2020 IEEE International Conference on Computational Photography

Modeling Defocus-Disparity in Dual-Pixel Sensors

Abhijith Punnappurath, Abdullah Abuolaim*, *Student Member, IEEE,* Mahmoud Afifi*, *Student Member, IEEE,* and Michael S. Brown, *Member, IEEE*

Abstract—Most modern consumer cameras use dual-pixel (DP) sensors that provide two sub-aperture views of the scene in a single photo capture. The DP sensor was designed to assist the camera's autofocus routine, which examines local disparity in the two sub-aperture views to determine which parts of the image are out of focus. Recently, these DP views have been used for tasks beyond autofocus, such as synthetic bokeh, reflection removal, and depth reconstruction. These recent methods treat the two DP views as stereo image pairs and apply stereo matching algorithms to compute local disparity. However, dual-pixel disparity is not caused by view parallax as in stereo, but instead is attributed to defocus blur that occurs in out-of-focus regions in the image. This paper proposes a new parametric point spread function to model the defocus-disparity that occurs on DP sensors. We apply our model to the task of depth estimation from DP data. An important feature of our model is its ability to exploit the symmetry property of the DP blur kernels at each pixel. We leverage this symmetry property to formulate an *unsupervised* loss function that does not require ground truth depth. We demonstrate our method's effectiveness on both DSLR and smartphone DP data.

Index Terms—Defocus, disparity, dual-pixel sensor, depth estimation

1 Introduction

Dual-pixel (DP) sensors use a unique design that splits each pixel site in half using two photodiodes. This split-pixel arrangement acts as a rudimentary light-field camera and allows the sensor to capture *two* sub-aperture views of the scene in a *single* exposure. Image regions that are outside the optical depth of field will observe a disparity between the two DP sub-aperture views. Camera systems use this information to determine how to move the lens in order to minimize the dual-pixel disparity in specific regions of interest, and thereby bring those regions into focus. While the DP sensor was designed for the purpose of autofocus, many recent papers have used these two DP views for other tasks, such as enhanced synthetic bokeh [1], reflection removal [2], and depth reconstruction [3]. These methods treat the DP images as stereo image pairs and estimate disparity using stereo matching techniques.

Despite the resemblance to classic stereo, the nature of the disparity in DP sensors has key differences from stereo disparity. Classic stereo treats disparity as an explicit shift in image content between the two images. This can be modeled as a PSF with a single impulse as shown in Fig. 1. The split-pixel arrangement in a DP sensor in conjunction with the camera optics has the effect that light rays passing through the right side of the lens are captured by the left half-pixels (left sub-aperture view) and those passing through the left side of the lens are collected by the right half-pixels (right sub-aperture view). For an in-focus region of the scene, there will be no disparity between the left and right DP views. However, for a region of the scene that is away from the focal plane, a *defocus-disparity* is induced by the out-of-focus blur being split across the two views in opposing directions (see Fig. 1(a)). Thus, unlike in stereo, it is the difference in the point spread functions (PSF) that produces disparity between DP views and not an explicit shift in image content. This is illustrated in Fig. 1(b) that shows the empirically measured PSFs for the left and right DP views.

Contribution This paper examines the nature of defocus-disparity in DP sensors and proposes a parameterizable PSF to model the blur kernels based on empirical observations of both DSLR and smartphone DP data. A useful property of this model is that the PSF in one DP view equals the PSF in the other DP view flipped about the axis perpendicular to the disparity axis. Using this symmetry property, we describe how to formulate an unsupervised loss function for the task of depth estimation from DP data. Our unsupervised loss has the advantage of using only data from the DP sensor, and circumvents the need for ground truth depth which is hard to acquire. Moreover, by explicitly formulating DP disparity as a defocus kernel in line with the image formation, our method makes judicious use of the depth cues embedded within the defocus blur. We establish the effectiveness of our loss function using a straightforward optimization-based approach. We demonstrate experimentally that our unsupervised optimization produces more accurate results than classic stereo matching algorithms (see Fig. 1(c)) and also compares favourably against state-of-the-art monocular and stereo-based deep learning methods for depth estimation that require supervised training. Additionally, we show that considerable speed-up can be achieved by replacing our optimization algorithm with a convolutional neural network (CNN).

- *A. Punnappurath, A. Abuolaim, and M. Afifi are with the Department of Electrical Engineering and Computer Science, York University, Toronto, Canada.*
  M. S. Brown is with the Department of Electrical Engineering and Computer Science, York University, Toronto, Canada, and Samsung Research AI Center, Toronto, Canada.
  ** denotes equal contribution.*
  E-mail: {pabhijith,abuolaim,mafifi,mbrown}@eecs.yorku.ca

978-1-7281-5230-1/20/$31.00 ©2020 IEEE

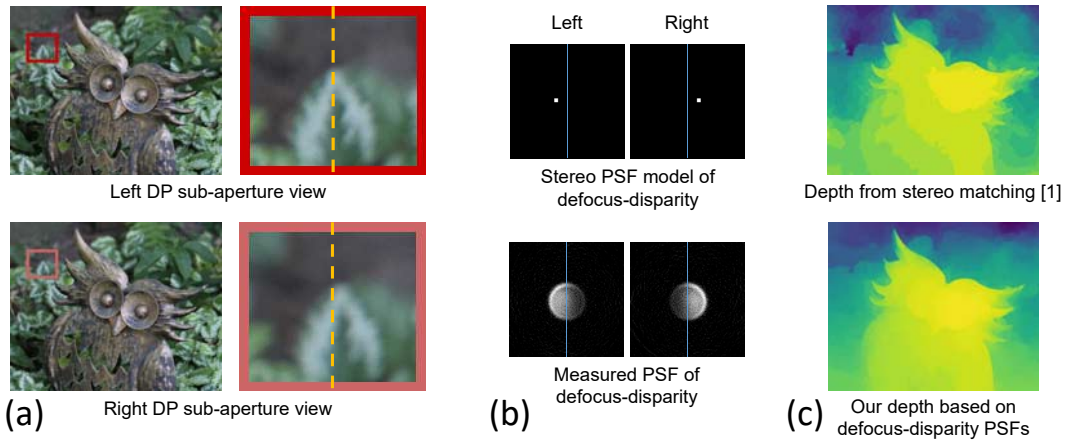


Fig. 1. (a) Shows the two sub-aperture views (denoted as *left* and *right*) from a DP sensor. An image region is zoomed in. The dotted reference line illustrates the disparity between the two views. (b) Shows the shifted impulse kernels for classic stereo and the empirically measured defocus kernels corresponding to the center pixels of the left and right zoomed-in DP patches. Classic stereo matching attempts to map one image to the other through an explicit shift in image content. This shift can be modeled as an impulse PSF. Disparity in a DP sensor, however, is the result of defocus blurring by two different PSFs acting in opposing directions. (c) Our method, which models the disparity in a DP sensor using defocus blur kernels, is able to estimate a more accurate depth map than algorithms based on traditional stereo matching (e.g., [1]).

Finally, we note DP data has an inherent ambiguity in that depth can be estimated only up to an unknown affine transformation [3]. This ambiguity is related to the optics of a DP sensor, and is similar to the well-understood scale ambiguity in depth from stereo (if the extrinsics are unknown). We refer readers to [3] for a detailed discussion on the connection between disparity and affine-transformed depth. Following [3], in this work, we also estimate depth up to an affine ambiguity.

2 Related Work

Dual-pixel sensors were introduced as a mechanism to provide fast and accurate autofocus [4]. The autofocus system on a DP camera exploits the defocus disparity induced between the left and right views for a region of the scene that is out of focus. By evaluating the *signed* average disparity value within a region of interest, the autofocus routine can determine the direction and extent by which the lens has to be moved to minimize disparity, and thereby bring that region into focus. Recent work has showed that DP data can be used for additional tasks, such as synthetic bokeh [1], reflection removal [2], and depth estimation [3]. In this work, we address the problem of depth estimation from DP sensor data. We briefly survey below works on depth estimation from images.

Early work on depth estimation relied on stereo pairs [5] or multi-view geometry [6]. These methods attempt to constrain the solution by assuming that multiple views of the scene of interest are available. Estimating depth from a single image is significantly more ill-posed as the same input image can project to multiple plausible depths. Classic monocular depth estimation methods leveraged cues such as shading [7], contours [8], and texture [9] to infer depth. However, these early methods could be applied only under certain constrained scenarios. Following the seminal monocular depth estimation work of Saxena *et al*. [10], several approaches using hand-crafted features have been proposed [11], [12], [13], [14], [15], [16], [17], [18].

The advent of deep learning saw rapid advancements in the area of monocular depth estimation. This was made possible mainly by end-to-end supervised training [19], [20], [21], [22], [23] on RGBD (RGB depth) datasets. Given the challenges in acquiring large amounts of ground truth depth in varied real-world settings, more recent work has explored the possibility of using synthetic depth data [24], [25], [26], [27], [28] or using self-supervision for training [29], [30], [31], [32], [33], [34]. The idea of self-supervision is to use stereo pairs or video sequences, and train the model to predict per-pixel disparities that map the input image to its nearby view.

Depth from defocus (DFD) [35] is another technique by which the depth of the scene can be recovered. Here, depth is estimated from a stack of images obtained by varying the camera's focus [36], [37]. Levin *et al*. [38] showed that replacing a conventional camera's aperture with a coded aperture enables depth recovery from a single image capture by better exploiting focus cues. Our method too requires a single image capture, and works by explicitly modeling the relation between depth and defocus on a DP sensor using a parameterized PSF.

Defocus blur can act as a complementary cue to disparity for stereo matching [39]. Existing work has incorporated defocus cues into stereo disparity estimation. Rajagopalan et al. [40] combine DFD and stereo matching using a Markov random field-based approach for robust depth estimation. However, their method requires two focal stack images from each stereo view. Bhavsar and Rajagopalan [41] generalize this framework to couple motion, blur, and depth. Methods for disparity estimation from a single pair of defocus stereo images have also been proposed [42], [43]. In comparison, the depth estimation algorithm of Paramanand and Rajagopalan [44] uses a blurred-unblurred image pair. To our knowledge, ours is the first work to explicitly model defocus blur for DP depth estimation. We exploit a symmetry property of the defocus kernels that is not present in a stereo configuration.

Depth can also be estimated using light field cameras [45], [46]. Light field cameras sample the 4D plenoptic function [47] using standard 2D images, and in so doing sacrifice spatial resolution for angular resolution. Due to their low spatial resolution, these cameras have not seen widespread consumer acceptance. A DP camera is also a rudimentary light field. However, the loss of spatial resolution in a DP sensor is minimal – only two angles from the light field are sampled, while light field cameras such as Lytro Illum sample 196 angles at the expense of considerable spatial resolution. Given their utility, this loss of spatial resolution is an acceptable compromise, and as such, DP sensors have been widely adopted on consumer cameras.

Recent work [1], [3] has explored the idea of depth estimation from DP sensor data. Wadhwa *et al*. [1] re-engineered classic stereo matching to estimate the depth map given the two DP views. The *folded loss* in the recently proposed deep learning method of Garg *et al*. [3], while having the advantage of being unsupervised, also models DP disparity as a simple per-pixel shift. This can lead to errors especially in regions far from the focal plane where the effective defocus PSFs vary vastly between the two views. Garg *et al*. [3] noted that their unsupervised loss is inadequate for the task and therefore propose a 3D supervised loss that assumes that the ground truth depth map is available during training. The requirement for ground truth depth poses additional challenges in terms of data acquisition, camera calibration and synchronization. Garg *et al*. [3] designed a custom-made calibrated rig containing five cameras, carefully synchronized capture, and used stereo techniques to estimate the "ground truth" depth map from the five views. We present an unsupervised loss function for depth estimation from DP data that does not require ground truth depth. Our method exploits an inherent symmetry of DP defocus kernels.

3 Dual-pixel image formation and kernel symmetry

In this section, we first examine the relationship between the PSFs corresponding to the left/right DP sub-aperture views and the image for an ideal DP sensor. Based on the image formation model in a DP camera, we describe a useful symmetry property of the left and right DP kernels. Next, we conduct a set of experiments on real data captured using DP cameras to study how the shape of the PSFs deviates in practice from the ideal case. We validate that the symmetry property holds equally well for real DP data.

A DP sensor splits a single pixel in half using two photodiodes housed beneath a microlens as shown in Fig. 2. The left and right photodiodes can detect light and record signals independently. The pixel intensity that is produced when these two signals are summed together is recorded as the final image, and will match the value from a traditional non-DP sensor. Also observe from Fig. 2 that left half-pixels integrate light over the right half of the aperture, and vice versa. We note that this can also be the upper and lower halves of the aperture depending on the sensor's orientation. Without loss of generality, we consider them to be the left and right views in the rest of the paper.

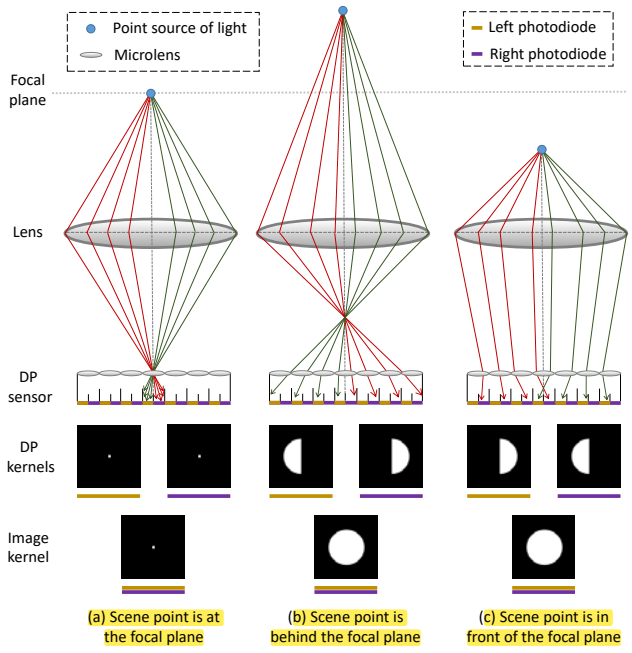


Fig. 2. An illustration of the image formation on an ideal DP camera. (a) A scene point that is at the focal plane produces no disparity – the PSF is an impulse in the left and right DP sub-aperture views as well as the image. (b-c) On the other hand, scene points that are away from the focal plane induce a *defocus-disparity* between the left and right DP views. For an ideal sensor and lens, the circle of confusion resulting from an out-of-focus scene point will be split across the two views leading to *half-circle* PSFs in the left and right views. The disparity is proportional to the blur circle radius, and the *sign* of the disparity can be determined from the direction of the half-circle PSFs, making it possible to disambiguate whether the scene point is behind or in front of the focal plane.

Let us now examine in detail the image formation in an *ideal* DP camera as illustrated in Fig. 2. Consider a scene point that is at the focal plane of the lens (see Fig. 2(a)). Light rays emanating from this point travel through the camera lens and are focused at a single pixel. There is no disparity and the blur kernel is an impulse in the left and right DP views as well as the image. Next, consider the scene points in Figs. 2(b) and (c) that are away from the focal plane. Light rays originating from the scene point in Fig. 2(b) that is behind the focal plane converge at a point in front of the sensor and produce a seven-pixel wide blur on the sensor. The circle of confusion that is induced by this out-of-focus scene point is split over the two DP views producing *half-circle* PSFs as shown. The sum of these two PSFs is equal to the full circle of confusion observed on the image since together, they account for all the light passing through the aperture. In a similar manner, rays from the scene point in Fig. 2(c) that is in front of the focal plane also create a seven-pixel wide blur on the sensor, and the left and right kernels are once again half-circles. While the combined kernel corresponding to the image is exactly the same as in Fig. 2(b), observe that the kernels corresponding to the left and right DP views have swapped positions, making it possible to disambiguate whether the scene point is behind or in front of the focal plane. Thus, the disparity is proportional to the (signed) blur circle radius. The blur circle radius itself is a function of the diameter of the aperture, the

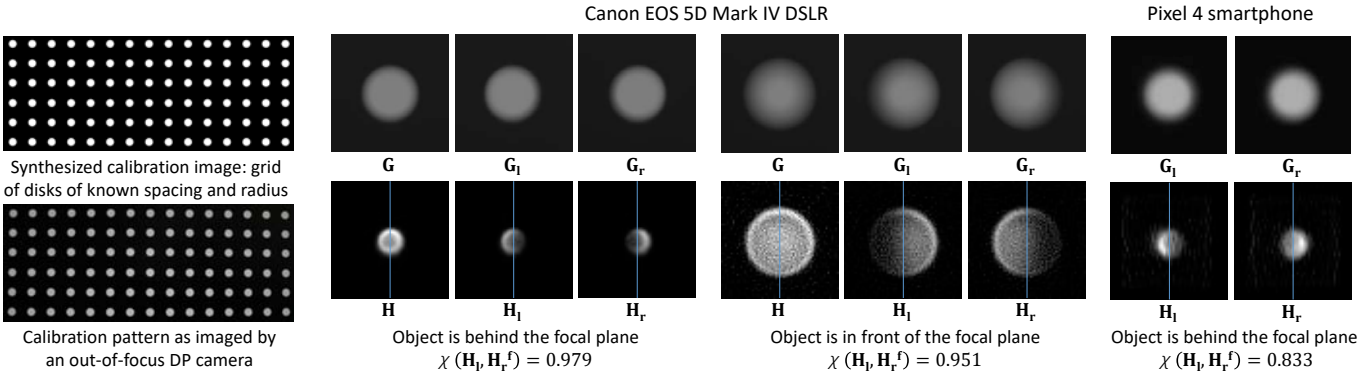


Fig. 3. We estimate "ground truth" kernels corresponding to the image as well as the left and right DP views independently. Using a grid of disks calibration pattern with known radius and spacing, we compute the sharp disk for non-blind kernel estimation. In the second and third columns, patches corresponding to the image, and the left and right DP views are shown for two different focus settings for data captured using the Canon EOS 5D Mark IV DSLR camera. Their associated estimated kernels are also shown in the second row. Notice that the direction of decay of the DP kernels is reversed depending on whether the object is behind or in front of the focal plane. The last column shows left and right DP patches and their corresponding estimated kernels for data captured using the Pixel 4 smartphone. The kernels are of size 61×61 pixels. The normalized 2D cross-correlation value χ between $\mathbf{H}_l$ and $\mathbf{H}_r^f$ is close to unity, indicating that our kernel symmetry property holds true in practice.

focal length of the lens, the focus distance of the camera, and the scene depth [3].

Consider an image patch $\mathbf{G}$ corresponding to a constant-depth region in the scene. Let $\mathbf{G}_l$ and $\mathbf{G}_r$ denote its corresponding left and right DP views, respectively, and $\mathbf{F}$ denote the underlying sharp patch (which is usually unknown). From the DP image formation model, we can write

$$\mathbf{G}_l = \mathbf{F} * \mathbf{H}_l, \quad (1)$$

$$\mathbf{G}_r = \mathbf{F} * \mathbf{H}_r, \quad (2)$$

$$\mathbf{G} = \mathbf{G}_l + \mathbf{G}_r = \mathbf{F} * (\mathbf{H}_l + \mathbf{H}_r) = \mathbf{F} * \mathbf{H}, \quad (3)$$

where $*$ denotes the convolution operation, $\mathbf{H}_l$ and $\mathbf{H}_r$ represent the PSFs that produce $\mathbf{G}_l$ and $\mathbf{G}_r$ when convolved with $\mathbf{F}$, and $\mathbf{H}$, which is the sum of $\mathbf{H}_l$ and $\mathbf{H}_r$, represents the combined kernel corresponding to the image $\mathbf{G}$.

Fig. 2 also reveals an interesting property of the kernels corresponding to the left and right DP views – the left kernel equals the right kernel flipped about the vertical axis (which is the axis perpendicular to the axis of disparity), and vice versa. Mathematically, we can express this as $\mathbf{H}_l = \mathbf{H}_r^f$, where $\mathbf{H}_r^f$ represents the right kernel flipped about the vertical axis. As we shall demonstrate in Section 4, this symmetry property of the kernels plays a central role in the construction of our unsupervised loss function.

3.1 Kernels on a real dual-pixel sensor

The half-circle kernels shown in Fig. 2 correspond to an *ideal* sensor and lens. On a real sensor, due to physical constraints in the positioning of the microlens, depth of the sensor wells, and other manufacturing limitations, it is to be expected that a part of the light ray bundle passing through the left half of the lens will leak into the left-half dual pixels, and vice versa. To investigate how the shape of the kernels deviates from the ideal theoretical case and whether the symmetry of the PSFs still holds, we perform the following experiment on real data captured using DP cameras.

We capture images of calibration patterns, and then estimate the "ground truth" kernels corresponding to the image as well as the left/right DP views using a *non-blind* kernel estimation technique (since the underlying sharp images are known from the calibration patterns). Towards this goal, following [48], we use a grid of disks with a known radius and spacing as the calibration image (see Fig. 3(a)). The calibration pattern is displayed on a 27-inch LED display of resolution 1920×1080. Following [2], we use the Canon EOS 5D Mark IV DSLR camera, which provides access to the sensor's DP data, to capture images. The monitor is placed at a fixed distance of about one meter fronto-parallel to the camera, and the focus distance is varied such that the focal plane is either behind or in front of the monitor introducing different levels of defocus blur. To avoid radial distortion effects at the periphery, we use only 20–30 disks at the center of the image for blur kernel estimation. Patches containing these disks are identified, the center of the disks estimated by finding the centroid of these patches, and the disk patches averaged. We round the estimated centroid coordinates to the nearest integer pixel value to avoid resampling the disks during averaging. The radius of the disks is a known fraction of the distance between disk centers. Thus, following [48], the latent sharp disk image for non-blind kernel estimation can be generated based on the size of the disk grid in the captured image. To estimate the kernel $\mathbf{K}$ from a sharp-blurred image pair $(\mathbf{F}, \mathbf{B})$, we solve the following equation:

$$\arg\min_{\mathbf{K}} \sum_p ||\mathbf{D}_p(\mathbf{F} * \mathbf{K} - \mathbf{B})||_2^2 + \gamma ||\mathbf{K}||_1, \quad \text{subject to } \mathbf{K} \geq \mathbf{0}, \quad (4)$$

where $\mathbf{D}_p \in \{\mathbf{D}_\delta, \mathbf{D}_x, \mathbf{D}_y, \mathbf{D}_{xx}, \mathbf{D}_{yy}, \mathbf{D}_{xy}\}$ denotes the spatial derivatives along the horizontal and vertical axes, and γ is a positive scalar which we set to 1. The l_1-norm encourages the kernel entries to be sparse. Additionally, we impose non-negativity constraint on the entries of $\mathbf{K}$. Note that we do not explicitly enforce that $\mathbf{F}$ and $\mathbf{B}$ have the same average intensity, and so we do not impose the sum to unity constraint while solving for $\mathbf{K}$.

We extract the DP views from the captured data, and estimate kernels *independently* for the image as well as the left and right views. The second and third columns of Fig. 3 show patches and estimated kernels corresponding to the image and the left and right DP views at two different focus distances. As expected, the shape of the DP kernels deviates from the ideal half-circles; the fall-off is gradual. To quantatively evaluate whether the symmetry property holds for real data, we compute normalized 2D cross-correlation χ (which is a commonly-employed metric for kernel similarity [49]) between the left kernel $\mathbf{H}_l$ and the flipped version of the right kernel $\mathbf{H}_r^f$. A value of χ close to one means that the kernels are a close match. As seen from the χ values in Fig. 3, our kernel symmetry assumption is valid even on real data. This can be attributed to sensor imperfections likely affecting both left and right halves in the same manner. We computed the value of χ by varying the focus distance, the aperture, and the focal length, and obtained consistent results.

While DP sensors are used by most modern consumer cameras, DP data is not accessible to users on the vast majority of these devices. This is because after the autofocus operation (which is one of the very early stages of the camera pipeline), the split-pixel data is combined to produce the image. To our knowledge, the Canon EOS 5D Mark IV is one of the few commercially available cameras in the DSLR segment that provide access to the sensor's DP data. Recently, the authors of [3] have released an Android application that allows DP data to be read from the Pixel 3 and Pixel 4 smartphones. We performed the same experiment as described above using DP images captured from a Pixel 4 phone. The result is shown in the last column of Fig. 3. Due to the small lens and sensor size, smartphone images have notably more noise than their DSLR counterparts. Radial distortion is also more pronounced. As a consequence, the kernel estimates are more noisy compared to the Canon DSLR. However, it can be observed that the symmetry of the kernels still holds to a fair extent.

4 Proposed Method

In this section, we first introduce our optimization-based algorithm for depth estimation from DP data. The loss to be minimized is unsupervised, and is constructed based on the kernel symmetry property described in Section 3. We also show that substituting our optimization method with a CNN offers considerable speed-up.

4.1 Unsupervised loss based on dual-pixel kernel symmetry

Using equations (1) and (2), and the associative and commutative properties of convolution, it can be shown that $\mathbf{G}_l * \mathbf{H}_r = \mathbf{G}_r * \mathbf{H}_l$ (see appendix for proof). This relationship [50] has been used for classic depth from defocus [51], [52]. Here, we apply it to the case of DP defocus kernels. Using the kernel symmetry property ($\mathbf{H}_l = \mathbf{H}_r^f$ defined earlier in Section 3), this can be rewritten as $\mathbf{G}_l * \mathbf{H}_r = \mathbf{G}_r * \mathbf{H}_r^f$. Formally, our unsupervised loss function can now be expressed as

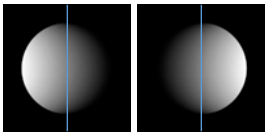


Fig. 4. Our parameterized translating disk kernels corresponding to the left and right DP sub-aperture views.

$$\arg\min_{\mathbf{H}_r} \{\mathbf{G}_l * \mathbf{H}_r - \mathbf{G}_r * \mathbf{H}_r^f\}, \quad \text{subject to } \mathbf{H}_r \geq \mathbf{0}, \sum \mathbf{H}_r = \frac{1}{2}, \tag{5}$$

where non-negativity and sum to half constraints are imposed since $\mathbf{H}_r$ is a blur kernel corresponding to one half of the aperture. The loss function states that the left DP patch blurred with the right DP kernel equals the right DP patch blurred with the *flipped* right DP kernel. To further simplify the optimization, we parameterize the blur kernel as a translating disk (see Fig. 4) such that the (signed) radius of the disk s is the only free parameter to be estimated. Formally, we express the parameterized blur kernel $\mathbf{H}_{r_s}$ as

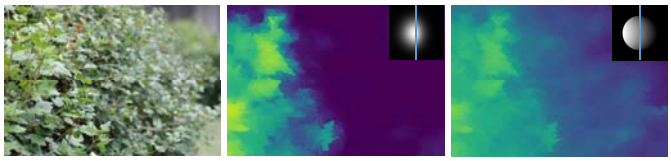
$$\mathbf{H}_{r_s} = \sum_{i=0}^{|2s|} \mathbf{C}\left(\frac{s}{|s|} i, 0\right), \tag{6}$$

where $|.|$ denotes absolute value, and $\mathbf{C}(x, y)$ represents a circular disk of radius $|s|$ centered at pixel coordinates (x, y). We normalize $\mathbf{H}_{r_s}$ to sum to half. Equation (5) can now be rewritten as

$$\arg\min_{s} \{\mathbf{G}_l * \mathbf{H}_{r_s} - \mathbf{G}_r * \mathbf{H}_{r_s}^f\}. \tag{7}$$

Note that the disparity is directly proportional to the signed blur kernel radius. The sign disambiguates whether the depth region corresponding to the patches $\mathbf{G}_l$ and $\mathbf{G}_r$ is in front of or behind the focal plane. As already discussed in Section 1, from DP data, it is only possible to recover this disparity which is related to the actual depth by an affine transformation [3].

Our unsupervised loss of equation (7) holds only for a constant-depth region of the scene. Therefore, given a test image, we adopt a sliding-window approach similar to [1] to estimate the depth map. The details of these steps are provided in Section 4.3.



(a) Input image (b) Depth map using a decaying Gaussian kernel (c) Depth map using the proposed kernel

Fig. 5. A comparison between a decaying Gaussian kernel and our proposed translating disk kernel. Our translating disk kernel more closely models the PSF observed on real DP data, and produces a more accurate depth map. Note that only the image is shown; the left and right DP views that are input to equation (7) have not been presented.

In the literature, defocus PSFs have most commonly been parameterized using a 2D Gaussian function. We also experimented with a *decaying* version of the Gaussian kernel (see Fig. 5) to parameterize $\mathbf{H}_{r_s}$. However, we found from our experiments that our proposed translating disk kernel yields more accurate depth estimates. An example is shown in Fig. 5.

4.2 CNN for faster inference

One limitation of our approach of Section 4.1 is the need to solve an optimization problem at each pixel location. This makes inference slow when performed over the entire image. To improve the runtime performance, we can substitute our optimization algorithm with a CNN.

To generate training data for our CNN, we imaged 12 printed postcards at different focus settings using the Canon EOS 5D Mark IV DSLR camera. Of these, 10 postcards were used for training, while the remaining two were used for validation. The postcards were placed at a fixed distance fronto-parallel to the camera, and the focus setting was varied to introduce different levels of defocus blur. We chose 10 focus settings to image each postcard, giving us a total of 120 images. The focus settings themselves were selected such that an approximately equal number had the focal plane behind and in front of the postcard plane. The left and right DP views were extracted from all 120 images. Patches lying in the central 66% region were cropped from both views (to avoid radial distortion effects) to generate training and validation data. We used patches of size 111×111 pixels. After filtering out homogeneous patches, we generated in total $\sim 17,500$ patches for training and another $\sim 2,100$ patches for validation.

Our network architecture is shown in Fig. 6. The network takes the left and right DP patches, $\mathbf{G}_l$ and $\mathbf{G}_r$, respectively, as input, and outputs the signed radius s of the disk kernel. The ground truth labels (i.e., the value of s) needed for training the network are generated by running our optimization algorithm (equation (7) of Section 4.1) on the 120 images. Note that the labels need to be generated only per image and not per patch. This is because all patches in a given image are at a constant depth from the camera and experience the same amount of defocus blur since the postcards are arranged fronto-parallel to the camera.

Similar to the optimization method of Section 4.1, we adopt a sliding-window approach at test time to recover the depth map of the scene. For comparison, while our optimization approach takes approximately 20 minutes to estimate the disparity values, our CNN takes under 10 seconds for disparity prediction on a 3-megapixel image. Following [1], we apply bilateral space techniques as a final processing step to ensure that the depth map is edge-aligned with the corresponding input image. We provide details of these steps in the next section.

4.3 Edge-aware filtering

Our kernel symmetry assumption holds only for a constant-depth region of the scene. To obtain the depth map given a test image, we apply a sliding-window approach similar to [1]. The patch size, for all experiments in this paper for both our optimization-based approach (Section 4.1) and our CNN-based approach (Section 4.2), is fixed to 111×111 pixels, and the stride is set to 33. Our disparity estimates can be noisy, particularly in homogeneous regions, or near depth boundaries. As a result, we compute a confidence value M for each window as:

$$M = S \times e^{-\beta E}, \tag{8}$$

where S is the average of the horizontal Sobel operator computed over the left and right DP patches, and E is the estimation error $\mathbf{G}_l * \mathbf{H}_{r_s} - \mathbf{G}_r * \mathbf{H}^f_{r_s}$. We use the horizontal Sobel operator because the disparity is along the horizontal axis, and only vertical edges provide meaningful information. While the Sobel value S weights down the confidence in homogeneous regions, the term $e^{-\beta E}$ decreases the confidence around depth boundaries where the error E is expected to be high. Following [1], we use the bilateral solver of [53] to make our estimated depth map smooth and edge-aware. Given the confidence map, the input image, and our estimated depth map, the bilateral solver outputs an edge-aware depth map. We perform guided filtering [54] of this depth map to produce our final output. We follow these same edge-aware filtering steps for both our optimization-based algorithm and our CNN.

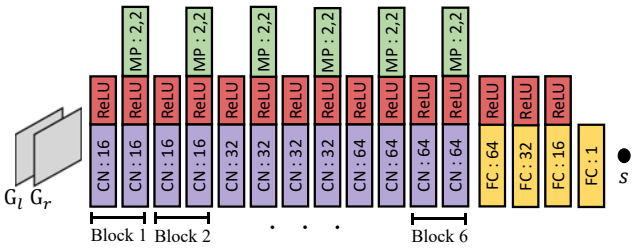


Fig. 6. Our network architecture. The network takes a six-channel input containing the left ($\mathbf{G}_l$) and right ($\mathbf{G}_r$) DP patches (each comprising three color channels) stacked together. CN : z denotes a 2D convolution layer with z filters of size 3×3 and a stride of one. ReLU represents a rectified linear unit activation layer. MP : p, q denotes a 2D max-pool layer with a kernel size p and a stride q. There are 6 blocks of CN, ReLU, and MP. FC : v is a fully connected layer of length v. The network outputs the signed radius value s of the disk kernel.

5 Experiments

To the best of our knowledge, there are no publicly available datasets that provide dual-pixel data with corresponding ground truth depth maps. Therefore, we captured a dataset using a dual-pixel camera to evaluate our algorithm's performance. We perform both quantitative and qualitative evaluation. We use the Canon camera for quantitative analysis since the DSLR provides greater flexibility and control in capturing focal stacks, which we use to obtain the ground truth depth maps. Qualitative comparisons are performed using data from both the Canon DSLR and the Pixel 4 smartphone.

The remainder of this section is organized as follows. We first provide details on CNN training. Next, we discuss comparison methods and the metrics used for evaluation. We then describe how we capture focal stacks and generate the ground truth depth maps for quantitative evaluation. Lastly, qualitative results on DSLR as well as smartphone DP data are presented.

5.1 CNN training and implementation details

We adopt He's weight initialization [55], and use the Adam optimizer [56] to train our model. The initial learning rate is set to 1×10^{-5}, which is decreased by half every 20 epochs. We train our model with minibatches of size 10 using the mean squared error (MSE) loss. Our network is trained using Keras [57] with TensorFlow [58] on an NVIDIA TITAN X GPU. Our model has approximately 176K parameters and is trained for 100 epochs.

We used a fixed value of $\beta = 10^{-6}$ in equation (8). The codes for the bilateral solver [53] and the guided filter [54] have been made publicly available by the authors. For the bilateral solver, the hyper-parameters were chosen as $\sigma_{xy} = 16, \sigma_l = 16, \sigma_{uv} = 8, \lambda = 128$ and 25 iterations of preconditioned conjugate gradient (PCG) for all experiments. We used $r = 10$ and $\epsilon = 10^{-6}$ for the guided filter.

5.2 Comparison with other methods

The work most closely related to ours is the DP depth estimation method of [3]. At the time of submitting this work, neither the dataset nor the trained model/code of [3] is publicly available. As such, we compare our results against the state-of-the-art deep learning-based stereoscopic and monocular depth estimation techniques of [24], [30], [31]. The codes for these three methods have been made available by the authors. Of these, the networks of [30], [31] are trained through self-supervision using stereo pairs, while the method of [24] uses ground truth depth for supervised training. In their paper, the authors of [3] retrained methods such as [30] that are based on stereo self-supervision on the two DP views from their dataset. However, the dataset of [3] is not publicly available. Moreover, our method, by virtue of being unsupervised, does not require the collection of a large dataset. Therefore, we report results using the best performing models for [30], [31] without any retraining. We also compare against the method of Wadhwa *et al*. [1] that retools classic stereo techniques to recover depth from dual pixels. Since their code is not available, we implement their method based on the description in their paper. We use the same parameters recommended by the authors for our experiments on the Pixel smartphone. For DSLR images, the disparity will be higher, and so we increase the tile size as well as the sum of squared differences (SSD) search window (see Section 4.1 of [1]) for optimal performance.

5.3 Error metrics

As mentioned in Section 1, dual-pixel data has a fundamental ambiguity in that depth be recovered only up to an unknown affine transformation [3]. Thus metrics such as mean absolute error (MAE) or root mean squared error (RMSE) that measure error between ground truth and estimated depth in absolute terms cannot be applied. Following [3], we use affine invariant versions of MAE and RMSE, denoted AI(1) and AI(2), respectively. We can define AI(p) as

$$\arg\min_{a,b} \left(\frac{\sum_{(x,y)} |D^*(x,y) - (a\hat{D}(x,y) + b)|^p}{N} \right)^{\frac{1}{p}}, \qquad (9)$$

where N is the total number of pixels, $\hat{D}$ is the estimated disparity, and D^* is the ground truth inverse depth. Note that [3] used a weighted variant AIWE(p) of the above equation, where the weights or confidence values are computed based on the coherence across views in their stereo rig. Since we compute our ground truth depth maps from focal stacks using DFD, we do not include this weighting term in our calculation. AI(2) can be formulated as a straightforward least-squares problem, while AI(1) can be computed using iterative reweighted least squares (we used five iterations in our experiments). We also use Spearman's rank correlation ρ_s, which evaluates ordinal correctness between the estimated depth and the ground truth, for evaluation. See [3] for more details. Once again, we differ from [3] in that they use a weighted variant of Spearman's ρ.

TABLE 1
Accuracy of different methods on our dataset. Lower is better. Best results are shown in bold. The right-most column shows the geometric mean of all the metrics.

Method	AI(1)	AI(2)	$1 - \lvert\rho_s\rvert$	Geometric Mean
CVPR'17 [30]	0.1175	0.1865	0.7454	0.2488
ACM ToG'18 [1]	0.0875	0.1294	0.2910	0.1443
CVPR'18 [24]	0.1082	0.1788	0.6235	0.2250
ICCV'19 [31]	0.1139	0.1788	0.6153	0.2285
Ours (optimization)	**0.0469**	**0.0742**	0.0817	0.0646
Ours (CNN)	0.0481	0.0743	**0.0779**	**0.0645**

5.4 Quantitative evaluation

To quantitatively evaluate our method's performance, we need DP data paired with ground truth depth information. We compute "ground truth" depth maps by applying well-established depth-from-defocus techniques on focal stacks captured using a DP camera. Alternate approaches to obtaining ground truth depth include using direct depth sensors, such as Kinect or LIDAR, or, as in [3], building a custom camera rig and applying multi-view stereo methods. However, both these approaches involve cumbersome registration and synchronization procedures – in the former case, between the DP camera and the direct depth sensors, and in the latter, between the central DP camera and the other stereo cameras in the rig. In contrast, DFD techniques can be applied directly to focal stacks captured using a single camera, and do not require additional cameras or sensors. One limitation of using DFD is that the scene being imaged has to remain unchanged for the entire duration of the capture of the stack. This precludes scenes or objects that are dynamic. While our quantitative evaluations are thereby limited to static scenes, we would like to note that we perform qualitative studies on images captured under unconstrained settings.

We captured 10 focal stacks corresponding to different scenes. We used the Canon EOS 5D Mark IV DSLR camera, which allows fine-grained control of the focus settings, to capture the stacks. Each stack contains between 75 and 90 images. Objects in the scenes were placed between 0.5 m to 2 m to ensure there is interesting nearby depth variation. To make the dataset challenging, we used printed posters (different from the postcards used in Section 4.2 to generate training data for the CNN) with fairly diverse and complex textures as background, while the foreground objects were

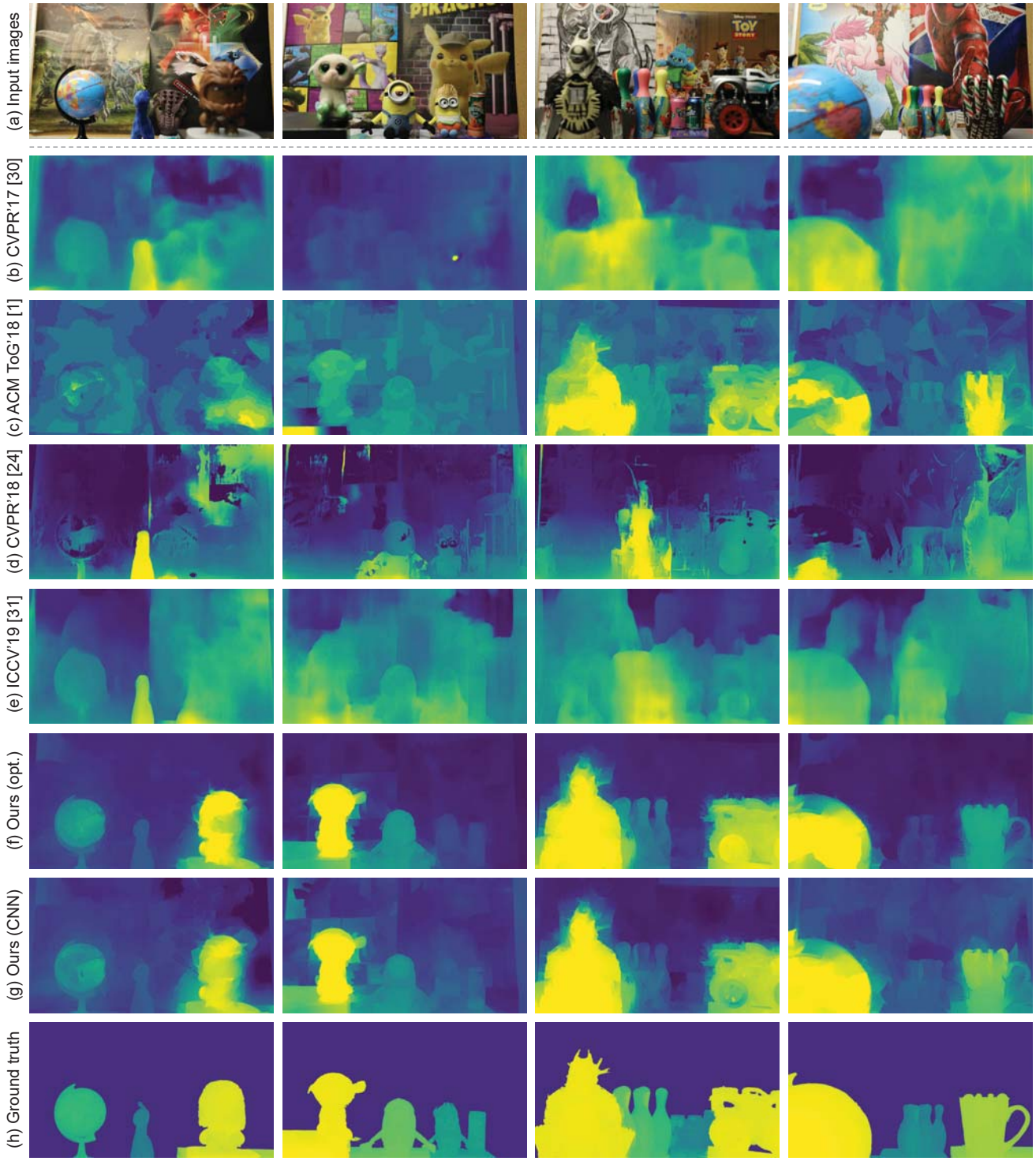


Fig. 7. Input images (a) from our focal stack dataset, the results of the deep learning-based methods of [30] (b), [24] (d), [31] (e), the output of the traditional stereo algorithm of [1] (c), the results of our optimization-based approach (f) and our CNN-based approach (g), and the ground truth (h). An affine transform has been applied to all visualizations to best fit the ground truth. The deep learning-based algorithms [24], [30], [31] do not perform well in general. The stereo approach of [1] fares better in comparison but has many errors in the background. In contrast, both our approaches are able to separate out the foreground objects from the background more accurately. Best seen zoomed-in in an electronic version.

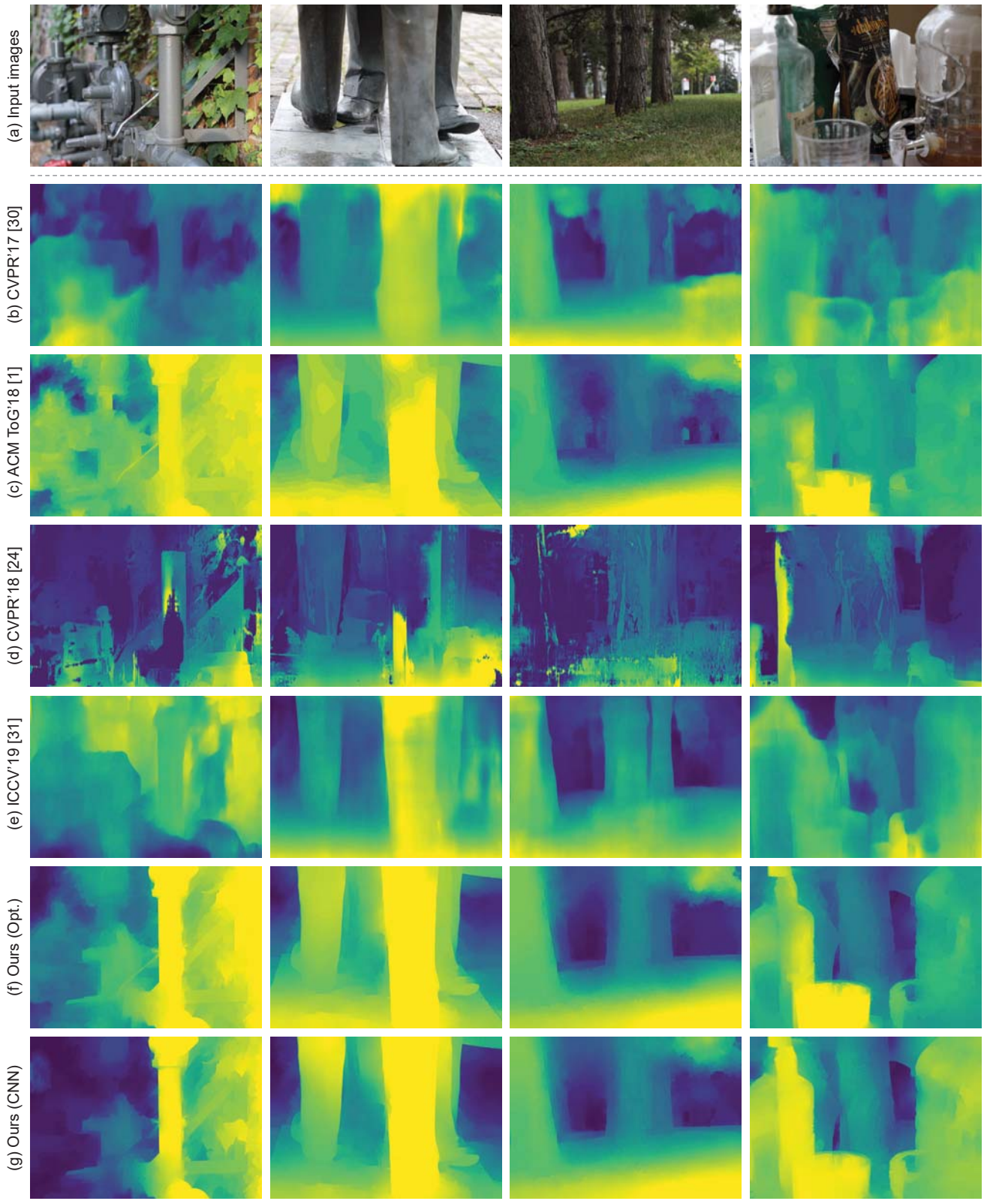


Fig. 8. Qualitative results of our proposed method as well as competing algorithms on data captured using the Canon EOS 5D Mark IV DSLR camera.

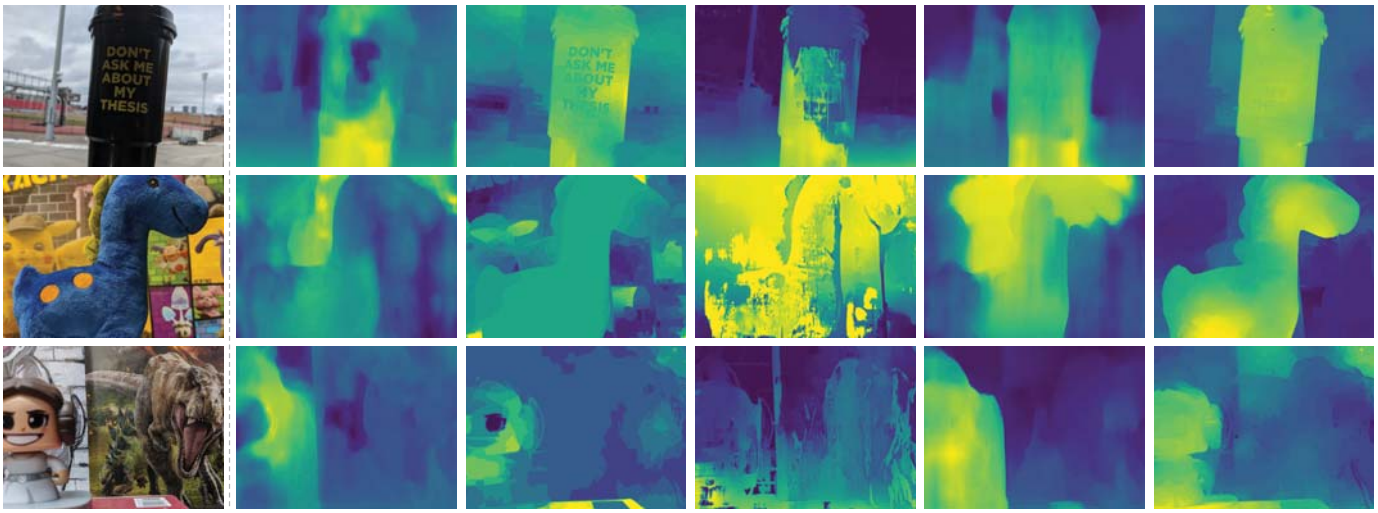


Fig. 9. Qualitative results of our proposed method as well as competing algorithms on data captured using the Pixel 4 smartphone.

designed to have occlusions and clutter. Sample images are shown in the first row of Fig. 7. To estimate the ground truth depth maps from the focal stacks, following [59], we used the commercially available and widely used HeliconSoft software. We noticed that the depth map produced by HeliconSoft tends to have errors in large homogeneous regions. This is a limitation of DFD methods in general since focus operators rely primarily on texture. Therefore, we manually annotated the depth values at these few pixels. The ground truth depth maps obtained from the focal stacks are shown in the last row of Fig. 7.

For quantitative evaluation, we selected 10 images at random from each focal stack for a total of 100 images. Although all images in the stack contain DP information, we extract the left and right DP views only from these selected images. For the purpose of estimating depth maps using DFD, we treat all images as conventional RGB images. The results of our method, as well as comparisons, on a few representative examples from our focal stack dataset are shown in Fig. 7. The deep learning methods of [24], [30], [31] are given the image as input, while [1] and our method take the left and right DP views as input. The methods of [24], [30], [31] perform poorly and fail to distinguish the foreground objects in most cases. The unsupervised stereo technique of [1] also produces erroneous depth estimates particularly in the first two examples. In comparison, our optimization-based and CNN-based approaches generate more accurate depth maps. Quantitative results averaged over the 100 images in our dataset are reported in Table 1. It can be observed that our proposed method outperforms all competing algorithms on all three metrics. While our optimization approach yields better performance in terms of AI(1) and AI(2), our CNN records a slightly lower (better) Spearman's rank correlation.

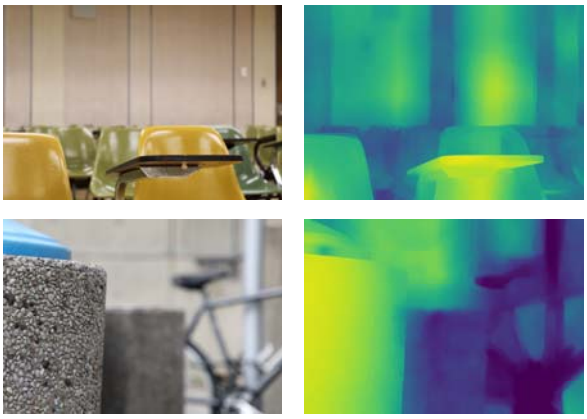
5.5 Qualitative evaluation

Qualitative results of our proposed method as well as competing algorithms on data from the Canon DSLR camera and the Pixel 4 smartphone are provided in Figs. 8 and 9, respectively. It can be seen that our proposed method produces more accurate depth estimates than competing methods. While our focal stack dataset was collected indoors in a controlled environment, the examples in Fig. 8 also include outdoor scenes captured in unconstrained settings. In the Pixel 4, the DP data is embedded in the green channel [3] – that is, the DP views are single-channel as against three color channels in the case of the Canon DSLR. We do not run our CNN method on Pixel data because our network was trained on Canon images with a six-channel input. In general, we found that Pixel data is more challenging, particularly for far-away scenes. This is mainly because the data has higher levels of noise compared to the DSLR (as observed in Fig. 3), and the disparity is lower due to the small aperture size. However, our method still produces fairly good depth estimates as seen from our results in Fig. 9.

Failure cases As with almost all algorithms that use focus/defocus cues for depth inference, our method too relies on the presence of texture. Large homogeneous regions and textureless surfaces that do not contain useful information for defocus estimation can lead to errors. Two failure cases of our method are shown in Fig. 10.

6 Conclusion

This paper has examined the image formation in dual-pixel sensors and proposed a parametric PSF to model the defocus-disparity in the two sub-aperture views. Currently, this DP sensor data can be obtained only on a limited number of devices – namely the Canon EOS 5D Mark IV and the Pixel 3 and 4 cameras. We have shown our model is applicable to both DLSRs with a large aperture and smartphone devices with a small and fixed aperture. Using our parametric model, we described how to leverage the symmetry property between the two corresponding PSFs to formulate an unsupervised loss. We demonstrated the efficacy of this loss in an optimization framework for the task of depth estimation from DP data. Experiments show



(a) Input images (b) Our results

Fig. 10. Example failure cases of our algorithm. Homogeneous depth planes such as the plain background walls in the two images can lead to erroneous depth estimates.

the effectiveness of our PSF formulation of disparity. While a CNN was introduced to speed up our inference, future work aims to use the unsupervised loss to directly train a CNN in an end-to-end manner.

ACKNOWLEDGMENTS

This study was funded in part by the Canada First Research Excellence Fund for the Vision: Science to Applications (VISTA) programme and an NSERC Discovery Grant. Dr. Brown contributed to this article in his personal capacity as a professor at York University. The views expressed are his own and do not necessarily represent the views of Samsung Research.

REFERENCES

[1] N. Wadhwa, R. Garg, D. E. Jacobs, B. E. Feldman, N. Kanazawa, R. Carroll, Y. Movshovitz-Attias, J. T. Barron, Y. Pritch, and M. Levoy, "Synthetic depth-of-field with a single-camera mobile phone," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 64:1–64:13, 2018.

[2] A. Punnappurath and M. S. Brown, "Reflection removal using a dual-pixel sensor," in *Computer Vision and Pattern Recognition*, 2019.

[3] R. Garg, N. Wadhwa, S. Ansari, and J. T. Barron, "Learning single camera depth estimation using dual-pixels," in *International Conference on Computer Vision*, 2019.

[4] A. Abuolaim, A. Punnappurath, and M. S. Brown, "Revisiting autofocus for smartphone cameras," in *ECCV*, 2018.

[5] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1–3, pp. 7–42, 2002.

[6] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[7] B. K. Horn, "Shape from shading: A method for obtaining the shape of a smooth opaque object from one view," Tech. Rep., 1970.

[8] M. Brady and A. Yuille, "An extremum principle for shape from contour," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 288–301, 1983.

[9] R. Bajcsy and L. Lieberman, "Texture gradient as a depth cue," *Computer Graphics and Image Processing*, vol. 5, no. 1, pp. 52 – 67, 1976.

[10] A. Saxena, S. H. Chung, and A. Y. Ng, "Learning depth from single monocular images," in *Neural Information Processing Systems*, 2005.

[11] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3D scene structure from a single still image," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 824–840, 2009.

[12] L. Ladický, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Computer Vision and Pattern Recognition*, 2014.

[13] X. Li, H. Qin, Y. Wang, Y. Zhang, and Q. Dai, "DEPT: Depth estimation by parameter transfer for single still images," in *Asian Conference on Computer Vision*, 2015.

[14] S. Choi, D. Min, B. Ham, Y. Kim, C. Oh, and K. Sohn, "Depth analogy: Data-driven approach for single image depth estimation using gradient samples," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5953–5966, 2015.

[15] J. Konrad, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee, "Learning-based, automatic 2D-to-3D image and video conversion," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3485–3496, 2013.

[16] J. Shi, X. Tao, L. Xu, and J. Jia, "Break ames room illusion: Depth from general single images," *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 225:1–225:11, 2015.

[17] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun, "Dense monocular depth estimation in complex dynamic scenes," in *Computer Vision and Pattern Recognition*, 2016.

[18] C. Hne, L. Ladicky, and M. Pollefeys, "Direction matters: Depth estimation with a surface normal classifier," in *Computer Vision and Pattern Recognition*, 2015.

[19] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Neural Information Processing Systems*, 2014.

[20] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *Computer Vision and Pattern Recognition*, 2018.

[21] J. Li, R. Klein, and A. Yao, "A two-streamed network for estimating fine-scaled depth maps from single rgb images," in *International Conference on Computer Vision*, 2017, pp. 3392–3400.

[22] A. Roy and S. Todorovic, "Monocular depth estimation using neural regression forest," in *Computer Vision and Pattern Recognition*, 2016.

[23] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Transactions on Pattern Analysis and Machince Intelligence*, vol. 38, no. 10, pp. 2024–2039, 2016.

[24] A. Atapour-Abarghouei and T. Breckon, "Real-time monocular depth estimation using synthetic data with domain adaptation," in *Computer Vision and Pattern Recognition*, 2018.

[25] Y. Zou, Z. Luo, and J.-B. Huang, "DF-Net: Unsupervised joint learning of depth and flow using cross-task consistency," in *European Conference on Computer Vision*, 2018.

[26] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox, "What makes good synthetic training data for learning disparity and optical flow estimation?" *International Journal on Computer Vision*, vol. 126, no. 9, pp. 942–960, 2018.

[27] J. N. Kundu, P. K. Uppala, A. Pahuja, and R. V. Babu, "AdaDepth: Unsupervised content congruent adaptation for depth estimation," in *Computer Vision and Pattern Recognition*, 2018.

[28] X. Guo, H. Li, S. Yi, J. Ren, and X. Wang, "Learning monocular depth by distilling cross-domain stereo networks," in *European Conference on Computer Vision*, 2018.

[29] R. Garg, B. V. Kumar, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision*, 2016.

[30] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Computer Vision and Pattern Recognition*, 2017.

[31] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth prediction," in *International Conference on Computer Vision*, 2019.

[32] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Computer Vision and Pattern Recognition*, 2017.

[33] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints," in *Computer Vision and Pattern Recognition*, 2018.

[34] H. Jiang, G. Larsson, M. Maire, G. Shakhnarovich, and E. Learned-Miller, "Self-supervised relative depth learning for urban scene understanding," in *European Conference on Computer Vision*, 2018.

[35] P. Grossmann, "Depth from focus," *Pattern Recognition Letters*, vol. 5, no. 1, pp. 63 – 69, 1987.

[36] S. Suwajanakorn, C. Hernandez, and S. M. Seitz, "Depth from focus with your mobile phone," in *Computer Vision and Pattern Recognition*, 2015.

[37] C. Hazirbas, S. G. Soyer, M. C. Staab, L. Leal-Taixé, and D. Cremers, "Deep depth from focus," in *Asian Conference on Computer Vision*, 2018.

[38] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 70:1–70:9, 2007.

[39] Y. Y. Schechner and N. Kiryati, "Depth from defocus vs. stereo: How different really are they?" *International Journal of Computer Vision*, vol. 39, pp. 141–162, 2000.

[40] A. N. Rajagopalan, S. Chaudhuri, and U. Mudenagudi, "Depth estimation and image restoration using defocused stereo pairs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1521–1525, 2004.

[41] A. V. Bhavsar and A. N. Rajagopalan, "Towards unrestrained depth inference with coherent occlusion filling," *International Journal of Computer Vision*, vol. 97, pp. 167–190, 2011.

[42] F. Li, J. Sun, J. Wang, and J. Yu, "Dual-focus stereo imaging," *Journal of Electronic Imaging*, vol. 19, 2010.

[43] C. Chen, H. Zhou, and T. Ahonen, "Blur-aware disparity estimation from defocus stereo images," in *International Conference on Computer Vision*, 2015.

[44] C. Paramanand and A. N. Rajagopalan, "Depth from motion and optical blur with an unscented Kalman filter," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2798–2811, 2012.

[45] H. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. Tai, and I. S. Kweon, "Accurate depth map estimation from a lenslet light field camera," in *Computer Vision and Pattern Recognition*, 2015.

[46] H. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. Tai, and I. S. Kweon, "Depth from a light field image with learning-based matching costs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 297–310, 2019.

[47] M. Landy and J. A. Movshon, *The Plenoptic Function and the Elements of Early Vision*. MITP, 1991.

[48] F. Mannan and M. S. Langer, "Blur calibration for depth from defocus," in *Conference on Computer and Robot Vision (CRV)*, 2016, pp. 281–288.

[49] Z. Hu and M.-H. Yang, "Good regions to deblur," in *European Conference on Computer Vision*, 2012.

[50] G. Harikumar and Y. Bresler, "Perfect blind restoration of images blurred by multiple filters: theory and efficient algorithms," *IEEE Transactions on Image Processing*, vol. 8, no. 2, pp. 202–219, 1999.

[51] T. Xian and M. Subbarao, "Depth-from-defocus: Blur equalization technique," in *SPIE: Society of Photo-Optical Instrumentation Engineers*, 2006.

[52] H. Tang, S. Cohen, B. Price, S. Schiller, and K. Kutulakos, "Depth from defocus in the wild," in *Computer Vision and Pattern Recognition*, 2017.

[53] J. T. Barron and B. Poole, "The fast bilateral solver," *European Conference on Computer Vision*, 2016.

[54] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *International Conference on Computer Vision*, 2015.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[57] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[58] M. et. al., "TensorFlow: Large-scale machine learning on heterogeneous systems," https://www.tensorflow.org/, 2015.

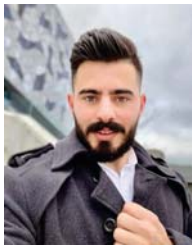
[59] A. Ito, S. Tambe, K. Mitra, A. C. Sankaranarayanan, and A. Veeraraghavan, "Compressive epsilon photography for post-capture control in digital imaging," *ACM Transactions on Graphics*, vol. 33, no. 4, pp. 88:1–88:12, 2014.

APPENDIX

Proof of $\mathbf{G}_l * \mathbf{H}_r = \mathbf{G}_r * \mathbf{H}_l$

$$
\begin{aligned}
\mathbf{G}_l * \mathbf{H}_r &= (\mathbf{F} * \mathbf{H}_l) * \mathbf{H}_r && \text{from equation (1)} \\
&= \mathbf{F} * (\mathbf{H}_l * \mathbf{H}_r) && \text{associative property} \\
&= \mathbf{F} * (\mathbf{H}_r * \mathbf{H}_l) && \text{commutative property} \\
&= (\mathbf{F} * \mathbf{H}_r) * \mathbf{H}_l && \text{associative property} \\
&= \mathbf{G}_r * \mathbf{H}_l && \text{from equation (2)}
\end{aligned}
$$

Abhijith Punnappurath is a Postdoctoral Fellow at the Electrical Engineering and Computer Science department, York University, Toronto, Canada. He received his Ph.D. degree from the Electrical Engineering department, Indian Institute of Technology Madras, India, in 2017. His research interests lie in the areas of low-level computer vision and computational photography. He has worked on face recognition, super-resolution, change detection, white-balancing, reflection removal, and image compression.



Abdullah Abuolaim is a Ph.D. candidate in the Electrical Engineering and Computer Science department at York University, Toronto, Canada. He received his Bachelor of Science in Computer Engineering from Jordan University of Science and Technology in 2015. Then he obtained his Masters of Science in Computer Science from the National University of Singapore in 2017. His research interests include computer vision, computational photography, and deep learning.

Mahmoud Afifi is a Ph.D. candidate at York University, Canada. He obtained an MSc degree and a BSc degree in information technology from Assiut University in Egypt in 2015 and 2009, respectively. His work received two best paper awards at Color and Imaging Conference (2019) and the IEEE International Conference on Mobile Data Management (2018), respectively. His research interests include computer vision and computational photography.

Michael S. Brown is a professor and Canada Research Chair in Computer Vision at York University in Toronto. His research interests include computer vision, image processing and computer graphics. He has served as program chair for WACV 2011/17/19 and 3DV 2015 and as general chair for ACCV 2014 and CVPR 2018/21. Dr. Brown is currently on leave from York University as a senior director at the Samsung AI Centre in Toronto.