ARShadowGAN: Shadow Generative Adversarial Network for Augmented Reality in Single Light Scenes

Daquan Liu[1], Chengjiang Long[2]*, Hongpan Zhang[1], Hanning Yu[1], Xinzhi Dong[1], Chunxia Xiao[1,3,4]*
[1]School of Computer Science, Wuhan University
[2]Kitware Inc., Clifton Park, NY, USA
[3]National Engineering Research Center For Multimedia Software, Wuhan University
[4]Institute of Artificial Intelligence, Wuhan University
chengjiang.long@kitware.com, {daquanliu,zhanghp,fishaning,dongxz97,cxxiao}@whu.edu.cn

Abstract

Generating virtual object shadows consistent with the real-world environment shading effects is important but challenging in computer vision and augmented reality applications. To address this problem, we propose an end-to-end Generative Adversarial Network for shadow generation named ARShadowGAN for augmented reality in single light scenes. Our ARShadowGAN makes full use of attention mechanism and is able to directly model the mapping relation between the virtual object shadow and the real-world environment without any explicit estimation of the illumination and 3D geometric information. In addition, we collect an image set which provides rich clues for shadow generation and construct a dataset for training and evaluating our proposed ARShadowGAN. The extensive experimental results show that our proposed ARShadowGAN is capable of directly generating plausible virtual object shadows in single light scenes. Our source code is available at `https://github.com/ldq9526/ARShadowGAN`.

1. Introduction

Augmented reality (AR) technology seamlessly integrates virtual objects with real-world scenes. It has broad application prospects in the fields of medical science, education and entertainment. In a synthetic AR image, the shadow of the virtual object directly reflects the illumination consistency between the virtual object and the real-world environment, which greatly affects the sense of reality. Therefore, it is very critical to generate the virtual object shadow and ensure it consistent with illumination constraints for high-quality AR applications.

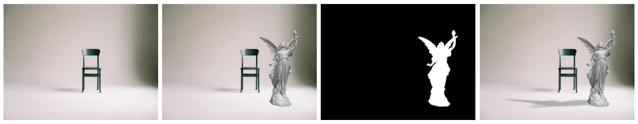


Figure 1. An example of casting virtual shadow for an inserted object in a single light scene. From left to right: the original image, the synthetic image without the virtual object shadow, the virtual object mask and the image with virtual object shadows.

Automatically generating shadows for inserted virtual objects is extremely challenging. Previous methods are based on inverse rendering [32] and their performances highly depend on the quality of the estimated geometry, illumination, reflectance and material properties. However, such an inverse rendering problem is very expensive and challenging in practice. What's worse, any inaccurate estimation may result in unreasonable virtual shadows. We aim to explore a mapping relationship between the virtual object shadow and the real-world environment in the AR setting without explicit inverse rendering. A shadow image dataset with clues to AR shadow generation in each image is desired for training and evaluating the performance of AR shadow generation. However, existing shadow-related datasets like SBU [41], SRD [38], and ISTD [44] , contain pairs of shadow image and corresponding shadow-free image, but most of the shadows lack occluders and almost all shadows are removed in shadow-free images. Such shadow datasets do not provide sufficient clues to generate shadows. Therefore, it is necessary to construct a new shadow dataset for AR applications.

In this work, we construct a large-scale AR shadow image dataset named Shadow-AR dataset where each raw image contains occluders, corresponding shadows and inserted 3D objects from public available datasets like ShapeNet [3]. We first annotate the real-world shadows and their corresponding occluders, and then determine the illumination and geometric information with camera and lighting cal-

[*] This work was co-supervised by Chengjiang Long and Chunxia Xiao.

ibration. Then we can apply 3D rendering to produce shadow for an inserted 3D object and take it as the ground-truth virtual shadow for both training and evaluation.

We observe that a straightforward solution like an image-to-image translation network cannot achieve plausible virtual shadows since it does not pay sufficient attention for handling the more important regions like real-world shadows and corresponding occluders. This observation inspires us to leverage the spatial attention information for real-world shadows and corresponding occluders to generate shadows for inserted virtual objects.

In this paper, we propose a generative adversarial network for directly virtual object shadow generation, which is called ARShadowGAN. As illustrated in Figure 1, ARShadowGAN takes a synthetic AR image without virtual shadows and the virtual object mask as input, and directly generates plausible virtual object shadows to make the AR image more realistic. Unlike inverse rendering-based methods [22, 23] perform geometry, illumination and reflectance estimation, our proposed ARShadowGAN produces virtual shadows without any explicit inverse rendering. Our key insight is to model the mapping relationship between the virtual object shadow and the real-world environment. In other words, ARShadowGAN automatically infers virtual object shadows with the clues provided by the real-world environment.

We shall emphasize that we adopt the adversarial training process [10] between the generator and the discriminator to generate an AR shadow image. With the number of epoches increases, both models improve their functionalities so that it becomes harder and harder to distinguish a generated AR shadow image from a real AR shadow image. Therefore, after a certain large number of training epochs, we can utilize the learned parameters in the generator to generate an AR shadow image.

To sum up, our main contributions are three-fold:

- We construct the first large-scale Shadow-AR dataset, which consists of 3,000 quintuples and each quintuple consists of a synthetic AR image without the virtual object shadow and its corresponding AR image containing the virtual object shadow, a mask of the virtual object, a labeled real-world shadow matting and its corresponding labeled occluder.

- We propose an end-to-end trainable generative adversarial network named ARShadowGAN. It is capable of directly generating virtual object shadows without illumination and geometry estimation.

- Through extensive experiments, we show that the proposed ARShadowGAN outperforms the baselines derived from state-of-the-art straightforward image-to-image translation solutions.

2. Related Work

The related work to shadow generation can be divided into two categories: *with* or *without* inverse rendering.

Shadow Generation with Inverse Rendering. Previous methods are based on inverse rendering to generate virtual object shadows, which require geometry, illumination, reflectance and material properties. Methods [39, 36, 48, 1] estimate lighting with known marker, which fail when the marker is blocked. Methods [22, 23, 25] estimate all the required properties, but inaccurate reconstruction results in odd-looking results. In recent years, deep learning has made significant breakthroughs, especially in visual recognition [13, 18, 26, 28, 17, 30, 27, 29, 16], object detection and segmentation [9, 42, 31], and so on. In particular, deep learning-based methods [7, 45, 8, 6, 14, 49] have been developed to estimate HDR illumination from a single LDR image but few of them work well for both indoor and outdoor scenes, and the rendering requires user interaction. Such heavy time and labor cost make this kind of methods infeasible for automatic shadow generation in AR.

Shadow Generation without Inverse Rendering. In recent years, generative adversarial network (GAN) [10] and its variants such as cGAN [33] and WGAN [2] have proven been applied successfully to various generative tasks such as shadow detection and removal [44, 46, 5, 50], of course also can be extended for shadow generation as a particular style transfer. It is worth mentioning that Hu *et al.*'s Mask-ShadowGAN [15] conducts shadow removal and mask-guided shadow generation with unpaired data at the same time. Zhang *et al.* extended image completion cGAN [19] to ShadowGAN [51] which generates virtual object shadows for VR images in which the scenes are synthesized with a single point light. Nonetheless, these methods dose not account for the occluders of real shadows. Unlike the previous methods, our proposed ARShadowGAN makes full use of spatial attention mechanism to explore the correlation between occluders and the corresponding shadows to cast plausible virtual shadows for inserted objects.

3. Shadow-AR Dataset

To cast shadow for an inserted virtual object in a single light scene, we need to explore a mapping relationship between the virtual object and the shadow in the AR setting. A necessary shadow image dataset with shadow clues for generating virtual shadow in each image is desired for training and evaluating the performance of virtual shadow generation. However, existing shadow-related datasets have many limitations. SBU [41] and UCF [52] consist of pairs of shadow images and corresponding shadow masks but no corresponding shadow-free images. SRD [38], UIUC [12], LRSS [11] and ISTD [44] contain pairs of shadow image and corresponding shadow-free image, but most of the

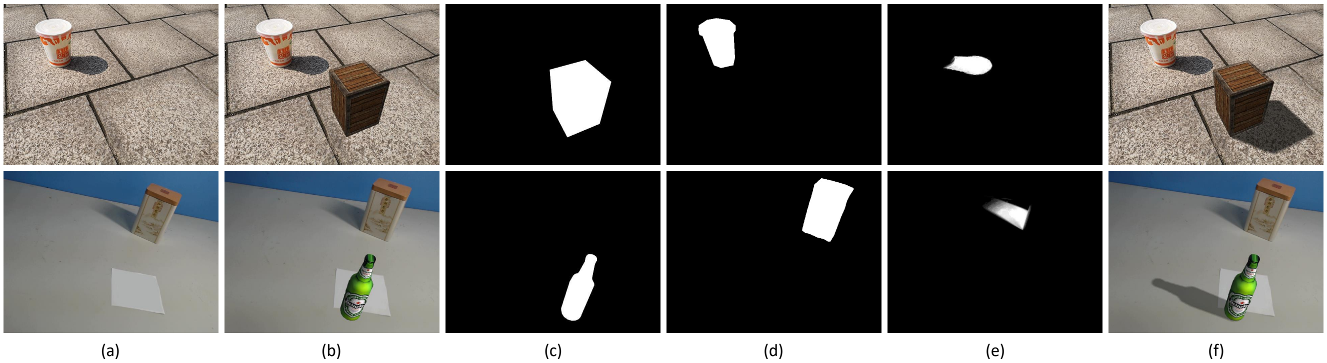


Figure 2. An illustration of two image examples in our Shadow-AR dataset. (a) is the original scene image without marker, (b) is the synthetic image without virtual object shadow, (c) is the mask of the virtual object, (d) is the real-world occluder, (e) is the real-world shadow, and (f) is the synthetic image containing the virtual object shadow.

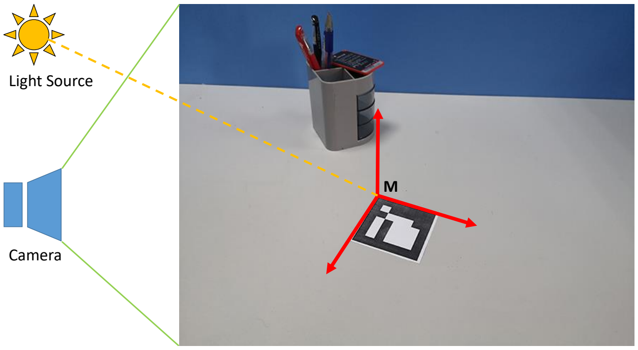


Figure 3. An illustration of data annotation. A 3D $Cartesian$ coordinate system M is established at the square marker. The camera pose is calculated by marker recognition. The light source position or direction is calibrated in the coordinate system **M**.

shadows lack occluders and almost all shadows are removed in shadow-free images. Such shadow datasets do not provide sufficient clues to generate shadows. Therefore, we have to construct a Shadow-AR dataset with shadow images and virtual objects.

3.1. Data Collection

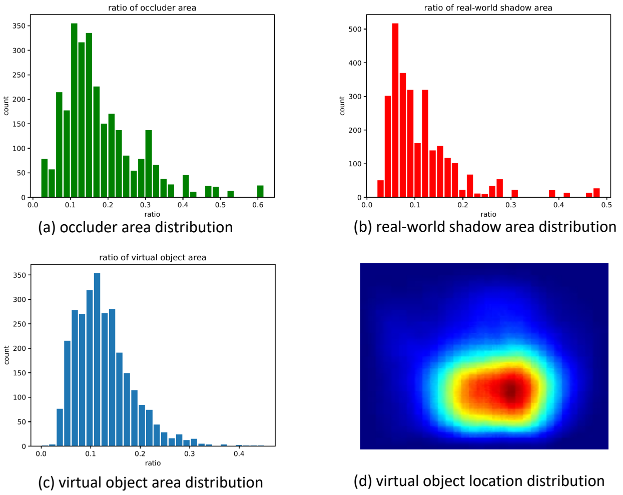
We collect raw images taken with a Logitech C920 Camera at 640×480 resolution, where scenes are taken with different camera poses. We keep real-world shadows and the corresponding occluders in photos because we believe that these can be used as *series clues* to shadow inference. We choose 9 models from ShapeNet [3], 4 models from Stanford 3D scanning repository and insert them into photos to produce different images of foreground (model) and background (scene) combinations. Our Shadow-AR dataset contains 3,000 quintuples. Each quintuple consists of 5 images: a synthetic image without the virtual object shadow and its corresponding image containing the virtual object shadow, a mask of the virtual object, a labeled real-world shadow matting and its corresponding labeled occluder. Figure 2 shows examples of our image data.

3.2. Mask Annotation and Shadow Rendering

We need to collect supervised information containing the real-world shadow matting, the corresponding occluder mask, and the synthetic images with plausible virtual object shadows. Note that insertion of a virtual 3D object requires geometric consistency and the virtual object shadow needs to be consistent with the real-world environment. This means that we need to calibrate the ***camera pose*** and the ***lighting*** in the real-world environment at the same time, which is very challenging. For convenience, we use a simple black-white square marker to complete the data annotation. As is shown in Figure 3, we establish such a 3D $Cartesian$ coordinate system **M** at the square marker as the world coordinate system.

Clues annotation. As is shown in Figure 2.(c)-(d), we annotate the real-world shadows and their corresponding occluders, which help to inference the virtual object shadow. We annotate real-world shadows with Robust-Matting software and annotate occluder with the LabelMe tool [43].

Camera and lighting calibration. We perform the square marker recognition and tracking by adaptive threshold with Otsu's [35] segmentation. With the extracted four marker corner points, camera poses are calculated by EPnP [24]. For indoor scenes, we consider a single dominant light and model it as a point light source with a three-dimensional position. To determine the most dominant light source, we manually block or turn off each indoor light (usually point or area light) sequentially and choose the one gives the most visible shadow. Then, we manually measure the dominant light geometric center coordinate X_m as the light position (as is shown in Figure 3). For outdoor scenes, the main light source is the sun and we model it as a directional light source. We measure the sunlight direction using interest point correspondences between a known



(a) occluder area distribution

(b) real-world shadow area distribution

(c) virtual object area distribution

(d) virtual object location distribution

Figure 4. Statistics of virtual objects and real-world clues. We show that our dataset have reasonable property distributions.

straight edge and its shadow.

Rendering. With the calibrated camera and lighting, we render 3D objects and the corresponding shadows. We render 3D objects with Phong shading [37]. We experimentally set ambient lighting as white with normalized intensity 0.25 for indoor and 0.35 for outdoor. We add a plane at the bottom of the 3D object and perform shadow mapping [47] along with alpha blending to produce shadows. To make the generated shadows have consistent appearances with real-world shadows, we apply a Gaussian kernel (5×5, $\sigma = 1.0$) to blur the shadow boundaries to get soft shadow borders.

Figure 4 shows statistical analysis of distribution properties of our dataset. The area distribution is expressed as the ratio between the target (shadows, occluders or virtual objects) area and image area. As we can see, majority of occluders falls in range of $(0.0, 0.3]$, majority of shadows falls in range of $(0.0, 0.2]$ and majority of virtual objects falls in range of $(0.0, 0.2]$. We found that clues falling in $(0.4, 0.6]$ occupy most of the image area, making it difficult to insert virtual objects. Similarly, inserted objects with too large area will block important clues. There are almost no such cases in our data set. In addition, we analyze the spatial distribution of virtual objects, we compute a probability map (Figure 4 (d)) to show how likely a pixel belongs to a virtual object. This is reasonable as virtual objects placed around human eyesight usually produce the most visual pleasing results.

4. Proposed ARShadowGAN

As illustrated in Figure 5, our proposed ARShadowGAN is an end-to-end network which takes a synthetic image without virtual object shadows and the virtual object mask as input, and produces the corresponding image with virtual object shadows. It consists of 3 components: an attention block, a virtual shadow generator with a refinement module, and a discriminator to distinguish whether the generated virtual shadow is plausible.

4.1. Attention Block

The attention block produces attention maps of real shadows and corresponding occluders. The attention map is a matrix with elements ranging from 0 to 1 which indicates varying attention of the real-world environments. The attention block takes the concatenation of the image without virtual object shadows and the virtual object mask as input. It has two identical decoder branches and one branch predicts the real shadow attention map and the other one predicts the corresponding ocluder attention map.

There are 4 down-sampling (DS) layers. Each DS layer extracts features by a residual block [13] which consists of 3 consecutive convolution, batch normalization and Leaky ReLU operations and halves the feature map with an average pooling operation. Then, features extracted by DS layers are shared by two decoder branches. The two decoder branches have the same architecture. Each decoder consists of 4 up-sampling (US) layers. Each US layer doubles the feature map by nearest interpolation followed by consecutive dilated convolution, batch normalization and Leaky ReLU operations. The last feature map is activated by a sigmoid function. Symmetrical DS-US layers are concatenated by skip connections.

4.2. Virtual Shadow Generator

The virtual shadow generator produces plausible virtual object shadows. It consists of a U-net followed by a refinement module. The U-net with 5 DS-US layers produces a coarse residual shadow image and then it is fine-tuned by the refinement module with 4 consecutive composite functions [18]. The final output is the addition of the improved residual shadow image and the input image.

In the virtual shadow generator, DS layers are the same as those in the attention block while US layers use convolutions instead of dilated ones. Each composite function produces 64 feature maps.

4.3. Discriminator

The discriminator distinguishes whether the virtual shadow shadows are plausible, thereby assisting the training of generator. We designed the discriminator in the form of Patch-GAN [20].

The discriminator contains 4 consecutive convolution with valid padding, instance normalization and Leaky ReLU operations. Then, a convolution produces the last feature map which is activated by sigmoid function. The final output of the discriminator is the global average pooling of the activated last feature map. In ARShadowGAN, the discriminator takes the concatenation of image without

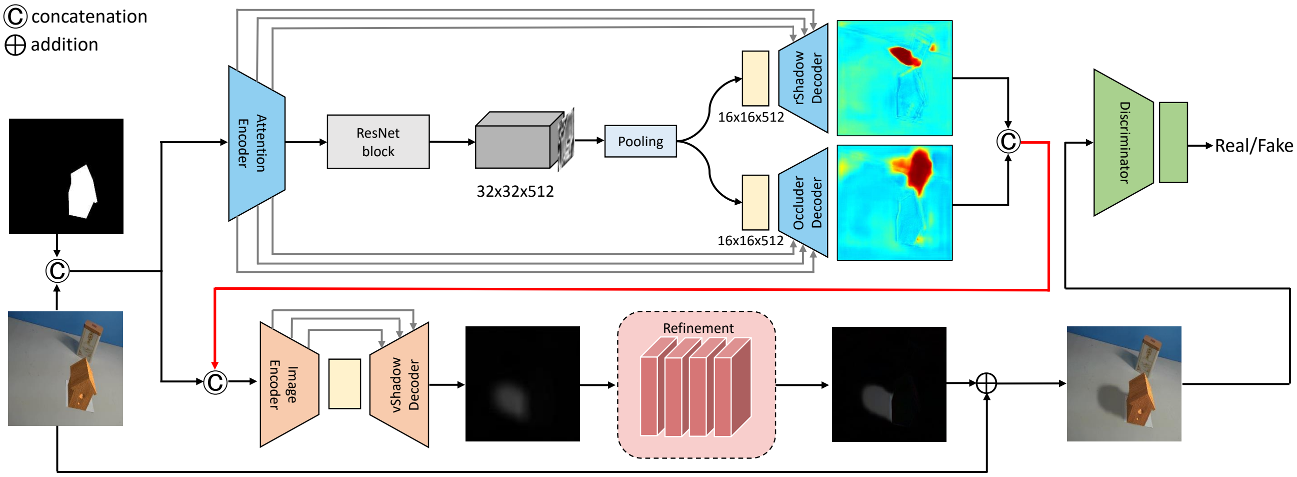


Figure 5. The architecture of our proposed ARShadowGAN. It consists of an attention block, a virtual shadow generator with a refinement module and a discriminator. Attention block has two branches producing attention maps of real-world shadows and occluders. The attention maps are leveraged by virtual shadow generator to produce a coarse residual shadow image. The coarse shadow image is fine-tuned by the refinement module. The final output is the addition of input image and the fine-tuned residual shadow image.

virtual object shadows, virtual object masks and the image with virtual object shadows as input.

4.4. Loss functions

Attention Loss. We use standard squared loss to measure the difference between the predicted attention maps and the ground truth masks. $\mathcal{L}_{attn}$ is defined as follows:

$$\begin{aligned}\mathcal{L}_{attn} &= \|\mathcal{A}_{robj}(x,m) - \mathcal{M}_{robj}\|_2^2 \\ &+ \|\mathcal{A}_{rshadow}(x,m) - \mathcal{M}_{rshadow}\|_2^2,\end{aligned} \tag{1}$$

where $\mathcal{A}_{rshadow}(\cdot)$ is the output attention map for real shadows and $\mathcal{A}_{robj}(\cdot)$ is the output attention map for real objects based on the input synthetic image x without virtual object shadows and the virtual object mask m. Note both $\mathcal{M}_{robj}$ and $\mathcal{M}_{rshadow}$ are the ground truth binary maps of the real-world shadows and their corresponding occluders. For $\mathcal{M}_{robj}$, 1 indicates that the pixel belongs to real objects and 0 otherwise. Similarly, 1 in $\mathcal{M}_{rshadow}$ indicates the pixel in the real shadow regions and 0 not.

Shadow Generation Loss. $\mathcal{L}_{gen}$ is used to measure the difference between the ground truth and the generated image with virtual object shadows. The shadow generation loss consists of three weighted terms, *i.e.*, $\mathcal{L}_2$, $\mathcal{L}_{per}$ and $\mathcal{L}_{adv}$, and the total loss is:

$$\mathcal{L}_{gen} = \beta_1\mathcal{L}_2 + \beta_2\mathcal{L}_{per} + \beta_3\mathcal{L}_{adv}, \tag{2}$$

where β_1, β_2 and β_3 are hyper-parameters which control the influence of terms.

$\mathcal{L}_2$ is the pixel-wise loss between the generated image and the corresponding ground truth. It is worth mentioning that our ARShadowGAN produces a coarse residual shadow image to generate a coarse virtual shadow image $\bar{y} = x + \mathbf{G}(x, m, \mathcal{A}_{robj}, \mathcal{A}_{rshadow})$. We further improve the residual image to form the final shadow image $\hat{y} = x + \mathbf{R}(\mathbf{G}(x, m, \mathcal{A}_{robj}, \mathcal{A}_{rshadow}))$ through the refinement module $\mathbf{R}(\cdot)$. Therefore, we can define $\mathcal{L}_2$ as follows:

$$\mathcal{L}_2 = \|y - \bar{y}\|_2^2 + \|y - \hat{y}\|_2^2, \tag{3}$$

where y is the corresponding ground truth shadow image.

$\mathcal{L}_{per}$ is the perceptual loss [21], which measures the semantic difference between the generated image and the ground truth. We use a VGG16 model [40] pre-trained on ImageNet dataset [4] to extract feature. The feature is the output of the 4^{th} max pooling layer ($14 \times 14 \times 512$), i.e. the first 10 VGG16 layers are used to compute feature map. $\mathcal{L}_{per}$ is defined as follows:

$$\mathcal{L}_{per} = \text{MSE}(V_y, V_{\bar{y}}) + \text{MSE}(V_y, V_{\hat{y}}), \tag{4}$$

where MSE is the mean squared error, and $V_i = \text{VGG}(i)$ is the feature map extracted by the well-trained VGG16 model.

$\mathcal{L}_{adv}$ describes the competition between the generator and the discriminator, which is defined as follows:

$$\mathcal{L}_{adv} = \log(\mathbf{D}(x, m, y)) + \log(1 - \mathbf{D}(x, m, \hat{y})), \tag{5}$$

where $\mathbf{D}(\cdot)$ is the probability that the image is “real”. During the adversarial training, the discriminator tries to maximize $\mathcal{L}_{adv}$ while the generator tries to minimize it.

4.5. Implementation details

Our ARShadowGAN is implemented in TensorFlow framework. In ARShadowGAN, all the batch normalization and Leaky ReLU operations share the same hyper parameters. We set decay as 0.9 for batch normalization and leak as 0.2 for Leaky ReLU. All images in our dataset are resized to 256×256 by cubic interpolation for training and testing.

Synthetic images and virtual object masks are normalized to $[-1, 1]$ while labeled clue images are normalized to $[0, 1]$. We randomly divide our dataset into three parts: 500 for attention block training, 2,000 for virtual shadow generation training and 500 for testing.

We adopt a two-stage training. At the 1^{st} stage, we train the attention block alone with the 500 training set. We optimize the attention block by minimizing $\mathcal{L}_{attn}$ with ADAM optimizer. Learning rate is initialized as 10^{-5} and $\boldsymbol{\beta}$ is set to $(0.9, 0.99)$. The attention block is trained for 5000 iterations with batch size 1. At the 2^{nd} stage, the attention block is fixed and we train virtual shadow generator and the discriminator with the 2,000 training set. We set $\beta_1 = 10.0$, $\beta_2 = 1.0$, $\beta_3 = 0.01$ for $\mathcal{L}_{gen}$. We adopt ADAM optimizer to optimize the generator and discriminator. The optimizer parameters are all same as those in the 1^{st} phase. The virtual shadow generator and discriminator is trained for 150,000 iterations with batch size 1. In each iteration, we alternately optimize the generator and discriminator.

5. Experiments

To evaluate the performance of our proposed ARShadowGAN, we conduct experiments on our collected Shadow-AR dataset. We calculate the average error on the testing set for quantitative evaluation. We calculate the root mean square error (RMSE) and structural similarity index (SSIM) with generated shadow images and the ground truth to measure the global image error. We calculate the balanced error rate [34] (BER) and accuracy (ACC) with generated shadow masks and ground truth shadow masks, which are obtained with ratio threshold, to measure the shadow area and boundary error. In general, the smaller RMSE and BER, the larger SSIM and ACC, the better the generated image. Note that all the images for visualization are resized to 4:3.

5.1. Visualization of Generated Attentions

Attention maps are used to assist the virtual shadow generator. As is shown in Figure 6, real-world shadows and their corresponding occluders are suggested more attention. It is worth mentioning that the virtual object itself is not a clue, and the mask prevents the virtual object from receiving more attention as real-world shadows and occluders. To verify the role of the mask, we replace the mask with a full black image which indicates no virtual object. The result is also shown in the 2^{nd} and 4^{th} row of Figure 6.

5.2. Comparison to Baselines

To our best knowledge, there are no existing methods proposed to directly generate AR shadows for inserted object without any 3D information. We still choose the following methods as baselines to compete since we can extend and adapt them on the our task:

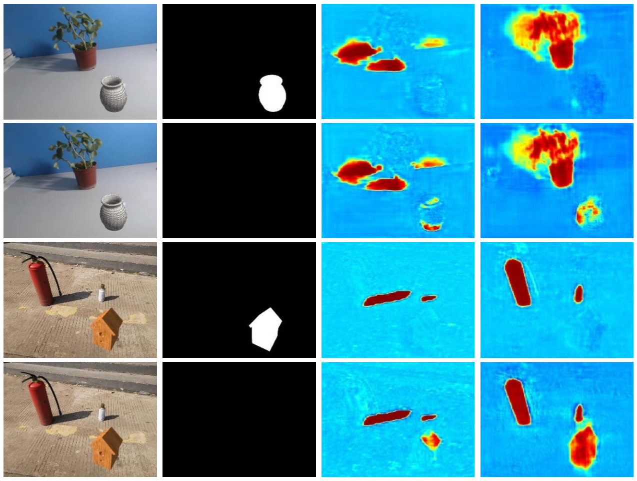


Figure 6. Examples of attention maps. From left to right: input images without virtual object shadows, input masks, attention maps of real-world shadows and their corresponding occluders. Corresponding cases without masks are also shown.

Pix2Pix [20] is a cGAN trained on paired data for general image-to-image translation. It is directly applicable to our shadow generation task. We make the Pix2Pix output shadow image directly.

Pix2Pix-Res is a variant of Pix2Pix whose architecture is the same as Pix2Pix but outputs the residual virtual shadow image like our ARShadowGAN.

ShadowGAN [51] synthesizes shadows for inserted objects in VR images. ShadowGAN takes exactly the same input items as our ARShadowGAN and generates shadow maps which are then multiplied to the source images to produce final images. We calculate shadow maps from our data to train ShadowGAN and we evaluate ShadowGAN with the produced final images.

Mask-ShadowGAN [15] performs both shadow removal and mask-guided shadow generation. We adapt this framework to our task. G_s and G_f are two generators of Mask-ShadowGAN and we adjust G_s to perform virtual shadow generation while G_f to perform mask-guided virtual shadow removal.

For fair comparison, we train all the models on the same training data with same training details and evaluate on the same testing data.

Models	RMSE	SSIM	S (%)	A (%)	ACC (%)
Pix2Pix	9.514	0.938	41.468	27.358	90.631
Pix2Pix-Res	8.043	0.959	29.597	26.476	96.689
ShadowGAN	8.041	0.961	28.347	24.547	97.122
Mask-ShadowGAN	7.493	0.959	23.261	21.131	98.443
ARShadowGAN	**6.520**	**0.965**	**22.278**	**19.267**	**98.453**

Table 1. Results of quantitative comparison. In the table, S represents BER of virtual shadow regions and A represents BER of the whole shadow mask. The best scores are highlighted in bold.

Quantitative comparison results are shown in Table 1.

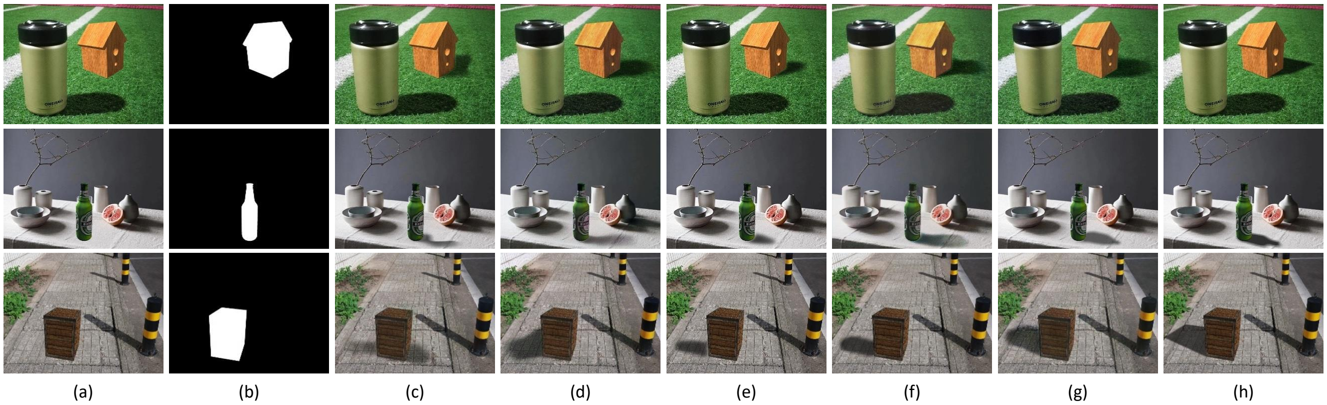


Figure 7. Visualization comparison with different methods. From left to right are input image (a), input mask (b), the results of Pix2Pix (c), Pix2Pix-Res (d), ShadowGAN (e), Mask-ShadowGAN (f), ARShadowGAN (g), and ground-truth (h).

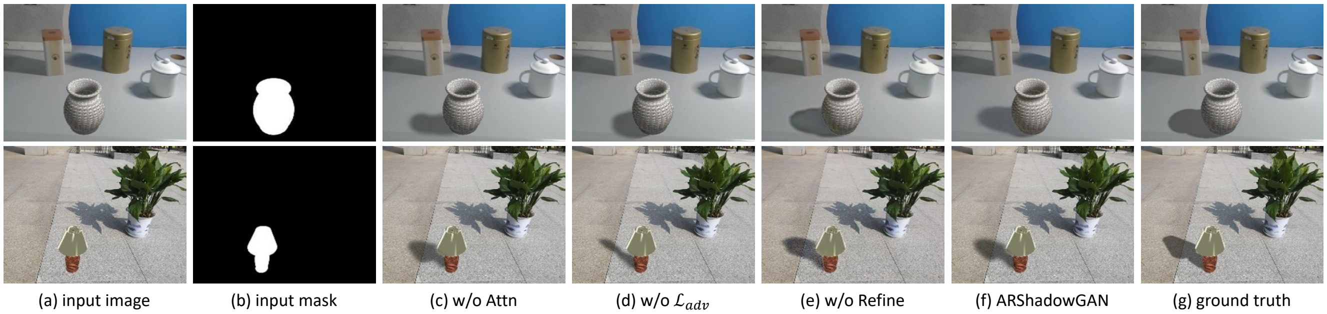


Figure 8. Examples of qualitative ablation studies of network modules.

Examples of qualitative comparison are shown in Figure 7. As we can see, the overall performances of Pix2Pix-Res and ShadowGAN are better than Pix2Pix, which indicates that the target of the shadow map or the residual shadow image makes the network focus on shadow itself rather than the whole image reconstruction. Mask-ShadowGAN performs a little better than Pix2Pix-Res and ShadowGAN, but it still produces artifacts. ARShadowGAN outperforms baselines with much less artifacts in terms of shadow azimuth and shape, which is partially because the attention mechanism enhances the beneficial features and make the most of them.

Models	RMSE	SSIM	S (%)	A (%)	ACC (%)
w/o Attn	7.175	0.962	23.162	21.079	98.446
w/o Refine	7.050	0.961	23.087	21.024	98.450
w/o $\mathcal{L}_{adv}$	7.781	0.959	29.093	26.354	97.487
w/o $\mathcal{L}_{per}$	8.001	0.963	29.576	26.399	97.152
w/o $\mathcal{L}_2$	9.696	0.924	50.748	30.829	88.548
ARShadowGAN	**6.520**	**0.965**	**22.278**	**19.267**	**98.453**

Table 2. Results of ablation studies. The best scores are highlighted in bold.

5.3. Ablation Studies

To verify the effectiveness of our loss function and network architecture, we compare our ARShadowGAN with its ablated versions:

- w/o Attn: we remove the attention block.
- w/o Refine: we remove the refinement module.
- w/o $\mathcal{L}_{adv}$: we remove the discriminator ($\beta_3 = 0$).
- w/o $\mathcal{L}_{per}$: we remove $\mathcal{L}_{per}$ from Equation 2 ($\beta_2 = 0$).
- w/o $\mathcal{L}_2$: we remove $\mathcal{L}_2$ from Equation 2 ($\beta_1 = 0$).

For models without attention blocks, the input to the virtual shadow generator is adjust to the concatenation of synthetic image (without virtual object shadows) and the object mask. We train these models on training set. Quantitative results of ablation studies are shown in Table 2 and examples of qualitative ablation studies are shown in Figure 8 and Figure 9.

Network modules. As we can see, our full model achieves the best performance. As is shown in Figure 8, the model without a discriminator mostly produces odd-looking virtual object shadows because the generator has not yet converge, which indicates that adversarial training does speed up the convergence of the generator. Our full model outperforms the version without attention block in overall virtual object shadow azimuth, which indicates that the attention block helps preserve features useful for shadow inference. The model without refinement module produces artifacts in the shadow area, suggesting that the refinement module fine-tunes virtual shadows from details by nonlinear activation functions.

Loss functions. As we can see, our full loss function achieves the best performance. As is shown in Figure 9, $\mathcal{L}_{per}$ has an important role in constraining the shadow

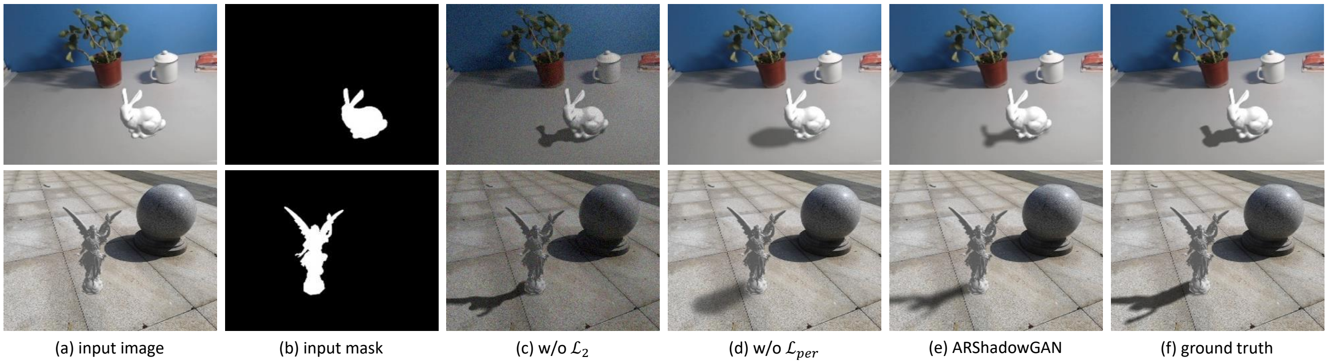


Figure 9. Examples of qualitative ablation studies of loss function.

shape. However, $\mathcal{L}_{per}$ is a global semantic constraint rather than a detail, so the pixel-wise intensity and noise are not well resolved. $\mathcal{L}_2$ maintains good pixel-wise intensity but produces blurred virtual object shadows which are not good in shape. $\mathcal{L}_{per} + \mathcal{L}_2$ outperforms both $\mathcal{L}_{per}$ and $\mathcal{L}_2$, which indicates that $\mathcal{L}_{per}$ and $\mathcal{L}_2$ promote each other.

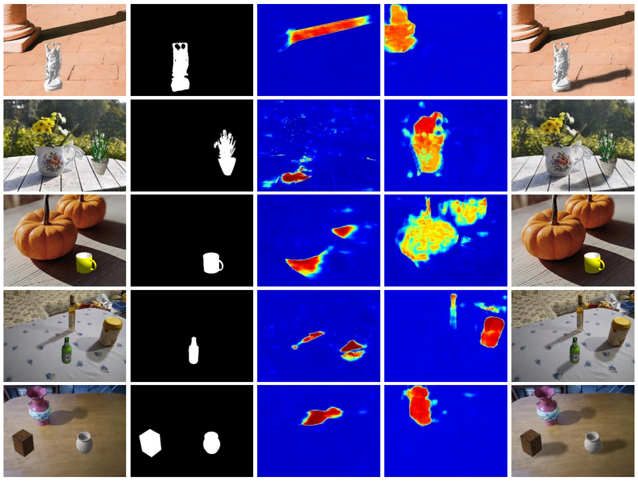


Figure 10. Robustness testing. From left to right: input images, input masks, attention maps of real-world shadows and their corresponding occluders and output images.

5.4. Robustness Testing

We test our ARShadowGAN with new cases outside Shadow-AR dataset in Figure 10 to show the robustness. All the images, model buddha, vase and mug are new and without the ground truth. The case with the model inserted in the real shadow is shown in the 3^{rd} row. Cases of multiple light sources and multiple inserted models are shown in the 4^{th} and 5^{th} row. Visualization results shows that ARShadowGAN is capable of producing plausible shadows.

6. Limitations

ARShadowGAN is subject to the following limitations:

(1) ARShadowGAN fails when there are large areas of dark or few clues. Examples are shown in Figure 11.

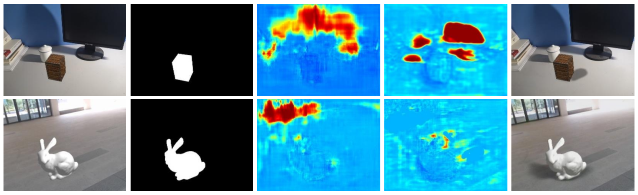


Figure 11. Failure cases of large dark areas and few clues. From left to right: input images without virtual shadows, input masks, attention maps of real-world shadows and their corresponding occluder and output images.

(2) ARShadowGAN only produces planar shadows which do not intersect with real-world shadows and do not exhibit multiple light source characteristics.

(3) ARShadowGAN does not change the shading of the inserted object.

Limitation (1) is because ARShadowGAN relies on clues to infer virtual object shadows while large dark areas seriously interfere with clues. Limitations (2) and (3) exist because the training data does not contain such examples. Extending the Shadow-AR dataset is a possible way to solve limitations (2) and (3).

7. Conclusion and Future Work

In this work, we construct a dataset and propose ARShadowGAN to directly generate plausible virtual object shadows consistent with real-world shading effects without any explicit estimation of the illumination and the geometry. The future work includes addressing the self-shading problem of inserted objects and extending the current Shadow-AR dataset and ARShadowGAN for more complex cases.

Acknowledgement

This work was partly supported by Key Technological Innovation Projects of Hubei Province (2018AAA062), Wuhan Science and Technology Plan Project (No. 2017010201010109), the National Key Research and Development Program of China (2017YFB1002600), the NSFC (No. 61672390, 61972298). The corresponding author is Chunxia Xiao.

References

[1] Ibrahim Arief, Simon McCallum, and Jon Yngve Hardeberg. Realtime estimation of illumination direction for augmented reality on mobile devices. In *Color and Imaging Conference*, volume 2012, pages 111–116. Society for Imaging Science and Technology, 2012.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[3] 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.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 248–255. IEEE, 2009.

[5] Bin Ding, Chengjiang Long, Ling Zhang, and Chunxia Xiao. Argan: Attentive recurrent generative adversarial network for shadow detection and removal. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[6] Marc-Andre Gardner, Yannick Hold-Geoffroy, Kalyan Sunkavalli, Christian Gagne, and Jean-Francois Lalonde. Deep parametric indoor lighting estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[7] 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 Transactions on Graphics (SIGGRAPH Asia)*, 9(4), 2017.

[8] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-Francois Lalonde. Fast spatially-varying indoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[9] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7036–7045, 2019.

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

[11] Maciej Gryka, Michael Terry, and Gabriel J Brostow. Learning to remove soft shadows. *ACM Transactions on Graphics (TOG)*, 34(5):153, 2015.

[12] Ruiqi Guo, Qieyun Dai, and Derek Hoiem. Paired regions for shadow detection and removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 35(12):2956–2967, 2013.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[14] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-Francois Lalonde. Deep sky modeling for single image outdoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[15] 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 (ICCV)*, 2019.

[16] Gang Hua, Chengjiang Long, Ming Yang, and Yan Gao. Collaborative active learning of a kernel machine ensemble for recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1209–1216. IEEE, 2013.

[17] Gang Hua, Chengjiang Long, Ming Yang, and Yan Gao. Collaborative active visual recognition from crowds: A distributed ensemble approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 40(3):582–594, 2018.

[18] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 4700–4708, 2017.

[19] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 36(4):107, 2017.

[20] 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 (CVPR)*, pages 1125–1134, 2017.

[21] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 694–711. Springer, 2016.

[22] Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics (TOG)*, 30(6):1–12, 2011.

[23] Kevin Karsch, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, Hailin Jin, Rafael Fonte, Michael Sittig, and David Forsyth. Automatic scene inference for 3d object compositing. *ACM Transactions on Graphics (TOG)*, 33(3):32, 2014.

[24] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision (IJCV)*, 81(2):155, 2009.

[25] Bin Liao, Yao Zhu, Chao Liang, Fei Luo, and Chunxia Xiao. Illumination animating and editing in a single picture using scene structure estimation. *Computers & Graphics*, 82:53–64, 2019.

[26] Chengjiang Long, Roddy Collins, Eran Swears, and Anthony Hoogs. Deep neural networks in fully connected crf for image labeling with social network metadata. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1607–1615. IEEE, 2019.

[27] Chengjiang Long and Gang Hua. Multi-class multi-annotator active learning with robust gaussian process for visual recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2839–2847, 2015.

[28] Chengjiang Long and Gang Hua. Correlational gaussian processes for cross-domain visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 118–126, 2017.

[29] Chengjiang Long, Gang Hua, and Ashish Kapoor. Active visual recognition with expertise estimation in crowdsourcing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3000–3007. IEEE, 2013.

[30] Chengjiang Long, Gang Hua, and Ashish Kapoor. A joint gaussian process model for active visual recognition with expertise estimation in crowdsourcing. *International Journal of Computer Vision (IJCV)*, 116(2):136–160, 2016.

[31] Chengjiang Long, Xiaoyu Wang, Gang Hua, Ming Yang, and Yuanqing Lin. Accurate object detection with location relaxation and regionlets re-localization. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 3000–3016. IEEE, 2014.

[32] Stephen Robert Marschner and Donald P Greenberg. *Inverse rendering for computer graphics*. Citeseer, 1998.

[33] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *Computer Science*, pages 2672–2680, 2014.

[34] Vu Nguyen, Yago Vicente, F Tomas, Maozheng Zhao, Minh Hoai, and Dimitris Samaras. Shadow detection with conditional generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4510–4518, 2017.

[35] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.

[36] Alexandros Panagopoulos, Chaohui Wang, Dimitris Samaras, and Nikos Paragios. Illumination estimation and cast shadow detection through a higher-order graphical model. In *Proceedings of the IEEE International Computer Vision and Pattern Recognition (CVPR)*, pages 673–680. IEEE, 2011.

[37] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[38] Liangqiong Qu, Jiandong Tian, Shengfeng He, Yandong Tang, and Rynson WH Lau. Deshadownet: A multi-context embedding deep network for shadow removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4067–4075, 2017.

[39] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Illumination from shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, (3):290–300, 2003.

[40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

[41] Tomás F Yago Vicente, Le Hou, Chen-Ping Yu, Minh Hoai, and Dimitris Samaras. Large-scale training of shadow detectors with noisily-annotated shadow examples. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 816–832. Springer, 2016.

[42] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9481–9490, 2019.

[43] Ketaro Wada. labelme: Image Polygonal Annotation with Python. `https://github.com/wkentaro/labelme`, 2016.

[44] Jifeng Wang, Xiang Li, and Jian Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1788–1797, 2018.

[45] Henrique Weber, Donald Prévost, and Jean-François Lalonde. Learning to estimate indoor lighting from 3d objects. In *International Conference on 3D Vision (3DV)*, pages 199–207. IEEE, 2018.

[46] Jinjiang Wei, Chengjiang Long, Hua Zou, and Chunxia Xiao. Shadow inpainting and removal using generative adversarial networks with slice convolutions. In *Computer Graphics Forum (CGF)*, volume 38, pages 381–392. Wiley Online Library, 2019.

[47] Lance Williams. Casting curved shadows on curved surfaces. In *Proceedings of the 5th annual conference on Computer graphics and interactive techniques*, pages 270–274, 1978.

[48] Ryan Christopher Yeoh and Steven ZhiYing Zhou. Consistent real-time lighting for virtual objects in augmented reality. 2009.

[49] Jinsong Zhang, Kalyan Sunkavalli, Yannick Hold-Geoffroy, Sunil Hadap, Jonathan Eisenman, and Jean-Francois Lalonde. All-weather deep outdoor lighting estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[50] Ling Zhang, Chengjiang Long, Xiaolong Zhang, and Chunxia Xiao. Ris-gan: Explore residual and illumination with generative adversarial networks for shadow removal. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[51] Shuyang Zhang, Runze Liang, and Miao Wang. Shadowgan: Shadow synthesis for virtual objects with conditional adversarial networks. *Computational Visual Media*, 5(1):105–115, 2019.

[52] Jiejie Zhu, Kegan GG Samuel, Syed Z Masood, and Marshall F Tappen. Learning to recognize shadows in monochromatic natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 223–230. IEEE, 2010.