

Du²Net: Learning Depth Estimation from Dual-Cameras and Dual-Pixels

Yinda Zhang, Neal Wadhwa, Sergio Orts-Escolano, Christian Haene, Sean Fanello, and Rahul Garg

Google Research

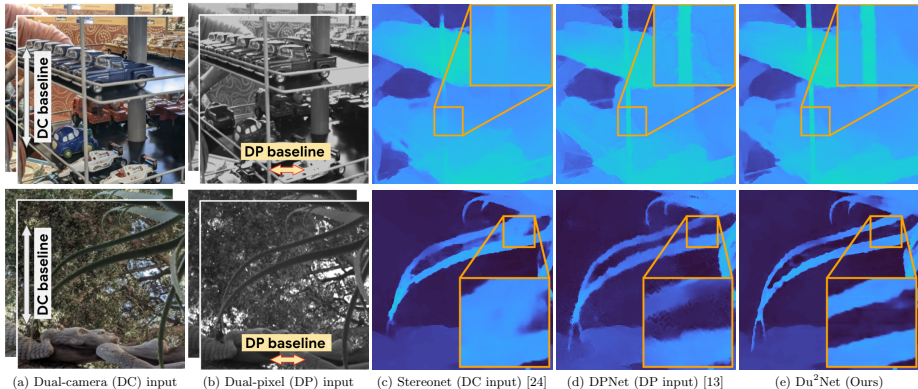


Fig. 1: Du²Net combines dual-camera (DC) and dual-pixel (DP) images to produce edge-aware disparities with high precision even near occlusion boundaries. The large vertical DC baseline complements the small horizontal DP baseline to mitigate the aperture problem (top) and occlusions (bottom).

Abstract. Computational stereo has reached a high level of accuracy, but degrades in the presence of occlusions, repeated textures, and correspondence errors along edges. We present a novel approach based on neural networks for depth estimation that combines stereo from dual cameras with stereo from a dual-pixel sensor, which is increasingly common on consumer cameras. Our network uses a novel architecture to fuse these two sources of information and can overcome the above-mentioned limitations of pure binocular stereo matching. Our method provides a dense depth map with sharp edges, which is crucial for computational photography applications like synthetic shallow-depth-of-field or 3D Photos. Additionally, we avoid the inherent ambiguity due to the aperture problem in stereo cameras by designing the stereo baseline to be orthogonal to the dual-pixel baseline. We present experiments and comparisons with state-of-the-art approaches to show that our method offers a substantial improvement over previous works.

Keywords: Dual-Pixels, Stereo Matching, Depth Estimation, Computational Photography

1 Introduction

Despite their maturity, modern stereo depth estimation techniques still suffer from artifacts in occluded areas, around object boundaries and in regions containing edges parallel to the baseline (the so-called aperture problem). These errors are especially problematic for applications requiring a depth map that is accurate near object boundaries, such as synthetic shallow depth-of-field or 3D photos.

While these problems can be mitigated by using more than two cameras, a recent improvement to consumer camera sensors allows us to alleviate them without any extra hardware. Specifically, camera manufacturers have added dual-pixel (DP) sensors to DSLR and smartphone cameras to assist with focusing. These sensors work by capturing two views of a scene through the camera’s single lens, thereby creating a tiny baseline binocular stereo pair (Fig. 2). Recent work has shown that it is possible to estimate depth from these dense dual-pixel sensors [13, 45]. Due to the tiny baseline, there are fewer occluded areas between the views, and as a result the depth from dual-pixels is more accurate near object boundaries than the depth from binocular stereo. However, the tiny baseline also means that the depth quality is worse than stereo at farther distances due to the quadratic increase in depth error in triangulation-based systems [43].

In this work, we consider a dual-camera (DC) system where one camera has a dual-pixel sensor, a common setup on recently-released flagship smartphones. We propose a deep learning solution to estimate depth from both dual-pixels and dual-cameras. Because depth from dual-cameras and depth from dual-pixels have complementary errors, such a setup promises to have accurate depth at both near and far distances and around object boundaries. In addition, in our setup, the dual-pixel baseline is orthogonal to the dual-camera baseline. This allows us to estimate depth even in regions where image texture is parallel to one of the two baselines (Fig. 1). This is usually difficult due to the well known aperture problem [33].

One key problem that prevents the trivial solution of multi-view stereo matching from working is a fundamental affine ambiguity in the depth estimated from dual-pixels [13]. This is because disparity is related to inverse depth via an affine transformation that depends on the camera’s focus distance, focal length and aperture size, which are often unknown or inaccurately recorded.

To address this issue, we propose an end-to-end solution that uses two separate low resolution learned *confidence volumes*, i.e. the softmax of a negative cost volume, to compute disparity maps from dual-cameras and dual-pixels independently. We then fit an affine transformation between the two disparity maps and use it to resample the dual-pixels’ confidence volume, so that it is in the same space as the dual-cameras’ confidence volume. The two are then fused to estimate a low resolution disparity map. A final edge-aware refinement [24] that leverages features computed from dual-pixels is then used to obtain the final high resolution disparity map.

To train and evaluate our approach, we capture a new dataset using a capture rig containing five synchronized Google Pixel 4 smartphones. Each phone has

two cameras and each capture consists of ten RGB images (two per phone) and the corresponding dual-pixel data from one camera on each phone. We use multi-view stereo techniques to estimate ground truth disparity using all ten views. We plan to release this dataset.

Via extensive experiments and comparisons with state-of-the-art approaches, we show that our solution effectively leverages both dual-cameras and dual-pixels. We additionally show applications in computational photography where precise edges and dense disparity maps are the key for compelling results.

2 Related Work

Stereo matching is a fundamental problem in computer vision and is often used in triangulation systems to estimate depth for various applications such as computational photography [45], autonomous driving [32], robotics [10], augmented and virtual reality [36] and volumetric capture [15]. Traditionally, stereo matching pipelines [40] follow these main steps: matching cost computation, cost aggregation, and disparity optimization, often followed by a disparity refinement (post-processing) step. This problem has been studied for over four decades [29] and we refer the reader to [16, 40, 41] for a survey of traditional techniques.

Recent classical approaches aim at improving the disparity correspondence search by using either global [3, 11, 26, 27] or local [4, 8, 44] optimization schemes. These methods usually rely on hand-crafted descriptors [4, 44] or learned shallow binary features [8, 9] followed by sequential propagation steps [4, 9] or fast parallel approximated CRF inference [8, 44]. However, these methods cannot compete with recent deep learning-based methods that use end-to-end training [5, 7, 23, 24, 28, 42, 46]. Such approaches were introduced by [22, 30], who used encoder-decoder networks for the problems of disparity and flow estimation.

Kendall et al. [23], inspired by classical methods, employed a model architecture that constructs a full cost-volume with 3D convolutions as an intermediate stage and infers the final disparity through a soft-arg min function. Khamis et al. [24] extended this concept by using a learned edge aware refinement step as the final stage of the model to reduce computational cost. More recently, PSMNet [5] used a multi-scale pooling approach to improve the accuracy of the predicted disparities. Finally [47], inspired by [20], used a semi-global matching approach to replace the expensive 3D convolutions.

Other end-to-end approaches use multiple iterative refinements to converge to a final disparity solution. Gidaris et al. [14] propose a generic architecture for labeling problems, such as depth estimation, that is trained end-to-end to predict and refine the output. Pang et al. [37] propose a cascaded approach to learn the depth residual from an initial estimate. Despite this progress, stereo depth estimation systems still suffer from limited precision in occlusion boundaries, imprecise edges, errors in areas with repeated textures, and the aperture problem. The aperture problem can be addressed with two orthogonal dual camera pairs [31]. Occlusions can be reduced by using trinocular stereo [34]. Both of these approaches require additional hardware and more complex calibration.

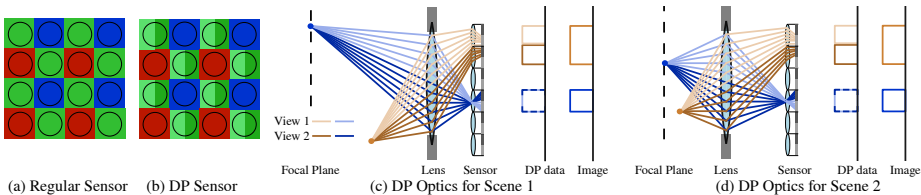


Fig. 2: In a regular Bayer sensor, each pixel has a microlens on top to collect more light (a). Dual-pixel sensors split some of the pixels underneath the microlens into two halves; the green pixels in (b). The two dual-pixel views get their light from different halves of the aperture, resulting in a slight depth-dependent disparity between the views (c). Different scenes can produce the same dual-pixel images if the focus distance changes ((c) vs. (d)). This is a fundamental ambiguity of dual-pixel sensors. (Reproduced with permission from Garg et al. [13].)

In this work, we combine binocular stereo with a dual-pixel sensor, a hardware available in most modern smartphone and DSLR cameras where they are used for autofocus.

Recently, a handful of techniques have been proposed to recover depth from a single camera using dual-pixels [13, 45]. Dual-pixels are essentially a two-view light field [35], providing two slightly different views of the scene. These two views can be approximated as a stereo pair except for a fundamental ambiguity identified by [13] discussed in the introduction. In addition to depth estimation, dual-pixels have been used for dereflection [38].

3 Dual-Pixel Sensors

Dual-pixel sensors work by splitting each pixel in half, such that the left half integrates light over the right half of the aperture and the right half integrates light over the left half of the aperture. Because the two half pixels see light from different halves of the aperture, they form a kind of “stereo pair”, whose centers of projection are in the centers of each half aperture. Since the two half pixel images account for all the light going through the aperture, when they are added together, the full normal image is recovered.

These sensors are becoming increasingly common in smartphone and DSLR cameras because they assist in auto-focus. The reason for this is that the zero-disparity distance corresponds exactly to the image being in-focus (e.g. the blue point in Fig. 2(c)) and disparity is exactly proportional to how much the lens needs to be moved to make the image in-focus. This property also implies that unlike rectified stereo image pairs, the range of disparities can be both negative and positive for DP data.

In addition to this dependence on focus distance, dual-pixels have a number of other key differences from stereo cameras. On the positive side, the dual-pixel views are perfectly synchronized and have the same white balance, exposure and focus, making matching easier. In addition, they are perfectly rectified. This

means that the baseline is perfectly horizontal for a sensor whose dual-pixels are split horizontally as in Fig. 2(b). Another advantage of this perfect rectification is that, dual-pixel sensors are not affected by rolling shutter or optical image stabilization [6], which shifts the principal point and center of projection of a camera. While we need to calibrate for this with stereo cameras, it does not cause problems for dual-pixel images.

Like in a stereo pair, the small baseline of dual-pixel images means that depth estimation at large distances is difficult. Please see the supplementary material for a visualization of the tiny parallax between the DP images. However, it also means there are fewer occlusions and it is possible to get accurate depth near occlusion boundaries in the image. This suggests that a system that combines dual-cameras and dual-pixels could recover depth at short distances and in occluded areas from dual-pixels and depth at larger distances from dual-cameras.

Another difference between dual-pixels and traditional stereo cameras is the interaction between defocus and disparity. Specifically, the amount of defocus is exactly proportional to the disparity between the views. This means that a learned model that makes use of dual-pixels could make use of defocus as well to resolve ambiguities that typically fool matching-based approaches, such as repeated textures.

Finally, we elaborate on the affine ambiguity of depth predictions discussed in the introduction (Fig. 2(c-d)). This happens because the mapping between disparity and depth depends on focus distance which is often inaccurate or unknown in cheap smartphone camera modules. Garg et al. [13] used the paraxial and thin-lens approximation to show that $D_{DP}(x, y) = \alpha + \beta/Z(x, y)$, where $Z(x, y)$ is the depth for pixel x, y ; $D_{DP}(x, y)$ is the dual-pixel disparity at (x, y) , and α and β are constants that depend on the aperture, point spread function, and the focus distance of the lens. Because these can be difficult to determine, inverse depth can be estimated only up to an unknown affine transform.

If there is a second camera in addition to the camera with dual-pixels, such as in our setup, the stereo disparity D_{DC} of the dual-cameras is bf/Z [43] where b is the baseline and f is the focal length. From this, it follows that D_{DC} and D_{DP} are also related via an affine transform

$$D_{DC}(x, y) = \alpha' + \beta' D_{DP}(x, y) \quad (1)$$

We use this observation in our network architecture to effectively integrate stereo and dual-pixel cues.

4 Fusing Dual-Pixels and Dual-Cameras

We describe our deep learning model to predict disparity from both dual-camera and dual-pixel data (Fig. 3). The input to our system is a pair of rectified dual-camera (DC) images, I_l and I_r corresponding to the left and the right cameras, and a pair of dual-pixel (DP) images from the right camera sensor, I_t^{DP} and I_b^{DP} corresponding to the top and bottom half-pixels on the sensor.

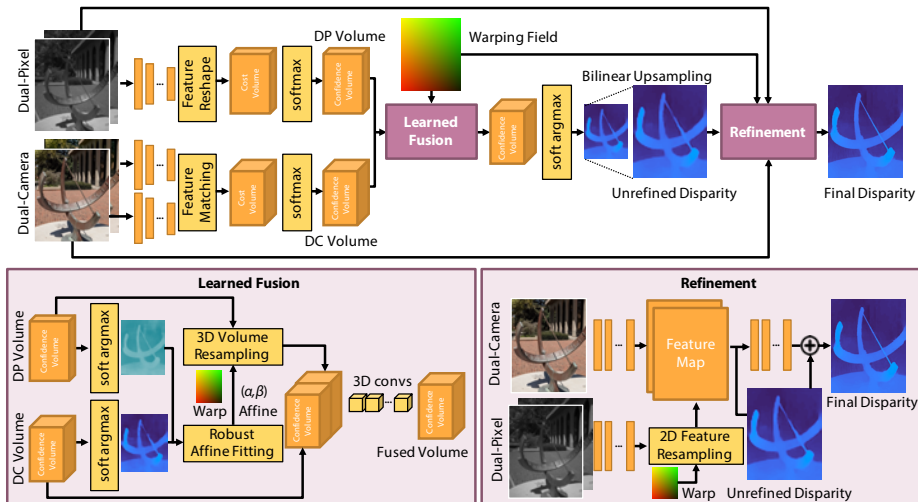


Fig. 3: Overview of Du²Net. Top: two disparity maps are separately inferred from the dual-camera and the dual-pixel branches. An affine transformation is fit between them and used to resample the dual-pixel confidence volume. It is then fused with the dual-camera volume and they are together used to infer the unrefined disparity. An edge-aware refinement step uses the dual-pixel features to predict the final disparity. Bottom: details of the volume fusion and refinement step. See text for more details.

At capture time, the DP images are perfectly aligned with the right DC image. However, after stereo rectification, the left and right dual-camera images are respectively warped by spatial homography transformations $\mathbf{W}_l(x, y)$ and $\mathbf{W}_r(x, y)$, which remaps every pixel to new coordinates in both images. As a result, the right image is no longer aligned with the dual-pixel images. In addition, as explained in Section 3, the two disparity maps coming from DC and DP respectively, are related via an affine transformation. Our method takes both of these issues into account when fusing information from the two sources.

Our model uses two building blocks from other state-of-the-art stereo matching architectures. Specifically, a cost volume [5, 23, 24] and refinement stages [24, 37]. Our main contributions are a method to fuse the confidence volumes (the softmax of the negative cost volume) computed from dual-cameras and dual-pixels, and to show the effectiveness of dual-pixels for refinement. Note that the proposed scheme can be used to give any stereo matching method that uses a cost-volume or that has a refinement stage, the benefits of the additional information in dual-pixels.

Our model consists of three stages, (a) extracting features and building cost volumes from DP and DC inputs independently, (b) building a fused confidence volume by fusing the DP and DC confidence volumes while accounting for the aforementioned spatial warp and affine ambiguity, and (c) a refinement stage

that refines the coarse disparity from the fused confidence volume using features computed from the DP and DC images. We will now explain these in detail.

4.1 Feature Extraction and Cost Volumes

Dual-Camera Cost Volume. Inspired by [24], we create features that are $1/8$ of the spatial resolution in each axis of the original DC images. We do this by running three 2D convolutions with stride 2 followed by six residual blocks [18]. Each convolutional layer uses a kernel of size 3×3 with 32 channels, followed by leaky ReLU activations. The resulting feature map from the left image is warped to the right feature map using multiple disparity hypotheses $d \in [0, 16]$. For each hypothesis, we calculate the distance between the two feature maps. Unlike [24], we use the ℓ_1 distance instead of subtracting the features since this increases the stability of training.

Dual-Pixel Cost Volume. Due to the tiny baseline ($< \sim 1$ mm), the disparities between DP views are usually in the range -4 to 4 pixels. Creating a cost volume with hypotheses that are one pixel apart (at reduced resolution) leads to unstable training. This is because all the possible disparities correspond to the same hypothesis. Instead, we concatenate the two DP images and implicitly produce the cost volume. To do so, we feed the DP images into a 2D network with six residual blocks. Each layer consists of convolutions of size 3×3 with 32 channels followed by leaky ReLU activations. A 2D convolution is attached to the end to produce a feature map with N_d channels, where $N_d = 17$ is the number of desired disparity hypotheses (0.5 pixels per sample from -4 to 4). We then reshape the feature map and convert it to a 3D volume by expanding the final dimension.

4.2 Fused Confidence Volume

It is not straightforward to merge the DP and DC cost volumes due to the rectification warp \mathbf{W}_r between the DP and DC images and the affine ambiguity in DP disparity (Sec 3). In addition, the costs in the two volumes may be scaled differently since they are predicted from different network layers.

We first normalize the two cost volumes to the range $[0, 1]$ by applying softmax to the negative cost volume along the disparity dimension. We call the resulting tensors confidence volumes. To handle the affine ambiguity, we first predict disparity maps D_{DP} and D_{DC} from the two confidence volumes using a soft-argmax operator [23], and fit an affine transformation between the two by solving a Tikhonov-regularized least squares problem that biases the solution to be close to $\alpha = 0, \beta = 1$:

$$\hat{\alpha}, \hat{\beta} = \underset{\alpha, \beta}{\operatorname{argmin}} \left\| (\alpha + \beta \cdot D_{DP}) - D_{DC} \right\|^2 + \gamma \|\beta - 1\|^2 + \gamma \|\alpha\|^2, \quad (2)$$

where $\gamma = 0.1$ is a regularization constant. Then, we use the known rectification warp $\mathbf{W}_r(x, y) = [W_r^x(x, y), W_r^y(x, y)]$ and the estimated affine transformation to warp the DP confidence volume into the DC space:

$$C_{warp}^{DP}(x, y, z) = C^{DP} \left(W_r^x(x, y), W_r^y(x, y), (z - \hat{\alpha})/\hat{\beta} \right), \quad (3)$$

where C^{DP} is the confidence volume built from DP images. The 3D warping uses differentiable bilinear interpolation and zero padding.

The warped DP confidence volume C_{warp}^{DP} is now normalized and aligned with the DC confidence volume. We stack these together to form a 4D tensor that is fused into a 3D confidence volume by a shallow network consisting of three layers of 3D convolutions with leaky ReLU activations and a softmax at the end. Finally, the fused volume is converted into a disparity map D_{unref} using a soft-arg max.

4.3 Disparity Refinement

The next step is to refine the low resolution disparity D_{unref} . Khamis et al. [24] use the RGB image as the guide image to upsample the disparity while applying a learned residual to improve edges and minimize the final error. A straightforward extension of [24] is to warp the DP images using the rectification warp \mathbf{W}_r and use them along with the RGB image as the guide image. However, we find that this yields inferior results compared to our method, presumably because warping and resampling makes it harder to extract disparity cues from DP images.

Instead, we extract features from the input DP images and then warp them using \mathbf{W}_r . This way, the model can easily extract disparity cues from the perfectly rectified DP image pair. The warped features are concatenated with the features extracted from the right RGB image and the unrefined disparity D_{unref} . These are fed into six residual blocks, with 3×3 convolutions followed by a leaky ReLU activation, to predict a residual R . The final output is set to $D_{ref} = D_{unref} + R$.

4.4 Loss Function

To train our network we use a weighted Huber loss [21]:

$$L(D) = \frac{\sum_{\mathbf{p}} \mathcal{H}(D(\mathbf{p}) - D^{gt}(\mathbf{p}), \delta) \cdot C^{gt}(\mathbf{p})}{\sum_{\mathbf{p}} C^{gt}(\mathbf{p})}, \quad (4)$$

where D^{gt} is the ground truth disparity, C^{gt} is the per-pixel confidence of the ground truth, and δ is the switching point between the quadratic and the linear function, which is set to 1 for disparity in range $[0, 128]$. The overall loss is a weighted sum of four terms:

$$L_{total} = \lambda_{DP} L(\hat{\alpha} + \hat{\beta} \cdot D_{DP}) + \lambda_{DC} L(D_{DC}) + \lambda_{unref} L(D_{unref}) + \lambda_{ref} L(D_{ref}), \quad (5)$$

where λ_{DC} is set to 10 and the other weights are set to 1.

5 Evaluation

In this section, we perform extensive experiments to evaluate our model. We conduct an ablation study to show the effectiveness of our design choices. We also compare to other stereo and dual-pixel methods. We focus our experiments on thin structures, edges and occlusion boundaries, to show the effectiveness of the complementary information coming from DP and DC data. For quantitative evaluations we report MAE, RMSE and the *bad δ* metric, i.e. the percentage of pixels with disparity error greater than δ . These are weighted by the ground-truth confidence. See the supplementary material for details.

5.1 Data Collection

We collect a new data set using the Google Pixel 4 smartphone, which has a dual camera system consisting of a main camera with a dual-pixel sensor and a regular telephoto camera. We refer to the main camera as the right camera and the telephoto camera as the left camera. We use a data acquisition set up similar to [13], i.e., a capture rig consisting of 5 phones (Fig. 4a) synchronized with [2]. Structure from motion [17] and multi-view stereo techniques are used to generate depth maps. Similar to [13], we also compute a per-pixel confidence for the depth by checking for depth coherence with neighboring views. The center phone in the rig is used for training and evaluation since its depth quality is higher than other views especially for occluded regions.

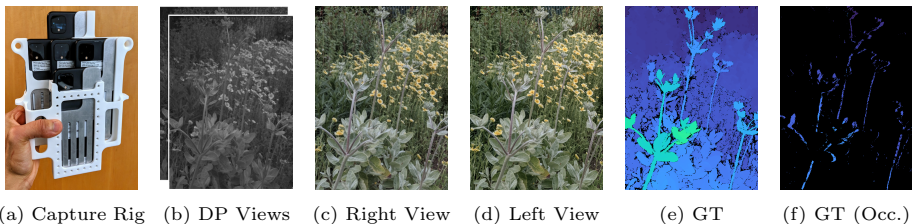


Fig. 4: Our capture rig (a) similar to [13] but with phones that can capture both dual-pixel (b) and dual-camera (c, d) data. The left and right views are rectified, and the ground truth disparity (e) corresponding to the right view is computed using multi-view stereo techniques on all 10 views captured by the rig. Low confidence depth samples are rendered in black. The multitude of views ensures that we have good quality depth in regions that are occluded in the left view. (f) shows the GT depth masked to regions that are occluded in the left view.

We rectify the stereo images from the center phone using estimated camera poses [12]. The estimated depth is converted to disparity and then rectified along with confidences to yield D_l^{gt}, C_l^{gt} and D_r^{gt}, C_r^{gt} for the left and right images respectively. As described in Section 4, dual-pixel images from the main (right) camera are not rectified or warped. Instead, we store the warp map \mathbf{W}_r that is

DP in Confidence Volume					DP in Refinement				
Conf. Volume	MAE	RMSE	$\delta > 2$	$\delta > 3$	Refinement	MAE	RMSE	$\delta > 2$	$\delta > 3$
DC	1.023	2.502	10.65	6.32	RGB	0.838	2.197	7.74	4.55
DP+DC (2D)	0.969	2.423	9.72	5.79	DP (I)	0.835	2.173	7.75	4.54
DP+DC (C)	0.964	2.372	9.79	5.80	DP	0.829	2.184	7.51	4.45
DP+DC (Ours)	0.889	2.263	8.78	5.18	RGB+DP (Ours)	0.802	2.147	7.17	4.25

Table 1: Ablation Study. Left: We compare different ways of fusing DP with the DC confidence volume. ‘(2D)’ indicates fusion of the 2D disparity maps extracted from the two confidence volumes. ‘(C)’ indicates fusing cost volumes instead of confidence volumes. Right: We compare the different ways of using DP to refine the best unrefined disparity from the left and show evaluation on the final disparity. ‘(I)’ indicates that input DP images are warped before computing features for refinement.

needed to warp and align the DP images with the rectified right camera image (see Section 4). In addition, to evaluate the quality of the estimated disparity in regions that are visible in only one of the cameras in the stereo pair, we also compute C_i^{occ} for $i \in l, r$, i.e., a per-pixel confidence indicating that the ground-truth disparity is correct but the pixel is occluded in the other camera. In total, we collect 3308 training examples and 1077 testing examples. Please refer to supplementary materials for details of the calculation of C_i^{occ} , data collection and ground truth calculation.

5.2 Training Scheme

We use Tensorflow [1] to implement the network and train using Adam [25] for 2 million iterations with a batch size of 1. The learning rate is set to 3×10^{-5} and then reduced to 3×10^{-6} after 1.5 million iterations. Training takes roughly 16 hours using 8 Tesla V100 GPUs. Inputs to the network I_l , I_r , and \mathbf{W}_r are resized to match the resolution of the predicted and the ground truth disparity, i.e., 448×560 . DP images I_l^{DP} and I_b^{DP} are of size 1000×1250 .

5.3 Ablation Study

We evaluate the effect of each component of the model. In particular we focus on the impact of dual-pixels on the fused volume and the refinement stage. We provide quantitative comparisons (Tab. 1) and qualitative comparisons (Fig. 5).

Dual-pixels in the confidence volume. Tab. 1 (left) shows the error of the unrefined disparity D_{unref} using different fusing strategies for the cost volume. Our method of merging the DP and DC volumes (DP+DC) significantly outperforms using only the DC cost volume. We also compare to fusing the 2D disparity maps instead of the the 3D confidence volumes. Specifically, we concatenate D_{DP} and D_{DC} after the affine transformation and use a 2D neural network with six residual blocks to predict D_{unref} . This is worse than our method according to all

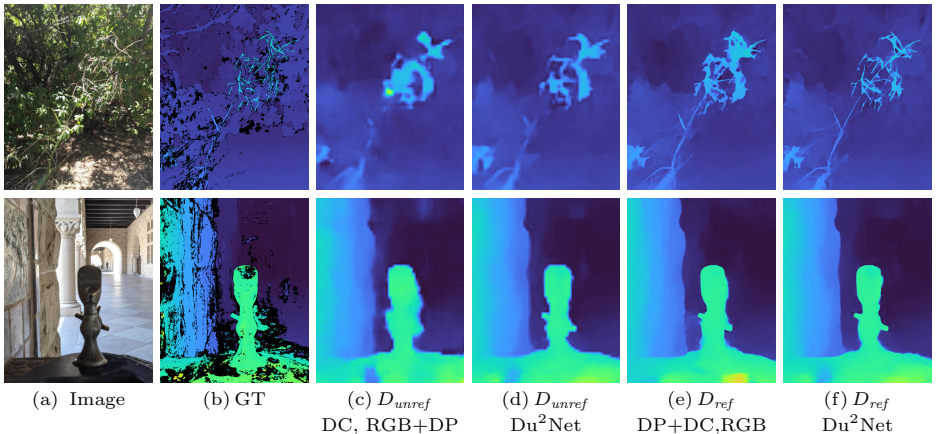


Fig. 5: Ablations of our method. The right camera image (a), ground truth disparity (b) with low confidence disparity in black, D_{unref} (c) from an ablation where only the DC input is used for the confidence volume, D_{unref} (d) from Du²Net, D_{ref} (e) from an ablation where only the RGB image is used for refinement, and D_{ref} (f) from Du²Net. DP input is useful for both the confidence volume and refinement stages to recover accurate depth for fine structures and occluded regions.

metrics (DP+DC (2D) vs DP+DC). Finally, we evaluate fusing cost volumes instead of confidence volumes. This, DP+DC (C), is also inferior to DP+DC.

Qualitative comparisons are shown in Fig. 5 (c) and (d). Compared to a DC only cost volume (c), fusing DP into the cost volume (d) adds more details to the unrefined disparity and prevents errors at object boundaries. This is critical for getting high quality disparities since the refinement is only able to make local adjustments to the disparity and cannot fix large errors.

Dual-pixels in refinement. We show that our method of using dual-pixels in the refinement stage is better than several baselines (Tab. 1 (right)). We use the fused DP + DC confidence volume for all cases and extract RGB and DP features for refinement using networks with the same capacity for a fair comparison. Using DP for refinement is better than just using the right RGB image. However, results are best when both are used (RGB+DP). Notably, if the DP is warped before feature extraction (instead of after), performance (DP (I) in Tab. 1) is not better than using only the RGB image. This suggests that DP cues are not effective after warping, and it is important to extract features before warping.

Qualitative comparisons are shown in Fig. 5 (e) and (f). While both methods use DP and DC to compute the unrefined disparity, (e) uses only the right RGB image for refinement, and (f) uses RGB and DP for refinement. Even though DP increases the quality of the unrefined disparity, using it during refinement further improves depth quality at thin structures and near object boundaries. This indicates that it is important to use DP for both the cost volume and the refinement to achieve the best performance.

Method	Input	All Pixels				Occluded Pixels			
		MAE	RMSE	$\delta > 2$	$\delta > 3$	MAE	RMSE	$\delta > 2$	$\delta > 3$
GA-Net [47]	DC	1.001	2.425	9.31	6.09	6.068	8.386	69.81	60.07
PSM-Net [5]	DC	0.815	2.289	7.98	4.97	2.799	5.188	34.73	26.78
StereoNet [24]	DC	0.935	2.432	9.07	5.41	3.123	5.632	38.13	28.49
Du ² Net(ours)	DC + DP	0.802	2.147	7.17	4.25	2.396	4.543	30.62	21.91
DPNet* [13]	DP	1.090	1.989	12.60	5.80	2.594	4.307	38.14	26.18
Du ² Net* (ours)	DC + DP	0.746	1.825	6.63	3.63	2.352	4.373	32.65	21.77

Table 2: Quantitative comparisons to the state-of-the-art. Note how the proposed approach substantially outperforms the StereoNet baseline [24], and GA-Net [13]. The method is on par with the more sophisticated PSM-Net [5] and it outperforms all the competitors in occluded regions. ‘*’ indicates a final affine transformation applied to the output disparity and the best results for these methods are highlighted separately (see text for details).

5.4 Comparison to State-of-the-art Methods

We compare to other stereo and dual-pixel depth estimation approaches (Tab. 2). All methods are trained on our data using code provided by the authors. We modified the original loss functions to use the confidence maps C_r^{gt} from our dataset in the same way that these are used in our method (as a per pixel weight). Additionally, for the stereo methods we compare to, we set the maximum disparity range to 128. The baseline methods were trained until convergence on the test set error. We used the hyper-parameters and training optimization strategies (including annealing schedules for learning rates) provided in the original implementations.

For stereo baselines, we compare to StereoNet [24] which is similar to our model with only the DC input, PSMNet [5] and GANet [47]. PSMNet [5] uses multi-scale feature extraction and cost volumes, and GANet [47] uses a sophisticated semi-global aggregation [47]. In Tab. 2, we report quantitative results. Our model uses a low-resolution cost volume and a refinement stage, with a runtime comparable with StereoNet, while achieving accuracy that is on par with or higher than more computationally expensive models, such as PSMNet (which uses 25 3D convolutional layers as opposed to our 8).

For DP input baselines, DPNet [13] predicts disparity *up to an unknown affine transformation*. To handle this, like [13], we find the best fit (according to MSE) affine transformation between the prediction and the ground truth and transform the prediction to compute the metrics. For a fair comparison, we apply the same post processing to our method (Du²Net* in Tab. 2), showing it consistently outperforms DPNet.

As mentioned in Sec. 5.1, we also compute an occlusion mask C_r^{occ} and evaluate the methods only on these pixels. The results are reported in Tab. 2 under ‘‘Occluded pixels’’. Our method outperforms the other competitors by a substantial margin in those areas showing the advantage of small baseline DP data.

Qualitative comparisons are provided in Fig. 6. Note how we better capture fine details and small structures, while correctly inferring disparity near occlusion boundaries. The orthogonal baselines of the dual-pixels and dual-cameras

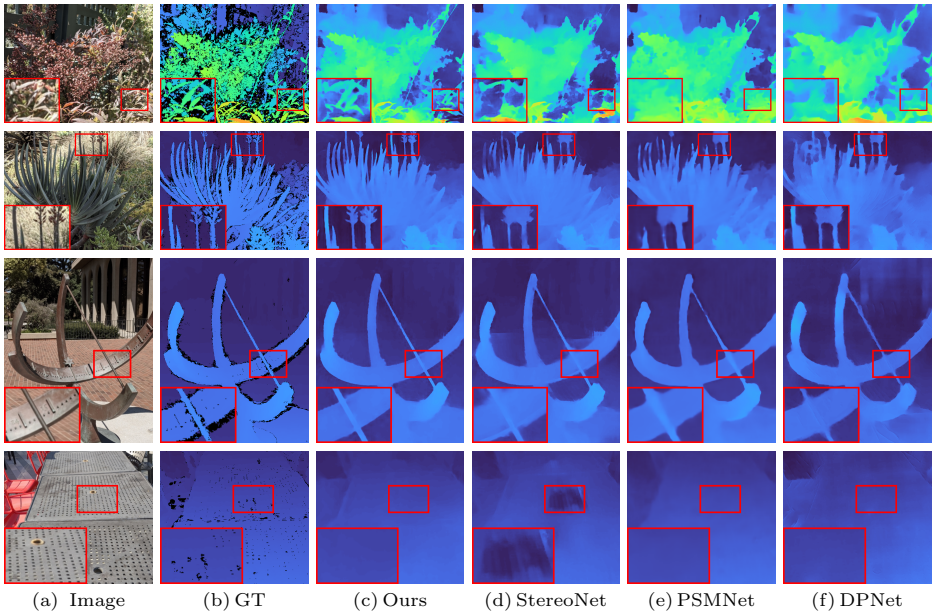


Fig. 6: Qualitative comparison to the state-of-the-art. Right camera image (a) from our test set, ground truth disparity with low confidence disparity in black (b), and results from our method (c), stereo only methods StereoNet [24] (d) and PSMNet [5] (e), and DP only method DPNet [13] (f). Stereo only methods fail for vertical structures due to the aperture problem, e.g., in the second image. They also fail in regions with fine structures and occlusions, e.g., in the first three images. StereoNet fails on the last image potentially due to repeated texture. DPNet’s accuracy falls quickly with distance due to the small dual-pixel baseline. Our method overcomes these problems by fusing the two cues.

also helps mitigate the aperture problem and issues due to repeated textures. Additional results are available in the supplementary material.

5.5 Applications in Computational Photography

Predicting accurate disparities, hence depth, is crucial for many applications in computational photography. These applications usually require accurate depth for fine structures and near occlusion boundaries. Fig. 7 and 8 show how our more accurate depth leads to fewer artifacts when used to produce synthetic shallow depth-of-field images and 3D photos [19] respectively.

6 Discussion

We presented the first method to combine dual-camera and dual-pixel data. The inherent affine ambiguity of disparity computed from dual-pixel images prevents

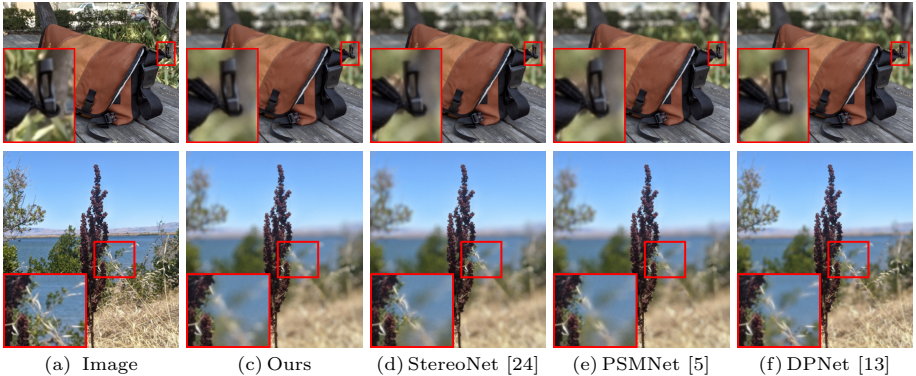


Fig. 7: Synthetic shallow depth-of-field results for different methods. Top: Accurate depth near occlusion boundaries is critical for avoiding artifacts near the subject boundary. Bottom: DPNet [13] is unable to resolve the small depth difference between the flower and the twigs in the background. As a result, parts of the background are incorrectly sharp.

a straightforward integration of the two modalities. Therefore, we proposed a novel solution that resamples the confidence volume computed from dual-pixels and concatenates it with the dual-camera volume. A refinement stage leverages dual-pixels to infer the final disparity map. We show the effectiveness of the proposed solution with experiments, comparisons to the state-of-the-art and applications. Our dataset will be released publicly and we hope it can advance the field. While the orthogonality of the baselines allows us to avoid the aperture problem, our method doesn't work on textureless regions. Perhaps this could be handled by combining information from additional modalities like active depth sensors. Another interesting direction for future work would be to consider dual-camera pairs where both cameras have dual-pixels.

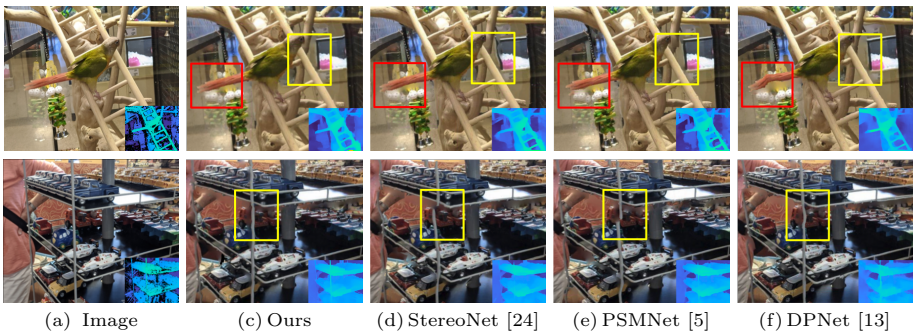


Fig. 8: 3D photos results [19]. Novel views of the scene are rendered by warping the image according to the estimated depth to new camera positions. Depth errors lead to unnatural distortion of rigid scene structures in the novel views.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), <https://www.tensorflow.org/>
2. Ansari, S., Wadhwa, N., Garg, R., Chen, J.: Wireless software synchronization of multiple distributed cameras. In: ICCP (2019)
3. Besse, F., Rother, C., Fitzgibbon, A., Kautz, J.: PMBP: patchmatch belief propagation for correspondence field estimation. IJCV (2014)
4. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo-stereo matching with slanted support windows. In: BMVC (2011)
5. Chang, J., Chen, Y.: Pyramid stereo matching network. In: CVPR (2018)
6. DiVerdi, S., Barron, J.T.: Geometric calibration for mobile, stereo, autofocus cameras. In: WACV (2016)
7. Duggal, S., Wang, S., Ma, W.C., Hu, R., Urtasun, R.: DeepPruner: learning efficient stereo matching via differentiable patchmatch. In: ICCV (2019)
8. Fanello, S.R., Valentin, J., Kowdle, A., Rhemann, C., Tankovich, V., Ciliberto, C., Davidson, P., Izadi, S.: Low compute and fully parallel computer vision with hashmatch. In: ICCV (2017)
9. Fanello, S.R., Valentin, J., Rhemann, C., Kowdle, A., Tankovich, V., Davidson, P., Izadi, S.: UltraStereo: efficient learning-based matching for active stereo systems. In: CVPR (2017)
10. Fanello, S., Pattacini, U., Gori, I., Tikhonoff, V., Randazzo, M., Roncone, A., Odone, F., Metta, G.: 3d stereo estimation and fully automated learning of eye-hand coordination in humanoid robots. In: Humanoids (2014)
11. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. IJCV (2006)
12. Fusiello, A., Trucco, E., Verri, A.: A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications* (2000)
13. Garg, R., Wadhwa, N., Ansari, S., Barron, J.T.: Learning single camera depth estimation using dual-pixels. In: ICCV (2019)
14. Gidaris, S., Komodakis, N.: Detect, replace, refine: Deep structured prediction for pixel wise labeling. In: CVPR (2017)
15. Guo, K., Lincoln, P., Davidson, P., Busch, J., Yu, X., Whalen, M., Harvey, G., Orts-Escolano, S., Pandey, R., Dourgarian, J., Tang, D., Tkach, A., Kowdle, A., Cooper, E., Dou, M., Fanello, S., Fyffe, G., Rhemann, C., Taylor, J., Debevec, P., Izadi, S.: The relightables: Volumetric performance capture of humans with realistic relighting. TOG (2019)
16. Hamzah, R.A., Ibrahim, H.: Literature survey on stereo vision disparity map algorithms. *Journal of Sensors* (2016)
17. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press (2003)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
19. Hedman, P., Kopf, J.: Instant 3D Photography. SIGGRAPH (2018)
20. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. TPAMI (2008)
21. Huber, P.J.: Robust estimation of a location parameter. In: *Breakthroughs in statistics*, pp. 492–518. Springer (1992)

22. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: CVPR (2017)
23. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: CVPR (2017)
24. Khamis, S., Fanello, S., Rhemann, C., Kowdle, A., Valentin, J., Izadi, S.: Stereonet: Guided hierarchical refinement for real-time edge-aware depth prediction. In: ECCV (2018)
25. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
26. Klaus, A., Sormann, M., Karner, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: ICPR (2006)
27. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: ICCV (2001)
28. Liang, Z., Feng, Y., Guo, Y., Liu, H., Chen, W., Qiao, L., Zhou, L., Zhang, J.: Learning for disparity estimation through feature constancy. In: CVPR (2018)
29. Marr, D., Poggio, T.: Cooperative computation of stereo disparity. *Science* (1976)
30. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: CVPR (2016)
31. Meier, L., Honegger, D., Vilhjalmsón, V., Pollefeys, M.: Real-time stereo matching failure prediction and resolution using orthogonal stereo setups. In: ICRA (2017)
32. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: CVPR (2015)
33. Morgan, M., Castet, E.: The aperture problem in stereopsis. *Vision research* (1997)
34. Mulligan, J., Isler, V., Daniilidis, K.: Trinocular stereo: A real-time algorithm and its evaluation. *IJCV* (2002)
35. Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P.: Light field photography with a hand-held plenoptic camera. Tech. rep., Stanford University (2005)
36. Orts-Escobedo, S., Rhemann, C., Fanello, S., Chang, W., Kowdle, A., Degtyarev, Y., Kim, D., Davidson, P.L., Khamis, S., Dou, M., Tankovich, V., Loop, C., Cai, Q., Chou, P.A., Mennicken, S., Valentin, J., Pradeep, V., Wang, S., Kang, S.B., Kohli, P., Lutchyn, Y., Keskin, C., Izadi, S.: Holoportation: Virtual 3d teleportation in real-time. In: UIST (2016)
37. Pang, J., Sun, W., Ren, J., Yang, C., Yan, Q.: Cascade residual learning: A two-stage convolutional neural network for stereo matching. In: ICCV Workshop (2017)
38. Punnappurath, A., Brown, M.S.: Reflection removal using a dual-pixel sensor. In: CVPR (2019)
39. Richardt, C., Orr, D., Davies, I., Criminisi, A., Dodgson, N.A.: Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In: ECCV (2010)
40. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV* (2002)
41. Sinha, S.N., Scharstein, D., Szeliski, R.: Efficient high-resolution stereo matching using local plane sweeps. In: CVPR (2014)
42. Song, X., Zhao, X., Hu, H., Fang, L.: EdgeStereo: A context integrated residual pyramid network for stereo matching. In: ACCV (2018)
43. Szeliski, R.: *Computer Vision: Algorithms and Applications*. Springer-Verlag, Berlin, Heidelberg, 1st edn. (2010)
44. Tankovich, V., Schoenberg, M., Fanello, S.R., Kowdle, A., Rhemann, C., Dzitsiuk, M., Schmidt, M., Valentin, J., Izadi, S.: Sos: Stereo matching in $O(1)$ with slanted support windows. In: IROS (2018)

45. Wadhwa, N., Garg, R., Jacobs, D.E., Feldman, B.E., Kanazawa, N., Carroll, R., Movshovitz-Attias, Y., Barron, J.T., Pritch, Y., Levoy, M.: Synthetic depth-of-field with a single-camera mobile phone. In: SIGGRAPH (2018)
46. Yang, G., Zhao, H., Shi, J., Deng, Z., Jia, J.: SegStereo: exploiting semantic information for disparity estimation. In: ECCV (2018)
47. Zhang, F., Prisacariu, V.A., Yang, R., Torr, P.H.S.: GA-Net: guided aggregation net for end-to-end stereo matching. In: CVPR (2019)

Supplementary Materials

In this supplementary material, we provide more information for data collection, implementation details, and quantitative and qualitative results.

A Data Collection

In this section, we provide information about our data capture rig and how we obtain the ground truth disparity, confidence, and the occlusion mask.

A.1 Data Capture



Fig. 9: Example capture from our data collection rig. Top: Views from the main cameras of the five phones on the rig. Bottom: Views from the telephoto cameras. All ten views are used to compute ground truth depth using multi-view stereo techniques.

As shown in Fig. 4(a) in the main paper, our capture rig consists of five Google Pixel 4 phones. Each phone captures a stereo pair (and dual-pixel data), giving us ten views of the same scene (Fig. 9). The five phones are synchronized using [2] allowing us to capture dynamic scenes, e.g., plants moving in the wind. Our dataset is captured both indoors and outdoors, and contains both man made and natural scenes.

A.2 Computing Ground Truth Disparity D^{gt} and Confidence C^{gt}

We now describe how we compute D^{gt} and C^{gt} given a capture from the rig.

Since the rig may not be perfectly rigid, we first compute camera poses, i.e., intrinsics and extrinsics, using structure from motion [17]. For computing ground truth depth using multi-view stereo, we use a method similar to [13] that is designed to give accurate depth for fine structures while avoiding edge fattening artifacts. We describe it in more detail below.

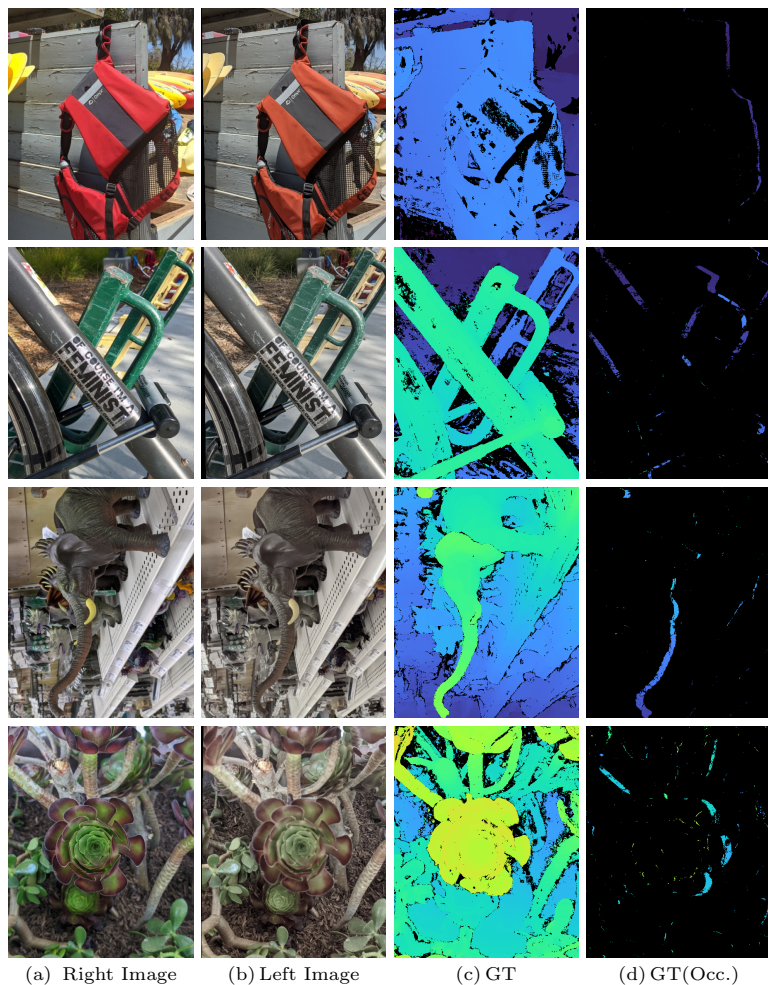


Fig. 10: Examples of collected data. The right view and left view (a,b) of the binocular stereo pair, the ground truth disparity (c), and the ground truth disparity for occluded pixels (d). Low confidence disparity is rendered in black in (c) and (d).

All ten RGB images are resized to 756×1008 . For each view, we use a plane sweep algorithm, with 256 planes sampled using inverse perspective sampling between 0.2m and 100m, and take the minimum of a filtered cost volume as each pixel’s depth. To compute the cost volume, for each pixel, we compute the sum of absolute differences for each of the warped neighbors and then bilaterally filter the cost volume using the grayscale reference image as the guide image thus avoiding edge fattening artifacts [39]. We use a spatial sigma of 3 pixels and a range sigma of 12.5 for the bilateral filter.

Following [13], we also estimate per-pixel confidence for depth, i.e., a scalar in the range $[0, 1]$. Specifically, we check for depth coherence across views by checking for left / right consistency [4]. We first compute consistency with each of the 9 neighboring images using the consistency measure in [13]. Then, under the assumption that a pixel must be visible in at least two other cameras for its depth to be reliable, we take the product of the largest two consistency values for each pixel to compute our final confidence.

Even though we capture data from all five phones, we only use the data from the center camera for training and testing since it’s likely to have the most accurate depth. We use the estimated camera poses for the center phone to rectify the stereo pair [12]. Specifically, we compute \mathbf{W}_l and \mathbf{W}_r , i.e., warp maps corresponding to the left and the right cameras that are applied to the RGB images, depth maps and confidences for the left and the right images. The camera poses are also used to convert depth into disparity between the rectified pair. See Fig. 10 for examples of rectified RGB images and the corresponding ground truth disparity.

Further, since the telephoto camera has a smaller field of view than the main camera, we apply a center crop of size 448×560 to all rectified images to restrict ourselves to the area of overlap.

A.3 Computing Occlusion Confidence C^{occ}

As mentioned in the main paper, we also compute C^{occ} , i.e., a per-pixel confidence where the ground truth disparity is accurate but the pixel is occluded in the other camera. This allows us to evaluate and compare the methods in regions that are occluded in one of the stereo views.

To compute it, we first estimate the set of pixels in the right image that are in field of view of the left image but are occluded by an occluder. This can be estimated as:

$$Occ_r = \left\{ (x, y) \quad s.t. \quad \begin{array}{l} 0 \leq x' < W, \\ |D_r^{gt}(x, y) + D_l^{gt}(x', y)| > \Delta \\ |D_l^{gt}(x', y) + D_r^{gt}(x' + D_l^{gt}(x', y), y)| \leq \Delta \\ |D_l^{gt}(x', y)| > |D_r^{gt}(x, y)| \end{array} \right\} \quad (6)$$

where $x' = x + D_r^{gt}(x, y)$, W is the width of the rectified image, and Δ is set to 1-pixel disparity. The first condition enforces that the pixel is in field of

view of the other (left) camera; the second condition ensures that the pixel is not visible in the other camera by checking for failed left-right consistency check; the final two conditions check that the pixel is occluded by an object that is visible in both the views (consistency check succeeds) and is in front of the occluded pixel. Finally, for pixels that are in the set Occ_r we set $C_r^{occ}(x, y)$ to be the product of the confidences of the pixel and the occluder, i.e.,

$$C_r^{occ}(x, y) = \begin{cases} C_r^{gt}(x, y) \cdot C_l^{gt}(x', y), & \text{if } (x, y) \in Occ_r \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

A few examples are shown in Fig. 10. Our conservative criterion for occlusion confidence ensures that we have few false positives.

B Implementation Details

In this section, we provide more details about confidence volume fusion, network architecture, and evaluation with affine fitting.

B.1 Cost Volume and Confidence Volume

In Sec. 4.2 of the main paper, we fuse confidence volume instead of the cost volume. Here we give more explanation and motivation.

The commonly used cost volume [23] is a 3D volume with two dimensions for the image space (H, W) and one dimension for the disparity space $R = [0, 1, \dots, d_{max}]$. Each voxel (x, y, d) is a floating number indicating the feature distance if the pixel (x, y) in one view is matched under the given disparity d with the other view. The distance can be computed using various distance metrics, such as ℓ_1 or ℓ_2 .

A soft-arg min, which is introduced in Eq. 1 in [23], is used to convert the cost volume into a disparity map. Specifically, a confidence volume is calculated as the softmax on the negative cost volume along the disparity dimension, and output disparity is the sum of disparity hypotheses weighted by the confidence. The operator to convert cost volume to confidence volume can also be written as:

$$\text{Confidence}(x, y, d) = \frac{e^{-\text{Cost}(x, y, d)/t}}{\sum_{d \in R} e^{-\text{Cost}(x, y, d)/t}}, \quad (8)$$

where t controls the sharpness of the softmax and is set to 0.5 in our implementation. The voxels along the disparity dimension for each (x, y) forms a probability distribution, i.e., $\sum_{d \in R} \text{Confidence}(x, y, d) = 1$, indicating the likelihood of each disparity proposal in R being correct (i.e. confidence). Therefore, the output disparity is

$$D(x, y) = \sum_{d \in R} \text{Confidence}(x, y, d) \cdot d. \quad (9)$$

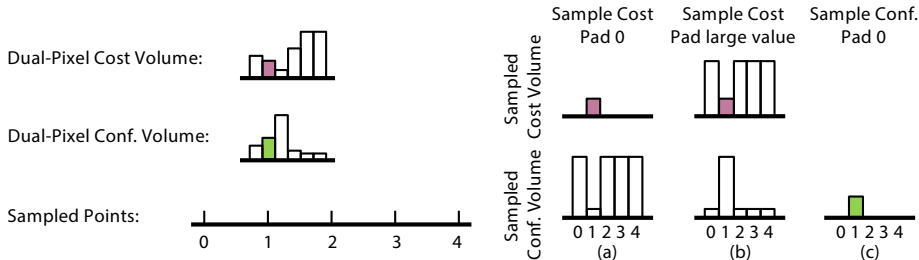


Fig. 11: Explanation of Volume Sampling. See text for details.

We now explain why it is better to sample confidence volume (Eq. 8) instead of the cost volume. Besides normalizing the scales of the two cost volumes, they may produce dramatically different results in some cases.

Fig. 11 illustrates one such case. For simplicity, we drop the image dimension (H, W) and only visualize the disparity dimension R for one pixel. On the left, we show an example of cost volume learned from DP inputs and the corresponding confidence volume. Note that the confidence is inversely related with the cost. Since the DC inputs covers larger range of disparity compared to DP, the warping process in Eq. 3 of the main paper usually samples many points out of the DP disparity range. In Fig. 11, we demonstrate the case where only one sample falls in the valid range of the DP disparity – sampling point with disparity 1. If we sample cost volume and pad 0 for samples out of range, we obtain a cost volume shown in (a) on the right. The corresponding confidence volume (according to Eq. 8) indicates that 1 is a bad disparity hypothesis and all the others are equally good, which is very inconsistent with the information provided in the original DP cost volume. In the second case (b), we sample cost volume but pad with a large value. Now, the disparity hypothesis 1 becomes the best hypothesis (unlike (a)) but the confidence in 1 is much higher than the original confidence in DP confidence volume. In contrast, if we sample the confidence volume and pad 0 as shown in (c), the produced confidence volume maintains exactly the same confidence from DP volume for disparity 1, while the others are set to zero.

B.2 Network Architecture

We provide detailed network architecture in Fig. 12.

B.3 Evaluation with Affine Fitting

In the Sec. 5.4 of the main paper, we compared to dual-pixel based depth estimation solution DpNet [13]. Since depth from DP can only be predicted up to an unknown affine transformation, Garg et al. [13] first estimate the affine transformation by solving a weighted least squares problem using the ground truth:

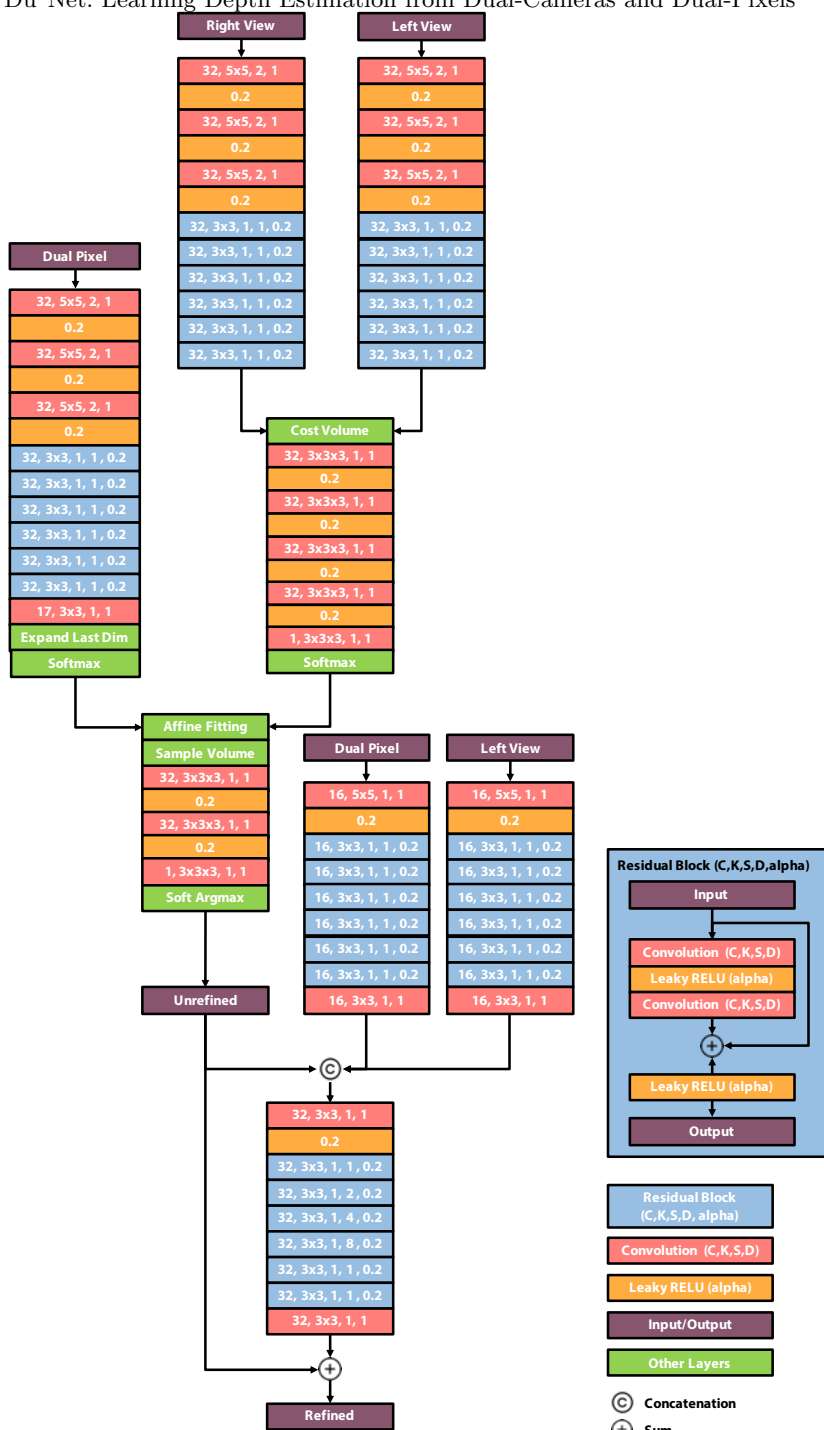


Fig. 12: Network Architecture. The numbers on convolution layers represent number of channels, size of filter, stride, and dilation respectively. The number on the leaky ReLU layer represents the slope for the negative input.

$$\hat{\alpha}, \hat{\beta} = \operatorname{argmin}_{\alpha, \beta} \left\| C^{gt} \cdot ((\alpha + \beta \cdot D_{raw}) - D^{gt}) \right\|^2, \quad (10)$$

where D_{raw} is the network output. $D_{fit} = \hat{\alpha} + \hat{\beta} \cdot D_{raw}$ is then used for computing the metrics. Even though Du²Net produces disparity in absolute scale and is free from this ambiguity, we apply the same post-processing when comparing to the DPNet for fairness (Du²Net* in Tab. 2 of the main paper).

C More Experiment Results

In this section, we provide more quantitative and qualitative evaluations.

C.1 Weighted Metrics

Our ground truth comes with a confidence mask (Sec. A.2), and we use it to calculate weighted evaluation metrics.

$$\text{MAE} = \frac{\sum_p |D(p) - D^{gt}(p)| \cdot C^{gt}(p)}{\sum_p C^{gt}(p)}, \quad (11)$$

$$\text{RMSE} = \sqrt{\frac{\sum_p (D(p) - D^{gt}(p))^2 \cdot C^{gt}(p)}{\sum_p C^{gt}(p)}}, \quad (12)$$

$$\delta > \epsilon = \frac{\sum_p \mathbb{1}(|D(p) - D^{gt}(p)| > \epsilon) \cdot C^{gt}(p)}{\sum_p C^{gt}(p)}, \quad (13)$$

where D is the predicted disparity, D^{gt} is the ground truth disparity, C^{gt} is the confidence map, $\mathbb{1}$ is an indicator function which equals 1 if the condition is true and 0 otherwise, and p is a pixel in the image.

C.2 More Quantitative Ablation Study

In the Sec. 5.3 of the main paper, we showed the quantitative evaluation of our method under different ablations on all pixels C^{gt} . We perform the same comparison on the occluded pixels using C^{occ} (Sec. A.3) to show the performance in occluded regions.

Tab. 3 shows the evaluation of the unrefined disparity (i.e. the output of the fused volume) under different fusion strategies. Consistent with the conclusion drawn from all pixels, our method outperforms all the others on the occluded regions.

Tab. 4 shows the evaluation of the refined disparity (i.e. the output of the refinement) under different settings. Refinement with DP consistently outperforms the case without DP on all the metrics. Using only DP is better than using both under some metrics, which is reasonable since RGB may not be very helpful to recover details in the occluded region and may even hurt the valuable information encoded in DP.

Conf. Volume	All Pixels					Occluded Pixels				
	MAE	RMSE	$\delta > 1.25$	$\delta > 2$	$\delta > 3$	MAE	RMSE	$\delta > 1.25$	$\delta > 2$	$\delta > 3$
DC	1.023	2.502	18.74	10.65	6.32	3.956	6.230	66.33	52.82	40.16
DP+DC (2D)	0.969	2.423	17.37	9.72	5.79	3.718	5.834	65.74	51.55	38.51
DP+DC (C)	0.964	2.372	17.51	9.79	5.80	3.671	5.768	65.49	51.63	38.62
DP+DC (Ours)	0.902	2.252	16.16	8.96	5.26	3.526	5.523	64.98	50.82	37.41

Table 3: Ablation study on volume fusion. We compare different ways of fusing DP with the DC confidence volume. ‘(2D)’ indicates fusion of the 2D disparity maps extracted from the two confidence volumes. ‘(C)’ indicates fusing cost volumes instead of confidence volumes.

Refinement	All Pixels					Occluded Pixels				
	MAE	RMSE	$\delta > 1.25$	$\delta > 2$	$\delta > 3$	MAE	RMSE	$\delta > 1.25$	$\delta > 2$	$\delta > 3$
RGB	0.838	2.197	14.17	7.74	4.55	2.627	4.866	46.17	33.70	24.18
DP (I)	0.835	2.173	14.25	7.75	4.54	2.518	4.600	46.07	33.25	23.47
DP	0.829	2.184	13.84	7.51	4.45	2.481	4.619	44.87	31.99	22.54
RGB+DP (Ours)	0.817	2.141	13.64	7.33	4.35	2.469	4.564	44.94	32.09	22.47

Table 4: Ablation study on refinement. We compare the different ways of using DP to refine the best unrefined disparity from the left and show evaluation on the final disparity. ‘(I)’ indicates that input DP images are warped before computing features for refinement.

C.3 More Qualitative Ablation Study

We show more qualitative comparison in Fig. 13. Our method fusing DP into the cost volume (d) significantly improves the quality of the unrefined disparity compared to the case using only DC (c). Based on this improved unrefined disparity with less error (d), refinement using both DP and DC (f) can further improve the object boundary and thin structures compared to the case using only DC (e).

C.4 More Qualitative Comparison to SOTA

We show more qualitative comparison to state-of-the-art stereo and DP based approaches in Fig. 14, 15, and 16. Compared to other stereo based approaches [5, 24] that only take DC as the input, our method performs better at object boundary and thin structures. Compared to dual-pixel only approach [13], our method produces significantly better depth for distant areas in the background while maintaining the foreground details (Fig. 16).

C.5 Analysis of Best and Worst Cases

In Fig. 17 we show representative images from the best (top 3 rows) and the worst (bottom 3 rows) results for our method as ranked by the MAE metric. As expected, the method performs very accurately on images with high frequency details and textured scenes whereas it does worse (along with other methods) in textureless areas.

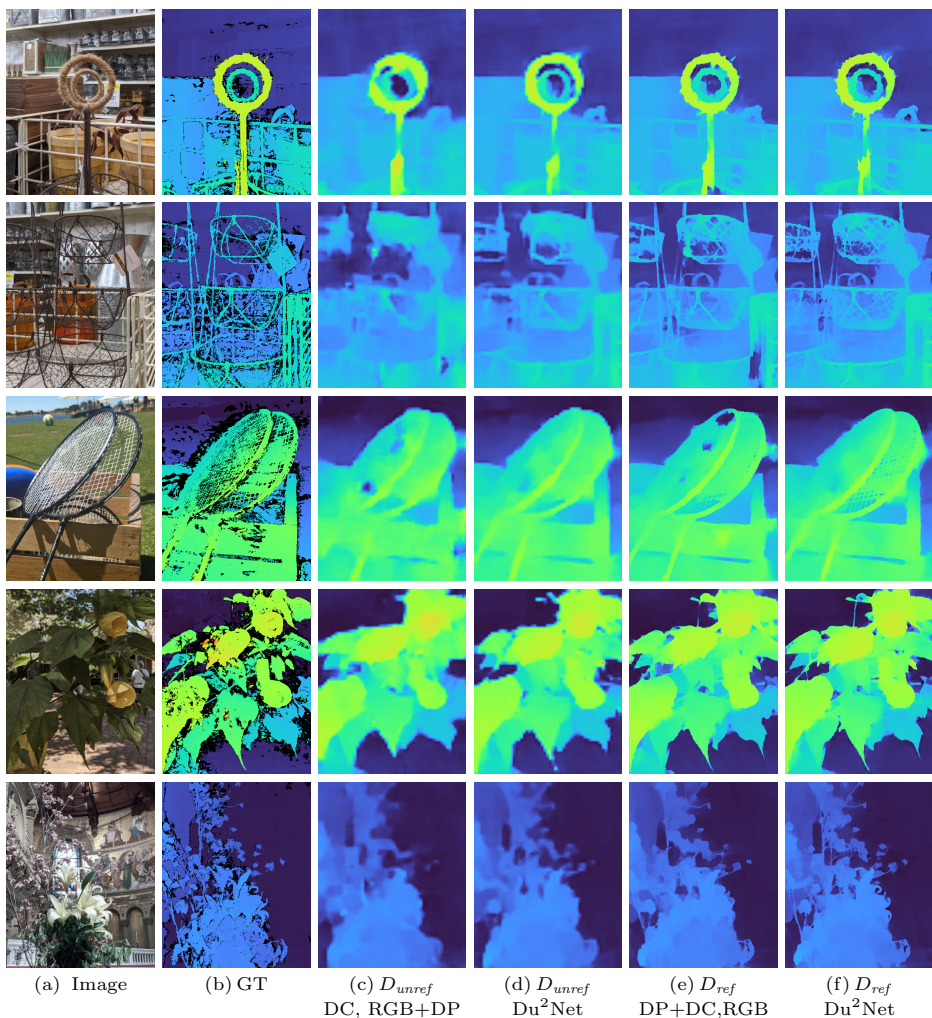


Fig. 13: Ablations of our method. The right camera image (a), ground truth disparity (b) with low confidence disparity in black, D_{unref} (c) from an ablation where only the DC input is used for the confidence volume, D_{unref} (d) from Du²Net, D_{ref} (e) from an ablation where only the RGB image is used for refinement, and D_{ref} (f) from Du²Net. DP input is useful for both the confidence volume and refinement stages to recover accurate depth for fine structures and occluded regions.

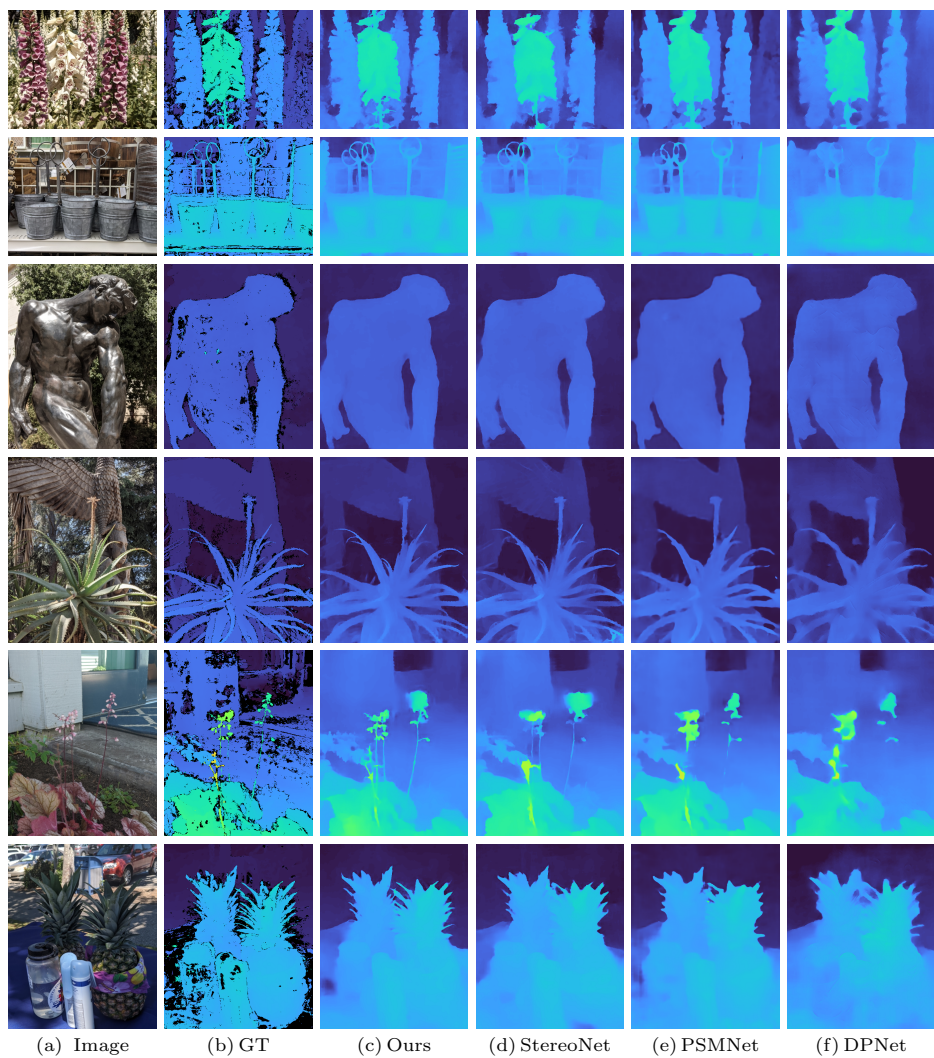


Fig. 14: Qualitative comparison to state-of-the-art stereo and DP based methods.

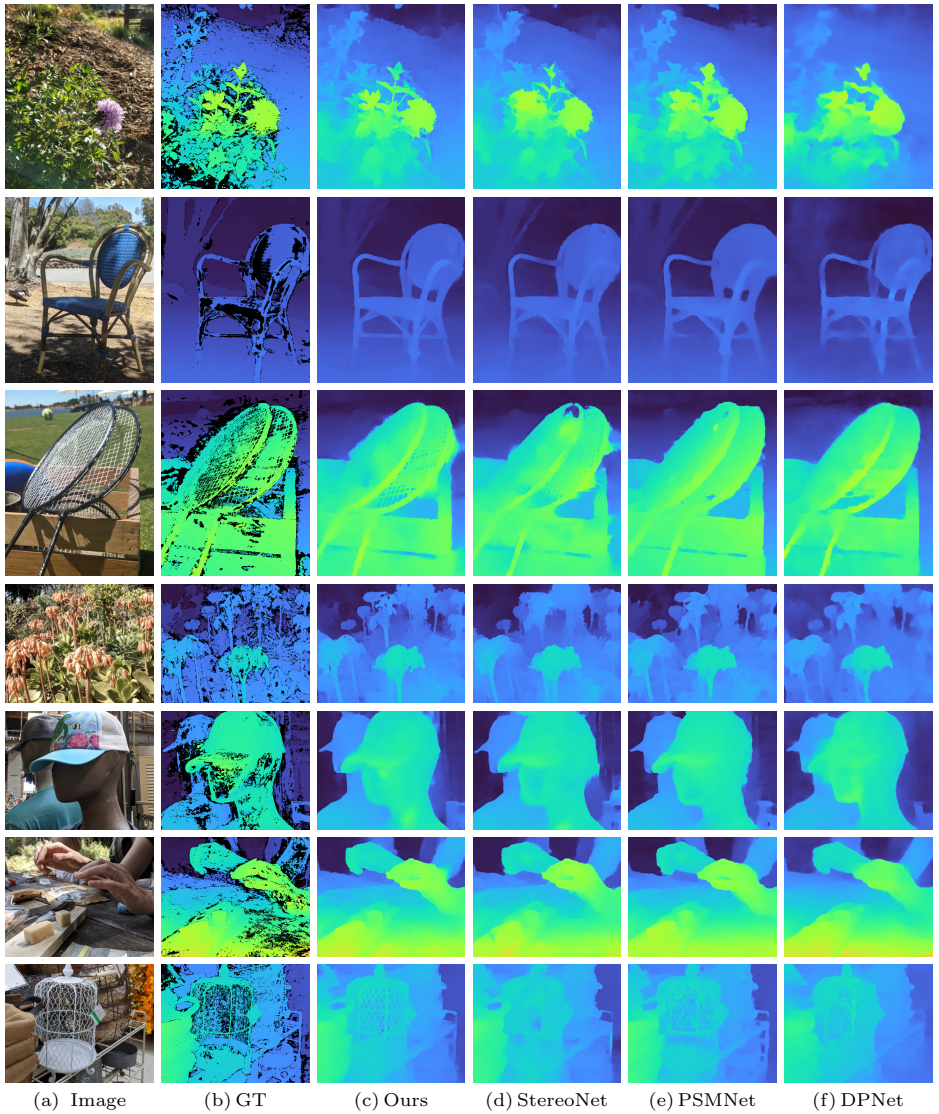


Fig. 15: Qualitative comparison to state-of-the-art stereo and DP based methods.

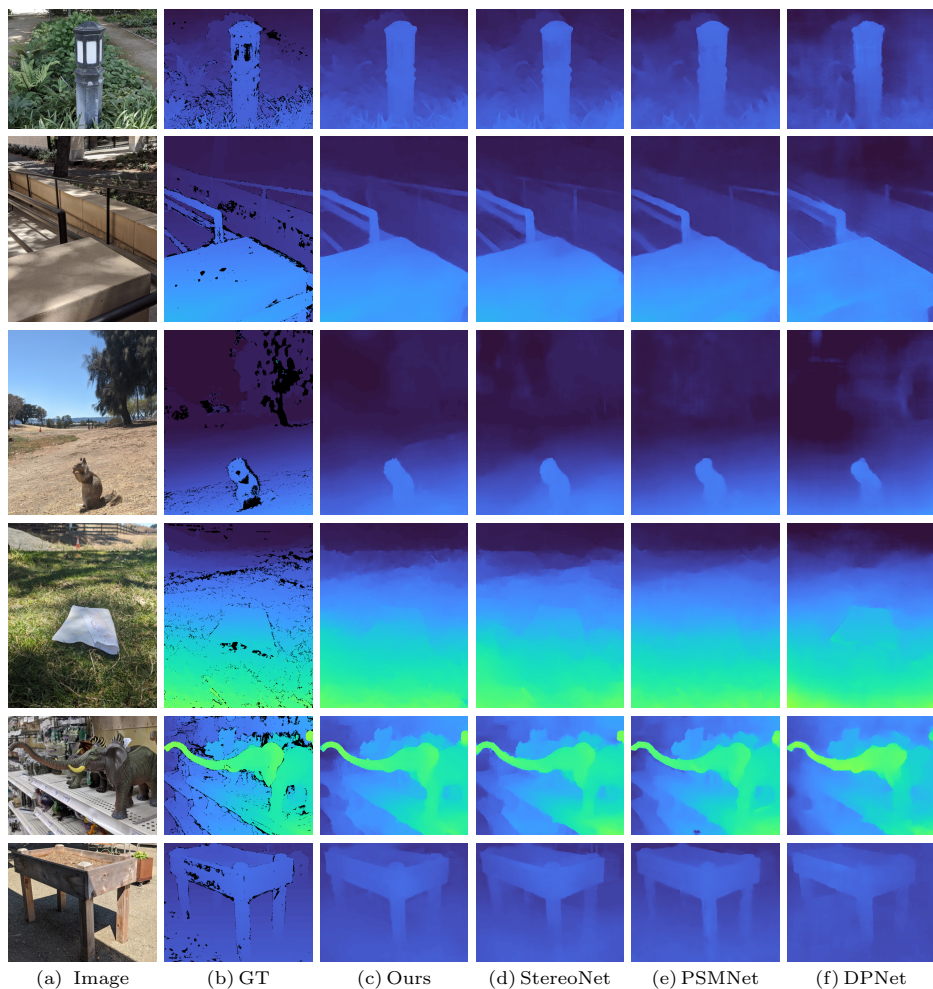


Fig. 16: DPNet (f) performs worse in distant areas compared to methods that take DC as input ((c), (d), (e)) due to the small baseline between the two dual-pixel images.

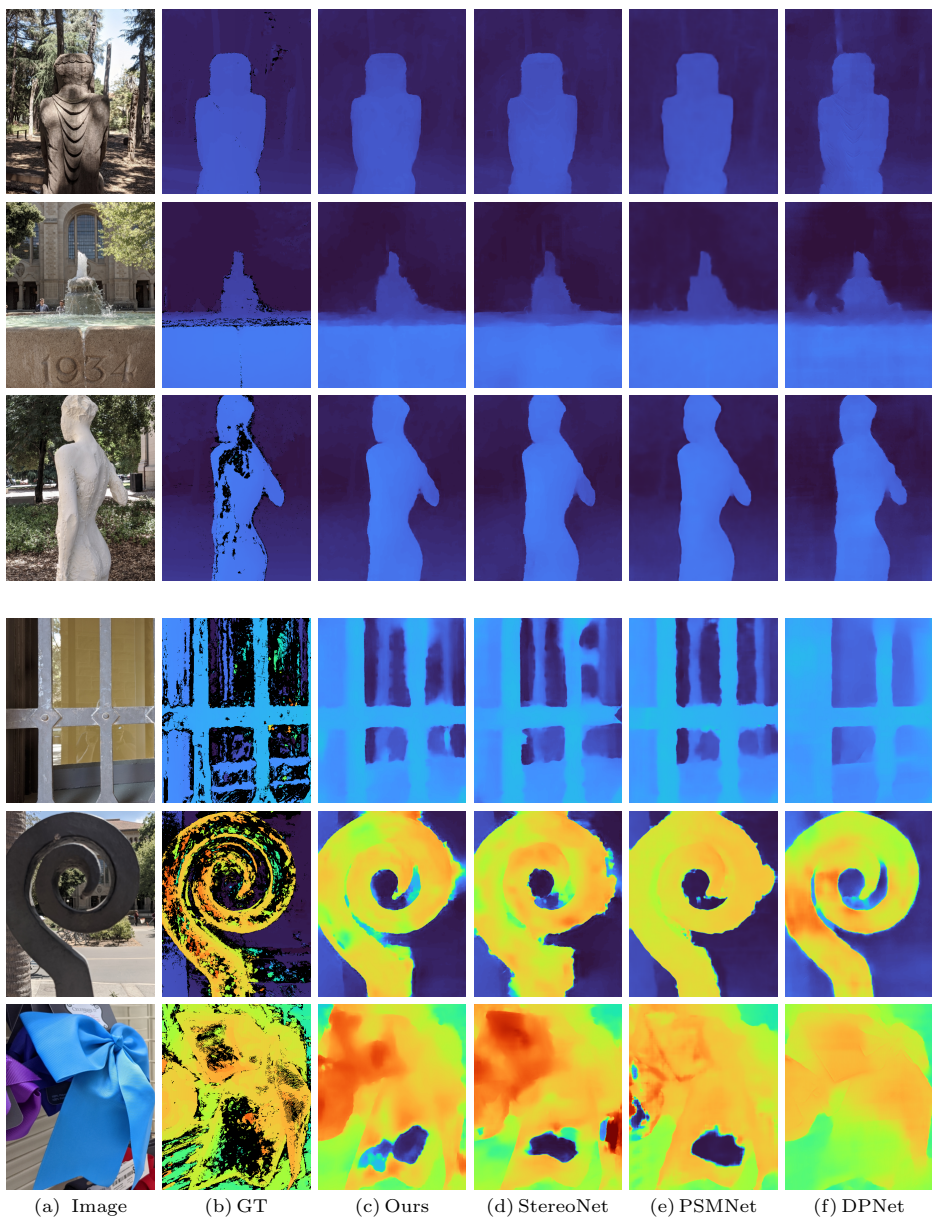


Fig. 17: Representative images from our best (top 3 rows) and worst (bottom 3 rows) results, as rated by MAE metric.

C.6 More Results for Applications

We show more examples of computational photography applications. Fig. 18 shows results of synthetic shallow depth-of-field effect using disparity from different models. Our method produces better details for object boundary and thin structure, which prevents artifacts near the subject boundary.

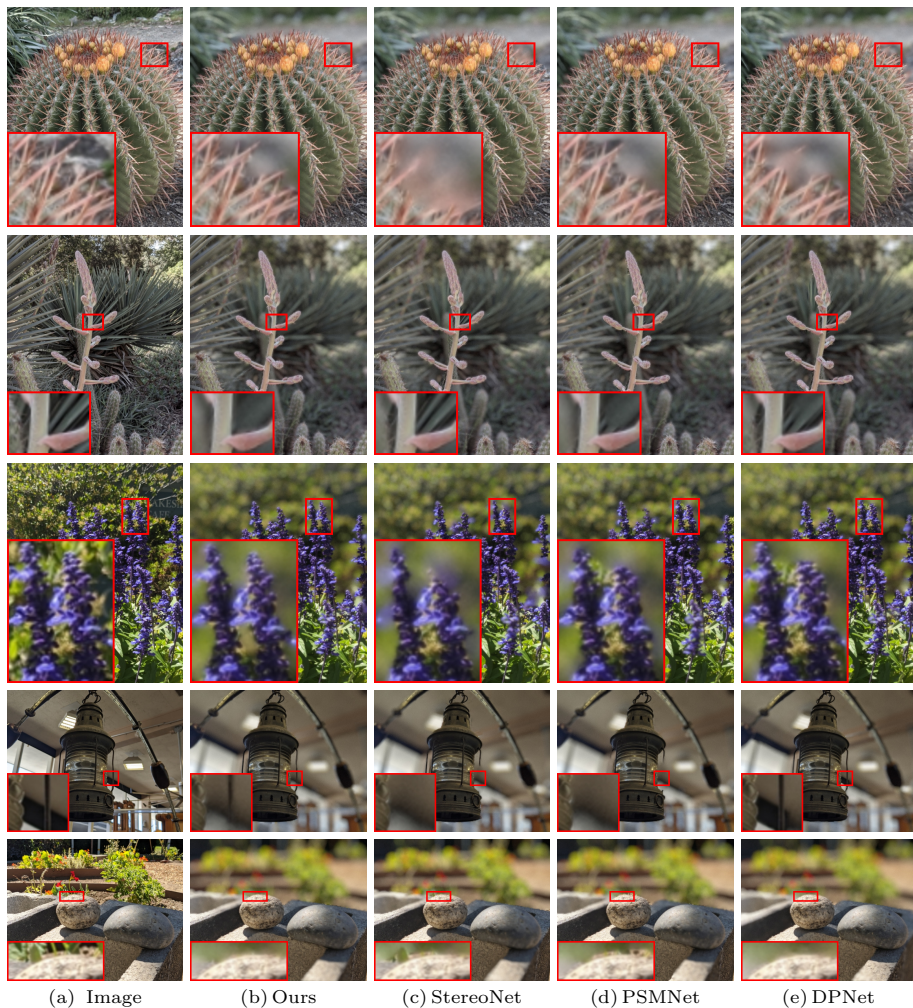


Fig. 18: Synthetic shallow depth-of-field results for different methods. Accurate depth near occlusion boundaries is critical for avoiding artifacts near the subject boundary.