DeepShadow: Neural Shape from Shadow

Asaf Karnieli, Ohad Fried, Yacov Hel-Or
Reichman University, Herzliya, Israel

Reichman University

ECCV TEL AVIV 2022

Introduction

Goal: Estimate accurate depth map using **only** cast-shadow and attached-shadow inputs.

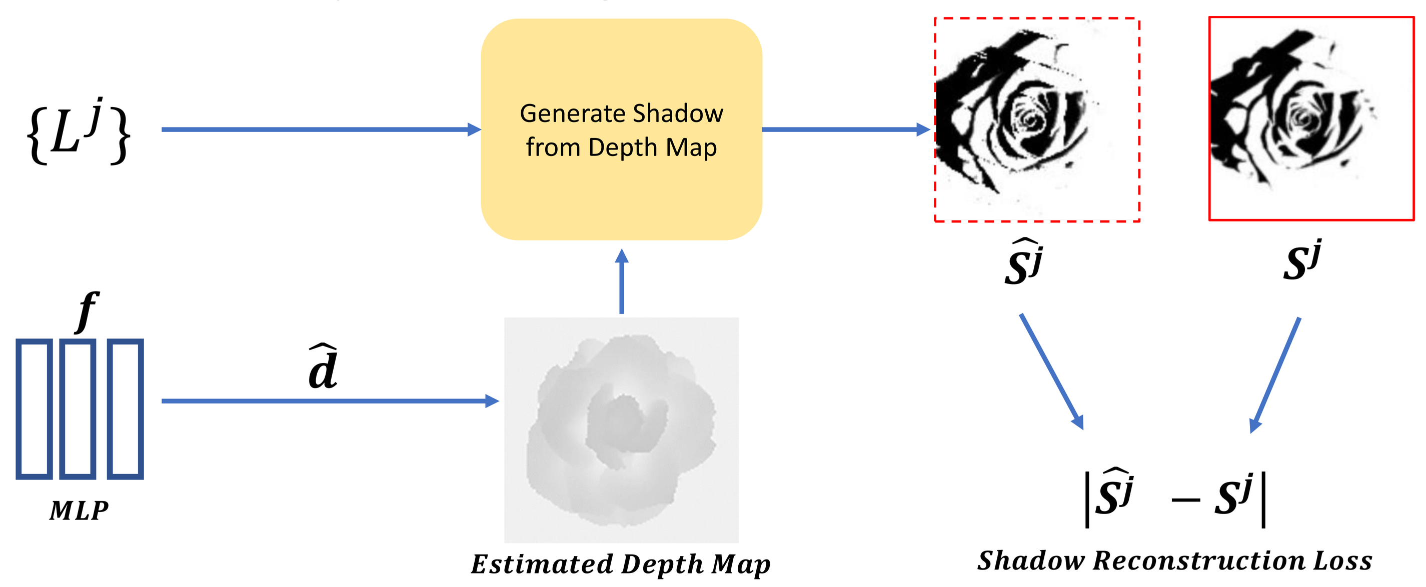
Input: Multiple shadow maps captured under varying light locations.
Output: Depth map and surface normals.

Contributions:

- First deep-learning based method for recovering depth and surface normals from shadow maps.
- Linear-time tracing calculations, as opposed to quadratic complexity used when tracing in NeRF-like methods.
- Insensitivity to specular highlights and varying light intensities.
- High quality results for both near-field and far-light settings.
- One-shot depth-map estimation, avoiding costly data collection & training.
- Novel shadow calculation globally aggregating spatial information.

Method

- Assumptions: Perspective projection and point-light source, known intrinsic and extrinsic camera parameters.
- Estimate the depth map $\hat{d}$ using an implicit network f_θ.
- For each light source L^j, calculate the corresponding shadow map $\hat{S}^j$ using the estimated depth.
- Use a loss between the estimated shadow $\hat{S}^j$ and the real shadow S^j.
- Optimize θ and $\hat{d}$ by minimizing the loss.



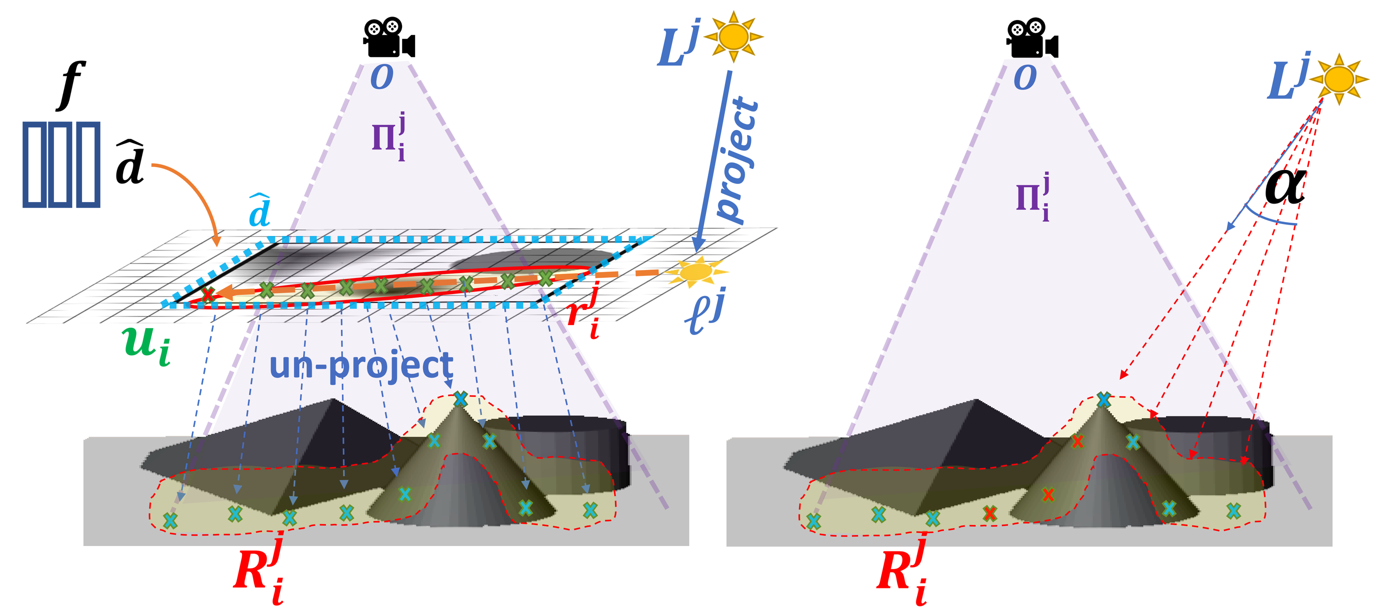
Loss: $\mathcal{L}$ is the sum of reconstruction loss L^j_{rec} and intensity–guided depth regularization $\mathcal{L}_d$. $\bar{I}$ is the mean of N input images, HW is the image size.

$$\mathcal{L} = \frac{1}{N}\sum_j \mathcal{L}^j_{rec} + \lambda \mathcal{L}_d \quad (1) \qquad \mathcal{L}^j_{rec} = \frac{1}{HW}|S^j - \hat{S}^j| \quad (2)$$

$$\mathcal{L}_d = \sum_{ij} \left|\partial_x \hat{d}_{ij}\right| e^{-\|\partial_x \bar{I}_{ij}\|} + \left|\partial_y \hat{d}_{ij}\right| e^{-\|\partial_y \bar{I}_{ij}\|} \quad (3)$$

Shadow Map Estimation

Left: Project the light source L^j onto the image plane to receive ℓ^j. Create a ray of points $\mathbf{r}^j_i$ between ℓ^j and a point on the image border $\mathbf{u}_i$. Estimate the depth $\hat{d}$ for each point in $\mathbf{r}^j_i$, then un-project to world coordinates (noted as R^j_i). Right: Use the Shadow Line Scan algorithm in 3D to calculate shadowed pixels. Red points are shadowed, since their angle to the light source is smaller than α.

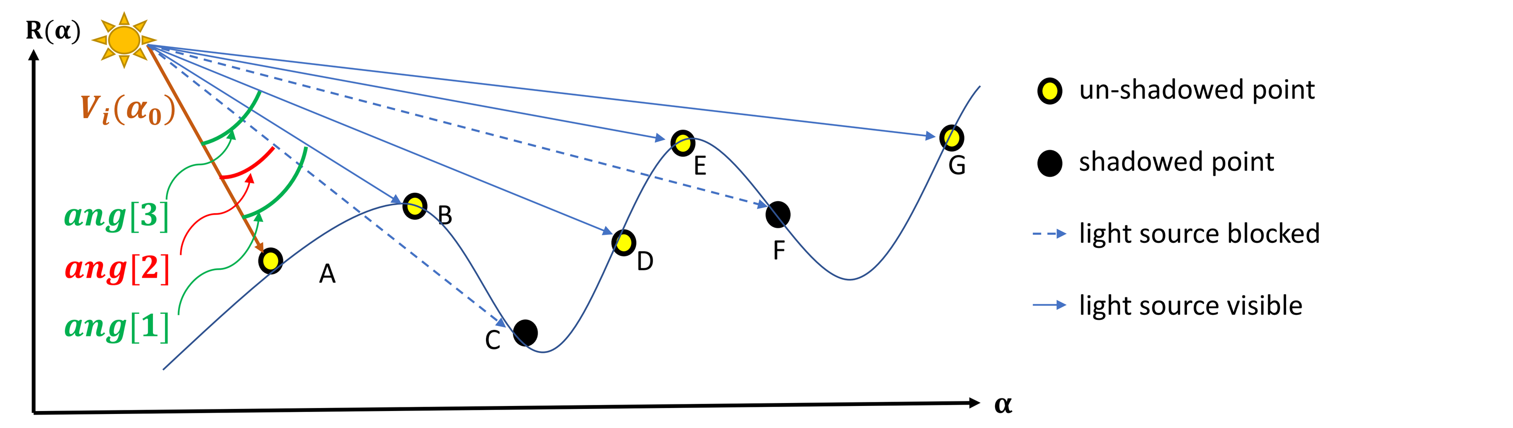


Shadow Line Scan Algorithm: Given a 1D height map slice, the vector between the light source L and point $R(a)$ is defined as $V(\alpha) = L - R(a)$, where α denotes the parameter of the un-projected ray R. We compare $ang[i]$ (the angle between $V[\alpha_i]$ and $V[\alpha_0]$) to all previous angles. If this angle is larger than all previous angles, this point is not shadowed.

$$s_L[\alpha_i] = \max\left(ang[\alpha_i], s_L[\alpha_{i-1}]\right) \quad (4)$$

The final shadow / no-shadow result (σ being the *sigmoid* function) -

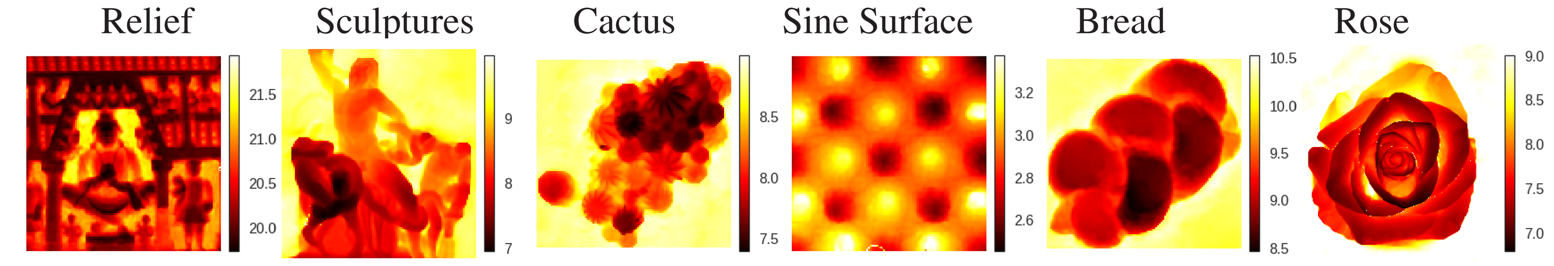
$$s[\alpha_i] = 2\sigma\Big(ang[\alpha_i] - s_L[\alpha_i]\Big) \quad (5)$$



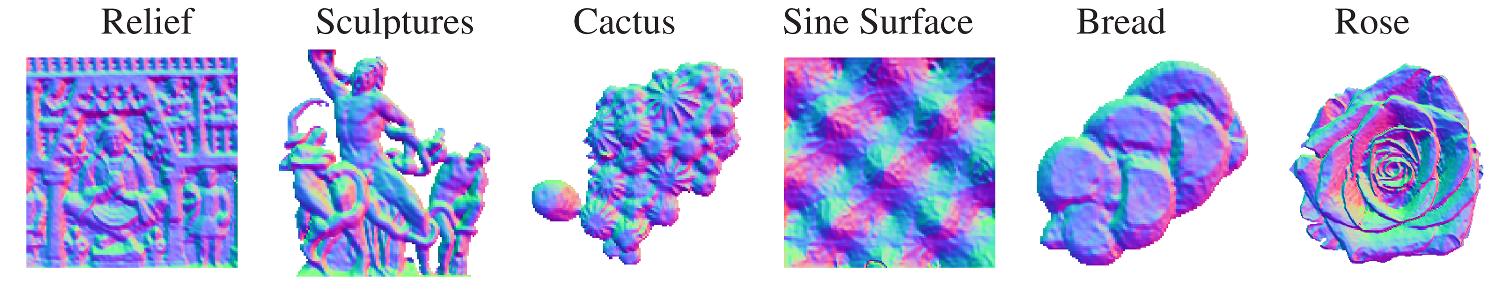
Experiments & Results

Shadow Maps Used:

Depth Estimations:



Surface Normal Estimations:



Quantitative Evaluation:

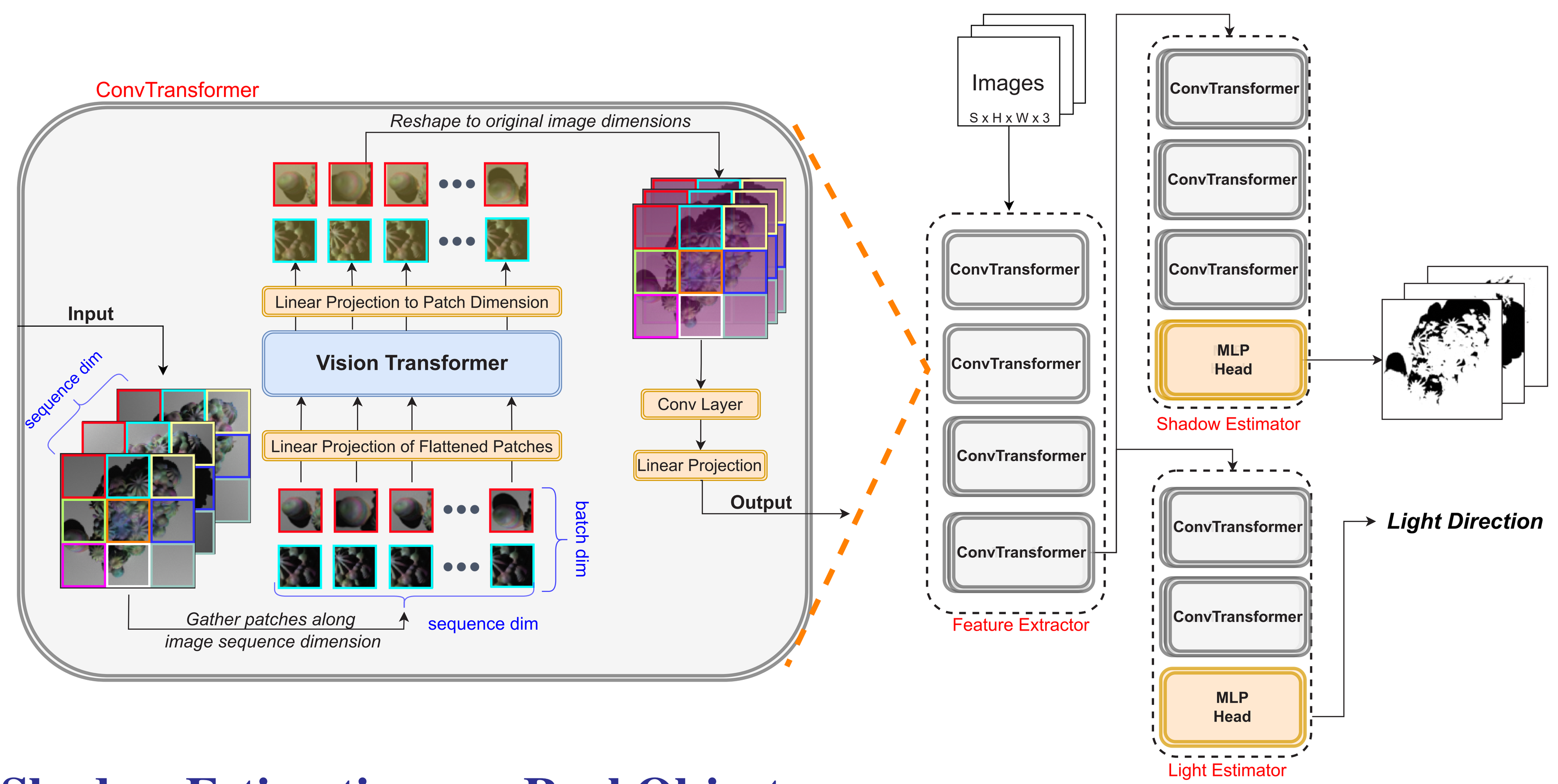
Method	Metric	Target	Cactus	Rose	Bread	Sculptures	Surface	Relief	Avg
Santo et al.	nMZE	Depth	0.96	1.16	0.77	0.81	0.75	0.75	0.87
Peng et al.	nMZE	Depth	0.43	**0.05**	0.40	0.20	0.33	0.42	0.31
Chen et al.	nMZE	Depth	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Ours	nMZE	Depth	**0.33**	0.11	**0.16**	**0.19**	**0.10**	**0.18**	**0.18**
Santo et al.	MAE	Surface Normals	32.79	50.23	33.95	54.49	22.21	21.93	35.93
Peng et al.	MAE	Surface Normals	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Chen et al.	MAE	Surface Normals	24.61	**25.12**	**18.31**	**26.43**	18.91	25.46	23.14
Ours	MAE	Surface Normals	**22.60**	26.27	20.43	27.50	**17.16**	**21.80**	**22.63**

Shadow Extraction Model

Goal: Train a network to estimate shadow maps from photometric stereo images.

Network Architecture:

- Input: Sequence of photometric stereo input images
- Output: Estimated light directions and shadow maps (also surface normals).
- Model is built from ConvTransformer blocks, which apply attention along patches of identical spatial location (while using multiple images), and uses convolution blocks on each image.



Shadow Estimations on Real Object: