DeepShadow: Neural Shape from Shadow — Supplementary Materials

Asaf Karnieli, Ohad Fried, and Yacov Hel-Or

School of Computer Science, Reichman University, Herzliya, Israel
asafkarnieli@gmail.com
{ofried,toky}@idc.ac.il

1 Shadow and light extraction network

Since shadow maps are not always available, we train a network to estimate them from input photometric stereo images. This network runs as a pre-processing step on the photometric stereo data, to produce shadow maps that DeepShadow can use as inputs. Our model also estimates the light direction, since this is also not always available. We use both a publicly available photometric stereo dataset, as well as our own renders — which are needed since there is no public dataset that has photometric stereo shadow ground-truth.

Please note that our goal is estimating depth from shadow maps. As we have shown, in certain cases DeepShadow with shadow maps as inputs may result in better shape estimation than shape-from-shading techniques. We trained the shadow and light extraction model solely to be able to use our method on datasets which do not have ground-truth shadow maps or light directions, so that we are able to test our method on more types of objects and scenes. Using the light extraction model, we estimate the direction of lights (located at infinity), and then convert these to point-lights by projecting onto the unit sphere and multiplying by a constant, empirically set to twice the distance between camera and object.

1.1 Network Architecture

The shadow estimation model is illustrated in Fig. 1. It is used for estimating light directions and shadow maps given photometric stereo image inputs. Although the model also outputs normal maps, these are only used during the training and are discarded during the inference. The input images have dimensions of $S \times C \times W \times H$, S being the number of input images (sequence dimension), C is the image color channel, and $H \times W$ is the spatial image size.

The complete model is composed of four hybrid Transformer-Convolution layers (ConvTransformers): the first block is a ConvTransformer for extracting features from the input images, and the second and third blocks are two ConvTransformers for estimating shadows and for estimating light directions. The last custom block is used for estimating the normals from the features (not illustrated in Fig. 1).

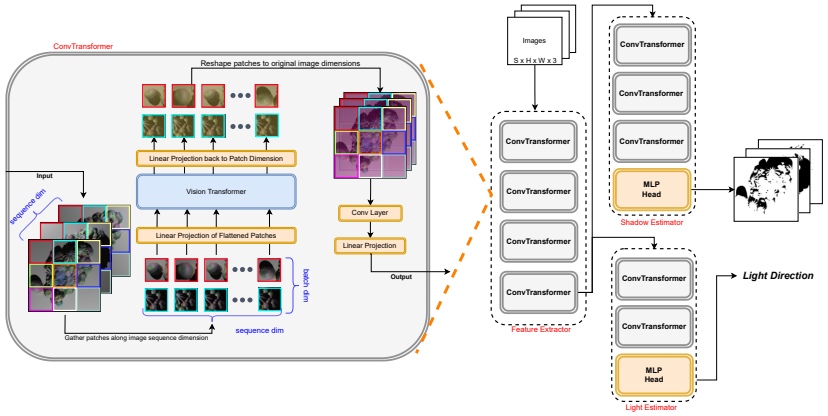

Fig. 1. Shadow Estimation Model. The model receives as inputs a sequence of photometric stereo images and outputs the estimated light directions and shadow maps. Each image is split to patches, and patches are gathered by the sequence index. Each such sequence is fed separately to the ConvTransformer. The model is composed of a four layer ConvTransformer which outputs intermediate features. These features are used to generate the final outputs, using 2 smaller ConvTransformers. The ConvTransformer can be viewed in detail in the bottom of the figure.

The feature-extraction ConvTransformer splits each input image into an 8x8 patch, as applied in Vision Transformer (ViT) [5]. In contrast to the regular ViT, we group all patches along the *sequence* dimension, since the main target of the model is predicting per-pixel output for each patch. Since predicting shadows from photometric stereo images is essentially a threshold-based problem, we choose to use an attention-based model and compare all spatially co-located patches, instead of comparing all patches in a single image. These patches are then used as a sequence of inputs to the transformer. Each sequence is passed separately through a ConvTransformer (sequences can be batched together) to produce a sequence of intermediate features extracted from the relevant patches. Once all the patches have passed through the transformer, they are reshaped back to the original image dimensions, and then passed through a convolution layer to output the features. The feature-extraction ConvTransformer is composed of 4 Transformer-Convolution pairs, with 16 attention heads and latent MLP dimension of 1024.

The light direction block uses 2 ConvTransformer blocks, with a 128 dim MLP and 4 attention heads. The shadow estimation block uses 3 ConvTransformer blocks with 6 attention heads and an MLP dimension of 128. It utilizes a Sigmoid function for outputting values between 0 and 1. We also estimate the surface normals from the features, using a linear projection layer and two convolution layers. This is done in order to be able to learn from datasets which have photometric stereo data and ground-truth normals.

1.2 Training Details

We use the Blobby and Sculptures datasets [1], which contain photometric stereo images, light directions and surface normals. We also render a shadow dataset composed of 10 objects downloaded from Sketchfab[1]. Each object was rendered in 32 different view angles and with 32 light angles for each view. We use Blender [3] to render our datasets. We use all three datasets during training in a ratio of 1:1:10, i.e., for every 10 iterations over the shadow dataset, we iterate once over the Blobby and Sculptures datasets.

During training, we randomly crop each input images to 64×64. We use random noise and color jitter augmentations, as well as randomize the sequence length between 16 and 32 inputs, in order to make the transformer agnostic to the number of input images.

We use the following loss function:

$$\mathcal{L} = \frac{1}{M} \sum_{m} (\mathcal{L}_N^m + \mathcal{L}_S^m + \mathcal{L}_L^m) \tag{1}$$

where $\mathcal{L}_N^m$ is the normal loss, $\mathcal{L}_S^m$ is the shadow reconstruction loss and $\mathcal{L}_L^m$ is the light direction loss. The sum is performed over all $m \in [0, M]$ sets of photometric images in the dataset. Each such set has $k \in [0, K]$ images of size $H \times W$ along with the associated light directions ℓ_k and a ground-truth normal map N. Our rendered dataset also has ground-truth shadow maps S_k. The complete loss combines between the loss of the ground-truth normals N and the predicted normals $\hat{N}$,

$$\mathcal{L}_N^m = \frac{1}{HW}(1 - N^m \cdot \hat{N}^m), \tag{2}$$

the L1 loss of the ground-truth and predicted shadow maps

$$\mathcal{L}_S^m = \frac{1}{KHW} \sum_{k} |S_k^m - \hat{S}_k^m|, \tag{3}$$

and the cosine embedding loss between the ground-truth light direction ℓ_k^m and the estimated direction $\hat{\ell}_k^m$

$$\mathcal{L}_L^m = \frac{1}{K} \sum_{k} (1 - cos(\ell_k^m, \hat{\ell}_k^m)). \tag{4}$$

We omit the supervision on the lights when using our dataset, and omit the shadow supervision when using Blobby and Scupltures datasets. We train using the Adam optimizer [7] for 1000 epochs with an initial learning rate of 1×10^{-4} which is decreased by a factor of 0.8 every 15 epochs.

The results of the shadow estimation can be seen in Fig. 2.

[1] https://sketchfab.com/

Fig. 2. Sculpture head object shadow estimation results.

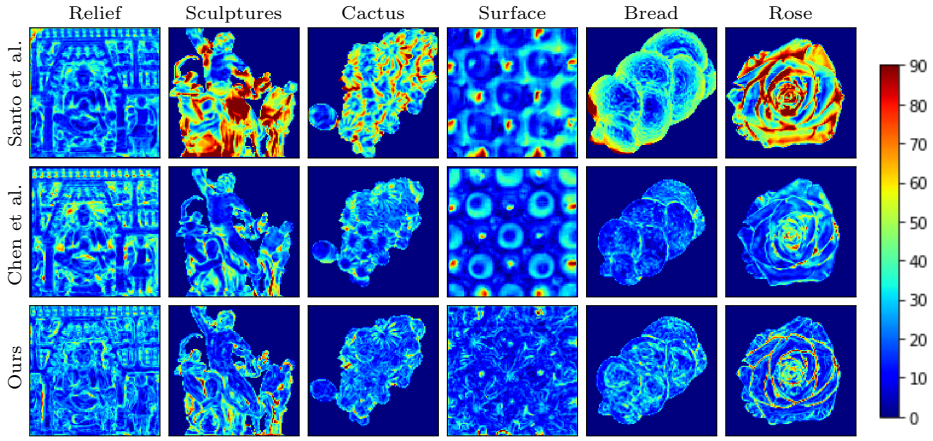

Fig. 3. Normal error maps comparing our method to [2] and [8].

2 Additional Results

In this section, we present more results of the DeepShadow method. We also examine the performance of the shadow estimation network on data which lacks ground-truth shadows.

2.1 Normal map errors

Fig. 3 shows the previously shown objects' normal map errors. Our method produces less normal errors on the *Relief*, *Cactus*, *Surface* and *Rose* objects compared to other methods.

2.2 Specular and diffuse objects

We test our method's resiliency to specular inputs and compare to that of a shape-from-shading method. We rendered a modified version of our *rose* object, with two different materials; highly specular and diffuse (Fig. 4).

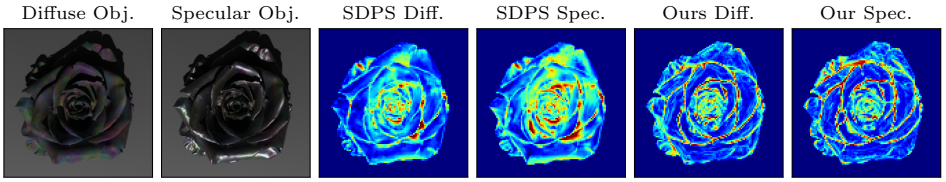

Fig. 4. Effect of specularities on angle error of output normals.

We evaluated our algorithm (including the shadow extraction model) on these objects, and compared the surface normal results to SDPS [2]. The diffuse object produces 25.02 MAE and the metallic object produces 34.35 MAE using SDPS, while our method achieves 26.50 MAE and 27.76 MAE, respectively. With specularity added, we can see a big drop in accuracy using SDPS, while our method achieves a smaller drop.

2.3 Shadow maps reconstruction error

We present the shadow reconstruction error on two objects from our rendered dataset. The reconstruction has two types of errors. The first is due to the nearest neighbor rounding used for the boundary sampling in the R2 method, and can be seen clearly in the third row *cactus'* shadow error, in the upper area, and in the *surface's* second, third and sixth rows. The second error can mostly be seen in the edges of the objects (e.g., last row in Fig. 5). The source of the error are edges in the depth map. Recall we generate a shadow line scan from the light source to each boundary pixel, and estimate the depth map value for every pixel in the image. The error can be minimized by sampling each line in a denser fashion rather than sampling a coordinate for every pixel, although it would come at a cost of computational cost.

2.4 Results on objects from Dome dataset

We present our results on *vase*, *face* and *golf ball* objects from the Dome dataset [6]. The *vase* results can be seen in Fig. 6. Previous shape-from-shadow methods [4, 9, 10] have used a threshold on gray-scale images to estimate shadows from images. As can be seen in Fig. 6, simple thresholds fail on an object such as the *vase*, which has specular highlights. We show the results for 3 different thresholds by taking values of 0.4, 0.5 and 0.6. Each threshold fails to produce accurate shadow maps in specific areas.

We also show our depth and normal estimation results in Fig. 7, which have been discussed in the main paper.

2.5 Results on real object

We present qualitative results on the *hand*[2] and *statue*[3] in Fig. 8. The objects were acquired in a half-dome setting with 34 different illuminations. Ground

[2] Work of Man Ray, sampled at the Museum of Israel.
[3] Privately sampled.

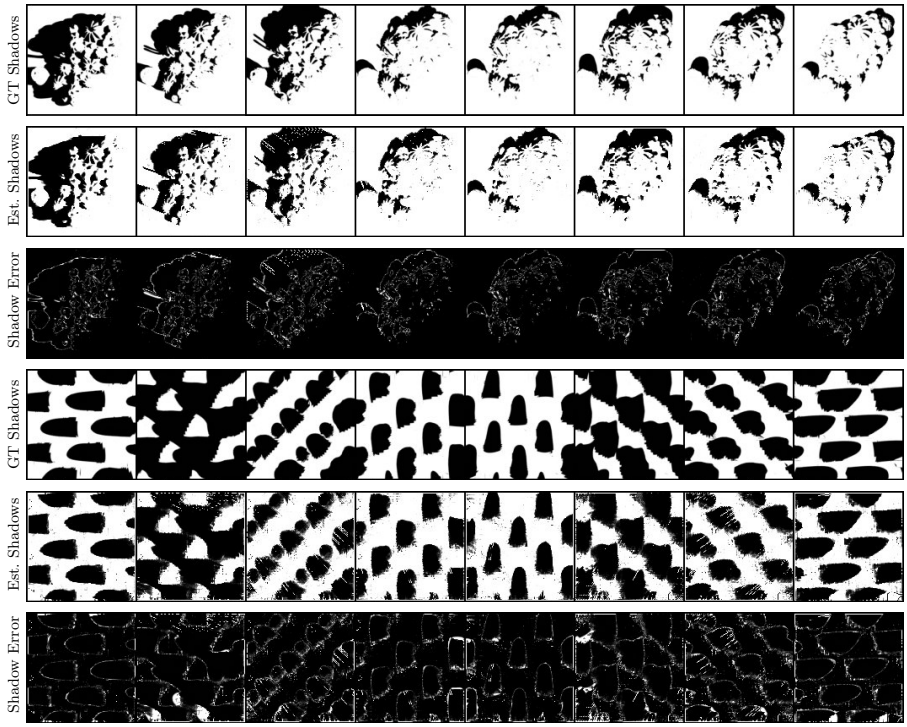

Fig. 5. Shadow reconstruction error. The top 3 rows are from the *cactus* object and the bottom 3 rows are from the *surface* object. Ground-truth and estimated shadows are shown, along with the L1 error between them. Each column represents a different illumination direction.

truth normals and depth are not available on this dataset, as well as light directions, which were estimated using the model described in Section 1. Our method requires the intrinsic camera parameters which were not known, thus had to be roughly estimated by guessing the object's size, and assuming a typical 50mm focal length.

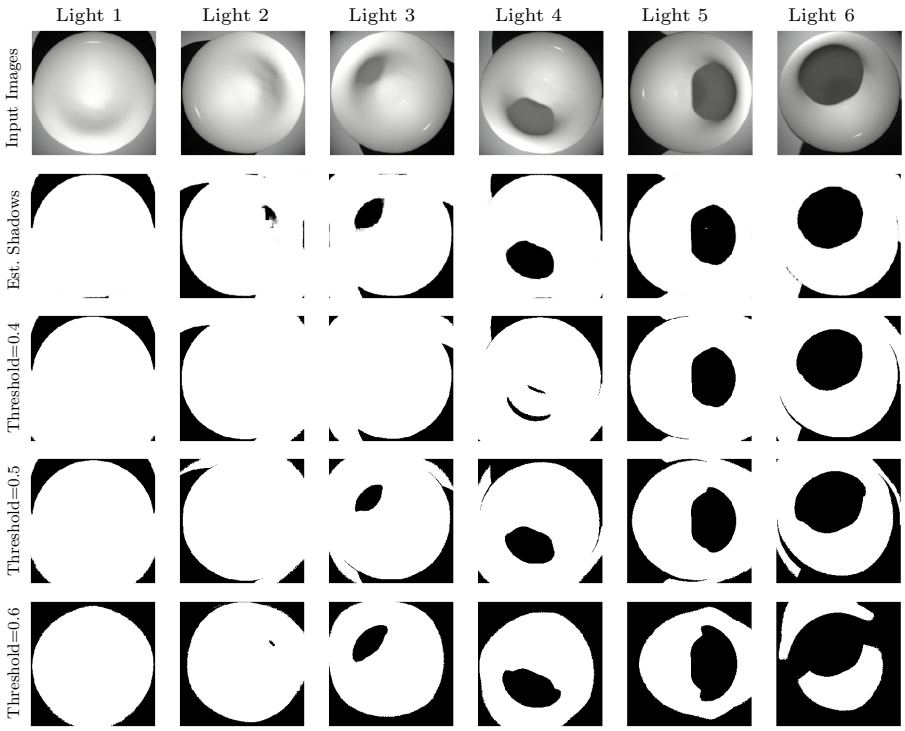

Fig. 6. Vase object shadow estimations. The top row contains the input images, the second row is our estimated shadow results using the model described in Section 1. The three bottom rows are a baseline result by taking thresholds of 0.4, 0.5 and 0.6 over the grayscale levels. Each column is a different light direction. A threshold of 0.4 fails on light directions 2, 3, and 4; a threshold of 0.5 fails on light directions 2 and 6 (external region); a threshold of 0.6 fails on light directions 4, 5, and 6.

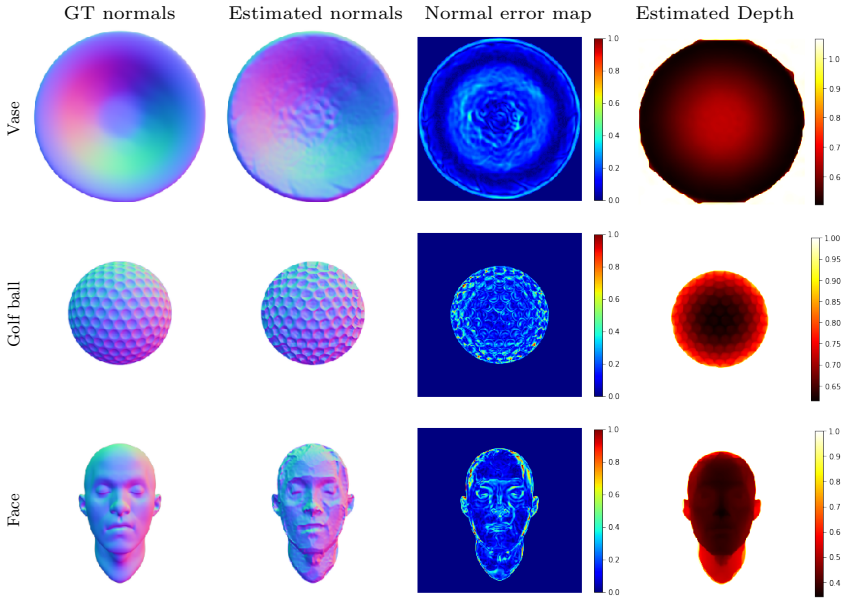

Fig. 7. Results on objects from [6]. Each row (from left to right) consists of ground-truth normals, estimated normals, normal error map and estimated depth. As described in the paper, the results of DeepShadow on the *vase* object outperform other attempted methods. The normal map produced from the *golf ball* has errors around the edges. The *face*'s estimated normal map has errors mostly around the forehead area, since that area is smooth and sparse in shadows.

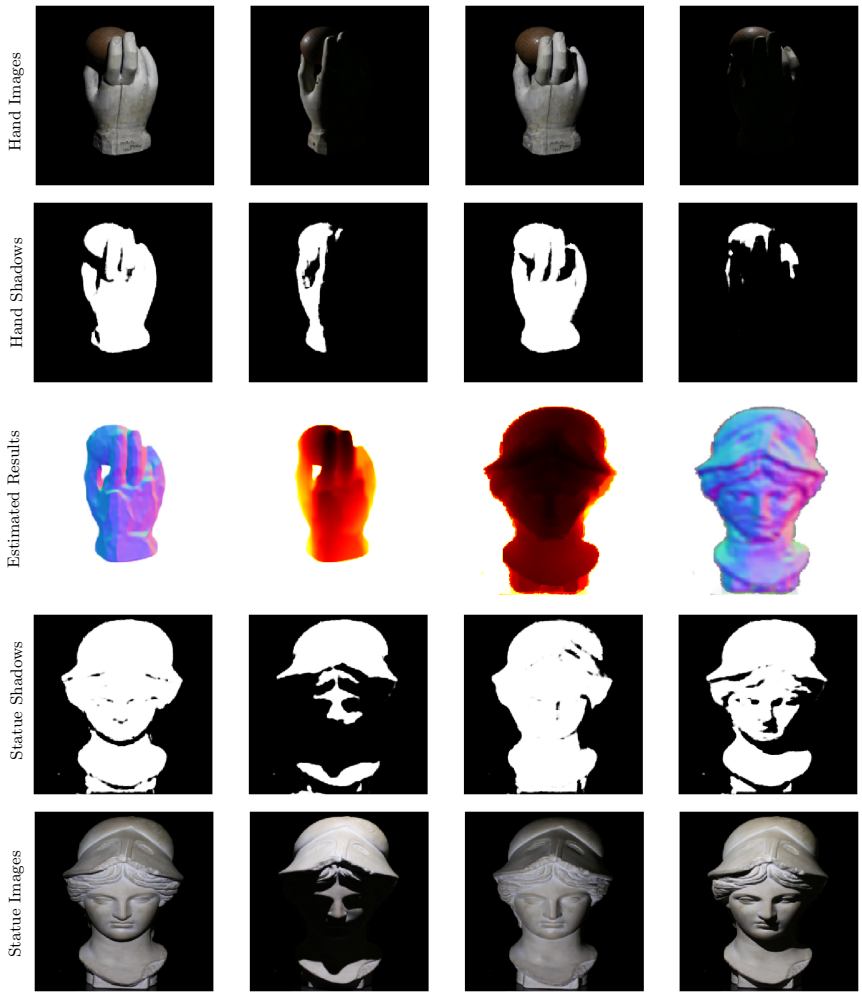

Fig. 8. Input images, shadows, estimated depth and normals. The upper row includes the *hand* object images, and the second row their estimated shadows. The third row contains the estimated depth and surface normals of both objects. The fifth row includes the *statue* image inputs and the forth row their estimated shadow maps. We can observe DeepShadow is able to extract fine detail in areas such as the helmet and hair of the statue, and fails in smooth areas such as its neck.

References

[1] Guanying Chen, Kai Han, and Kwan-Yee K. Wong. "PS-FCN: A Flexible Learning Framework for Photometric Stereo". In: *ECCV*. 2018.

[2] Guanying Chen et al. *Self-calibrating Deep Photometric Stereo Networks*. 2019. arXiv: 1903.07366 [cs.CV].

[3] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam, 2018. URL: http://www.blender.org.

[4] M. Daum and G. Dudek. "On 3-D surface reconstruction using shape from shadows". In: *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*. 1998, pp. 461–468. DOI: 10.1109/CVPR.1998.698646.

[5] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *CoRR* abs/2010.11929 (2020). arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929.

[6] Berk Kaya et al. *Uncalibrated Neural Inverse Rendering for Photometric Stereo of General Surfaces*. 2021. arXiv: 2012.06777 [cs.CV].

[7] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[8] Hiroaki Santo, Michael Waechter, and Yasuyuki Matsushita. "Deep Near-Light Photometric Stereo for Spatially Varying Reflectances". In: *Computer Vision – ECCV 2020*. Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 137–152. ISBN: 978-3-030-58598-3.

[9] Silvio Savarese et al. "3D Reconstruction by Shadow Carving: Theory and Practical Evaluation". In: *International Journal of Computer Vision* 71 (Mar. 2007), pp. 305–336. DOI: 10.1007/s11263-006-8323-9.

[10] Yizhou Yu and Johnny Chang. "Shadow Graphs and 3D Texture Reconstruction". In: *International Journal of Computer Vision* 62 (Apr. 2005), pp. 35–60. DOI: 10.1023/B:VISI.0000046588.02227.3b.