Supplementary material: PhaseCam3D — Learning Phase Masks for Passive Single View Depth Estimation

Yicheng Wu[1], Vivek Boominathan[1], Huaijin Chen[1], Aswin Sankaranarayanan[2], and Ashok Veeraraghavan[1]
[1]Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005 USA
[2]Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA

I. Derivation of back-propagation in the optical layer

During the training process, the height map of the phase mask and the parameters in the U-Net are updated by the back-propagation algorithm. Given the forward model and the loss function, the gradient can be calculated by the chain rule. Although the back-propagation is done by the automatic differentiation implemented in TensorFlow [1] in our system, we would like to show the analytic form for the optical layer. This is not trivial since the derivation involves complex-valued variables and element-wise operations.

Without loss of generality, we focus on the 1D height map with a single scene depth and a single wavelength. All the coordinates and subscripts are removed. The following equations describe how the height map affects the PSF.

$$\phi^M = k\Delta n h \tag{1}$$

$$\phi = \phi^M + \phi^{DF} \tag{2}$$

$$P = A \odot \exp(i\phi) \tag{3}$$

$$\begin{aligned} PSF &= |\mathcal{F}P|^2 \\ &= (\mathcal{F}P)^* \odot (\mathcal{F}P) \end{aligned} \tag{4}$$

In the back-propagation step, we define the error from the digital network as δ, which describes how the final loss function L (defined in Eq. 8 in the main text) changes when PSF changes.

$$\delta := \frac{\partial L}{\partial PSF} \tag{5}$$

Based on the chain rule, the derivative of L with respect to each variable is shown below. Following are notations we will use. For a matrix or an operator O, O^* is the complex conjugate of O, and O^T is the transpose of O. Particularly for the Fourier operator, $(\mathcal{F}^*)^T = \mathcal{F}^{-1}$. $\odot$ means element-wise multiplication. $\text{diag}(v)$ returns a square diagonal matrix with the elements of vector v on the main diagonal. $\text{Im}(v)$ returns the imaginary part of v.

$$\frac{\partial L}{\partial \phi} = \frac{\partial L}{\partial PSF}\frac{\partial PSF}{\partial \phi} \tag{6}$$

$$\begin{aligned} \frac{\partial PSF}{\partial \phi} &= \frac{\partial PSF}{\partial (\mathcal{F}P)}\frac{\partial (\mathcal{F}P)}{\partial \phi} + \frac{\partial PSF}{\partial (\mathcal{F}P)^*}\frac{\partial (\mathcal{F}P)^*}{\partial \phi} \\ &= \text{diag}((\mathcal{F}P)^*)\mathcal{F}\text{diag}(iP) - \text{diag}(\mathcal{F}P)\mathcal{F}^*\text{diag}(iP^*) \end{aligned} \tag{7}$$

Plugging Eq. 7 into Eq. 6, we get:

$$\begin{aligned} \frac{\partial L}{\partial \phi} &= [\text{diag}((\mathcal{F}P)^*)\mathcal{F}\text{diag}(iP) - \text{diag}(\mathcal{F}P)\mathcal{F}^*\text{diag}(iP^*)]^T\delta \\ &= \text{diag}(iP)\mathcal{F}^T\text{diag}((\mathcal{F}P)^*)\delta - \text{diag}(iP^*)\mathcal{F}^{-1}\text{diag}(\mathcal{F}P)\delta \\ &= 2\,\text{Im}[\text{diag}(P^*)\mathcal{F}^{-1}\text{diag}(\mathcal{F}P)\delta] \\ &= 2\,\text{Im}[P^* \odot \mathcal{F}^{-1}((\mathcal{F}P) \odot \delta)] \end{aligned} \tag{8}$$

$$\begin{aligned} \frac{\partial L}{\partial h} &= \frac{\partial L}{\partial \phi}\frac{\partial \phi}{\partial h} \\ &= 2k\Delta n\,\text{Im}[P^* \odot \mathcal{F}^{-1}((\mathcal{F}P) \odot \delta)] \end{aligned} \tag{9}$$

This form is also correct for a 2D height map.

In our case, the 2D height map is a combination of Zernike polynomials. If we define the vectorization operator as $\mathcal{V}$, then we can represent the height map as

$$\mathcal{V}\{h\} = Za \tag{10}$$

where Z is written in a matrix form.

Then the derivative can be written as

$$\begin{aligned} \frac{\partial L}{\partial a} &= \frac{\partial L}{\partial h}\frac{\partial h}{\partial a} \\ &= Z^T\mathcal{V}\{\frac{\partial L}{\partial h}\} \\ &= Z^T\mathcal{V}\{2k\Delta n\,\text{Im}[P^* \odot \mathcal{F}^{-1}((\mathcal{F}P) \odot \delta)]\} \end{aligned} \tag{11}$$

This gradient can be used to update the learning variable a by gradient descent or Adam [2] optimizer.

II. Experimental Comparison with the Fisher mask

The Fisher mask is optimized by minimizing L_{CRLB} in our given depth range ($W_m = [-10, 10]$), which is shown in the main text Fig. 4(b). To make an experimental comparison between the proposed PhaseCam3D mask and the Fisher mask, we fabricated the Fisher mask, as well, with the same size as the proposed mask (Fig. 1a). Fig. 1(b) shows the experimental PSFs at different depths, which is used for fine-tuning the digital network.

To make a fair comparison, we captured the same scene with both masks and estimate the depth respectively. As shown in Fig. 2(a), the depth estimation with the PhaseCam3D mask is better. Besides, we quantified the estimation error by capturing planar targets at known depths. The root-mean-square (RMS) error is 0.0125 m for the PhaseCam3D mask, and 0.0308 m for the Fisher mask (Fig. 2b).

This section shows that our learning-based PhaseCam3D mask outperforms the model-based Fisher mask experimentally.

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *Symposium on Operating Systems Design and Implementation (OSDI)*, 2016.

[2] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

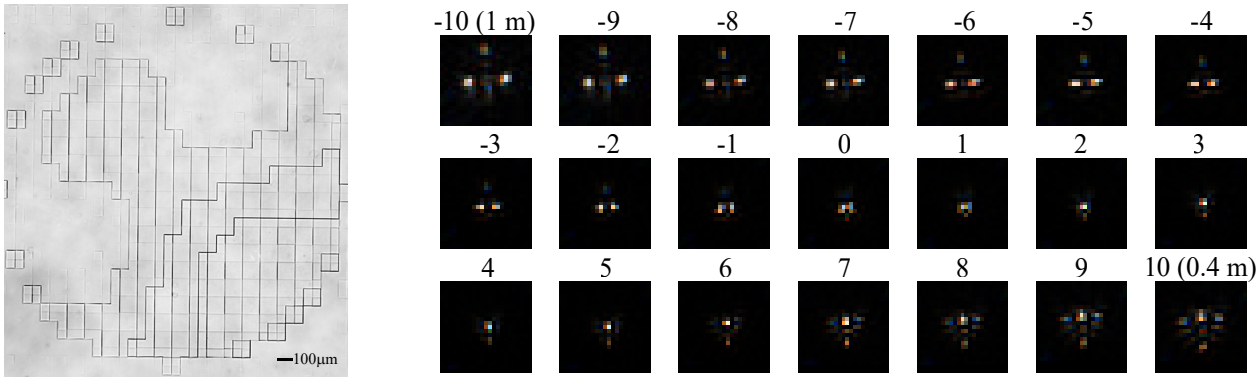

Fig. 1. **The fabricated Fisher mask and its PSFs. (a)** A close-up image of the fabricated Fisher mask taken using a 2.5× microscope objective. **(b)** Calibrated PSFs of the Fisher mask. The depth range is 0.4 m to 1 m, which corresponds to the normalized W_m range for an aperture size of 2.835 mm.

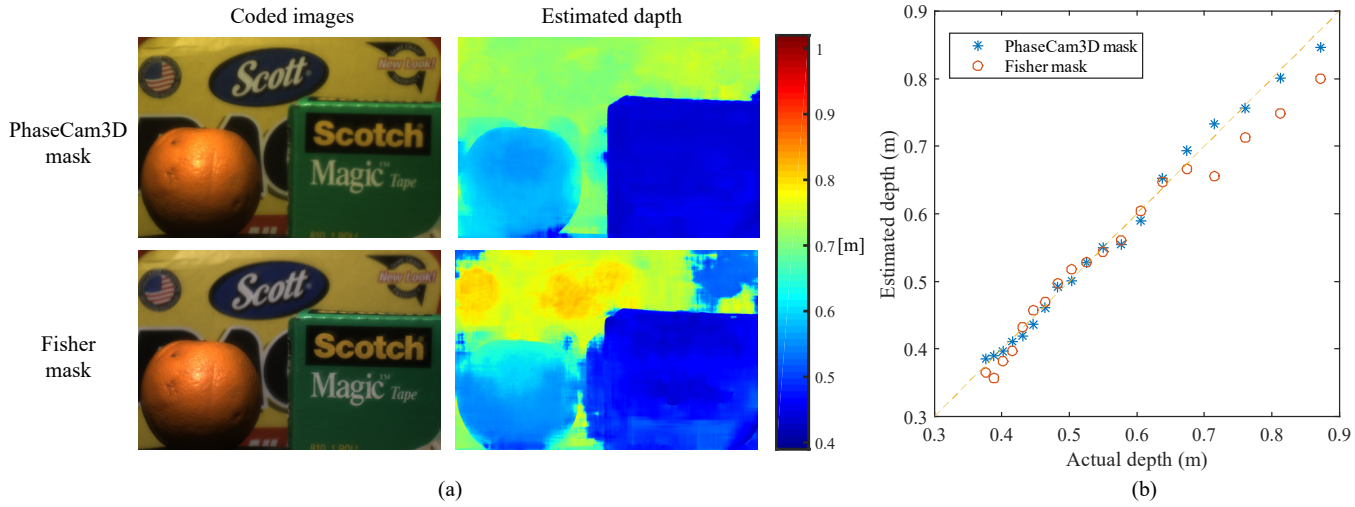

Fig. 2. **Depth estimation performance comparison between the proposed PhaseCam3D mask and the Fisher mask. (a)** A same scene is captured for depth estimation using two masks. Based on the estimated depth, the PhaseCam3D provides a better result. **(b)** A quantitative comparison is done by capturing targets at known depths. The RMS for the PhaseCam3D mask is 0.0125 m, while the RMS for the Fisher mask is 0.0308 m.