This supplemental document provides technical details of the blur kernel experiment, network architecture, datasets, and training. We also present additional results.

1. Blur Kernel Experiments

For our blur kernel measurements, we use a Canon EOS 5D Mark IV DSLR camera equipped with SIGMA Art 50mm f/1.4 lens, using the widest aperture (i.e., \( N = 1.4 \)). To generate point-light patterns, we display white dot patterns with black background on a large LED display (Dell P4317Q), place the camera 2 m away from the display, and capture the dot pattern on the display (see Figure 1). In Figure 2 of the main paper, we show how kernels change their shape depending on the focus plane depth of a point source. In this experiment, we fix the depth of the point source and change the focus depth of the lens instead so that we can avoid the source-camera registration problem. Note that changing the depth of a source is virtually equivalent to changing the focus position of optics. All dual-pixel images in the first column of Figure 2 are captured in a dark room to minimize the effect of ambient illumination. We then separate the left and right pairs of the dual-pixel images by Canon’s official software (Digital Photo Professional 4) using its Dual Pixel RAW Optimizer. We fix the same camera setup and environment for all captures, then vary the focus plane depth by controlling the lens’ focus distance. Figure 2 shows the results. Note that we convert three color channels into a gray channel and tonemap it for better visualization. The result shows that the blur kernel is isotropic when the pixels are gathered from all diodes (the second column in Figure 2), whereas it is split into two reflection-symmetric anisotropic kernels (the third and fourth columns in Figure 2) when collected separately. In addition, the reflection-symmetric kernels get inverted based on the in-focus plane of depth. This blur kernel experiment shows that our model in the main paper supports the optical characteristic of dual-pixel imaging.

2. Network Details

From the baseline binocular stereo network [6], we use a single backbone to reduce the number of parameters. The network takes two images as input and computes the correlation pyramid and context information. Then, from the generated correlation pyramid and context information, it performs an iterative update of disparity using a gated recurrent unit (GRU). For all experiments, we use 22 GRU update iterations for training and 32 GRU update iterations for testing. And to handle the large blurriness in dual-pixel images, we enlarge the correspondence search range that can enough cover the blurriness. The correspondence search range is enlarged by increasing the correlation pyramid level and lookup range of correlation. In particular, a correlation pyramid level of 5 and a lookup range of 8, where the original backbone uses 4 and 4, respectively.

3. Dataset Details

SceneFlow [8]. This dataset is used in the pretraining stage of our method. It provides synthetic stereo image pairs with corresponding ground truth disparity. We use 31,100 image pairs from the train splits of the two finalpass datasets of the
Aubolaim et al. [1]. This dataset is used in our self-supervised learning. Originally this dataset is created for deblurring, containing 350 training dual-pixel image pairs, including blurry and corresponding clean images. We only use the blurred image to train and test our method. The size of the original image is 1680 × 1120, and we use half of the size as input for training, and testing with the original size, removing some border pixels following the Punnappurath et al. [12].

Punnappurath et al. [12]. This dataset is only used in our testing. It provides 100 dual-pixel left and right image pairs with corresponding ground-truth inverse depth. The ground-truth inverse depth is estimated by a depth-from-defocus method using commercial software (HeliconSoft). In this dataset, there are 10 scenes, and each scene has 10 different focus plane depths, resulting in 100 images in total. The size of the original image and its ground truth inverse depth is 5180 × 2940. For testing, we use half of the size of an image and remove some border pixels following the original paper’s procedure [12].

4. Training Details

We implement our method using PyTorch [11]. We train each stage with different schemes. In the pretraining stage, we perform training for 40,000 iterations with a batch size of 40. Same as baseline method [6], we use one cycle learning rate scheduling policy [13] with a peak learning rate of 1.0e−4 and optimize using the AdamW optimizer [7]. We use a mixture of multiple synthetic stereo datasets, SceneFlow [8], Sintel stereo [3], Falling Things [17], and Tartan Air [19]. These stereo datasets provide left and right binocular stereo images along with corresponding unidirectional disparity maps. For those having blur information, we include blurry data (named final pass) to learn blurred dual-pixel images. For the random crop augmentation, we use a crop size of 640 × 348.

In the second stage, our self-supervised learning, we train our model for 1,500 iterations with a batch size of 8. The learning rate scheduling policy and optimizer remain the same as the first step, but with a reduced peak learning rate of 1.0e−5. Since the second stage is self-supervised, we only need dual-pixel left and right image pairs for training. We use DSLR dual-pixel images from Aubolaim et al. [1]. Note that this dataset includes enough coverage of large defocus blur for the typical dual-pixel using scenario, which enables the robustness of our method with blurry input. For all of these images, we use the 1/2 size of the images from the raw dataset. In this step, we crop the input image to have 800 × 512. For testing, the input image is bilinearly resized to a width of 640, and the output disparity is resized and scaled to fit the input image.
Table 1. Quantitative evaluation result of our method trained with and without occlusion mask in the photometric loss. The same evaluation metrics presented in the main paper are used. We do not observe clear improvement even with an occlusion mask.

| Method | $A_s(1)$ | $A_s(2)$ | $1 - |\rho_s|$ |
|--------|----------|----------|-----------------|
| Ours, w/o occlusion mask | 0.0391 | 0.0682 | 0.2619 |
| Ours, occlusion mask [20] | 0.0391 | 0.0681 | 0.2620 |
| Ours, occlusion mask [16] | 0.0391 | 0.0681 | 0.2621 |

5. Creating Synthetic Dual-pixel image

Before developing the proposed self-supervised learning approach, we exploit the potential of supervised learning by evaluating two possible ways to create a dual-pixel dataset. First, any RGB-D dataset can be used to create a dual-pixel dataset [2]. Naïve shifting of depth layers lacks the blur effects; also, artificial blurring without camera information is not physically faithful. Second, we can create synthetic dual-pixel rendering images by modifying the center of projection. It requires a custom camera model that simulates light transport with respect to lens and dual-pixel on top of a conventional path-tracing light simulation. In this way, we create a synthetic dual-pixel image shown in Figure 1 of the main paper. However, this is an arduous task to do. Therefore, we have decided to use a self-supervised learning approach rather than generating the supervised dual-pixel dataset with this laborious approach.

Main paper Figure 1. We generate the dual-pixel image using the dual-pixel light transport path tracing method mentioned above. We start from the micro-lens array-based light field camera model [4]. Note that a dual-pixel image can be considered as a two-sampled light field camera [10,18], since split pixels under micro-lens is equivalent to separating the aperture in a micro-lens array-based light field camera [9]. Based on this, we first render the light field with $2N \times 2N$ sub-aperture images by properly modifying each sub-aperture image’s center of projection. Then, by accumulating each half $(2N \times N)$ of the sub-aperture images [14], we obtain the corresponding left/right dual-pixel image. We perform this process with Blender’s Cycles path tracer with python script where $N = 8$.

6. Occlusion Mask in Photometric Loss

It is a common practice to use an occlusion mask in the learning-based stereo and optical flow methods [5, 15, 16, 20]. However, we observe that the occlusion mask shows an insignificant effect on the dual-pixel disparity estimation, since the edges are blurred if it has disparity, and the horizontal shift is much smaller compared with the traditional binocular stereo case. Figure 3 shows the occlusion mask estimation examples on traditional binocular stereo images and dual-pixel images. We show two occlusion mask estimation methods: range map [20] and forward-backward consistency check [16]. As shown in Figure 3, there are a significant amount of occluded pixels in the binocular stereo pair, whereas nearly no occluded pixels are observed in the dual-pixel setup. We also show this quantitatively in Table 1, as the occlusion mask shows no impact on our disparity estimation.

7. Additional Bidirectional Results

Figures 4 and 5 show additional bidirectional disparity estimation results. We compare our method with the others: Wadhwa et al. [18] and Punnappurath et al. [12]. The results show that our method successfully estimate bidirectional disparities with dual-pixel images.

8. Additional Unidirectional Results

Figures 6 and 7 show additional inverse depth estimation results. We compare our method with the others: Wadhwa et al. [18], Punnappurath et al. [12], and Pan et al. [10]. Our method is consistent at homogeneous depth regions...
Figure 4. Additional bidirectional disparity estimation results with varying focus plane depth on multiple scenes. In our setup, the bidirectional disparity’s sign is positive if its direction is from left to right (color-coded with blue), and negative the other way around (red).
Figure 5. Additional bidirectional disparity estimation results with varying focus plane depth on multiple scenes.
Figure 6. Additional inverse depth estimation results on different scenes.
9. Results on Google Pixel dataset

As mentioned in the Discussion section in the main paper, different from the DSLR dataset, the image blur kernels in the Google Pixel smartphone dataset [21] do not hold the reflection symmetry. Even though the reflection symmetry does not hold, which is the key observation of our method, we compare the depth estimation accuracy of our method on the Google Pixel smartphone dataset with other methods [10, 12, 18, 21]. When the optical aberrations in the dataset are not corrected by calibration, our method outperforms the other methods tested with the uncalibrated dataset. On the other hand, it is not surprising that when a physically-based optical calibration is applied to the dataset, Xin et al. [21] shows the highest performance. See Table 2 for the result.

Table 2. Unidirectional depth evaluation results on the Google Pixel smartphone dataset. The same metrics are used as those in the main paper Table 2. The lower, the better.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncalibrated</td>
<td>Wadhwa et al. [19]</td>
<td>0.1304</td>
<td>0.1694</td>
<td>0.5386</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Punnappurath et al. [12]</td>
<td>0.1437</td>
<td>0.1869</td>
<td>0.6359</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pan et al. [10]</td>
<td>0.1358</td>
<td>0.1825</td>
<td>0.6186</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>0.1246</td>
<td>0.1586</td>
<td>0.4688</td>
<td></td>
</tr>
<tr>
<td>Calibrated</td>
<td>Xin et al. [21]</td>
<td>0.0488</td>
<td>0.0733</td>
<td>0.1189</td>
<td></td>
</tr>
</tbody>
</table>
References


