Passive Snapshot Coded Aperture Dual-Pixel RGB-D Imaging

Supplementary Material

Bhargav Ghanekar[1], Salman Siddique Khan[1], Pranav Sharma[2],
Shreyas Singh[2], Vivek Boominathan[1], Kaushik Mitra[2], Ashok Veeraraghavan[1]
[1] Rice University [2] IIT Madras

1. Methods

1.1. Realistic naive (no code) DP PSFs in simulations

For ours CADS framework, the coded DP PSFs are generated using the mask pattern and the naive (no code) DP PSFs. Modelling the left, right naive DP PSFs accurately is crucial for realistic simulations. The left, right dual-pixels receive light from different halves of the lens. Thus in an ideal scenario, the left, right naive DP PSFs are shaped as semi-circular kernels. However, this is rarely seen in real-world DP PSFs. Errors in manufacturing, optical aberrations, and physical constraints for placement of microlenses and sensor well depths can cause light leakage from the other lens half [2, 10], making it look more like tapered semi-circular halves. In [2], the authors designed a heuristic model to simulate DP PSFs that look closer to the real-world DP PSFs. The DP PSFs (h_z^L, h_z^R) are modeled as the Hadamard product of a 2D Butterworth filter with the circle-of-confusion. We generate our simulated naive DP PSFs in the same manner, choosing $n = 1$ as the filter order, $\alpha = 2.5$, $\beta = 0.4$, and smoothing strength of 7. We generate PSFs for $N_z = 21$ depth planes spanning equally both sides of the defocus. The left, right PSF z-stack corresponds to defocus blur sizes (or circle-of-confusion sizes) ranging from -40 to +40 pixels (signed blur size). Our no-code DP PSFs h_z^L, h_z^R that were used in simulations are depicted in Fig. 1.

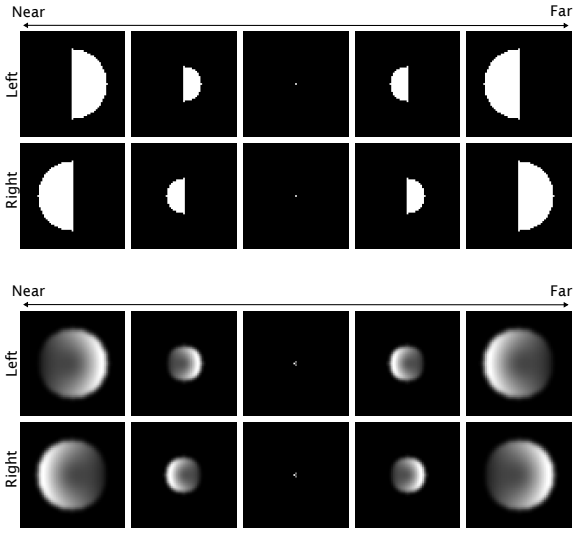

Figure 1. **Modelling parametric DP PSFs for realistic simulations**. (Top row) An ideal dual-pixel sensor would produce left, and right semi-circular PSFs. However, real-world DP PSFs look significantly different. The parametric model from [2] is used to simulate more realistic no-code DP PSFs (Bottom row).

1.2. Occlusion-aware dual-pixel image rendering

For rendering accurate dual-pixel images, we adopt a multi-plane representation of the scene, where the scene is divided into discrete depth planes. Given the coded DP PSFs $h_z^{L,C}$ and $h_z^{R,C}$, the coded DP left, right images I_L, I_R of a 3D scene can be expressed as a sum of 2D convolutions

$$I_L = \sum_{k=0}^{K-1} h_{z_k}^{L,C} * s_{z_k}, \quad I_R = \sum_{k=0}^{K-1} h_{z_k}^{R,C} * s_{z_k}, \tag{1}$$

where s_{z_k} is the scene intensity map which falls into the depth layer k (which is at depth $z = z_k$), K are the number of depth planes (or MPI planes), and $*$ is the 2D convolution operator. To remove artifacts at the edges of the MPI depth layers, we adopt the modifications to Eqn. 1 from Ikoma *et al.* [6], thus we have a differentiable non-linear image formation model as follows

$$\begin{aligned} I_L &= \sum_{k=0}^{K-1} \frac{h_{z_k}^{L,C} * s_{z_k}}{E_k^L} \prod_{k'=k+1}^{K-1} \left(1 - \frac{h_{z_{k'}}^{L,C} * \alpha_{z_{k'}}}{E_{k'}^L}\right), \\ I_R &= \sum_{k=0}^{K-1} \frac{h_{z_k}^{R,C} * s_{z_k}}{E_k^R} \prod_{k'=k+1}^{K-1} \left(1 - \frac{h_{z_{k'}}^{R,C} * \alpha_{z_{k'}}}{E_{k'}^R}\right), \end{aligned} \tag{2}$$

where α_{z_k} is the binary depth mask corresponding to depth layer k (which is at depth $z = z_k$), $E_k^{L,R}$ are normalization factors equal to $h_{z_k}^{L,C} * \sum_{k'=0}^{k} \alpha_{k'}$ and $h_{z_k}^{R,C} * \sum_{k'=0}^{k} \alpha_{k'}$ respectively. We further add a minor modification to the above Eq 2 to the binary depth masks (or alpha maps) α_k. We first expand the alpha maps α_k into 3x3 2D max-pooling and then blend the maps into 2 adjacent depth layers instead of 1 (using 2x1x1 3D avg-pooling), and then normalize the alpha maps. Using 21 depth planes is not a very coarse division of the scene into MPI layers, thus the blending still keeps the simulation realistic, while improving the rendering at edges of individual (and consecutive) MPI layers having fewer fringes/artifacts.

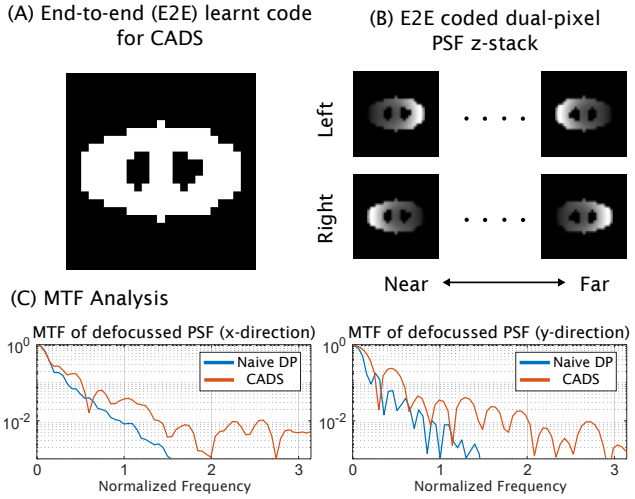

Figure 2. **CADS Learnt Mask.** (A) shows the learned amplitude mask and (B) shows the corresponding PSFs for different depths. (C) shows the MTF of CADS PSF vs. Naive DP PSF (at a defocussed depth, showing only left DP PSF). The higher MTF of the CADS PSF indicates better conditioning of the PSF.

1.3. Evaluation Metrics

We evaluate our simulation results and compare our proposed coded dual-pixel sensing approach (CADS) to naive DP and naive standard-pixel cases, as well as to previous works. We describe our evaluation metrics' definitions here

Depth Metrics. For comparison with naive dual-pixel and naive standard-pixel, we use absolute metrics -

- RMSE (RMS Error): $\frac{1}{N}\sum_{i=1}^{N}|\hat{D}_i - D_i|^2$
- MAE (Mean Absolute Error): $\frac{1}{N}\sum_{i=1}^{N}|\hat{D}_i - D_i|$
- δ^1 with threshold T: $\frac{1}{N}\sum_{i=1}^{N}\left(\max\left(\frac{\hat{D}_i}{D_i}, \frac{D_i}{\hat{D}_i}\right) < T^1\right)$

For comparison with previous works, we use affine-invariant metrics, as used in [5].

- AI(1): $\min_{p,q}\left(\frac{\sum_{i=1}^{N}|D_i-(p\hat{D}_i+q)|}{N}\right)$
- AI(2): $\min_{p,q}\left(\frac{\sum_{i=1}^{N}|D_i-(p\hat{D}_i+q)|^2}{N}\right)^{(1/2)}$
- $1-|\rho_s|$: ρ_s denotes the Spearman's Rank Correlation Coefficient.

AIF Metrics. For AIF deblurred predictions we compare results using PSNR, SSIM, and LPIPS [14] metrics.

2. Results

2.1. End-to-end training results

We perform end-to-end training in simulations as outlined in the CADS Section (Sec. 3). We initialize the mask with a circular open aperture of pixelated size 21x21 (as mentioned in Sec 3.2.3). The mask resolution (in pixels) was chosen based on competing factors. A coarser pixelation (e.g., 7x7, 11x11) would mean the resulting blur would be less informative for small defocus blurs and perform worse. With finer pixelation one risks creating small feature sizes — thereby causing severe diffraction-based distortion and having additional computational costs for end-to-end training — hence the choice of 21x21. Different mask initializations were also tested; the circular open aperture initialization performed the best (on average, 0.35dB better in AIF and 4% better in depth estimation).

Thus, with the above-mentioned mask initialization, we train for 80 epochs on 20k FlyingThings3D scenes with a batch size of 8. The mask learning phase is only for the first 30 epochs, after which the mask is fixed. Cosine decay scheduling is applied for the learning of mask parameters (θ_C) and CADNet weights (θ_D) as well. In our mask parameterization, we initialize α to 0 and increase it using the schedule $\alpha_t = \alpha_0 + \frac{t}{8000}$, where t is the total number of iterations. This is done to ensure smooth learning of a binary coded mask [11]. An initial learning rate of 3×10^{-5} and 3×10^{-4} are used for CADNet weights (θ_D) and for the mask parameters (θ_C), alongwith cosine decay scheduling. During testing/inference, we explicitly threshold the mask, setting it to be binary.

2.2. End-to-end CADS Learnt Mask

The learned mask and the corresponding left, right coded DP PSFs are shown in Fig. 2. The CADS PSFs are better conditioned as compared to the naive DP PSFs, as indicated by the MTF plots in Fig. 2(C). The learned coded aperture has the shape of a flattened ellipse (with a larger horizontal diameter), with two dots inside. The possible reason for this can be explained as follows. The intended goal is to learn a mask that gives the best defocus map prediction and AIF image prediction when coupled with dual-pixel sensors. In order to do so, the coded DP PSFs should be able to show a disparity effect with defocus, while having the shortest possible blur size. Since the disparity between the DP PSFs is horizontal, the learned coded aperture maintains the horizontal opening of the aperture, while reducing the vertical opening. Furthermore, the two opaque dots near the center on the horizontal axis potentially add to better conditioning of the mask and ensure recovery of high-frequency texture (mainly in the horizontal/x-direction).

2.3. Simulation results

2.3.1 Ablations on coded aperture

Coded aperture masks have been used previously for PSF engineering to gain optimal deblurring and depth estimation performance [3, 8, 11, 12]. These have been used in the naive standard-pixel settings, and hence may not necessarily translate to optimal performance in the dual-pixel sensor setting. We compare the naive (no-code) DP case and our end-to-end learned code with the following codes in the dual-pixel sensor setting:

CADS Mask	Depth Pred.	AIF Pred.	
	MAE(mm) ↓	PSNR ↑	SSIM↑
No code	5.51	29.7	0.83
No code (50%)	5.77	30.6	0.85
Levin *et al.* [8]	5.54	30.6	0.85
MLS code [3]	5.64	30.3	0.85
Shedligeri *et al.* [11]	5.49	30.4	0.85
E2E learnt (ours)	5.15	31.2	0.87

Table 1. **Coded Mask Ablation.** Coded aperture DP outperforms naive DP for AIF and depth prediction. Among coded DP designs, the proposed end-to-end learned design offers the best performance. Red indicates best, orange indicates second best.

Figure 3. **Coded apertures considered for ablations**. We test out various coded apertures in conjunction with dual-pixel sensors in simulations. From left to right, showing no code mask, no code (50% size) mask, Levin *et al.* [8] mask, separable MLS code [15], Shedligeri *et al.* [11] mask, and our proposed end-to-end learned CADS mask.

- Open aperture (no-code) that is 50% smaller in area
- Code from Levin *et al.* [8], which was derived based on an optimization problem formulated for constructing a desirable depth estimation mask.
- Separable MLS code from [3] that has flat singular value spectrum and has been used for lensless imaging
- Code from Shedligeri *et al.* [11]. This code was learned end-to-end for the purposes of depth estimation, albeit in a standard pixel setting (not for dual-pixel sensors).

The coded apertures are depicted in Fig. 3. We trained CADNet-Mono for the above masks while keeping the mask fixed (no learning). The results of depth estimation and deblurring performance are given in Table 1. All the coded masks perform similarly in depth estimation and show gains in deblurring performance. Our proposed end-to-end learnt code possesses the novelty of being the first one to be trained specifically for dual-pixel sensor imaging. Hence, our end-to-end learnt code outperforms all the above coded masks to give the best depth estimation and deblurring performance.

2.3.2 Comparison with existing DP-sensing works

We test existing DP-sensing works on simulated naive DP captures based on FlyingThings3D scenes and also on simulated naive DP captures based on NYUv2 scenes. Since existing methods were designed for reconstructing from naive DP captures, we created a simulated dataset of naive (no code) DP captures, using the FlyingThings3D dataset scenes and another one using NYUv2 dataset scenes. We compare the following works

- **DPDNet** - the authors in [1] designed a UNet-based neural network to reconstruct the deblurred (all-in-focus) image from DP captures, called DPDNet. DPDNet was trained using supervised real-world AIF GT data. We use the same trained model weights as given in their repository.
- **DDDNet** - the authors in [9] designed a two-stage neural network (called DDDNet) to predict the disparity map and the deblurred (all-in-focus) image of the scene from DP captures. We use the same model weights as given in their repository.
- **Xin *et al.* 2021**. In [13], an optimization problem was formulated for simultaneous defocus map and all-in-focus image recovery. We use the code given in their repository and pass our simulated DP PSFs as arguments for the optimization problem. Owing to the fact that the runtime for this was slow, we evaluate for a randomly selected set of 16 DP captures.
- **Punnappurath *et al.* 2020**. In [10], an optimization problem was formulated for recovering the disparity map from DP captures, exploiting left-right kernel symmetry to do so. We use the code given in their repository. Due to slow inference time, we evaluate for a randomly selected set of 16 DP captures.
- **Kim *et al.* 2023**. In [7], the authors trained a stereo disparity estimation network to handle bi-directional disparity, and then devised a self-supervised loss to learn disparity based on the DPDBlur [1] dataset. We evaluate the code given on their repository.

Table 2 shows the quantitative results of previous methods with our CADS method, and with our method for the naive DP case as well. The simulated DP captures showed bidirectional disparity in the left, right captures. Thus, methods designed for uni-directional disparity did not work well [9, 13]. Punnappurath *et al.* [10] outputs disparity maps that resemble the ground truth but are not as accurate as our methods. While Kim *et al.* [7] is trained to handle bi-directional disparity, the error is higher, possibly due to the fact that the self-supervised loss is not trained on the simulated captures.

2.4. DSLR Photography results

2.4.1 Experimental setup

For real-world DSLR photography experiments, we use the Canon EOS 5D Mark IV DSLR. It is equipped with a 30MP color sensor (6880x4544 pixels) with a R-G-G-B Bayer pattern, with pixel pitch $p = 5.36$ μm. For DSLR photography, we capture naive DP images and CADS images with a Yongnuo 50 mm focal length lens with aperture $L = f/4 = 12.5$ mm. We print our binary amplitude mask having 12.5 mm diameter on transparency sheets, and place

FlyingThings3D dataset	AIF Predictions		Disparity Predictions		
Method	PSNR(dB)↑	SSIM↑	AI(1)↓	AI(2)↓	$1-\lvert\rho_s\rvert$ ↓
DPDNet [2]	24.34	0.63	N.A.	N.A.	N.A.
DDDNet [9]	21.35	0.54	0.277	0.381	0.561
Xin *et al.* [13]†	18.13	0.30	0.302	0.415	0.951
Punnappurath *et al.* [10]†	N.A.	N.A.	0.194	0.264	0.243
Kim *et al.* [7]	N.A.	N.A.	0.287	0.382	0.694
Naive DP (ours)	29.72	0.83	0.019	0.050	0.092
CADS (ours)	31.20	0.87	0.018	0.046	0.087
NYUv2 dataset					
DPDNet [2]	26.90	0.80	N.A.	N.A.	N.A.
DDDNet [9]	20.42	0.56	0.303	0.392	0.595
Xin *et al.* [13]†	16.70	0.37	0.302	0.412	0.579
Punnappurath *et al.* [10]†	N.A.	N.A.	0.149	0.204	0.170
Kim *et al.* [7]	N.A.	N.A.	0.306	0.386	0.489
Naive DP (ours)	29.33	0.87	0.017	0.030	0.058
CADS (ours)	31.32	0.91	0.016	0.029	0.057

Table 2. **Comparison with existing DP methods.** CADS offers the best AIF and disparity estimation quality on our simulated DP captures based on FlyingThings3D scenes, and on our simulated DP captures based on the NYUv2 scenes. Red highlights best, orange highlights second best. † indicates metrics computed over 16 samples since these methods had a slow runtime. For methods where AIF/disparity is not predicted, metrics are marked as N.A.

it inside the DSLR lens, as done in [8]. Fig. 4 illustrates the same. We set the focus distance of the camera to 40 cm, and we set up toy scenes 32–53 cm away from the camera. These imaging parameters of the setup were chosen such that the defocus blur sizes will approximately be within 0-40 pixel size (on both sides of the defocus).

Canon Dual-pixel RAW data capture. We capture images using the Canon camera, under the DP-RAW setting with the lowest possible ISO setting of 100. We process the raw .CR2 files using Adobe DNG converter to convert them to the DNG format. We extract the combined (L+R) and left (L) images from the DNG files using the Tiff() capabilities in MATLAB. For simplicity we do not perform demosaicking, thus we obtain RGB DP captures of size 3440x2272x3. The 14-bit RAW data is appropriately scaled and the black-level is subtracted to get the left, right DP images.

Scene capture details. For a given scene, we capture a naive (no code) DP measurement, a CADS measurement, and also a $f/22$ measurement to obtain the ground truth deblurred all-in-focus image of the scene. Furthermore, we also capture a coarse ground truth depth map using an Intel RealSense D415 stereo sensor (see Fig. 4). We pre-calibrate the Intel RealSensor sensor with our Canon DSLR sensor with the help of a 10x12 checkerboard pattern, so as to transform the depth map into the DSLR's frame of reference. We crop the captured DP measurements and only reconstruct the central 1696x1522x3 region. We show a few more example results in Fig. 6.

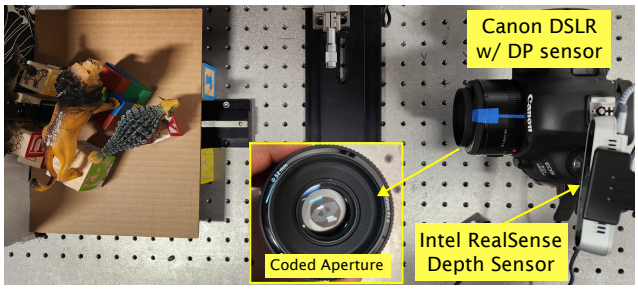

Figure 4. **Experimental setup for DSLR photography captures.** The coded aperture is placed in the aperture plane inside the Yongnuo lens.

2.4.2 PSF capture, calibration, and fine-tuning

Real-world PSF capture. To capture PSFs, we capture pinhole images (illuminated with white light) for 21 depths ranging from 32 cm to 53 cm. We capture these for the no-code (open aperture) case and for the E2E learnt CADS case. Fig. 5 illustrate the same.

Fine-tuning. To reconstruct real-world captures, we perform fine-tuning of a trained CADNet-RGB model (trained on simulated FlyingThings3D scenes using simulated DP PSFs). We first capture real-world PSFs as mentioned above. The real-world PSFs are used to simulate DP captures based on FlyingThings3D scenes, and CADNet-RGB model weights are fine-tuned on the new captures for 30 epochs. During fine-tuning phase, we train for a variable amount of heteroscedastic noise [4] levels ranging from 0.7%–1.5%, along with extra data augmentations (random)

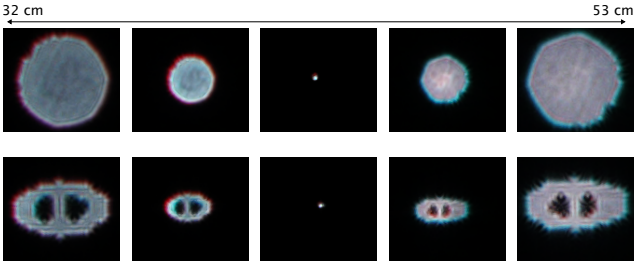

Figure 5. **DSLR PSF captures**. Showing combined (L+R) real-world PSFs captured at different depths. (Top row) no-code case. (Bottom row) CADS case.

on brightness, contrast, gamma, and hue. This is done to remove certain sim-to-real mismatches to enable better depth and AIF reconstructions.

Calibration. In [13], the authors describe a vignetting calibration scheme. Images of a white sheet are captured to model the vignetting profile for left, right DP images. Scenes are pre-processed by dividing the left, right scene captures with the corresponding vignetting images. We follow the same procedure in our case as well.

2.5. Endoscopy, Dermoscopy results

2.5.1 Experimental setup details

Endoscopy setup and capture details. We use a Karl Storz 26003ARA Rubina rigid endoscope of 10 mm diameter. A rigid endoscope consists of several relay lenses to relay the image from the patient side to the surgeon side (eyepiece). For making a prototype CADS endoscope, we mount a 2.5mm coded mask in front of the Canon DSLR lens and align its optical axis to that of the endoscope. We focus the DSLR such that it is focussed at a point 20 mm away from the other end of the endoscope. Fig. 8 illustrates ours CADS endoscope prototype. We capture a 2D PSF array 2.5 mm away from the scope, and scale it down to obtain PSFs at 21 depths ranging from 2.5 mm to 20 mm. We perform fine-tuning as outlined before but with some more modifications – (1) we implicitly model PSF spatial variance across the FoV (see ahead for more details), (2) we only reconstruct for negative defocus, and (3) along with data augmentations, we add random bias and random attenuation to one of the channels (because the vignetting correction is not perfect). With such a setup, we are able to capture ≤ 40 μm features over an extended depth-of-field, as illustrated in Fig. 7.

Dermoscopy setup and capture details. We built a prototype CADS dermoscope using the Pixel 4 camera, 12x macro lens, and a 2.5 mm diameter CADS mask. We focus at the closest distance possible i.e. 45 mm. We capture PSFs at 32 mm and scale them accordingly to obtain PSFs at 21 depth planes from 32 mm to 76 mm. We perform fine-tuning in the same way as outlined for the endoscopy case. We capture DP data using an open-source Android app https://github.com/google-research/google-research/tree/master/dual_pixels. Since the data obtained has rectangular-shaped pixels (2:1) we re-size the smaller dimension accordingly and reconstruct for a central FoV of 512x432 pixels.

2.5.2 Modelling spatially-varying PSFs

For our CADS endoscopy and dermoscopy setup, we observe that the coded DP PSFs vary over the entire field of view (FoV), as shown in Fig. 9. This leads to improper reconstruction results if we fine-tune CADNet with the central PSF only. To account for this spatial variance, we capture a 2D array of PSFs across the FoV. During the fine-tuning phase, we randomly sample one of the PSFs (out of many) in every iteration and use its corresponding PSF z-stack to simulate and render out the coded dual-pixel images. By doing so, the CADNet network sees all the variations in the PSF for the given system and thus can correct for those reliably. Fig. 10 illustrates the same. For the case in which we fine-tune only with the central PSF z-stack, we obtain incorrect depth maps and slightly incorrect AIF maps. Our fine-tuning using the captured PSF 2D array gives better AIF reconstruction and much better depth reconstruction. We use the same fine-tuning procedure (using a captured PSF 2D array) for our dermoscopy experiments.

References

[1] Abdullah Abuolaim and Michael S Brown. Defocus deblurring using dual-pixel data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 111–126. Springer, 2020. 3

[2] Abdullah Abuolaim, Mauricio Delbracio, Damien Kelly, Michael S Brown, and Peyman Milanfar. Learning to reduce defocus blur by realistically modeling dual-pixel data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2289–2298, 2021. 1, 4

[3] M Salman Asif, Ali Ayremlou, Aswin Sankaranarayanan, Ashok Veeraraghavan, and Richard G Baraniuk. Flatcam: Thin, lensless cameras using coded aperture and computation. *IEEE Transactions on Computational Imaging*, 3(3): 384–397, 2016. 2, 3

[4] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE transactions on image processing*, 17(10):1737–1754, 2008. 4

[5] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T. Barron. Learning single camera depth estimation using dual-pixels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[6] Hayato Ikoma, Cindy M Nguyen, Christopher A Metzler, Yifan Peng, and Gordon Wetzstein. Depth from defocus with

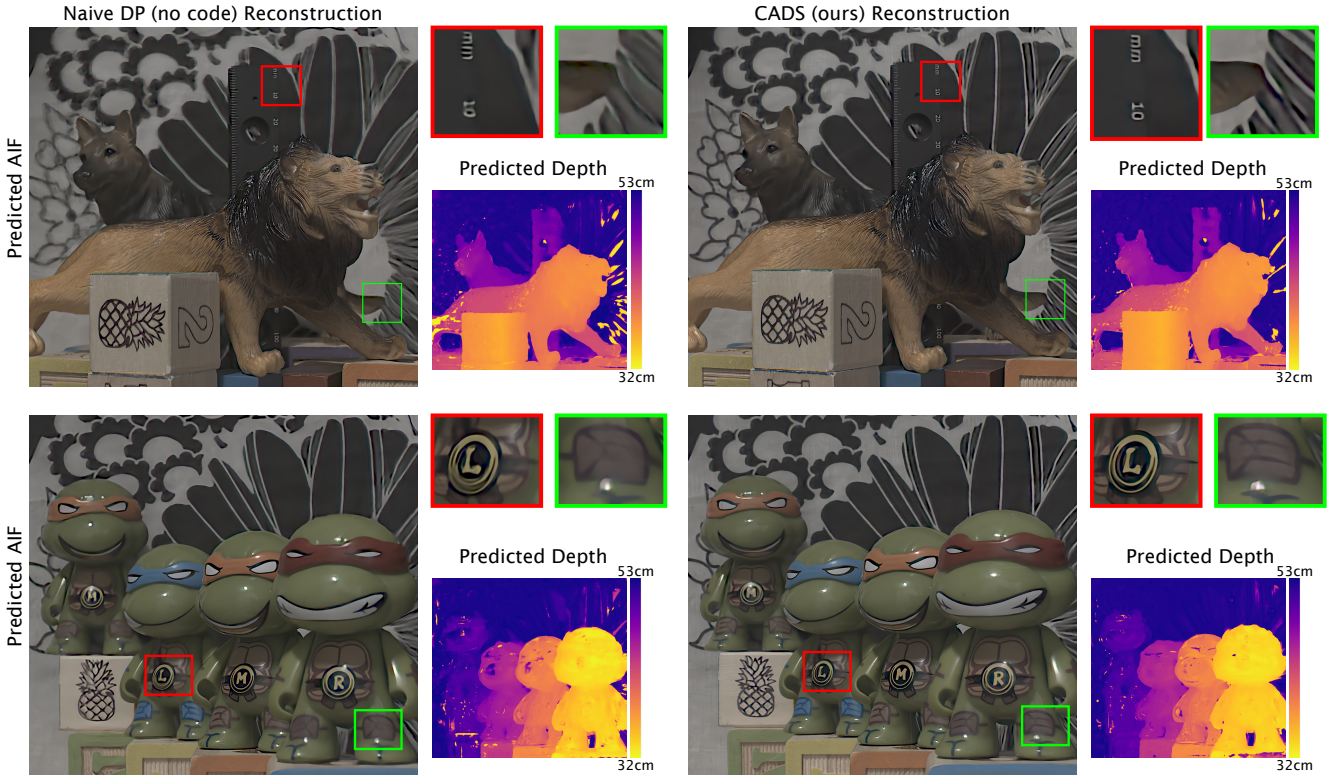

Figure 6. **DSLR photography additional results**. Showing Naive DP and CADS (ours) reconstructions (depth and AIF). Zoom-in for better comparison.

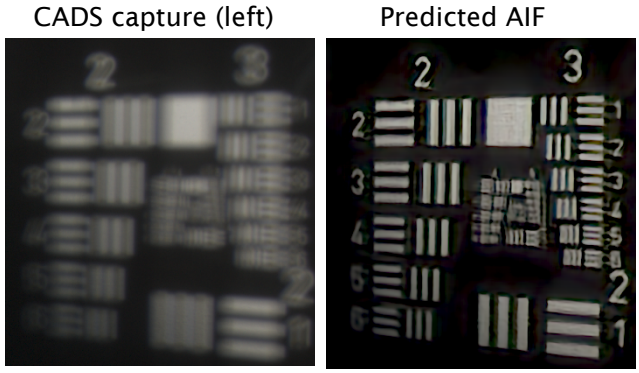

Figure 7. **USAF reconstruction result**. With our CADS prototype endoscope, we are able to resolve ≤ 40 μm features, as line pairs in Group (3,5) are visible in the reconstruction. USAF target placed $\sim$10mm away from scope.

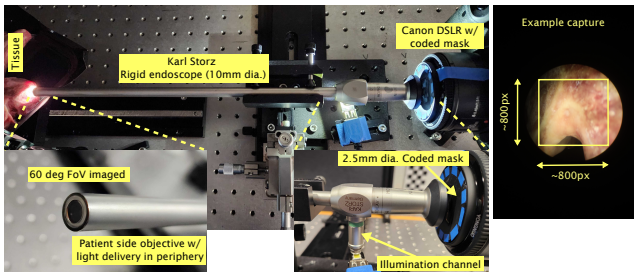

Figure 8. **CADS Endoscopy setup**. (Left) CADS endoscope setup, with zoom-in insets. (Right) Showing an example RAW capture.

learned optics for imaging and occlusion-aware depth estimation. In *2021 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2021. 1

[7] Donggun Kim, Hyeonjoong Jang, Inchul Kim, and Min H. Kim. Spatio-focal bidirectional disparity estimation from a dual-pixel image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5023–5032, 2023. 3, 4

[8] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26 (3):70–es, 2007. 2, 3, 4

[9] Liyuan Pan, Shah Chowdhury, Richard Hartley, Miaomiao Liu, Hongguang Zhang, and Hongdong Li. Dual pixel exploration: Simultaneous depth estimation and image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4340–4349, 2021. 3, 4

[10] Abhijith Punnappurath, Abdullah Abuolaim, Mahmoud Afifi, and Michael S Brown. Modeling defocus-disparity in

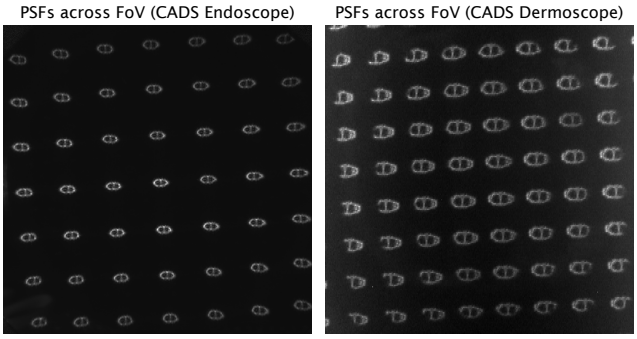

Figure 9. **Spatial variance of PSFs across FoV**. Showing for endoscopy and dermoscopy cases (left and right respectively).

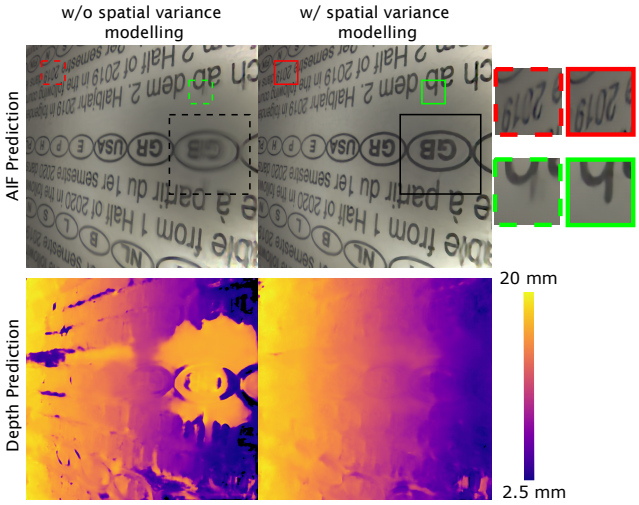

Figure 10. **Modelling spatial variance in PSFs across FoV.** For coded dual-pixel endoscopy captures, we observe that accounting for several PSFs across the FoV during fine-tuning phase enables better reconstruction in AIF and especially in depth predictions.

dual-pixel sensors. In *2020 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2020. 1, 3, 4

[11] Prasan A Shedligeri, Sreyas Mohan, and Kaushik Mitra. Data driven coded aperture design for depth recovery. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 56–60. IEEE, 2017. 2, 3

[12] Ashok Veeraraghavan, Ramesh Raskar, Amit Agrawal, Ankit Mohan, and Jack Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph.*, 26(3):69, 2007. 2

[13] Shumian Xin, Neal Wadhwa, Tianfan Xue, Jonathan T Barron, Pratul P Srinivasan, Jiawen Chen, Ioannis Gkioulekas, and Rahul Garg. Defocus map estimation and deblurring from a single dual-pixel image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2228–2238, 2021. 3, 4, 5

[14] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 2

[15] Changyin Zhou, Stephen Lin, and Shree Nayar. Coded aperture pairs for depth from defocus. In *2009 IEEE 12th international conference on computer vision*, pages 325–332. IEEE, 2009. 3