hard shadows generated by a GPU-based renderer by leveraging light's linearity property. Our approach can generate much more diverse soft shadow than a naïve sampling several soft shadows from some directions.

We prepare our shadow bases once during the dataset generation stage. We assume that only the top half regions in the 256 × 512 ELMs cast shadows since the shadow receiver is a plane. For each 16 × 16 non-overlapping patch in the ELM, we sample the hard shadows cast by each pixel included in the patch and sum the group of shadows as a soft shadow base, which is used during training stages.

Each model silhouette mask has a set of soft shadow bases. During training, the soft shadow is rendered by weighting the soft shadow bases with the ELM composed of randomly sampled 2D Gaussian mixtures.

5. Learning to Render Soft Shadow

We want the model to learn a function $\phi(\cdot)$ that takes cutout mask I_m and environment light map I_e as input and predicts the soft shadow I_s cast on the ground plane:

$$\hat{I}_s = \phi(I_m, L_e) \tag{4}$$

During training, the input ELM L_e, as described in Sec. 4.2, is generated randomly to ensure the generalization ability of our model.

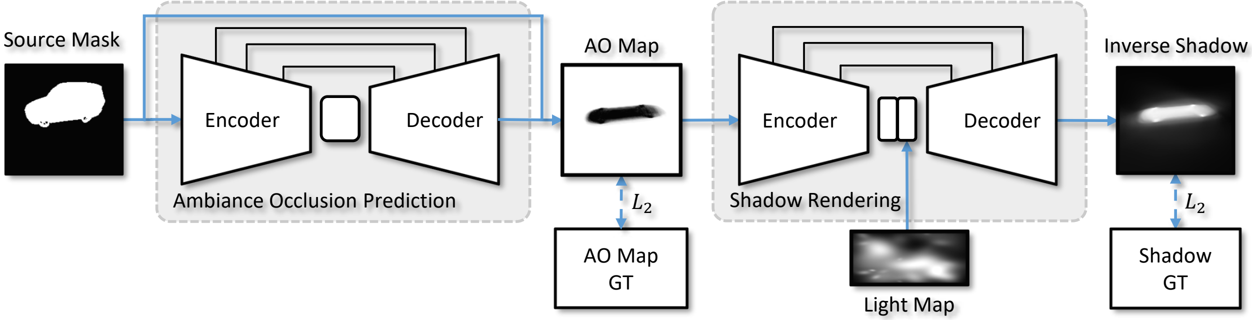


Figure 5: SSN Architecture is composed of two sub-modules: ambient occlusion prediction (AOP) module and shadow rendering (SR) module. AOP module has a U-net shape architecture. It takes a binary mask as input and outputs the ambient occlusion map. SR module has a similar architecture except that we inject the ELM into the bottleneck. Soft shadows are rendered as an output of the SR module. Please refer to supplementary materials for details.

During training, we observe that the model has a hard time to converge. This is because the shadow maps have a very wide dynamic range in most of the areas. To address this issue, we propose a simple transform to invert the ground truth shadow map during training:

$$\hat{S} = max(S_s) - S_s. \tag{5}$$

Therefore, most of the areas except the shadow region is near zero. Inverting the shadows makes the model focus on the final non-zero valued shadow region instead of the high dynamic range of the radiance on the plane receiver. The exact values of the lit radiance are useless for the final shadow prediction. This simple transformation does not bring or lose any information for shadows, but it significantly improves the converging speed and training performance. Please refer to Table 2 for quantitative results.

5.1. Network Architecture

SSN architecture (Fig. 5) has two modules: an ambient occlusion prediction (AOP) and a shadow rendering (SR). The overall design of the two modules is inspired by the U-Net [34], except that we inject light source information into the bottleneck of the SR module. In the two modules, both the encoder and decoder are fully convolutional. The AOP module takes masks as input and outputs the ambient occlusion (AO) maps. Then the source masks and the predicted AO maps are passed to the shadow rendering (SR) module. ELM is flattened, repeated in each spatial location, and concatenated with the bottleneck code of the shadow rendering module. The SR module renders soft shadows. The AOP module and the SR module share almost the same layer details. The encoders of both the AOP module and the SR module are composed of a series of 3 × 3 convolution layers. Each convolution layer during encoding follows conv-group norm-ReLU fashion. The decoders applied bilinear upsampling-convolution-group norm-ReLU fashion. We skip link the corresponding activations from corresponding encoder layers to decoder layers for each stage.

5.2. Loss Function and Training

The loss for both AOP module and SR module is a per-pixel L_2 distance. Let's denote the ground truth of AO map as $\hat{A}$. The inverse shadow map (Eqn(5)) is $\hat{S}$, and the prediction of ambient occlusion map as A and soft shadow map as S:

$$L_a(\hat{A}, A) = ||\hat{A} - A||_2, \tag{6}$$

$$L_s(\hat{S}, S) = ||\hat{S} - S||_2. \tag{7}$$

To use big batch size, the AOP and the SR modules are trained separately. For AOP module training, we use the mask as input and compute loss using Eqn (6). While for SR module training, we perturb the ground truth AO maps by random erosion and dilation. Then the mask and the perturbed AO map are fed into the SR module for training. During training, we randomly sample the environment light map ELM from Eqn (3) using our Gaussian mixtures and render the corresponding soft shadow ground truth on the fly to compute shadow loss using Eqn (7). This training routine efficiently helps our model generalize for diverse lighting conditions. The inverse shadow representation also helps the net to converge much faster and leads to better performance.

We provide a fully automatic pipeline to render soft shadows given a source mask and a target light during the inference stage. We further allow the user to manipulate the target light and modify the predicted ambient occlusion map to better the final rendered soft shadows interactively in real-time.

6. Results and Evaluation

6.1. Training Details

We implemented our deep neural network model by using PyTorch [32]. All results were generated on a desktop computer equipped with Intel Xeon W-2145

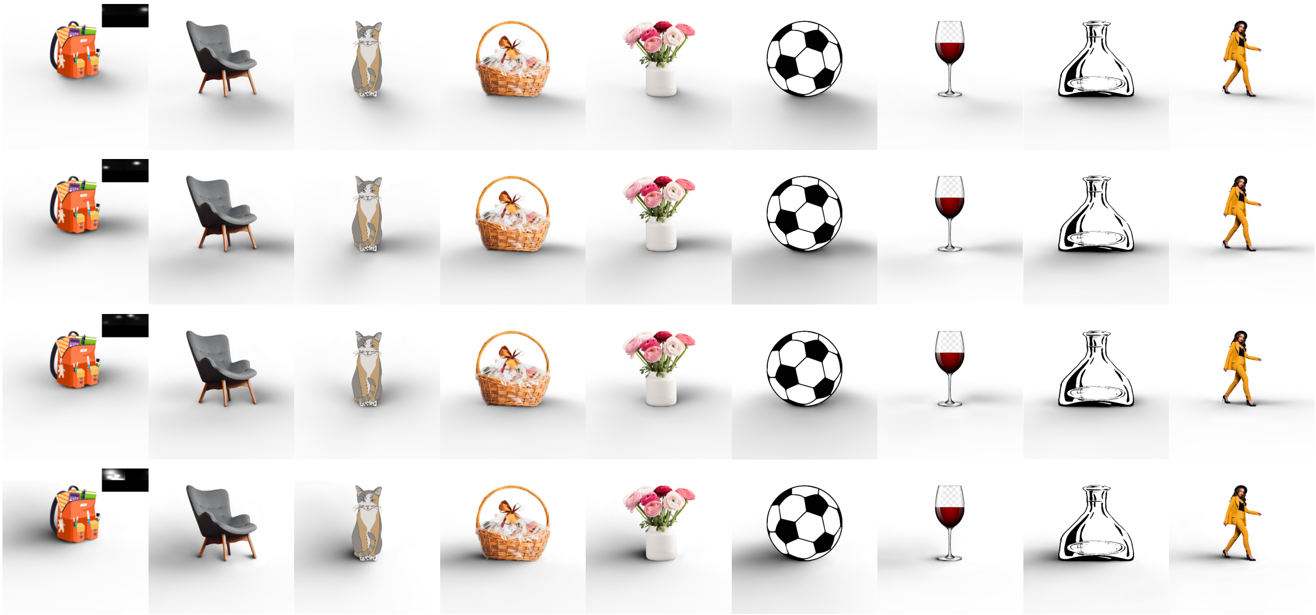


Figure 6: Soft shadows generated by SSN using four different light maps. Each row shares the same light map shown in the upper corner of the first image. All light maps have a weak ambient light. The four light maps also have one, two, four, seven strong area lights. Note some objects, *e.g*. cat, football, wine glass, etc, are not covered in our training set.

CPU(3.70GHz), and we used three NVIDIA GeForce GTX TITAN X GPUs for training. We used Adam optimizer [27] with an initial learning rate of $1e^{-3}$. For each epoch, we run the whole dataset 40× to sample enough environment maps. Our model converged after 80 epochs and the overall training time was about 40 hours. The average time for soft shadow inference was about 5 ms.

The animated example in Fig. 1 shows that our method generates smooth shadow transitions for dynamically changing lighting conditions.

Fig. 2 shows an example of compositing an existing input scene by inserting several 2D cutouts with rendered soft shadows. Note that once the ELM has been created for one cutout, it is reused by the others. Adding multiple cutouts to an image is simple. Several other examples generated by the users are shown in supplementary materials.

6.2. Quantitative Evaluation

Benchmark We separate the benchmark dataset into two different datasets: a general object dataset and a human dataset, 29 other general models from ModelNet [50] and ShapeNet [9] with different geometries and topologies, *e.g*. airplane, bag, basket, bottle, car, chair, etc., are covered in the general dataset, nine human models with diverse poses, shapes, and clothes are sampled from Daz3D studio, and we render 15 masks for each model with different camera settings. Soft shadow bases are rendered using the same method in training. We also used the same environment lightmap generation method as described aforementioned to randomly sample 300 different ELMs. Note that all the models are not shown in the training dataset.

Metrics We used four metrics to evaluate the testing performance of SSN: 1) RMSE, 2) RMSE-s [44], 3) zero-normalized cross-correlation (ZNCC), and 4) structural dissimilarity (DSSIM) [37]. Since the exposure condition of the rendered image may vary due to different rendering implementations, we use scale-invariant metric RMSE-s, ZNCC, and DSSIM in addition to RMSE. Note that all the measurements are computed in the inverse shadow domain.

Ablation study To evaluate the effectiveness of inverse shadow representation and AO map input, we perform ablation study on the aforementioned benchmark and evaluate results by the four metrics. *Non-inv-SSN* denotes a baseline that is the same as SSN but uses the non-inverse shadow representation. *SSN* is our method with inverse shadow representation. *GT-AO-SSN* is a baseline for replacing the predicted AO map with the ground truth one for the SR module. This is to show the upper-bound of improvement by refining the AO map when the geometry of the object is ambiguous to SSN.

Table 2 shows that Non-inv-SSN has a significantly worse performance for each metric. By comparing the metric difference between SSN and GT-AO-SSN in Table 2 and Table 3, it is observed that in some specific dataset, *e.g*. human dataset, our SSN has a reasonably good performance without refining the ambient occlusion map. We further

Table 2: Quantitative shadow analysis on general object benchmark. Non-inv-SSN uses the same architecture with our SSN except that the training shadow ground truth is not inverted. GT-AO-SSN uses ground truth ambient occlusion map as input for SR module. For RMSE, RMSE-s, DSSIM, the lower the values, the better the shadow prediction, while ZNCC is on the opposite.

Method	RMSE	RMSE-s	ZNCC	DSSIM
Non-inv-SSN	0.0926	0.0894	0.7521	0.2913
SSN	0.0561	0.0506	0.8192	0.0616
GT-AO-SSN	0.0342	0.0304	0.9171	0.0461

Table 3: Quantitative shadow analysis on human benchmark. The difference between the two methods is much smaller than the same methods in Table 2, indicating SSN can have a good performance for some specific object.

Method	RMSE	RMSE-s	ZNCC	DSSIM
SSN	0.0194	0.0163	0.8943	0.0467
GT-AO-SSN	0.0150	0.0127	0.9316	0.0403

validate it by a user study discussed in qualitative evaluation. In a more diverse test dataset, Table 2 shows that soft shadow quality is improved with better AO map input. Also, Fig. 6 shows that SSN generalizes well for various unseen objects with different ELMs.

6.3. Qualitative Evaluation

We performed two perceptual user studies. The first one measures the perceived level of realism of shadows generated by the SSN; the second one tested the shadow generator's ease-of-use.

Perceived Realism (user study 1) We have generated two sets of images with soft shadows. One set, called MTR, was generated from the 3D object by rendering it in Mitsuba renderer, a physics-based rendered and was considered the ground truth when using enough samples. The second set, called SSN, used binary masks from the same objects from MTR and estimated the soft shadows. Both sets have the same number of images $|MTR| = |SSN| = 18$, resulting in 18 pairs. In both cases, we used 3D objects that were not present in the training set or the SSN validation set during its training. The presented objects were unknown to the SSN. The ELM used were designed to cover a wide variety of shadows ranging from a single shadow, two shadows to a very subtle shape and intensity. Fig. 7 shows an example of the pair of images used in our study. Please refer to supplementary materials for some other user study examples.

The perceptual study was a two-alternative forced-choice (2AFC) method. To validate the rendered images' perceived realism, we have shown pairs of images in random order and random position (left-right) to multiple users and asked the participants of an online study which of the two images is a fake shadow.

Figure 7: A sample pair of images from our Perceived Realism user study. We show the output generated from 3D objects rendered by Mitsuba (left) and the output generated by SSN from binary masks (right).

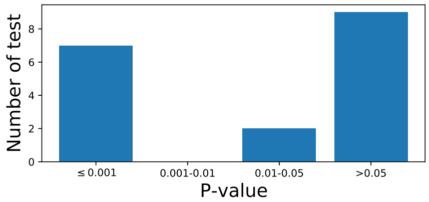


Figure 8: **p-value distributions:** In our first user study 7 questions have p-value≤ 0.001, two questions $0.01 <$ p-value ≤ 0.05, and 9 questions have p-value > 0.05.

The study was answered by 56 participants (73% male, 25% female, 2% did not identify). We discarded all replies that were too short (under three minutes) or did not complete all the questions. We also discarded answers of users who always clicked on the same side. Each image pair was viewed by 46 valid users. In general, the users were not able to distinguish SSN-generated shadows from the ground truth. In particular, the result shows that the average accuracy was 48.1% with a standard deviation of 0.153. T-test for each question in Fig. 8 shows that there are half of the predictions that do not have a significant difference with the Mitsuba ground truth.

Ease-of-Use (user study 2): In the second study, human subjects were asked to recreate soft shadows by using a simple interactive application. The result shows users can generate specified soft shadows in minutes using our GUI. Refer to supplementary materials for more details.

6.4. Discussion

With the impressive results from recent 2D-to-3D object reconstruction methods, *e.g*. PIFuHD [36], one may argue that rendering soft shadows can be straightforward by first getting the 3D object model from an image and then using the traditional shadow rendering methods in computer graphics. However, a 3D object reconstruction task from a single image is still challenging. Moreover, existing methods like PIFuHD are trained in the natural image domain. Thus it can be difficult for them to generalize to other image domains such as cartoons and paintings.

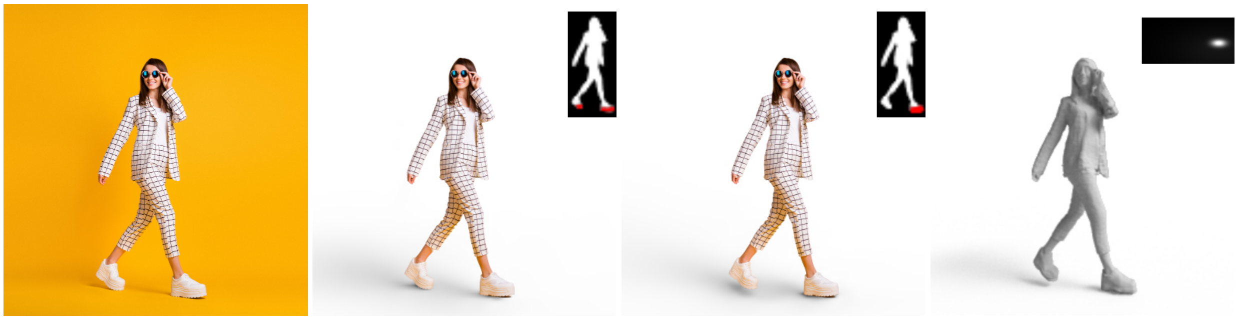


Figure 9: Shadow generation comparison with PIFuHD-based approach. SSN (2nd) renders the soft shadow given a cutout from an image (1st). The intermediate AO map prediction from our AO prediction module is visualized by the red regions overlaying the cutout mask shown in the top-right corner. SSN can also render a different soft shadow using a different AO input to change the 3D relationship between the occluder and shadow receiver (3rd). Mitsuba (4th) renders a soft shadow from the reconstructed 3D geometry of the object by PIFuHD, but it is hard to adjust the foot contact to match the original image. The ELM used for the three examples is in the corner of the 4th image.

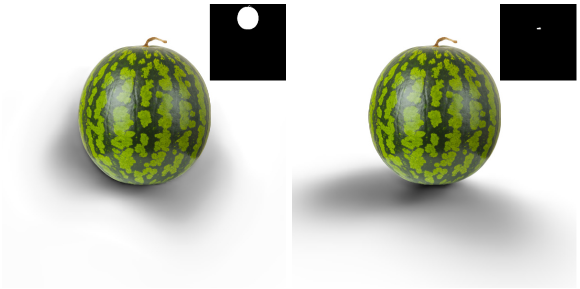


Figure 10: An example of the AO refinement. The AO map for SR module is shown in the corner of the composition results. Although AOP predicts a wrong AO map due to the ambiguity of the mask input (left), our SSN can render a much more realistic soft shadow with simple AO refinement (right).

Besides, there is an even more critical issue for soft shadow rendering using the 2D-to-3D reconstruction methods. In Fig. 9, we show an example using PIFuHD and the Mitsuba [26] renderer to generate the soft shadow for a 2D person image. Due to the inaccuracy in the 3D shape and the simplified camera assumption in PIFuHD, the feet' pose may not sufficiently align with the ground plane, making it less controllable to generate desirable shadow effects near the contact points. In contrast, our method can cause different shadow effects near the contact points by modifying the AO map, which provides more control over the 3D geometry interpretation near the contact points. Another example is shown in Fig. 10. The mask input for the watermelon is almost a disk which is very ambiguous. With a simple refinement of the AO input, the 3D relationship between the cutout object and the ground is visually more reasonable from the hint of a more realistic soft shadow.

Limitations The input to SSN is an object mask that is a general representation of the 2D object shape. Still, it can be ambiguous for inferring the 3D shape and pose for some objects. This can be improved by refining the AO map, while others may not be easily resolved in our framework. Some examples are shown in the supplementary materials. Also, the shape of soft shadows depends on camera parameters. However, the mask input is ambiguous for some extreme camera settings. For instance, a very large field-of-view distorts the shadows that the SSN cannot handle. Moreover, we assume our objects are always standing on a ground plane, and the SSN cannot handle the cases where objects are floating above the ground or the shadow receiver is more complicated than a ground plane.

7. Conclusion

We introduced Soft Shadow Network to synthesize soft shadows given 2D masks and environment map configurations for image compositing. Naively generating diverse soft shadow training data is cost expensive. To address this issue, we constructed a set of soft shadow bases combined with fast ELM sampling which allowed for fast training and better generalization ability. We also proposed the inverse shadow domain that has significantly improved the convergence rate and overall performance. A controllable pipeline is proposed to alleviate the generalization limitation by introducing a modifiable ambient occlusion map as input. Experiments demonstrated the effectiveness of our method. User studies confirmed the visual quality and showed that the user can quickly and intuitively generate soft shadows even without any computer graphics experience.

Acknowledgments: We appreciate constructive comments from reviewers. Thank Lu Ling for help in data analysis. This research was funded in part by National Science Foundation grant #10001387, *Functional Proceduralization of 3D Geometric Models*. This work was supported partly by the Adobe Gift Funding.

References

[1] Maneesh Agrawala, Ravi Ramamoorthi, Alan Heirich, and Laurent Moll. Efficient image-based methods for rendering soft shadows. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, page 375–384, USA, 2000. 2

[2] Thomas Annen, Zhao Dong, Tom Mertens, Philippe Bekaert, Hans-Peter Seidel, and Jan Kautz. Real-time, all-frequency shadows in dynamic scenes. *ACM Trans. Graph.*, 27(3):1–8, Aug. 2008. 2

[3] Ulf Assarsson and Tomas Akenine-Möller. A geometry-based soft shadow volume algorithm using graphics hardware. *ACM Trans. Graph.*, 22(3):511–520, 2003. 2

[4] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Trans. Graph.*, 38(4), July 2019. 2

[5] Stefan Brabec and Hans-Peter Seidel. Single sample soft shadows using depth maps. In *Graphics Interface*, volume 2002, pages 219–228. Citeseer, 2002. 2

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018. 2

[7] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional reflection functions from surface bump maps. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 273–281, 1987. 2

[8] Eric Chan and Fredo Durand. Rendering Fake Soft Shadows with Smoothies. In Philip Dutre, Frank Suykens, Per H. Christensen, and Daniel Cohen-Or, editors, *Eurographics Workshop on Rendering*. The Eurographics Association, 2003. 2

[9] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3, 6

[10] Robert L Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 137–145, 1984. 2

[11] Franklin C. Crow. Shadow algorithms for computer graphics. *SIGGRAPH Comput. Graph.*, 11(2):242–248, July 1977. 2

[12] William Donnelly and Andrew Lauritzen. Variance shadow maps. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 161–165, 2006. 2

[13] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 2

[14] Tobias Alexander Franke. Delta voxel cone tracing. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 39–44. IEEE, 2014. 2

[15] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *ACM Trans. Graph.*, 36(6):1–14, 2017. 2

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[17] Paul Green, Jan Kautz, Wojciech Matusik, and Frédo Durand. View-dependent precomputed light transport using nonlinear gaussian function approximations. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, I3D '06, page 7–14, New York, NY, USA, 2006. Association for Computing Machinery. 3, 4

[18] Gaël Guennebaud, Loïc Barthe, and Mathias Paulin. Real-time soft shadow mapping by backprojection. In *Rendering techniques*, pages 227–234, 2006. 2

[19] Gael Guennebaud, Loic Barthe, and Mathias Paulin. High-quality adaptive soft shadow mapping. *Comp. Graph. Forum*, 26(3):525–533, 2007. 2

[20] Tal Hassner and Ronen Basri. Example based 3d reconstruction from single 2d images. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 15–15. IEEE, 2006. 2

[21] Eric Heitz, Jonathan Dupuy, Stephen Hill, and David Neubelt. Real-time polygonal-light shading with linearly transformed cosines. *ACM Transactions on Graphics (TOG)*, 35(4):1–8, 2016. 2

[22] Yannick Hold-Geoffroy, Kalyan Sunkavalli, Sunil Hadap, Emiliano Gambaretto, and Jean-François Lalonde. Deep outdoor illumination estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7312–7321, 2017. 2

[23] Xiaowei Hu, Yitong Jiang, Chi-Wing Fu, and Pheng-Ann Heng. Mask-shadowgan: Learning to remove shadows from unpaired data. In *Proceedings of the*

IEEE International Conference on Computer Vision, pages 2472–2481, 2019. 3

[24] N Inoue, D Ito, Y Hold-Geoffroy, L Mai, B Price, and T Yamasaki. Rgb2ao: Ambient occlusion generation from rgb images. In *Computer Graphics Forum*, volume 39, pages 451–462. Wiley Online Library, 2020. 3

[25] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[26] Wenzel Jakob. Mitsuba renderer, 2010. 8

[27] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 2, 6

[28] Vladimir A Knyaz, Vladimir V Kniaz, and Fabio Remondino. Image-to-voxel model translation with conditional adversarial networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 2

[29] Daquan Liu, Chengjiang Long, Hongpan Zhang, Hanning Yu, Xinzhi Dong, and Chunxia Xiao. Arshadowgan: Shadow generative adversarial network for augmented reality in single light scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8139–8148, 2020. 3

[30] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017. 2

[31] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 2

[32] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[33] Julien Philip, Michaël Gharbi, Tinghui Zhou, Alexei A. Efros, and George Drettakis. Multi-view relighting using a geometry-aware network. *ACM Trans. Graph.*, 38(4), July 2019. 2

[34] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 3, 5

[35] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2304–2314, 2019. 2

[36] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 84–93, 2020. 7

[37] Tiago A Schieber, Laura Carpi, Albert Díaz-Guilera, Panos M Pardalos, Cristina Masoller, and Martín G Ravetti. Quantification of network structural dissimilarities. *Nature communications*, 8(1):1–10, 2017. 6

[38] Michael Schwarz and Marc Stamminger. Bitmask soft shadows. In *Comp. Graph. Forum*, volume 26, pages 515–524, 2007. 2

[39] Pradeep Sen, Mike Cammarano, and Pat Hanrahan. Shadow silhouette maps. *ACM Trans. Graph.*, 22(3):521–526, July 2003. 2

[40] Zhixin Shu, Sunil Hadap, Eli Shechtman, Kalyan Sunkavalli, Sylvain Paris, and Dimitris Samaras. Portrait lighting transfer using a mass transport approach. *ACM Trans. Graph.*, 36(4):1, 2017. 2

[41] Françis X Sillion, James R Arvo, Stephen H Westin, and Donald P Greenberg. A global illumination solution for general reflectance distributions. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 187–196, 1991. 2

[42] Cyril Soler and François X Sillion. Fast calculation of soft shadow textures using convolution. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 321–332, 1998. 2

[43] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Trans. Graph.* 2

[44] Tiancheng Sun, Jonathan T. Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *ACM Trans. Graph.*, 38(4), July 2019. 6

[45] Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. Multi-scale image har-

monization. *ACM Trans. Graph.*, 29(4):1–10, 2010. 2

[46] Michael W Tao, Micah K Johnson, and Sylvain Paris. Error-tolerant image compositing. In *European Conference on Computer Vision*, pages 31–44. Springer, 2010. 2

[47] Yusuke Tokuyoshi. Virtual spherical gaussian lights for real-time glossy indirect illumination. *Comput. Graph. Forum*, 34(7):89–98, Oct. 2015. 3, 4

[48] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3789–3797, 2017. 2

[49] Stephen H Westin, James R Arvo, and Kenneth E Torrance. Predicting reflectance functions from complex surfaces. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 255–264, 1992. 2

[50] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 3, 6

[51] Qingyuan Zheng, Zhuoru Li, and Adam Bargteil. Learning to shadow hand-drawn sketches. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2