Supplementary Material:
Deep Optics for Monocular Depth Estimation and 3D Object Detection

Julie Chang
Stanford University
jchang10@stanford.edu

Gordon Wetzstein
Stanford Univsersity
gordon.wetzstein@stanford.edu

1. Optical Model Details

1.1. Implementation Details

We model three wavelengths, corresponding to the three RGB color channels: 635, 550, and 450 nm. In simulations, the optical material used for index of refraction parameters is PMMA; in our physical prototype, our lens is an N-BK7 lens. In simulations the sensor pixel size is 8 microns; in our physical protoype, the sensor pixel size is 4.29 microns.

1.2. Phase Masks for Custom PSFs

In our optical model, we separate the standard thin lens profile from the customizable phase mask profile, which can be used to add in additional aberrations to the PSF. Note that naive chromatic aberration is not modeled with an additional phase mask, but rather with the wavelength parameter itself and the wavelength-dependent index of refraction of the lens material. We parametrize the additional phase mask with a Zernike polynomial basis, Z1-Z36 [9]. Zernike polynomials have been used previously for several optical engineering applications, including vision correction in opthalmology [6], extended-depth-of-field imaging [11], and superresolution microscopy [2]. An example of a randomly generated phase mask built from Z1-Z10 and the resulting PSF stack is shown in Fig. S1.

Astigmatism In our simulations, we include the case of a fixed achromatic lens with astigmatism. By adjusting the Zernike coefficient corresponding to polynomial Z5, we can control the degree of astigmatism introduced in the resulting lens system. The surface profile of the astigmatism phase mask and corresponding PSFs are shown in Fig. S1.

Annular phase mask Inspired by Haim *et al*. [3], we also evaluate an annular phase mask in our depth estimation simulations. This mask consists of concentric rings at different optimizable heights (Fig. S1). Haim *et al*. additionally optimize the ring radii, but for our version of the annular mask, we fix the ring radii to their reported optimal normalized ring radii ($r = \{0.55, 0.8, 1\}$).

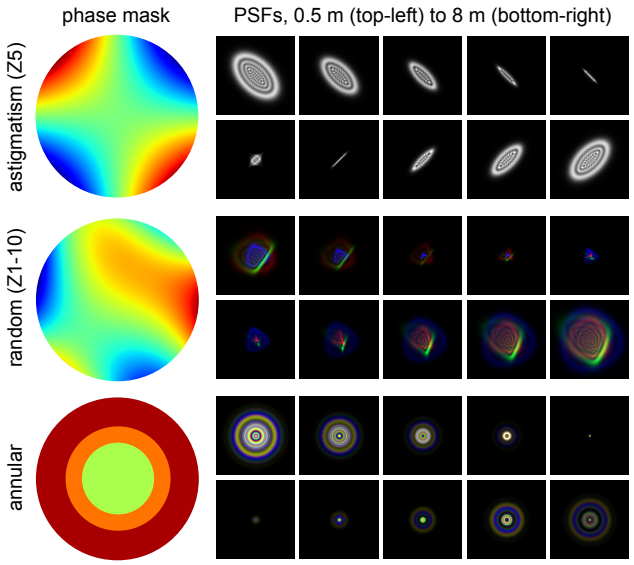

Figure S1. **Phase mask surface profiles and PSFs**. Examples shown for achromatic astigmatism, random Zernike coefficients Z1-Z10, and an annular phase mask.

1.3. Image Formation

We use a layered representation that models the scene as a set of planar surfaces at a discrete number of depth planes [4]. In the original layered model, the all-in-focus image was segmented out by each layer's depth mask and inpainted to fill occluded regions before convolution. In our simulations, instead of using extra computations to inpaint occluded regions, we simply use the full image when convolving with each PSF.

For an all-in-focus image $\mathbf{L}$, a set of $j = 1 \dots J$ discrete depth layers with pre-commputed PSFs $\{\text{PSF}_j\}$, and occlusion masks $\{\mathbf{M}_j\}$, we calculate our final image by:

$$\mathbf{I}_\lambda = \sum_{j=1}^{J} (\mathbf{L}_\lambda * \text{PSF}_{\lambda,j}) \circ \mathbf{M}_{\lambda,j} \tag{1}$$

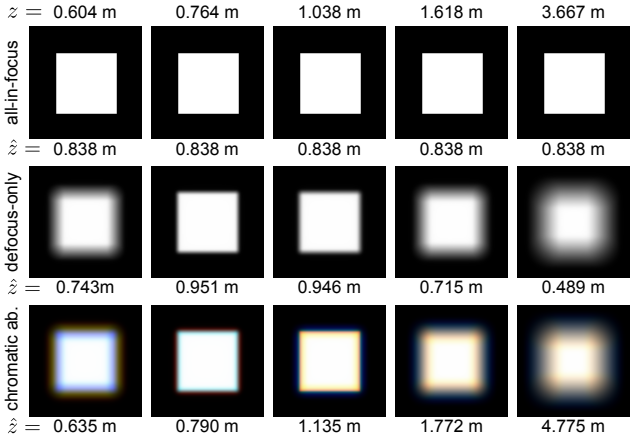

Figure S2. **Rectangles dataset**. Sensor images of white rectangles placed at different depths (z) against a black background from various optical models (all-in-focus, defocus-only, and chromatic aberration). Average predicted depths ($\hat{z}$) of the object are listed below each image. During training and testing, rectangles are randomly sized.

where $*$ denotes 2D convolution for each color channel centered on λ, and $\circ$ denotes element-wise multiplication. The cumulative occlusion masks $\{\mathbf{M}_j\}$ are alpha masks that modulate how much light from each layer is captured by the sensor. The masks are generated with the blurred binary depth masks, $\{\mathbf{A}_k\}$, from current and preceding layers:

$$\mathbf{M'}_j = (1 - \mathbf{M}_{j+1})(\mathbf{A}_j * \mathrm{PSF}_j), \quad j < J \tag{2}$$

$$\mathbf{M'}_J = \mathbf{A}_J * \mathrm{PSF}_J \tag{3}$$

Here, layer J is the layer closest to the camera that is not occluded by any additional layers, so $\mathbf{M}'_J$ simply consists of the regions of the depth map that fall into this layer, $\mathbf{A}_J$, blurred by PSF_J. Each layer behind J is occluded by the layers in front of it. Finally, $\{\mathbf{M}'_j\}$ are normalized to $\{\mathbf{M}_j\}$ such that the sum of occlusion mask weights at each pixel location sums to 1.

For our experiments, we used $J = 12$, and depth intervals were spaced evenly in inverse depth. The depth ranges for each dataset were: Rectangles, 0.5 m to 10 m; NYU Depth v2, 0.5 m to 8 m; KITTI, 5 m to 80 m.

2. Depth Estimation

2.1. Rectangles Dataset

In addition to two standard datasets, we created a custom dataset consisting of randomly shaped white rectangles placed at random depths against a black background. For this dataset, depth estimation error was only considered in the object region so that the constant-depth background would not skew the results.

The Rectangles dataset represents an extreme situation where pictorial cues are insufficient for depth estimation.

Figure S3. **Optical prototype**. (Left) Our optical prototype, which consists of a Canon camera and a Thorlabs singlet lens. (Right) Images of the lens and aperture mask for chromatic aberration images, followed by the pinhole mask for all-in-focus images.

From Fig. S2, the all-in-focus images of this dataset make it intuitive that depth cannot be predicted from a single RGB image without additional information, and the constant mean-depth-valued predicted depths support this.

With defocus blur, there is some encoding of depth information into the sensor images, but there is still some ambiguity whether an object is in front of or behind the focal plane (1 m in this example). With the addition of chromatic aberrations, this ambiguity is resolved, and the predicted depths match the grount truth depths much more closely. The remaining error stems from the limited number of depth intervals used during our simulations.

2.2. Prototype PSF Calibration

For our real-world experiments, it was first necessary to calibrate the PSFs from the optical prototype (Fig. S3) with the PSFs from our simulated model. We captured a series of images with the camera placed at increasing distances from a point white light source (Thorlabs MWWHL3) and compared these with the simulated PSFs at the same distances. Fig. S4 shows the effects of adding in each calibration step of matching focus distance, downsampling, and spherical aberration. Primary spherical aberration was tuned by adjusting Zernike polynomial Z_0^4, or equivalently the 11th in the Noll index [7, 9]. Some differences remain since we only model three wavelengths in the simulation whereas the light source consists of a continuous spectrum of wavelengths. More complex modeling of the spatially-varying nature of the PSF could further improve accuracy and would be valuable to explore in future work. The final calibrated PSFs were then used to retrain the depth estimation network on the NYU Depth v2 dataset.

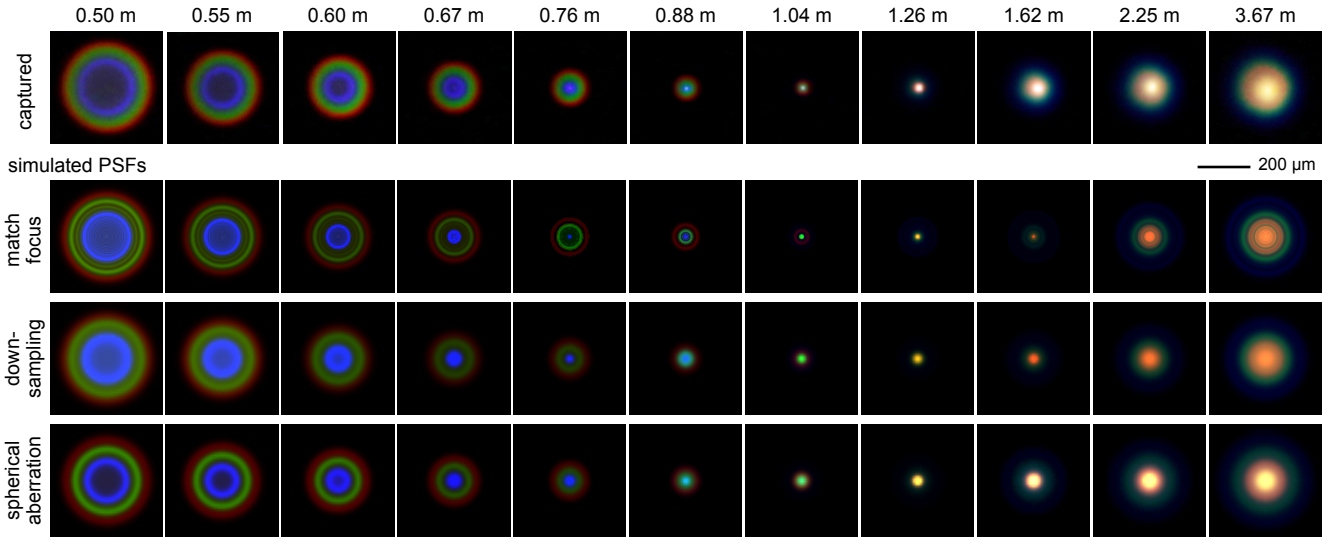

Figure S4. **PSF calibration.** Captured PSFs and simulated PSFs with calibration adjustments, over a range of depths. Each consecutive line of simulated PSFs includes the calibration adjustments of those above it. Scale bar applies to all PSFs and refers to size on sensor.

Model	RMSE_{lin}	RMSE_{log}	δ_1
All-in-focus	2.9100	0.1083	0.9444
Optimized	1.9288	0.0621	0.9864
Optimized (DORN)	2.5999	0.0923	0.9630

Table S1. Depth estimation performance on our validation set for the all-in-focus model, optimized mask model, and optimized mask model with DORN depth. Root-mean-square-error (RMSE) is reported for linear and log scaling of depth (m or log(m)). Thresholds δ_1 are calculated as in [1].

2.3. Extended Results

In the Supplemental Materials, we include a sample video of a KITTI drive sequence with our predicted depth maps using the optimized lens. Due to file size constraints, the full set of comparison videos can be found online (`http://www.computationalimaging.org/publications/deep-optics-depth/`). These sequences consist of images from both the training and validation sets. The depth colormaps are scaled as in Fig. 4 of the main paper.

We show extended real-world results that also include the all-in-focus captured images in Fig. S5.

3. 3D Object Detection

3.1. 2D Object Detection

Since our 3D object detection network (adapted from FPointNet, [10]) relies on the outputs of a 2D object detection network during region proposal, we separately train 2D object detection networks on the KITTI dataset for use during 3D object detection. We download a Faster R-CNN model (with Inception Resnet v2, Atrous version [5]), pretrained on the MS-COCO dataset, and fine-tune it for the KITTI categories. We train one network using the original all-in-focus images from our training split of the object detection dataset, and another network using the corresponding blurred images from the lens optimized for depth estimation. We train for 100,925 iterations with a batch size of 1, using the Momentum optimizer at a learning rate of 3e-4 that decays to 3e-5 after 500,000 iterations.

3.2. 3D Object Detection

To use FPointNet with our RGB and predicted depth inputs, we project the dense predicted depth maps into a 3D point cloud using camera calibration information from the KITTI dataset. Since the top section of the predicted depth maps tends to be unreliable, we ignore the first 100 rows of the predicted depth map when creating the point cloud. We also ignore any negative predicted depth values. Once depth information is in the point cloud format, the original FPointNet implementation can be used.

Again, we train one model using the all-in-focus RGB images and predicted depths, and another using the blurred RGB images from the lens optimized for depth estimation and their predicted depths. We train for 101 epochs with a batch size of 32, using the ADAM optimizer and a decay rate of 0.5 every 800,000 steps. We set the number of points sampled in each frustrum to be 1024. During training, we use ground truth 2D bounding boxes with perturbation. During validation and testing, we use 2D bounding boxes from our retrained 2D object detection networks. Results would likely be improved with more dedicated attention to the architecture and hyperparameters of these detection networks.

Method	3D object localization			3D object detection		
	Easy	Moderate	Hard	Easy	Moderate	Hard
All-in-focus (val)	26.71	19.87	19.11	16.86	13.82	13.26
Optimized (val)	**37.51**	**25.83**	**21.05**	**25.20**	**17.07**	**13.43**
All-in-focus (test)	**13.78**	4.73	4.36	**10.58**	2.34	1.82
Optimized, DORN (test)	9.99	**5.43**	**5.05**	5.40	**3.14**	**2.70**

Table S2. 3D object localization AP % (bird's eye view) and 3D object detection AP % (IoU= 0.7) for the car class. The table compares the performance of our validation split, for which we were able to use a more accurate dense depth map, and the KITTI test set, for which we used dense depth maps generated by DORN. The higher AP values for each dataset comparison are bolded.

A sample video with output 3D bounding boxes from a KITTI drive sequence is included in the Supplementary Materials. The top panel shows bounding box predictions for the all-in-focus model, and the bottom panel shows bounding box predictions for the optimized lens model.

3.3. Assessment on KITTI Test Set

In the main paper, we reported results on our validation split of the KITTI object detection dataset. In our image formation model, we require dense depth maps to accurately simulate sensor images with optical blur. We were able to obtain more accurate dense depth maps for the subset of the official object detection training set that overlapped with the depth estimation dataset, since these images had more ground truth depth points to facilitate sparse to dense depth completion [8].

However, for the official object detection testing dataset, we did not have this ground truth information (only sparse LIDAR) and instead used DORN, a leading monocular depth estimation network, to generate the pseudo-truth depth maps we input into our model. We retrain the depth estimation network with the optimized lens using the DORN depth maps to simulate sensor images. As listed in Table S1, depth estimation performance decreases due to the inaccuracy in the sensor images, but errors are still lower than the all-in-focus model (recall, the all-in-focus model does not require dense depth maps since the original dataset image is used).

We also retrain the FPointNet model using these outputs. The same 2D object detection model is used to predict 2D bounding boxes to feed into the 3D network for testing. In Table S2, we report the 3D object detection results on the test set using the DORN-based dense depth maps. The depth estimation errors carry through to the object detection task, resulting in less accuracy in the predicted bounding boxes. The results show that the average precision (AP) in bounding box estimation for the easy, or least occluded, division is lower for the optimized model than the all-in-focus model, whereas the AP for the moderate and hard divisions is higher for the optimized model than the all-in-focus model. We include these results in the supplement for thoroughness, though they do not fully reflect the capabilities of the optimized element. With more accurate dense depth maps for the test dataset, more accurate 3D bounding box predictions would likely also be obtained.

References

[1] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[2] Travis J Gould, Daniel Burke, Joerg Bewersdorf, and Martin J Booth. Adaptive optics enables 3d sted microscopy in aberrating specimens. *Optics express*, 20(19):20998–21009, 2012.

[3] Harel Haim, Shay Elmalem, Raja Giryes, Alex M Bronstein, and Emanuel Marom. Depth estimation from a single image using deep learned phase coded mask. *IEEE Transactions on Computational Imaging*, 4(3):298–310, 2018.

[4] Samuel W Hasinoff and Kiriakos N Kutulakos. A layer-based restoration framework for variable-aperture photography. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[5] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

[6] D Robert Iskander, Michael J Collins, and Brett Davis. Optimal modeling of corneal surfaces with zernike polynomials. *IEEE Transactions on biomedical engineering*, 48(1):87–95, 2001.

[7] Virendra N Mahajan. Zernike circle polynomials and optical aberrations of systems with circular pupils. *Applied optics*, 33(34):8121–8124, 1994.

[8] Fangchang Mal and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[9] Robert J Noll. Zernike polynomials and atmospheric turbulence. *JOsA*, 66(3):207–211, 1976.

[10] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-

d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[11] Vincent Sitzmann, Steven Diamond, Yifan Peng, Xiong Dun, Stephen Boyd, Wolfgang Heidrich, Felix Heide, and Gordon Wetzstein. End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging. *ACM Transactions on Graphics (TOG)*, 37(4):114, 2018.

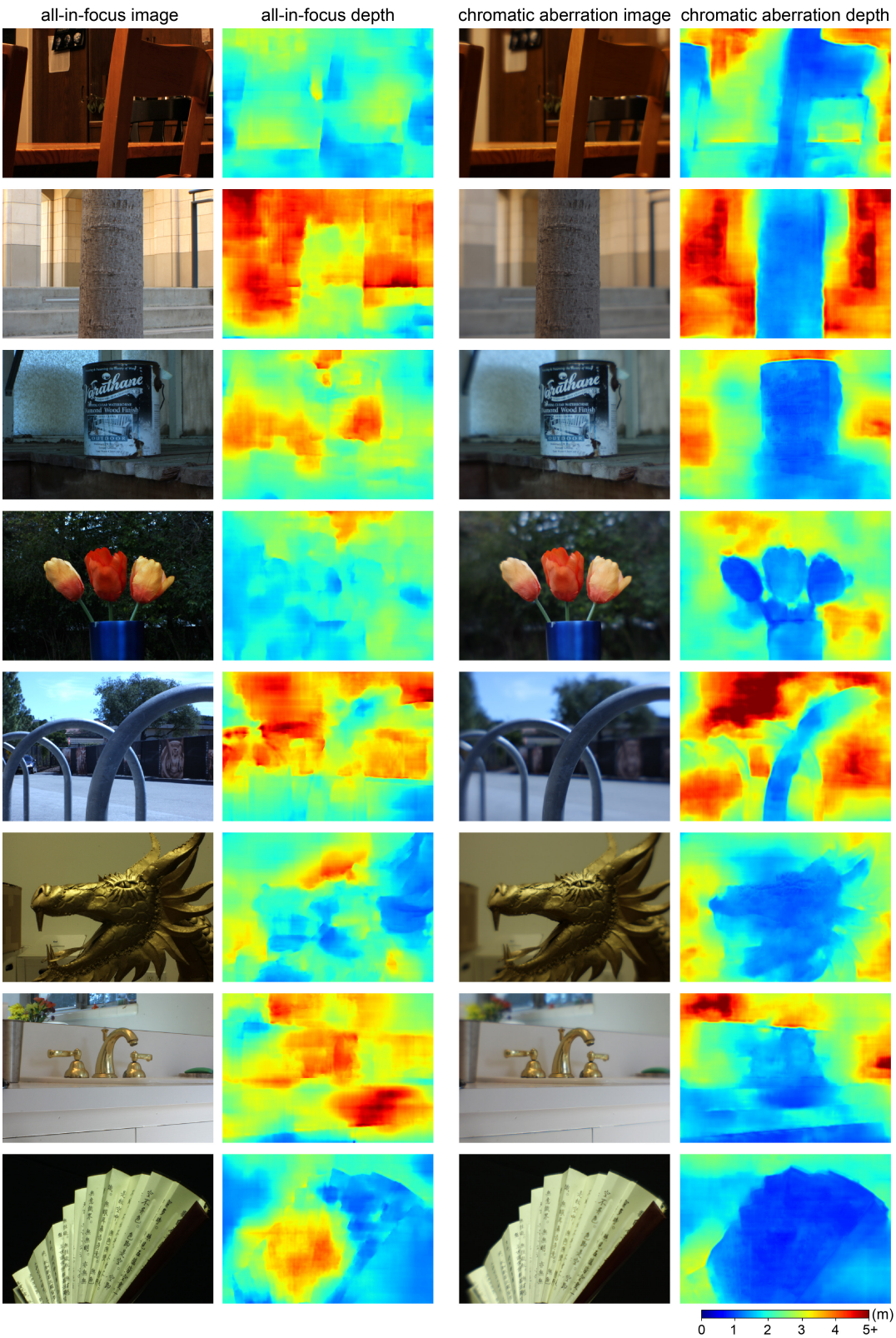

Figure S5. **Extended real-world results.** Examples of all-in-focus and chromatic aberration captured images and the predicted depth maps from each. All-in-focus images are obtained by inserting a pinhole in front of the imaging lens while the camera is kept in the same position. We encourage the electronic viewer to zoom-in for comparison of captured images. Colormap for is the same for all depth maps.