
Under the Shadow: Exploiting Opacity Variation for Fine-grained Shadow Detection

Xiaotian Qiao^{1,2}, Ke Xu¹, Xianglong Yang², Ruijie Dong¹, Xiaofang Xia^{1*}, Jiangtao Cui^{1*}

¹School of Computer Science and Technology, Xidian University, China

²Guangzhou Institute of Technology, Xidian University, China

Appendix

This Appendix provides extended technical details and experimental analyses to support the main paper. We elaborate on the construction of the FSD dataset across diverse scenarios: indoor/outdoor environments, single/multiple light sources, shadow overlaps, and varying illumination intensities. We describe the physics-based rendering pipeline in Blender Cycles, leveraging path tracing to simulate realistic shadow interactions governed by radiometric principles. We elaborate in detail on the calculation process of the ground truth values for the opacity shadow mask and supplement the qualitative experimental results of shadow detection. Additionally, we further analyze and present the qualitative experimental results of shadow removal using the opacity shadow mask.

A FSD Dataset

A.1 Additional details

When constructing the dataset, we considered a variety of scenarios encompassing indoor and outdoor environments, varying from single to multiple light sources, weak to strong lighting conditions, and situations with overlapping shadows. The dataset comprises shadow-free images, instance masks, binary shadow masks, self-shadow masks, and opacity shadow masks.

For indoor object models, we source 1030 models from Google Scanned Objects [2]. To prevent interference within the dataset, we manually exclude cases where a model consists of multiple separate parts. Human models are selected from free models on Sketchfab [6] based on quality, resulting in a manual selection of 59 models. In outdoor settings, we incorporate HDRI textures from Ambient CG [1] and Poly Haven. To ensure data quality, scenes with extremely weak lighting or existing shadows are excluded, leading to a final selection of 26 scenes. Below are some examples of scenes included in the dataset.

Indoor, Single Light Scenario. In indoor settings, a cubic space serves as the base for constructing the scenes. A total of 428 floor textures from Poly Haven are utilized to adorn the indoor space. Subsequently, 1-5 objects are randomly positioned near the world coordinate origin, with their sizes randomly adjusted to diversify the dataset. The placement strategy for objects in this single-light indoor scenario (refer to Figure. 1) involves distributing the $xyxy$ coordinates within a narrow strip area to simulate common real-world object arrangements. For this scenario, a single light source (such as parallel light or SUN in Blender) is employed to replicate indoor reflection effects, aligning with typical indoor scene configurations. The variability in light sources encompasses factors like intensity, blur level (indicating the softness of shadows), light direction, position, and other parameters. Regarding the camera settings in both indoor and outdoor scenes, the position and angle of the camera are randomized, excluding specifications such as focal length and sensor size.



Figure 1: Indoor, single light

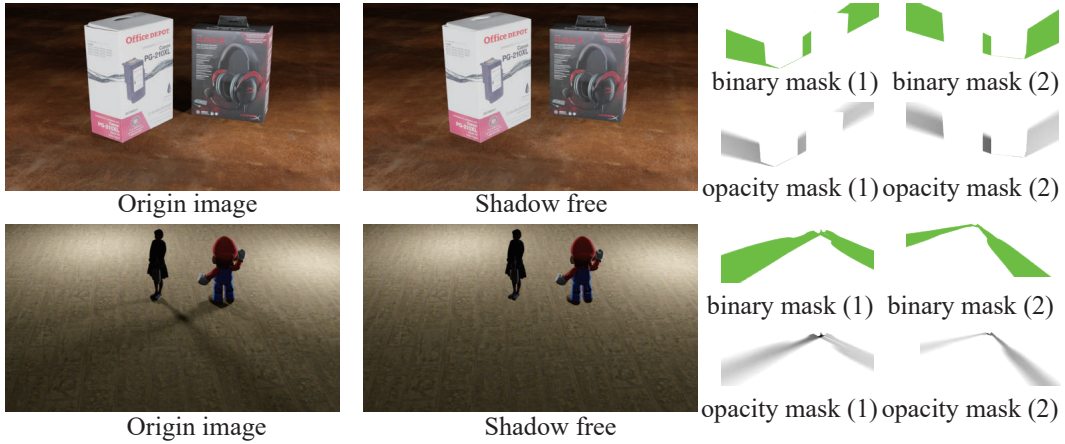


Figure 2: Indoor, multiple lights

Indoor, multiple lights. In more intricate scenarios (refer to Figure. 2), the presence of multiple light sources may give rise to various shadow phenomena. This particular scenario has received limited attention in prior research. To enrich the dataset’s scope, we have included this scenario to expand its comprehensiveness. In this setting, objects are placed with completely random xyy coordinates. As for lighting, we have implemented a strategy to randomly generate lighting effects, encompassing point light (POINT in Blender), sunlight (parallel light, SUN), spotlight (SPOT), and area light (AREA). The selection of light types and quantities is randomized, and the attributes of the lights are determined in a similar random fashion as previously described.

Shadow overlaps. When multiple light sources are utilized, shadow intersections occur (see Figure. 2), a phenomenon that cannot be adequately elucidated by a binary shadow mask alone. Our dataset intentionally includes scenarios featuring shadow intersections. The distinction between this and multiple-light scenarios lies in the placement strategy of objects and lights. Specifically, two objects are positioned to form a specific angle, after which two point light sources are placed in the directions of these objects (from the origin to each object) to generate intersecting shadows.

Outdoor, weak light. For all outdoor scenes, we create a hemisphere and utilize HDRI textures to establish the environment, thereby integrating lighting details from the textures organically. In instances of low lighting intensity (refer to Figure. 3), shadows exhibit a blurred and ghosted appearance, rendering binary shadow masks less efficient for shadow elimination. Given that the lighting details are predetermined by the texture, the only elements subject to randomness are the camera angles and the placement of objects.



Figure 3: Outdoor, weak light

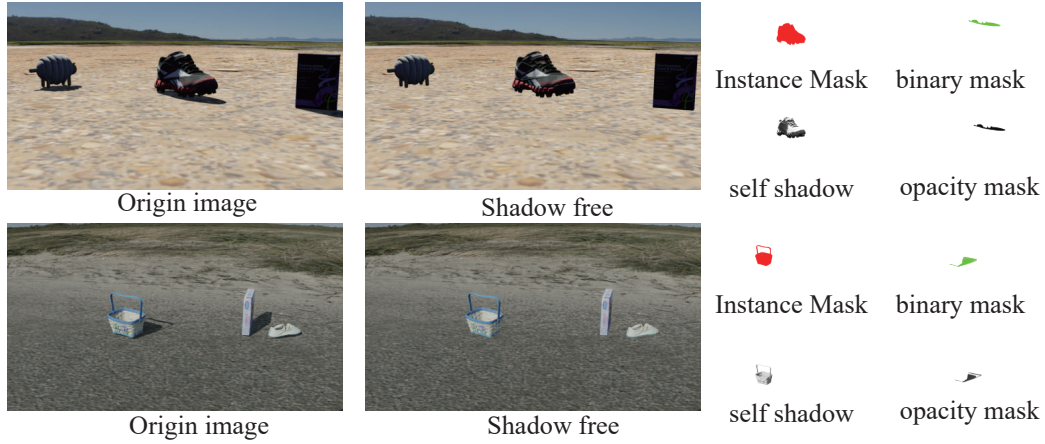


Figure 4: Outdoor, strong light

Outdoor, strong light. In outdoor environments characterized by intense lighting conditions (see Figure. 4), representative of real-world settings, shadows tend to be distinct and deep. Our dataset comprises a substantial quantity of such scenes. The primary contrast between these two outdoor setups lies in the HDRI textures employed.

Once the scene is set up, we employ keyframe animation to produce the initial image, a shadow-free rendition, and all desired annotations for each object. Blender sequentially generates the animation frame by frame, which is later subjected to post-processing. Due to inherent noise in the original opacity shadow mask from Blender Cycles, we begin by applying noise reduction techniques. Subsequently, we derive the binary shadow mask based on the opacity shadow mask. Following the processing of all samples, we compile a JSON file to document our datasets in the format specified by SOBA [10]. Comprehensive details regarding the scene setup will be released.

B Additional Shadow Detection Results

B.1 Shadow Rendering Process

Blender Cycles uses the path tracing algorithm, which is based on the following equation:

$$L_o(p, \omega_o) = L_e(p, \omega_o) + \int_{\Omega} f_r(p, \omega_i, \omega_o) L_i(p, \omega_i) n \cdot \omega_i d\omega_i \quad (1)$$

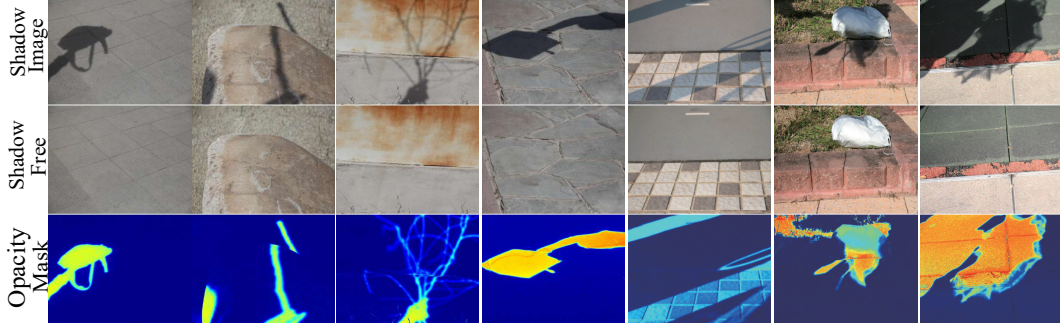


Figure 5: The figure demonstrates the shadow image, the shadow-free image, and the extracted opacity shadow mask. Compared to the binary mask, the opacity shadow mask not only indicates the location of the shadows but also reveals the intensity of the shadows and the degree of background.

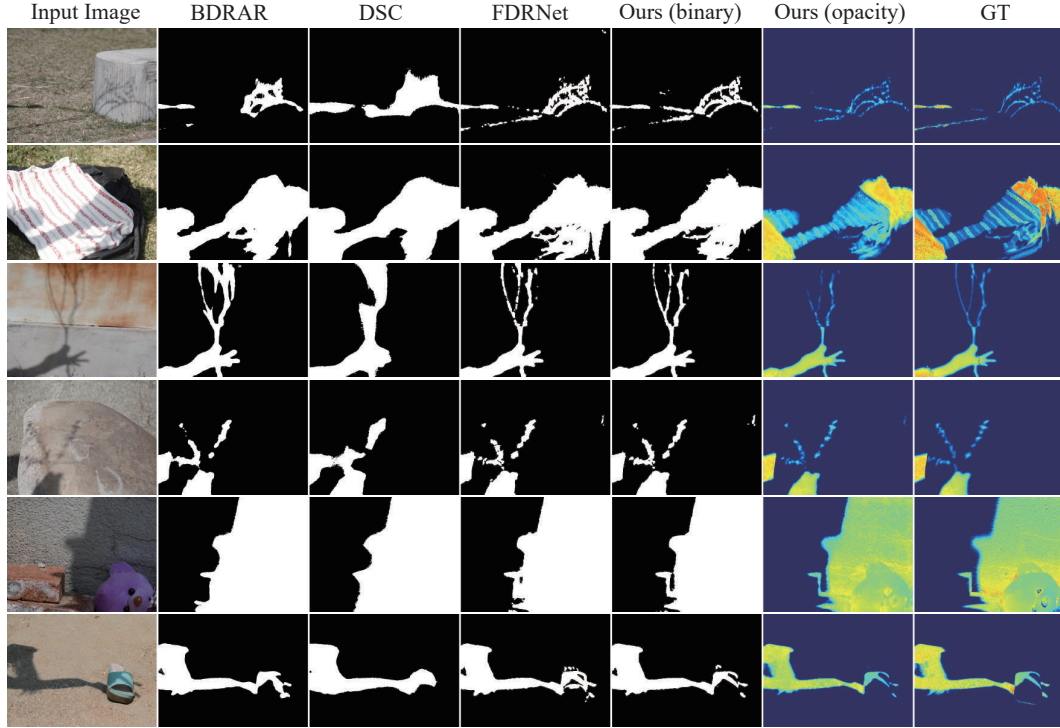


Figure 6: The figure illustrates qualitative comparison results between ours and the state-of-the-art methods on the SRD dataset [8], our method exhibits significant advantages in visual quality.

It uses radiometry to describe lighting more accurately, resulting in more realistic images. The term $L_i(p, \omega_i)n \cdot \omega_i$ describes the radiance of the light coming from solid angle ω_i in the environment, contributing to the radiance at point p . Thus, $\int_{\Omega} L_i(p, \omega_i)n \cdot \omega_i d\omega_i$ represents the contribution of all light rays in the hemisphere Ω , i.e. all light in the environment. The term f_r determines how much light will be reflected. Although with numerous BSDF, BRDF models available, they only determine the material of the object. The determinant of shadows is the term L_i . The term L_e represents the object's self-emission.

The ground truth opacity shadow mask α_{gt} is calculated from the shadow image S and the shadow-free image F . The specific steps are as follows: first, convert S and F to the YCbCr color space and extract the Y channel, then compute the ratio of Y_S to Y_F , apply a low-pass filter to remove noise, and finally obtain α_{gt} through thresholding.

$$\alpha_{gt} = \max(t, f(\frac{Y_S}{Y_F})) \quad (2)$$

Here, Y_S and Y_F represent the Y channels of the shadow image and the shadow-free image in the YCbCr color space, respectively. f denotes a low-pass filtering operation with $\sigma = 0.5$, and t is the threshold, which is set to 0.1 in this paper.

Figure 6 presents additional experimental results on the SRD dataset [8]. Our method outperforms the state-of-the-art approaches across multiple examples, our method demonstrates exceptional performance in fine-grained shadow detection. For instance, in rows 2 and 4, our method can predict the transparency changes of shadows, while in rows 1 and 5, it achieves more precise detection of soft shadow boundaries. These results further validate the reliability of the shadow detection approach guided by shadow opacity.

B.2 Ablation Study

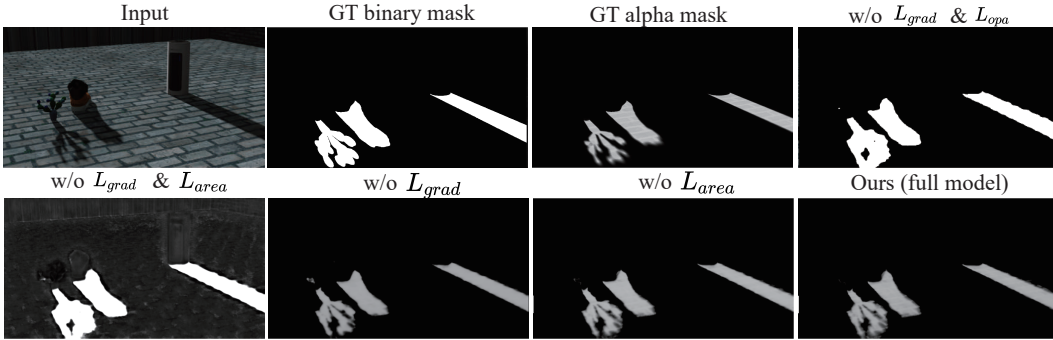


Figure 7: Qualitative comparison results of different ablated versions of our model.

Qualitative comparison. The ablation results in Figure 7 validate our design: the opacity mask enables robust shadow positioning and transparency estimation, complemented by \mathcal{L}_{grad} for regulating gradient smoothness and \mathcal{L}_{area} for boosting accuracy in extreme scenarios.

Quantitative Evaluation. We additionally conducted an ablation study of our loss functions on the ISTD dataset [9]. As it lacks ground truth annotations for shadow opacity, we employed the Balanced Error Rate (BER) metric for evaluation. The results, presented in the table below, demonstrate that our full model outperforms all ablated versions, validating the effectiveness of our design choices.

Method	BER ↓	Shadow ↓	Non Shad. ↓
w/o \mathcal{L}_{grad} and \mathcal{L}_{opa}	1.73	0.92	2.53
w/o \mathcal{L}_{grad} and \mathcal{L}_{area}	1.55	2.21	0.88
w/o \mathcal{L}_{grad}	1.50	2.25	0.75
w/o \mathcal{L}_{area}	<u>1.46</u>	2.15	<u>0.77</u>
Ours (full)	1.32	<u>1.67</u>	0.96

Table 1: Ablation study on ISTD.

C Additional Shadow Removal Results

In our shadow removal experiments, we improved upon the state-of-the-art ShadowDiffusion [4] method. Specifically, we replaced the binary shadow mask with the detection results from our method for training, while keeping other hyperparameters and optimization strategies consistent with the original ShadowDiffusion. The experiments were conducted on an RTX 4090 GPU, with the training epochs set to 1000. We employed the Adam optimizer (with momentum parameters of (0.9, 0.999)) and an initial learning rate of 3×10^{-5} . The model weights were initialized using the Kaiming initialization technique [5], and an exponential moving average (EMA) strategy with a decay rate of 0.9999 was applied across all experiments.

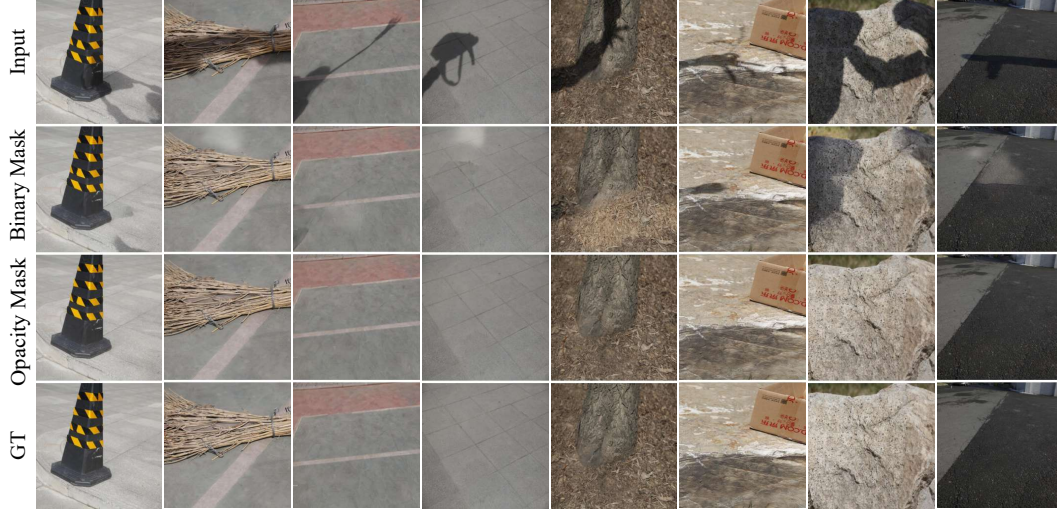


Figure 8: We compared the results of using binary masks and opacity masks for shadow removal in training the ShadowDiffusion[4] model on SRD Dataset [7]. The results indicate that using opacity masks instead of binary masks for annotation leads to better visual effects.

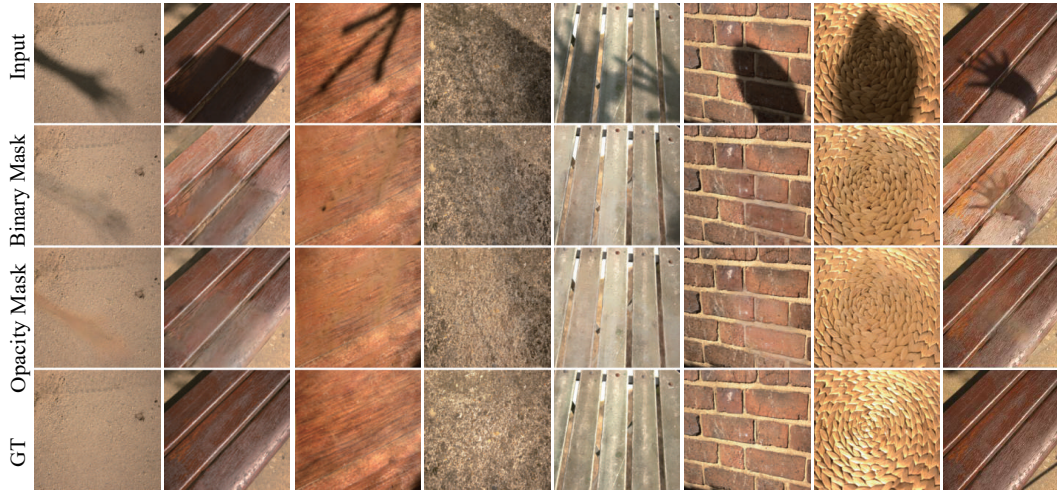


Figure 9: Results of different inputs on the LRSS [3] dataset using ShadowDiffusion [4].

Figure 8 showcases the performance superiority of the model trained using opacity shadow masks compared to the model trained with binary shadow masks across various test images. The qualitative analysis demonstrates that, under the same training conditions, the utilization of opacity shadow masks enables the model to effectively eliminate shadows while maintaining more detailed information about the underlying scene. Particularly in complex scenarios with blurred shadow boundaries, the model trained with opacity shadow masks shows notably enhanced performance.

To validate the effectiveness of the opacity shadow mask, we conducted supplementary experiments on the LRSS soft shadow dataset [3], which contains 46 pairs of shadow and shadow-free images. The experiments adopted a strategy of training on the SRD dataset [7] and evaluating on the LRSS dataset [3]. As illustrated in Figure. 9, the opacity shadow mask demonstrated significant advantages in restoring soft shadow edges and background textures, thereby fully validating its effectiveness.

References

- [1] Lennart Demes. ambientcg - cc0 textures, hdris and models, July 2024. URL <https://ambientcg.com/>.

- [2] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items, 2022. URL <https://arxiv.org/abs/2204.11918>.
- [3] Maciej Gryka, Michael Terry, and Gabriel J Brostow. Learning to remove soft shadows. *ACM TOG*, 2015.
- [4] Lanqing Guo, Chong Wang, Wenhan Yang, Siyu Huang, Yufei Wang, Hanspeter Pfister, and Bihan Wen. Shadowdiffusion: When degradation prior meets diffusion model for shadow removal. In *CVPR*, 2023.
- [5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.
- [6] Sketchfab Inc. Sketchfab - the best 3d viewer on the web, July 2024. URL <https://sketchfab.com/>.
- [7] Hieu Le and Dimitris Samaras. Shadow removal via shadow image decomposition. In *ICCV*, 2019.
- [8] Liangqiong Qu, Jiandong Tian, Shengfeng He, Yandong Tang, and Rynson Lau. Deshadownet: A multi-context embedding deep network for shadow removal. In *CVPR*, 2017.
- [9] Jifeng Wang, Xiang Li, and Jian Yang. Stacked conditional generative adversarial networks for jointly learning shadow detection and shadow removal. In *CVPR*, 2018.
- [10] Tianyu Wang, Xiaowei Hu, Qiong Wang, Pheng-Ann Heng, and Chi-Wing Fu. Instance shadow detection. In *CVPR*, 2020.