Supplementary material – CodedStereo: Learned Phase Masks for Large Depth-of-field Stereo

Shiyu Tan[1,*] Yicheng Wu[1,*] Shoou-I Yu[2] Ashok Veeraraghavan[1,†]
[1]Rice University [2]Facebook Reality Labs
{shytan, yicheng.wu, vashok}@rice.edu Shoou-I.Yu@fb.com

We organize our supplementary material into five appendix sections, consisting of: a) *PSFs simulation*: derivation of the disparity-dependent PSFs; b) *Reconstruction networks*: architecture details of the RGB and depth reconstruction networks; c) *Mask fabrication*: details of mask fabrication using Nanoscribe; d) *System calibration*: stereo rectification and PSFs calibration procedures; e) *More comparison results*.

Appendix A. PSFs simulation

In this section, we derive our disparity-dependent PSF based on the Fourier Optics theory [4], according to which the PSF is computed as the squared Fourier transform of the pupil function. The pupil function depends on the mask pattern inserted in the aperture, and can be represented as a complex-valued function.

$$P(x_1, y_1) = A(x_1, y_1) \exp i(\phi^M(x_1, y_1) + \phi^{DF}(x_1, y_1)). \quad (1)$$

where (x_1, y_1) is the spatial coordinate on the mask plane, ans $A(x_1, y_1)$ is a circular amplitude function with respect to aperture radius R

$$A(x_1, y_1) = \begin{cases} 1, & x_1^2 + y_1^2 \leq R^2 \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

$\phi^M(x_1, y_1)$ denotes the phase shift induced by the phase mask

$$\phi^M(x_1, y_1) = k\Delta n h(x_1, y_1). \quad (3)$$

where $k = 2\pi/\lambda$ is the wave vector, $h(x_1, y_1)$ is the phase mask height map, and Δn is the reflective index difference between the air and the mask material. $\phi^{DF}(x_1, y_1)$ denotes the quadratic defocus phase, which is related to the in-focus depth z_0 and the actual depth z of a scene point.

$$\phi^{DF}(x_1, y_1) = \frac{k}{2}(\frac{1}{z} - \frac{1}{z_0})(x_1^2 + y_1^2). \quad (4)$$

Given the inverse relation between depth z and binocular disparity d, i.e. $d = bf/z$, the defocus phase can be further derived as

$$\phi^{DF}(x_1, y_1) = \frac{k}{2fb}(d - d_0)(x_1^2 + y_1^2). \quad (5)$$

where f is the focal length, b is the baseline between the left/right cameras, and d_0 is the corresponding disparity value at the in-focus depth z_0. Thus, our disparity-dependent PSF can be computed as

$$PSF(x_1, y_1) \propto |\mathcal{F}\{P(x_1, y_1)\}|^2. \quad (6)$$

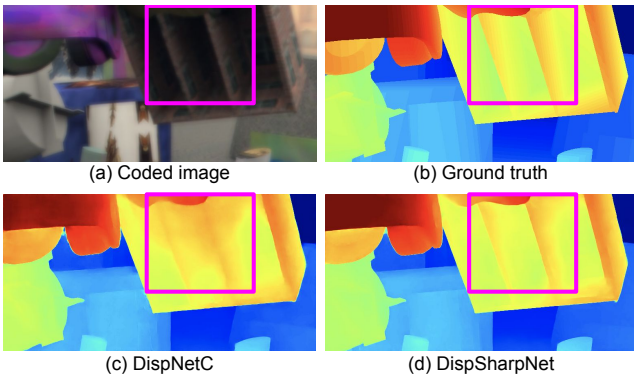

Figure 1. **Disparity prediction using DispSharpNet with an extra encoder-decoder module.** (a) Input coded images. (b) Disparity ground truth. (c) Prediction using DispNetC [5] without the encoder-decoder module. (d) Prediction using DispSharpNet with the encoder-decoder module. DispSharpNet encourages the disparity estimation with clearer details and sharper boundaries.

Appendix B. Reconstruction networks

We use a U-Net and a DispSharpNet to reconstruct the disparity map and sharp texture images, respectively. The detailed architectures of the networks are shown in Table 1 and 3.

The DispSharpNet adopts the structure of DispNetC [5], which is computationally efficient both in terms of memory and runtime providing the potential to achieve real-time

[1]These two authors contributed equally. [†]Corresponding author.

Name	Layer description	Output dimension	Input
down1_1	3×3, 32 conv	H×W×32	input image
down1_2	3×3, 32 conv	H×W×32	down1_1
down2_0	2×2, max_pool	1/2H×1/2W×32	down1_2
down2_1	3×3, 64 conv	1/2H×1/2W×64	down2_0
down2_2	3×3, 64 conv	1/2H×1/2W×64	down2_1
down3_0	2×2, max_pool	1/4H×1/4W×64	down2_2
down3_1	3×3, 128 conv	1/4H×1/4W×128	down3_0
down3_2	3×3, 128 conv	1/4H×1/4W×128	down3_1
down4_0	2×2, max_pool	1/8H×1/8W×128	down3_2
down4_1	3×3, 256 conv	1/8H×1/8W×256	down4_0
down4_2	3×3, 256 conv	1/8H×1/8W×256	down4_1
down5_0	2×2, max_pool	1/16H×1/16W×256	down4_2
down5_1	3×3, 512 conv	1/16H×1/16W×512	down5_0
down5_2	3×3, 512 conv	1/16H×1/16W×512	down5_1
up4_0	2×2, upsamp	1/8H×1/8W×256	down5_2
up4_1	3×3, 256 conv	1/8H×1/8W×256	[up4_0,down4_2]
up4_2	3×3, 256 conv	1/8H×1/8W×256	up4_1
up3_0	2×2, upsamp	1/4H×1/4W×128	up4_2
up3_1	3×3, 128 conv	1/4H×1/4W×128	[up3_0,down3_2]
up3_2	3×3, 128 conv	1/4H×1/4W×128	up3_1
up2_0	2×2, upsamp	1/2H×1/2W×64	up3_2
up2_1	3×3, 64 conv	1/2H×1/2W×64	[up2_0,down2_2]
up2_2	3×3, 64 conv	1/2H×1/2W×64	up2_1
up1_0	2×2, upsamp	H×W×32	up2_2
up1_1	3×3, 32 conv	H×W×32	[up1_0,down1_2]
up1_2	3×3, 32 conv	H×W×32	up1_1
up1_3	1×1, 3 conv	H×W×3	up1_2
output	tanh(up1_3)+input	H×W×3	up1_3

Table 1. **Architecture of U-Net for RGB image reconstrucion.** The residual image, tanh(up1_3), is learned to encourage high-frequency information recovery.

		F8 lens		CodedStereo	
	Time	EPE	3px(%)	EPE	3px(%)
DispNetC[5]	60ms	1.921	10.44%	1.738	9.18%
DispSharpNet	111ms	1.815	9.79%	1.512	7.85%
DeepPruner[3]	182ms	1.649	8.07%	1.494	6.69%
PSMNet[1]	410ms	1.613	8.17%	1.488	6.86%

Table 2. **Comparison between conventional stereo and our CodedStereo.** Independent of the network architecture used for stereo reconstruction, CodedStereo results in significant performance improvements. Shown above are performance comparisons using four different base network architectures.

inference. DispNetC makes use of an explicit 1D correlation layer that can provide sharper edge estimation and smoother area filling. We modify it by adding extra deconvolution layers to predict full-resolution disparity maps. We also apply an encoder-decoder module in the feature extraction step to encourage disparity estimations with clearer details and sharper boundaries. Comparisons of disparity prediction using DispNetC and DispSharpNet are shown in Figure 1. We also compare our CodedStereo with the conventional stereo on various stereo reconstruction networks. As shown in Table 2, CodedStereo results in significant performance improvements on all the network architectures we tested.

Appendix C. Mask fabrication

We fabricated our mask using two-photon lithography (Photonic Professional GT Nanoscribe 3D printer). The mask was fabricated using IP-Dip resist with a 63× immersion objective (Drill mode) on a 170μm-thick fused silica

Name	Layer description	Output dimension	Input
input		H×W×3	
Feature Extraction			
down1_1	3×3, 32 conv	H×W×32	input
down1_2	3×3, 32 conv	H×W×32	down1_1
down2_0	2×2, max_pool	1/2H×1/2W×32	down1_2
down2_1	3×3, 64 conv	1/2H×1/2W×64	down2_0
down2_2	3×3, 64 conv	1/2H×1/2W×64	down2_1
down3_0	2×2, max_pool	1/4H×1/4W×64	down2_2
down3_1	3×3, 128 conv	1/4H×1/4W×128	down3_0
down3_2	3×3, 128 conv	1/4H×1/4W×128	down3_1
down4_0	2×2, max_pool	1/8H×1/8W×128	down3_2
down4_1	3×3, 256 conv	1/8H×1/8W×256	down4_0
down4_2	3×3, 256 conv	1/8H×1/8W×256	down4_1
down5_0	2×2, max_pool	1/16H×1/16W×256	down4_2
down5_1	3×3, 512 conv	1/16H×1/16W×512	down5_0
down5_2	3×3, 512 conv	1/16H×1/16W×512	down5_1
up4_0	2×2, upsamp	1/8H×1/8W×256	down5_2
up4_1	3×3, 256 conv	1/8H×1/8W×256	[up4_0,down4_2]
up4_2	3×3, 256 conv	1/8H×1/8W×256	up4_1
up3_0	2×2, upsamp	1/4H×1/4W×128	up4_2
up3_1	3×3, 128 conv	1/4H×1/4W×128	[up3_0,down3_2]
up3_2	3×3, 128 conv	1/4H×1/4W×128	up3_1
up2_0	2×2, upsamp	1/2H×1/2W×64	up3_2
up2_1	3×3, 64 conv	1/2H×1/2W×64	[up2_0,down2_2]
up2_2	3×3, 64 conv	1/2H×1/2W×64	up2_1
up1_0	2×2, upsamp	H×W×32	up2_2
up1_1	3×3, 32 conv	H×W×32	[up1_0,down1_2]
up1_2	3×3, 32 conv	H×W×32	up1_1
up1_3	1×1, 3 conv	H×W×3	up1_2
conv1	7×7, 64 conv, str2	1/2H×1/2W×64	tanh(up1_3)+input
conv2	5×5, 128 conv, str2	1/4H×1/4W×128	conv1
conv_rdi	1×1, 32 conv	1/4H×1/4W×32	conv2
1D Correlation Layer			
corr_lr	left/shifted right	1/4H×1/4W×48	[conv2_l,conv2_r]
Disparity Regression			
conv3_1	5×5, 256 conv, str2	1/8H×1/8W×256	[corr_lr,conv_rdi]
conv3_2	3×3, 256 conv	1/8H×1/8W×256	conv3_1
conv4_1	3×3, 512 conv, str2	1/16H×1/16W×512	conv3_2
conv4_2	3×3, 512 conv	1/16H×1/16W×512	conv4_1
conv5_1	3×3, 512 conv, str2	1/32H×1/32W×512	conv4_2
conv5_2	3×3, 512 conv	1/32H×1/32W×512	conv5_1
conv6_1	3×3, 1024 conv, str2	1/64H×1/64W×1024	conv5_2
conv6_2	3×3, 1024 conv	1/64H×1/64W×1024	conv6_1
pr6+loss6	3×3, 1 conv	1/64H×1/64W×1	conv6_2
iconv5_1	4×4, 512 deconv, str2	1/32H×1/32W×512	conv6_2
iconv5_2	3×3, 512 conv	1/32H×1/32W×512	[i5_1,pr6,c5_2]
pr5+loss5	3×3, 1 conv	1/32H×1/32W×1	conv5_2
iconv4_1	4×4, 256 deconv, str2	1/16H×1/16W×256	conv5_2
iconv4_2	3×3, 256 conv	1/16H×1/16W×256	[i4_1,pr5,c4_2]
pr4+loss4	3×3, 1 conv	1/16H×1/16W×1	conv4_2
iconv3_1	4×4, 128 deconv, str2	1/8H×1/8W×128	conv4_2
iconv3_2	3×3, 128 conv	1/8H×1/8W×128	[i3_1,pr4,c3_2]
pr3+loss3	3×3, 1 conv	1/8H×1/8W×1	conv3_2
iconv2_1	4×4, 64 deconv, str2	1/4H×1/4W×64	conv3_2
iconv2_2	3×3, 64 conv	1/4H×1/4W×64	[i2_1,pr3,c2_2]
pr2+loss2	3×3, 1 conv	1/4H×1/4W×1	conv2_2
iconv1_1	4×4, 32 deconv, str2	1/2H×1/2W×32	conv2_2
iconv1_2	3×3, 32 conv	1/2H×1/2W×32	[i1_1,pr2,c1_2]
pr1+loss1	3×3, 1 conv	1/2H×1/2W×1	conv1_2
iconv0	4×4, 16 deconv, str2	H×W×16	conv1_2
pr0+loss0	5×5, 1 conv	H×W×1	[i0,pr2,input]

Table 3. **Architecture of DispSharpNet for disparity prediction.** The encorder-decorder module consists of the contracting part ($down1\_1$ to $down5\_2$) and the expanding part ($up4\_0$ to $up1\_3$), followed by the 1D correlation layer $corr\_lr$ and the disparity regression ($conv3\_1$ to $iconv0/pr0$). The final prediction output is $pr0$ (the same size as input images). In the input description, iN is short for $iconvN$, and cN is short for $convN$.

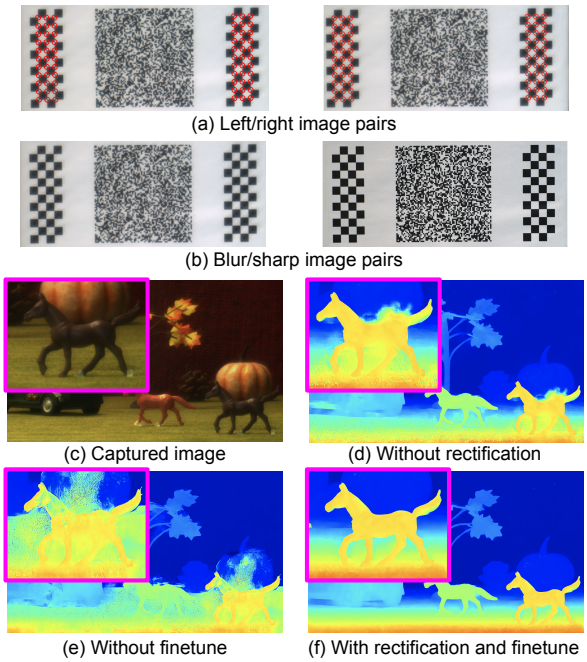

Figure 2. **Rectification and PSFs calibration in real experiment.** The checkerboards in the left/right image pair are used for rectification, and the binary patterns in the blur/sharp image pair are used for PSFs estimation. Artifacts are shown in the predicted disparity map without the rectification and/or the networks finetuning (with the calibrated PSFs).

substrate. Both the hatching distance (along x-y) and the slicing distance (along z) were set to be $200nm$. In particular, the printing laser scanned at a speed of $10mm/s$ with a power of 50%. After printing, the mask was developed in SU-8 developer for 10mins and cleaned in isopropanol (IPA) for 5mins. The refractive indices of the fabricated mask (after polymerization) are around 1.5414 (at $620nm$), 1.5472 (at $540nm$) and 1.5562 (at $460nm$) [2].

Appendix D. System calibration

As our system couples stereo correspondence and blur information together, we calibrate both the correspondence and the PSFs simultaneously. In particular, we use a calibration target as shown in Figure 2. The random binary pattern in the center is used for the PSF estimation, and the checkerboard around it is used for stereo rectification. We follow the PSF calibration procedure in [7] that estimates the PSF from a pair of blur/sharp images by solving a deconvolution problem. The blur image is captured using our lens with the phase mask, and the sharp image is captured by a reference lens with focus adjusted to different depths. To count for the misalignment between simulation and real experiment, we finetune our networks with the calibrated PSF for the best performance.

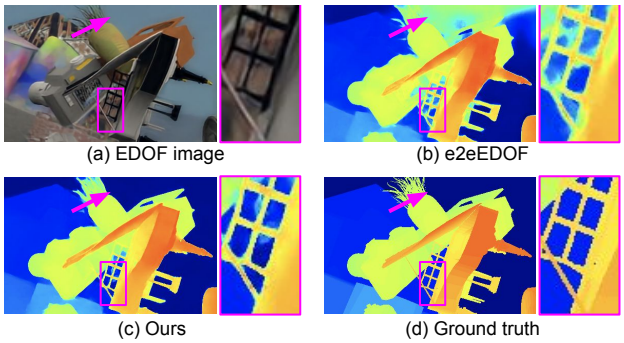

Figure 3. **Comparison with e2eEDOF mask [6] in simulation.** Our phase mask is end-to-end optimized together with both the RGB reconstruction network and the disparity prediction network, and thus can produce precise, pixel-to-pixel correspondence with clearer details and sharper edges than the e2eEDOF mask. Additionally, our optimized PSFs come with some variations along the disparity axis, providing a complementary cue to assist the disparity prediction of problematic areas, as pointed to by the pink arrow.

For rectification, a pair of left/right images, captured with our lens sliding from left to right, are used to estimate the misalignment across two views. Comparisons of disparity prediction results with and without rectification/finetune are shown in Figure 2.

Appendix E. More comparison results

We show more qualitative results of the reconstructed RGB images and the disparity maps of our CodedStereo system, and compare them to the other methods.

In Figure 3, we compare our design with the e2eEDOF mask [6] in simulation. Results show that our design outperforms the e2eEDOF mask by providing precise, pixel-to-pixel correspondence with clearer details and sharper edges. Additionally, our optimized PSFs come with some variations along the disparity axis, providing a complementary cue to assist the disparity prediction of problematic areas.

In Figure 4, we compare conventional stereo and our CodedStereo under different exposures in real experiments, and show our significant improvements over the conventional design.

References

[1] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, 2018. 2

[2] Stephan Dottermusch, Dmitry Busko, Malte Langenhorst, Ulrich W Paetzold, and Bryce S Richards. Exposure-dependent refractive index of nanoscribe ip-dip photoresist layers. *Opt. letters*, 2019. 3

[3] Shivam Duggal, Shenlong Wang, Wei-Chiu Ma, Rui Hu, and Raquel Urtasun. Deeppruner: Learning efficient stereo matching via differentiable patchmatch. In *ICCV*, 2019. 2

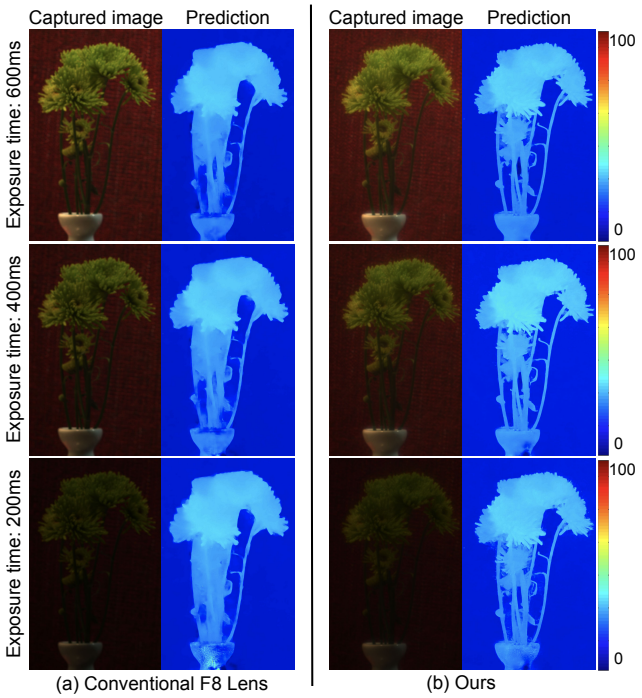

Figure 4. **Comparison with F8 conventional system under different exposures (in experiment). Left:** the captured images and predicted disparity maps of the conventional F8 system. **Right:** the captured images and predicted disparity maps of our Coded-Stereo system. Our system outperforms conventional F8 under different exposure levels.

[4] Joseph W Goodman. *Introduction to Fourier optics*. 2005. 1

[5] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 1, 2

[6] 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. *TOG*, 2018. 3

[7] Yicheng Wu, Vivek Boominathan, Huaijin Chen, Aswin Sankaranarayanan, and Ashok Veeraraghavan. Phasecam3d—learning phase masks for passive single view depth estimation. In *ICCP*, 2019. 3