FreeCam3D: Snapshot Structured Light 3D with Freely-Moving Cameras Supplementary Document

Yicheng Wu[1], Vivek Boominathan[1], Xuan Zhao[1], Jacob T. Robinson[1], Hiroshi Kawasaki[2], Aswin Sankaranarayanan[3], and Ashok Veeraraghavan[1]

[1] Rice University, Houston TX, USA
{yicheng.wu, vivekb, xz61, jtrobinson, vashok}@rice.edu
[2] Kyushu University, Fukuoka, Japan
kawasaki@ait.kyushu-u.ac.jp
[3] Carnegie Mellon University, Pittsburgh PA, USA
saswin@andrew.cmu.edu

1 Derivation of the PSFs

In this section, we derive how the PSF of the projected pattern depends on the phase mask height h and the scene depth z based on Fourier optics theory [1].

The mask is inserted on the pupil plane of the projection lens. The pupil function can be represented as a complex-valued 2D matrix P.

$$P = A\exp(i\phi) \tag{1}$$

The amplitude part A is a constant disk function. The phase part ϕ is consist of two components.

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

ϕ^M is introduced by the phase mask. It depends on the height map of the phase mask h.

$$\phi^M = k\Delta n h \tag{3}$$

where k is the wave vector $k = 2\pi/\lambda$ and Δn is the reflective index difference between air and the material of the phase mask.

ϕ^{DF} is introduced by defocus, which is a quadratic phase related to the in-focus depth z_0 and the actual depth z of a scene point.

$$\phi^{DF} = k\frac{x_1^2 + y_1^2}{2}\left(\frac{1}{z} - \frac{1}{z_0}\right) \tag{4}$$

where (x_1, y_1) are the coordinates in the pupil plane.

For an incoherent system, the PSF is the squared magnitude of the Fourier transform of the pupil function.

$$PSF(h, z) = \left|\mathcal{F}\{P(h, z)\}\right|^2 \tag{5}$$

The PSF is a function of the mask height map h and the depth of the scene z. We do not include the PSF dependence on the wavelength because a narrow-band light source is used here.

2 Networks detail

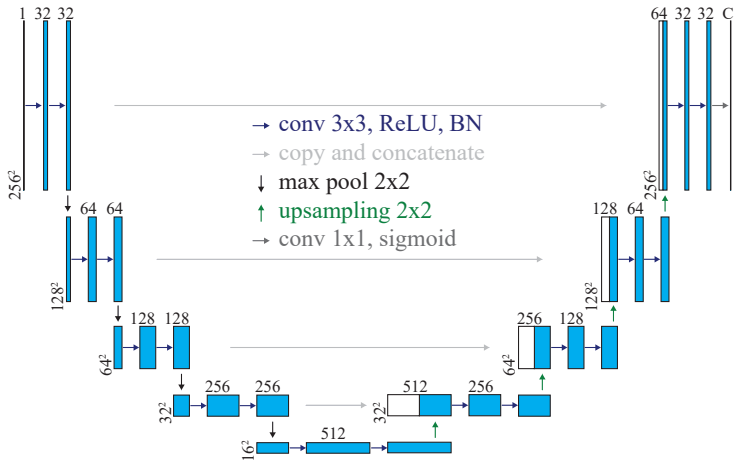

Fig. S1. Network architecture. Both the XYNet and ZNet have this encoder-decoder architecture, with only the difference that C = 2 for XYNet and C = 1 for ZNet.

For the reconstruction networks (Sec. 4.2 in the main text), we use XYNet for x, y estimation, and ZNet for z estimation. We found that two separate networks provide the best result. One possible reason is that XYNet should learn from the global context, while ZNet should focus on the local pattern. We tried various networks with different receptive fields. But these networks show similar performance.

In our paper, we have almost the same encoder-decoder architecture (Fig. S1) for XYNet and ZNet. The input image is I_c^{LN} $(256 \times 256 \times 1)$. Each convolution operator consists of a 3×3 convolution, a rectified linear unit (ReLU) and a batch normalization (BN) [2]. The downsampling is achieved using a 2×2 max-pooling operation, and the upsampling is achieved using resize-convolution. Concatenation is applied between the encoder and decoder to avoid the vanishing gradient problem. At the final layer, a 1×1 convolution is used with a sigmoid function to map each pixel to the given range. The only difference between XYNet and ZNet is that the number of the output channels C is 2 for XYNet and 1 for ZNet.

3 Derivation of image reprojection

As mentioned in Sec. 4.1, we include the reprojection loss to enforce the network to learn the depth from perspective distortion. To calculate the reprojected image $\hat{I}_c$, we use a similar formula as mentioned in Eq. 5 of the main text.

For the image warping part, we have I_p^B and T_{cp} from the ground truth and calculate $\hat{z}_c$ from $\hat{z}_p^{cView}$. Both $\hat{z}_c$ and $\hat{z}_p^{cView}$ are in the camera view. But $\hat{z}_c$ is the depth to the camera, while $\hat{z}_p^{cView}$ is the depth to the projector. Based on the pixel location and the camera intrinsic parameters, it is straightforward to represent the 3D location of each pixel in the camera view as $(x_c z_c, y_c z_c, z_c)^T$ (x_c and y_c are in the projected coordinates). The relationship between $\hat{z}_c$ and $\hat{z}_p^{cView}$ are as follows.

$$z_p^{cView} = T_{cp}^{(3)} \begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \begin{pmatrix} t_{31} & t_{32} & t_{33} & t_{34} \end{pmatrix} \begin{pmatrix} x_c z_c \\ y_c z_c \\ z_c \\ 1 \end{pmatrix} \tag{6}$$

Only the third row of the transformation $T_{cp}^{(3)}$ is required for z. So we can find $\hat{z}_c$ from $\hat{z}_p^{cView}$ directly.

$$\hat{z}_c = \frac{\hat{z}_p^{cView} - t_{34}}{t_{31} x_c + t_{32} y_c + t_{33}} \tag{7}$$

For the occlusion mask, we directly pick the region with a threshold of I_c to reduce the computational complexity.

Finally, the formula of the reprojected image is

$$\hat{I}_c = \mathcal{W}^I(I_p^B, \hat{z}_c, T_{cp}) \cdot (I_c > \epsilon) \tag{8}$$

4 Simulation results analysis

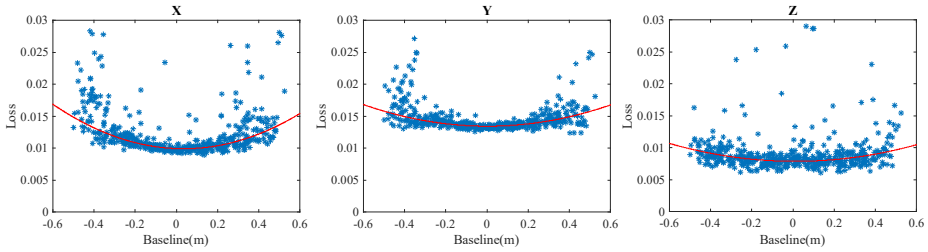

Fig. S2. Relation between the camera-projector baseline and the estimation losses $L_{rms}^x, L_{rms}^y, L_{rms}^z$.

Our proposed method is able to recover the 3D scene from a freeform camera. Here, we investigate how the camera-projector baseline affects the estimation accuracy. 500 testing scenes are generated using our simulation pipeline with different baselines along the x-axis. For each scene, the losses $L_{rms}^x, L_{rms}^y, L_{rms}^z$ are

calculated. The scatter plots are shown in Fig. S2. For x estimation, it gives the best accuracy when there is no baseline. As the baseline increases, the average and the variation of the loss increase. This is mainly because that the perspective distortion for large-baseline cases makes the global pattern estimation more challenging. Similar phenomenon exists fo y estimation. But since there is no baseline in the y axis, the accuracy is less affected by the distortion. For z estimation, the loss is almost independent on the baseline. We think the reason is that the ZNet mainly estimates the depth from the defocused blur, which is not related to the baseline. For all three losses, there are outliers corresponding to challenging scenes, which are not included in the robust fitting.

5 Mask fabrication and calibration

5.1 Mask fabrication

We fabricate the designed phase mask using a multi-layer photolithography technique on a fused-silica glass substrate. Briefly, the fabrication involves three steps, repeated multiple times. Step 1 is to spin coat photoresist onto the substrate. Step 2 is transferring pattern from a photomask to the photoresist through exposure to the UV light and then removing the UV-exposed photoresist. Step 3 is etching the exposed glass substrate using reactive ion etching (RIE) process. Each round of fabrication results in a binary profile change on the substrate. Hence, by repeating the process N-times, a 2^N-level phase mask can be achieved [3].

Our mask has 15 height levels of 73 nm each, with a total height of 1095 nm. The fabrication is achieved by repeating the steps mentioned above 4 times using four different photomasks. Our substrate was a 0.5mm-thick 4 inch Fused Silica wafer, and we used MICROPOSIT S1818 photoresist. After fabrication, the phase mask was cut out from the wafer to a manageable size.

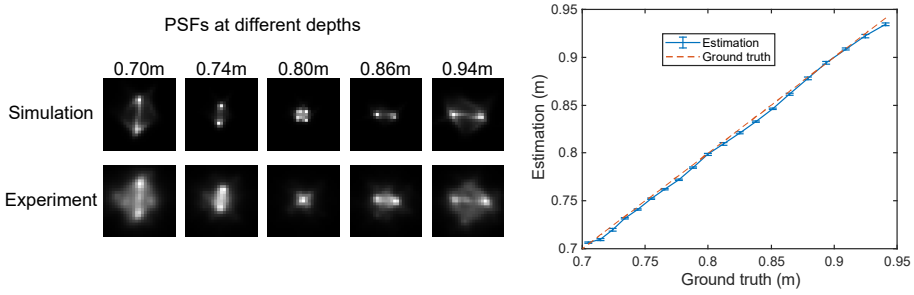

Fig. S3. (Left) PSFs comparison between simulation and experiment. (Right) Experimental depth accuracy evaluation.

5.2 PSFs calibration

Although the PSF at different depth is calculated from the simulation as shown in Fig. 4 of the main text, we calibrate our prototype experimentally to account for any mismatch born out of physical implementation such as aberrations in fabricated phase mask and misalignment during the mask installation.

We used sparse grid dots as the projector pattern, and placed the camera to be co-located with the projector with a beam splitter so that there is no perspective distortion. The scene is a planar whiteboard perpendicular to the optical axis. During the calibration, the whiteboard was placed at 21 depths ranging from 0.7m to 0.95m, and the corresponding image was captured at each depth. For each image, we averaged the local pattern generated from grid dots as the PSF at that depth.

The comparison between the simulated PSFs and calibrated PSFs are shown on the left side of Fig. S3. As we can see, they follow similar structures. But the Experimental PSFs look more blurry due to imperfection in the experiment. We believe a better fabrication process can further improve our experiment performance.

5.3 Depth accuracy evaluation

To make sure the experimental setup working properly, we evaluate the depth estimation accuracy. We measure the depth reconstruction of a whiteboard at different depths ranging from 0.7m - 0.95m. The camera and projector is co-located using a beam splitter. We calculate the average and stand deviation for each depth and plot the results in Fig. S3. Our estimation matches the ground nicely with small errors. The estimation is slightly worse near the depth limit, which might be caused by a lack of training data. The average root mean square error for the whole depth range is 3.7 mm.

6 Analysis for textured scenes

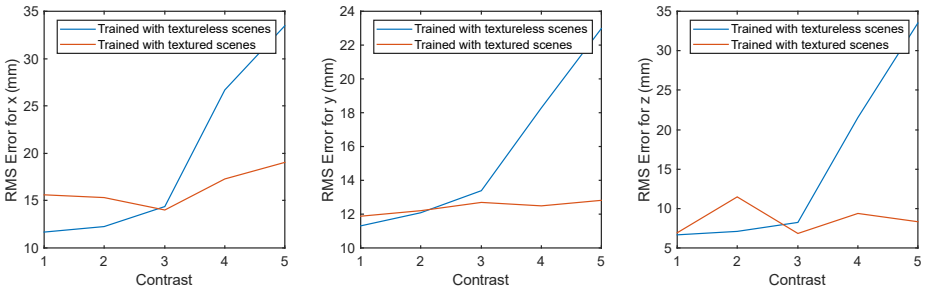

Fig. S4. Simulation evaluation on textured scenes

As mentioned in the main text, intensity variation caused by the texture of the scene might affect the reconstruction accuracy. Nevertheless, our method is able to work for textured scene, as shown in Fig. 4 of the main text.

To evaluate how the reconstruction accuracy depends on texture, we test on data with checkerboard pattern as the texture. The contrast is defined by the ratio between the bright and dark region of the checkerboard. Fig. S4 shows the rms error for x, y, z. As we can see, even for the algorithm trained with textureless scenes, it still works well for small contrast. The main reason is the local normalization that we applied on the image as the pre-processing step, to reduce the intensity variation. And the results can be further improved when the algorithm is trained with textured scenes.

References

1. Goodman, J.W.: Introduction to Fourier optics. Roberts and Company Publishers (2005)
2. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167 (2015)
3. Peng, Y., Fu, Q., Heide, F., Heidrich, W.: The diffractive achromat full spectrum computational imaging with diffractive optics. In: SIGGRAPH ASIA 2016 Virtual Reality meets Physical Reality: Modelling and Simulating Virtual Humans and Environments, pp. 1–2 (2016)