Learning Single Camera Depth Estimation using Dual-Pixels

Rahul Garg Neal Wadhwa Sameer Ansari Jonathan T. Barron
Google Research

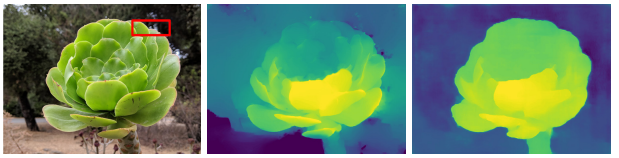
Abstract

Deep learning techniques have enabled rapid progress in monocular depth estimation, but their quality is limited by the ill-posed nature of the problem and the scarcity of high quality datasets. We estimate depth from a single camera by leveraging the dual-pixel auto-focus hardware that is increasingly common on modern camera sensors. Classic stereo algorithms and prior learning-based depth estimation techniques underperform when applied on this dual-pixel data, the former due to too-strong assumptions about RGB image matching, and the latter due to not leveraging the understanding of optics of dual-pixel image formation. To allow learning based methods to work well on dual-pixel imagery, we identify an inherent ambiguity in the depth estimated from dual-pixel cues, and develop an approach to estimate depth up to this ambiguity. Using our approach, existing monocular depth estimation techniques can be effectively applied to dual-pixel data, and much smaller models can be constructed that still infer high quality depth. To demonstrate this, we capture a large dataset of in-the-wild 5-viewpoint RGB images paired with corresponding dual-pixel data, and show how view supervision with this data can be used to learn depth up to the unknown ambiguity. On our new task, our model is 30% more accurate than any prior work on learning-based monocular or stereoscopic depth estimation.

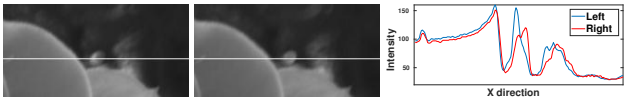
1. Introduction

Depth estimation has long been a central problem in computer vision, both as a basic component of visual perception, and in service to various graphics, recognition, and robotics tasks. Depth can be acquired via dedicated hardware that directly senses depth (time-of-flight, structured light, etc) but these sensors are often expensive, power-hungry, or limited to certain environments (such as indoors). Depth can be inferred from multiple cameras through the use of multi-view geometry, but building a stereo camera requires significant complexity in the form of calibration, rectification, and synchronization. Machine learning techniques can be used to estimate depth from a single image, but the under-constrained nature of image formation often results in inaccurate estimation.

Recent developments in consumer hardware may provide an opportunity for a new approach in depth estimation. Cameras have recently become available that allow a single camera to simultaneously capture two images that resemble a stereo pair with a tiny baseline (Fig. 1), through the use of dense dual-pixel (DP) sensors (Fig. 2). Though this technology was originally developed in service of camera auto-focus, dual-pixel images can also be exploited to recover dense depth maps from a single camera, thereby obviating any need for additional hardware, calibration, or synchronization. For example, Wadhwa *et al*. [50] used classical stereo techniques (block matching and edge aware smoothing) to recover depth from DP data. But as shown in Fig. 1, the quality of depth maps that can be produced by conventional stereo techniques is limited, because the interplay between disparity and focus in DP imagery can cause classic stereo-matching techniques to fail. Existing monocular learning-based techniques also perform poorly on this

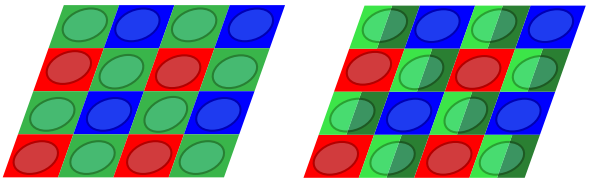


(a) RGB image (b) Depth from [50] (c) Our depth



(d) Left DP view (e) Right DP view (f) Left vs Right intensity

Figure 1. Here we have an RGB image (a) containing dual-pixel data. Crops of the left and right dual-pixel images corresponding to the marked rectangle in (a) are shown in (d), (e), and their intensity profiles along the marked scanline are shown in (f). While the profiles matches for the in-focus flower, they are considerably different for the out of focus background. Because [50] uses traditional stereo matching that assumes that intensity values differ by only a scale factor and a local displacement, it fails to match the background accurately, and produces the depth shown in (b). Our technique learns the correlation between depth and differences in dual-pixel data thereby estimates an accurate depth map (c).



(a) Traditional Bayer Sensor (b) Dual-Pixel Sensor

Figure 2. A modern Bayer sensor consists of interleaved red, green, and blue pixels underneath a microlens array. (a). In dual-pixel sensors, the green pixel under each microlens is split in half (b), resulting in two green images that act as a narrow-baseline stereo camera, much like a reduced light field camera.

task. In this paper, we analyze the optics of image formation for dual-pixel imagery and demonstrate that DP images have a fundamentally ambiguous relationship with respect to scene depth — depth can only be recovered up to some unknown affine transformation. With this observation, we analytically derive training procedures and loss functions that incorporate prior knowledge of this ambiguity, and are therefore capable of learning effective models for affine-invariant depth estimation. We then use these tools to train deep neural networks that estimate high-quality depth maps from DP imagery, thereby producing detailed and accurate depth maps using just a single camera. Though the output of our learned model suffers from the same affine ambiguity that our training data does, the affine-transformed depths estimated by our model can be of great value in certain contexts, such as depth ordering or defocus rendering.

Training and evaluating our model requires large amounts of dual-pixel imagery that has been paired with ground-truth depth maps. Because no such dataset exists, in this work we also design a capture procedure for collecting "in the wild" dual-pixel imagery where each image is paired with multiple alternate views of the scene. These additional views allow us to train our model using view supervision, and allow us to use multi-view geometry to recover ground-truth estimates of the depth of the scene for use in evaluation. When comparing against the state-of-the-art in depth estimation, our proposed model produces error rates that are 30% lower than previous dual-pixel and monocular depth estimation approaches.

2. Related Work

Historically, depth estimation has seen the most attention and progress in the context of stereo [44] or multi-view geometry [24], in which multiple views of a scene are used to partially constrain its depth, thereby reducing the inherent ambiguity of the problem. Estimating the depth of a scene from a single image is significantly more underconstrained, and though it has also been an active research area, progress has happened more slowly. Classic monocular depth approaches relied on singular cues, such as shading [29], texture [7], and contours [12] to inform depth, with some success in constrained scenarios. Later work attempted to use learning to explicitly consolidate these bottom-up cues into more robust monocular depth estimation techniques [10, 27, 43], but progress on this problem accelerated rapidly with the rise of deep learning models trained end-to-end for monocular depth estimation [16, 18], themselves enabled by the rise of affordable consumer depth sensors which allowed collection of large RGBD datasets [31, 39, 46]. The rise of deep learning also yielded progress in stereoscopic depth estimation [51] and in the related problem of motion estimation [15]. The need for RGBD data in training monocular depth estimation models was lessened by the discovery that the overconstraining nature of multi-view geometry could be used as a supervisory cue for training such systems [17, 19, 21, 34, 38, 52], thereby allowing "self-supervised" training using only video sequences or stereo pairs as input. Our work builds on these monocular and stereo depth prediction algorithms, as we construct a learning-based "stereo" technique, but using the impoverished dual-pixel data present within a single image.

An alternative strategy to constraining the geometry of the scene is to vary the camera's focus. Using this "depth from (de)focus" [23] approach, depth can be estimated from focal stacks using classic vision techniques [48] or deep learning approaches[25]. Focus can be made more informative in depth estimation by manually "coding" the aperture of a camera [36], thereby causing the camera's circle of confusion to more explicitly encode scene depth. Focus cues can also be used as supervision in training a monocular depth estimation model [47]. Reasoning about the relationship between depth and the apparent focus of an image is critical when considering dual-pixel cameras, as the effective point spread functions of the "left" and "right" views are different. By using a flexible learning framework, our model is able to leverage the focus cues present in dual-pixel imagery in addition to the complementary stereo cues.

Stereo cameras and focal stacks are ways of sampling what Adelson and Bergen called "the plenoptic function": a complete record of the angle and position of all light passing through space [3]. An alternative way of sampling the plenoptic function is a light field [37], a 4D function that contains conventional images as 2D slices. Light fields can be use to directly synthesize images from different positions or with different aperture settings [40], and light field cameras can be made by placing a microlens array on the sensor of a conventional camera [4, 41]. Light fields provide a convenient framework for analyzing the equivalence of correspondence and focus cues [49]. While light fields have been used to recover depth [32, 33], constructing a light field camera requires sacrificing spatial resolution in favor of angular resolution, and as such light field cameras have not

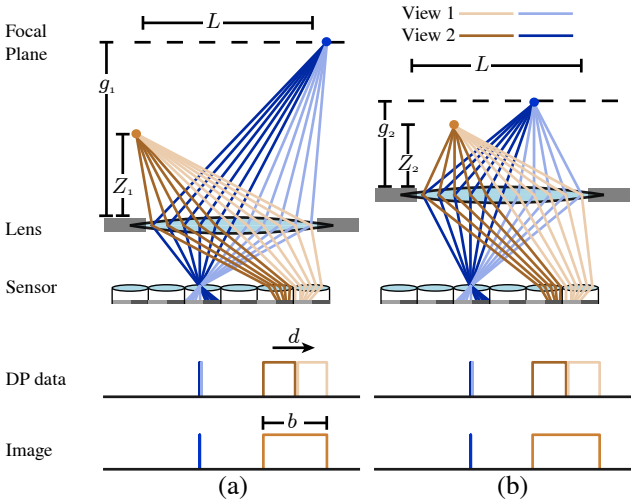


Figure 3. Dual-pixel views see different halves of the aperture, which provides a depth cue. However, due to a fundamental ambiguity, different scenes can yield the same dual-pixel images if the focus distance (or the aperture size, or the focal length) of the camera changes. In (a), a camera with focus distance g_1 images an in-focus blue point and an out-of-focus orange point a distance Z_1 away. Light refracting through the left half of the aperture (dark blue and orange rays) arrives at the right half of each dual-pixel, and vice versa. This results in a dual-pixel image in which the out-of-focus orange point is displaced by d pixels (a, "DP Data") and blurred by b pixels (a, "Image"). In (b), a different focus distance and set of scene depths yields the same dual-pixel and RGB images. However, as shown in the text, this scene is related to the one in (a) by an affine transformation on inverse depth.

seen rapid consumer adoption. Dual-pixel cameras appear to represent a promising compromise between more ambitious light field cameras and conventional cameras — DP cameras sacrifice a negligible amount of spatial resolution to sample two angles in a light field, while true monocular cameras sample only a single angle, and light field cameras such as the Lytro Illum sample 196 angles at the cost of significant spatial resolution. As a result, they have seen wider adoption in consumer cameras and in space-constrained applications like endoscopy [6].

3. Dual-Pixel Geometry and Ambiguity

Dual-pixel (DP) sensors work by splitting each pixel in half, such that the left half integrates light over the right half aperture and vice versa (Fig. 3). Because each half of a dual-pixel integrates light over one half of the aperture, the two halves of a pixel together form a kind of stereo pair, in which nearby objects exhibit some horizontal disparity between the two views in accordance with their distance. This effect interacts with the optical blur induced by the lens of the camera, such that when image content is far from the focal plane, the effects of optical blur are spread across the two "views" of each dual-pixel (Fig. 3(a, DP data)). The sum of the two views accounts for all the light going through the aperture and is equal to the ordinary full-pixel image that would be captured by a non dual-pixel sensor. As a result, the disparity d between the two views in a dual-pixel image is proportional to what the defocus blur size b would be in an equivalent full-pixel image. Dual-pixel sensors are commonly used within consumer cameras to aid in auto-focus: the camera iteratively estimates disparity from the dual-pixels in some focus region and moves the lens until that disparity is zero, resulting in an image in which the focus region is in focus.

While dual-pixel imagery can be thought of as a stereo pair with a tiny baseline, it differs from stereo in several ways. The views are perfectly synchronized (both spatially and temporally) and have the same exposure and white balance. In addition, the two views in DP images have different point-spread functions that can encode additional depth information. Traditional stereo matching techniques applied to dual-pixel data will not only ignore the additional depth information provided by focus cues, but may even fail in out-of-focus regions due to the effective PSFs of the two views being so different that conventional image matching fails (Figs. 1(d)-1(f)). As an additional complication, the relationship between depth and disparity in dual-pixel views depends not only on the baseline between the two views, but also on the focus distance. Thus, unlike depth from stereo, which has only a scale ambiguity if the extrinsics are unknown, depth from dual-pixel data has *both* scale and offset ambiguities if the camera's focus distance is unknown (as is the case for most current consumer cameras, such as those we use). Addressing the ambiguity caused by this unknown scale and offset is critical when learning to estimate depth from dual-pixel imagery, and is a core contribution of this work. As we will demonstrate, for a network to successfully learn from dual-pixel imagery, it will need to be made aware of this affine ambiguity.

We will now derive the relationship between depth, disparity, and blur size according to the paraxial and thin-lens approximations. Consider a scene consisting of point light sources located at coordinates $(x, y, Z(x, y))$ in camera space. As stated previously, the disparity of one such point on the image plane $d(x, y)$ is proportional to the (signed) blur size $\bar{b}(x, y)$, where the sign is determined by whether the light source is in front or behind the focal plane. Therefore, from the paraxial and thin-lens approximations:

$$d(x,y) = \alpha \bar{b}(x,y) \tag{1}$$

$$\approx \alpha \frac{Lf}{1 - f/g}\left(\frac{1}{g} - \frac{1}{Z(x,y)}\right) \tag{2}$$

$$\triangleq A(L,f,g) + \frac{B(L,f,g)}{Z(x,y)}, \tag{3}$$

where α is a constant of proportionality, L is the diameter of

the aperture, f is the focal length of the lens and g is the focus distance of the camera. We make the affine relationship between inverse depth and disparity explicit in Eqn. 3 by defining image-wide constants $A(L, f, g)$ and $B(L, f, g)$. This equation reflects our previous assertion that perfect knowledge of disparity d and blur size b only gives enough information to recover depth Z if the parameters L, f and g are known. Please see the supplement for a derivation.

Eqn. 3 demonstrates the aforementioned affine ambiguity in dual-pixel data. This means that different sets of camera parameters and scene geometries can result in identical dual-pixel images (Fig. 3(b)). Specifically, two sets of camera parameters can result in two sets of affine coefficients (A_1, B_1) and (A_2, B_2) such that the same image-plane disparity is produced by two different scene depths

$$d(x, y) = A_1 + \frac{B_1}{Z_1(x, y)} = A_2 + \frac{B_2}{Z_2(x, y)}. \quad (4)$$

Consumer smartphone cameras are not reliable in recording camera intrinsic metadata [14], thereby eliminating the easiest way that this ambiguity could be resolved. But Eqn. 3 does imply that it is possible to use DP data to estimate some (unknown) affine transform of inverse depth. This motivates our technique of training a CNN to estimate inverse depth only up to an affine transformation.

Though absolute depth would certainly be preferred over an affine-transformed depth, the affine-transformed depth that can be recovered from dual-pixel imagery is of significant practical use. Because affine transformations are monotonic, an affine-transformed depth still allows for reasoning about relative ordering of scene depths. Affine-invariant depth is a natural fit for synthetic defocus (simulating wide aperture images by applying a depth dependent blur to a narrow aperture image [9, 50]) as the affine parameters naturally map to the user controls — the depth to focus at, and the size of the aperture to simulate. Additionally, this affine ambiguity can be resolved using heuristics such as the likely sizes of known objects [28], thereby enabling the many uses of metric depth maps.

4. View supervision for Affine Invariant Depth

A common approach for training monocular depth estimation networks from multi-view data is to use self supervision. This is typically performed by warping an image from one viewpoint to the other according to the estimated depth and then using the difference between the warped image and the actual image as some loss to be minimized. Warping is implemented using a differentiable spatial transformer layer [30] that allows end-to-end training using only RGB views and camera poses. Such a loss can be expressed as:

$$\mathcal{L}(I_0, \Theta) = \sum_{(x,y)} \Delta\left(I_0(x, y), I_1\left(M\left(x, y; F\left(I_0, \Theta\right)\right)\right)\right) \quad (5)$$

Where I_0 is the RGB image of interest, I_1 is a corresponding stereo image, $F(I_0, \Theta)$ is the (inverse) depth estimated by a network for I_0, $M(x, y; \hat{D})$ is the warp induced on pixel coordinates (x, y) by that estimated depth $\hat{D} = F(I_0, \Theta)$ and by the known camera poses, and $\Delta(\cdot, \cdot)$ is some arbitrary function that scores the per-pixel difference between two of RGB values. $\Delta(\cdot, \cdot)$ will be defined in Sec. 6.2, but for our current purposes it can be any differentiable penalty. Because we seek to predict inverse depth up to an unknown affine transform, the loss in Eqn. 5 cannot be directly applied to our case. Hence, we introduce two different methods of training with view supervision while predicting inverse depth up to an affine ambiguity.

4.1. 3D Assisted Loss

If we assume that we have access to a ground truth inverse depth D^* and corresponding per-pixel confidences C for that depth, we can find the unknown affine mapping by solving

$$\arg\min_{a,b} \sum_{(x,y)} C(x, y)\left(D^*(x, y) - \left(aF\left(I_0, \Theta\right)(x, y) + b\right)\right)^2 \quad (6)$$

While training our model Θ, during each evaluation of our loss we solve Eqn. 6 using a differentiable least squares solver (such as the one included in TensorFlow) to obtain a and b, which can be used to obtain absolute depth that can then be used to compute a standard view supervision loss. Note that since we only need to solve for two scalars, a sparse ground truth depth map with a few confident depth samples suffices.

4.2. Folded Loss

Our second strategy does not require ground truth depth and folds the optimization required to solve the affine parameters into the overall loss function. We associate variables a and b with each training example I_0 and define our loss function as:

$$\mathcal{L}_f(I_0, \Theta, a, b) = \sum_{(x,y)} \Delta\left(I_0(x, y), I_1\left(M\left(x, y; aF\left(I_0, \Theta\right) + b\right)\right)\right) \quad (7)$$

and then let the gradient descent optimize for Θ, $\{a^{(i)}\}$ and $\{b^{(i)}\}$ by solving

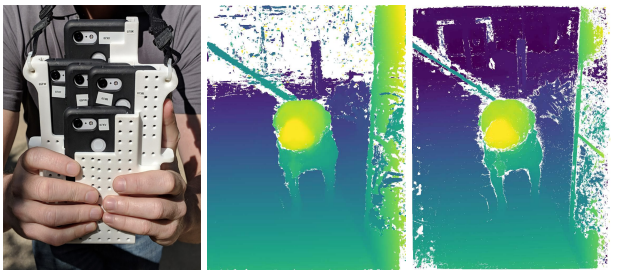
$$\arg\min_{\Theta, \{a^{(i)}\}, \{b^{(i)}\}} \sum_i \mathcal{L}_f(I_0^{(i)}, \Theta, a^{(i)}, b^{(i)}). \quad (8)$$

To avoid degeneracies as $a^{(i)}$ approaches zero, we parameterize $a^{(i)} = \epsilon + \log(\exp(a_\ell^{(i)}) + 1)$ where $\epsilon = 10^{-5}$. We initialize $\{a_\ell^{(i)}\}$ and $\{b^{(i)}\}$ from a uniform distribution in $[-1, 1]$. To train this model, we simply construct one optimizer instance in which Θ, $\{a_\ell^{(i)}\}$, and $\{b^{(i)}\}$ are all treated as free variables and optimized over jointly.

5. Data Collection

To train and evaluate our technique, we need dual-pixel data paired with ground-truth depth information. We therefore collected a large dataset of dual-pixel images captured in a custom-made capture rig in which each dual-pixel capture is accompanied by 4 simultaneous images with a moderate baseline, arranged around the central camera (Figure 4(a)). We compute "ground truth" depths by applying established multi-view geometry techniques to these 5 images. These depths are often incomplete compared to those produced by direct depth sensors, such as the Kinect or LIDAR. However, such sensors can only image certain kinds of scenes — the Kinect only works well indoors, and it is difficult to acquire LIDAR scans of scenes that resemble normal consumer photography. Synchronization and registration of these sensors with the dual-pixel images is also cumbersome. Additionally, the spatial resolutions of direct depth sensors are far lower than the resolutions of RGB cameras. Our approach allows us to capture a wide variety of high-resolution images, captured both indoors and outdoors, that resemble what people typically capture with their cameras: pets, flowers, etc (we do not include images of faces in our dataset, due to privacy concerns). The plus-shaped arrangement means that it is unlikely that a pixel in the center camera is not visible in at least one other camera (barring small apertures or very nearby objects) thereby allowing us to recover accurate depths even in partially-occluded regions. The cameras are synchronized using the system of [5], thereby allowing us to take photos from all phones at the same time (within ~ 16 milliseconds, or half a frame) which allows us to reliably image moving subjects. Though the inherent difficulty of the aperture problem means that our ground-truth depths are rarely perfect, we are able to reliably recover high-precision *partial* depth maps, in which high-confidence locations have accurate depths and inaccurate depths are flagged as low-confidence (Figures 4(b), 4(c)). To ensure that our results are reliable and not a function of some particular stereo algorithm, we compute two separate depth maps (each with an associated confidence) using two different algorithms: the established COLMAP stereo technique [45, 1], and a technique we designed for this task. See the supplement for a detailed description.

Our data is collected using a mix of two widely available consumer phones with dual-pixels: The Google Pixel 2 and the Google Pixel 3. For each capture, all 5 images are collected using the same model of phone. We captured 3,573 scenes resulting in $3{,}573 \times 5 = 17{,}865$ RGB and DP images. Our photographer captured a wide variety of images that reflect the kinds of photos people take with their camera, with a bias towards scenes that contain interesting nearby depth variation, such as a subject that is 0.5 - 2 meters away. Though all images contain RGB and DP information, for this work we only use the DP signal of the center camera. All other images are treated as conventional RGB images. We process RGB and DP images at a resolution of 1512×2016, but compute "ground truth" depth maps at half this resolution to reduce noise. We use inverse perspective sampling in the range 0.2 - 100 meters to convert absolute depth to inverse depth D^*. Please see the supplement for more details.



(a) Our capture rig (b) COLMAP depth (c) Our depth

(d) An example capture

Figure 4. Our portable capture rig with synchronized cameras (a) can be used to capture natural in-the-wild photos, where each central image is accompanied with 4 additional views (d). These multiple views allow us to use multi-view stereo algorithms to compute "ground truth" depths and confidences, as shown in (b) and (c) (low confidence depths are rendered as white).

Though our capture rig means that the relative positions of our 5 cameras are largely fixed, and our synchronization means that our sets of images are well-aligned temporally, we were unable to produce a single fixed intrinsic and extrinsic calibration of our camera rig that worked well across all sets of images. This is likely due to the lens not being fixed in place in the commodity smartphone cameras we use. As a result, the focus may drift due to mechanical strain or temperature variation, the lens may jitter off-axis while focusing, and optical image stabilization may move the camera's center of projection [14]. For this reason, we use structure from motion [24] with priors provided by the rig design to solve for the extrinsics and intrinsics of the 5 cameras individually for each capture, which results in an accurate calibration for all captures. This approach introduces a variable scale ambiguity in the reconstructed depth for each capture, but this is not problematic for us as our training and evaluation procedures assume an unknown scale ambiguity.

	AIWE(1)	AIWE(2)	$1 - \lvert\rho_s\rvert$
Folded Loss	.0225	.0318	.195
3D Assisted Loss	**.0175**	**.0264**	**.139**

Table 1. Accuracy of DPNet trained with two different methods. Our "3D Assisted Loss", which has access to ground truth depth to fully-constrain the ambiguity, tends to outperform the alternative approach of our "Folded Loss", which circumvents the lack of known depth by folding an optimization problem into the computation of the loss function during training.

6. Experiments

We describe our data, evaluation metrics and method of training our CNN for depth prediction. In addition, we compare using affine-invariant losses to using scale-invariant and ordinary losses and demonstrate that affine-invariant losses improve baseline methods for predicting depth from dual-pixel images.

6.1. Data Setup

Following the procedure of [50], we center crop our dual-pixel images to 66.67% of the original resolution to avoid spatially varying effects in dual-pixel data towards the periphery, and to remove the need for radial distortion correction. We do not downsample the dual-pixel images, as doing so would destroy the subtle disparity and blur cues they contains. After cropping, the input to our network is of resolution 1008×1344 while the output is 504×672, i.e., the same resolution as our ground truth depth. Our evaluation metrics are computed on a center crop of the output of size 384×512, as the center of the image is where our additional stereo views are most likely to overlap with the center view. We randomly split our data into train and test sets, under the restriction that all images from each capture session are contained entirely within one of the two sets. Our training and test sets contains $2{,}757$ and 718 images respectively. During training we use only our own ground-truth depth, though we evaluate on both our depth and COLMAP's depth. COLMAP's SfM failed to converge on 47 images in our test set, so we report the mean error of the remaining 671 images.

6.2. Training a Neural Net for Depth Prediction

Now that we have defined our loss function and our dataset, we can construct a neural network architecture for our task for predicting depth from dual-pixel and RGB images. We use both the VGG model architecture similar to [21] and a lightweight network (DPNet) similar to a U-Net [42] with residual blocks [26]. While the VGG model has ~ 19.8 million parameters and ~ 295 billion flops per inference, the DPNet has only ~ 0.24 million parameters and ~ 5.5 billion flops. The architectures are detailed in the supplement.

For our difference $\Delta\left(I_0, I_j\right)$ between the source image I_0 and the warped image I_j from the j^{th} neighbor, we use a weighted combination of a DSSIM loss and a Charbonnier loss with weights set to 0.8 and 0.2 respectively. Our DSSIM loss is the same as that of [21]: a window of size 3×3, with $c_1 = 0.01^2$ and $c_2 = 0.03^2$. The Charbonnier loss is computed by setting $\alpha = 1$ and $c = 0.1$ in the parametrization described in [8]. Images are normalized to $[0, 1]$ range and the losses are computed on three channel RGB images with the losses per channel averaged together. Similar to [21, 52], we predict depths at multiple resolutions (5 for DPNet and 3 for VGG), each scaled down a factor of 2, and aggregate losses across them.

To adapt the view supervision loss for stereo images (Eqn. 5) to multi-view data, we use the approach of [22] and compute $\Delta\left(I_0, I_j\right)$ for each neighbor and then take per-pixel minimum using the heuristic that a pixel must be visible in at least one other view, which applies for our case since the neighboring views surround the center view in the capture rig.

Our implementation is in Tensorflow [2] and trained using Adam [35] with a learning rate of 0.001 for 2 million steps with a batch size of 4 for the lightweight model and 2 for the VGG model. Our model weights are initialized randomly using Tensorflow's default initialization [20]. We perform data augmentation by applying uniformly random translations to our imagery, limiting maximum translation in either direction to 10 pixels.

6.3. Evaluation Metrics

The optics of dual-pixel cameras means that that we should not expect the depth estimated from dual-pixel imagery to be accurate in absolute terms — at best, it should be accurate up to some unknown affine transformation. This ambiguity prohibits the use of conventional metrics (such as those used by the Middlebury Stereo benchmark [44]) for evaluating the depth maps estimated from dual-pixel imagery, and requires that we construct metrics that are invariant to this ambiguity.

Instead, we use a weighted-variant of Spearman's rank correlation ρ_s, which evaluates the ordinal correctness of the estimated depth with ground truth depth confidences as weight. In addition, we use affine invariant weighted versions of MAE and RMSE, denoted AIWE(1) and AIWE(2) respectively. Please see the supplemental for details.

6.4. Folded Loss vs 3D Assisted Loss

Our first experiment is to investigate which of our two proposed solutions for handling affine invariance during training performs best. Training our DPNet with both approaches, as shown in Table 1, shows that the 3D assisted loss (Sec. 4.1) converges to a better solution than the folded loss (Sec. 4.2). We therefore use our 3D assisted loss in all

Method	Invariance	Evaluated on Our Depth			Evaluated on COLMAP Depth			Geometric
		AIWE(1)	AIWE(2)	$1-\lvert\rho_s\rvert$	AIWE(1)	AIWE(2)	$1-\lvert\rho_s\rvert$	Mean
RGB Input								
DPNet	None	.0602	.0754	.631	.0607	.0760	.652	.1432
	Scale	.0409	.0544	.490	.0419	.0557	.514	.1047
	Affine	**.0398**	**.0530**	**.464**	**.0410**	**.0546**	**.493**	**.1014**
DORN [18] (NYUDv2 model)		.0421	.0555	.407	.0426	.0557	.419	.0990
DORN [18] (KITTI model)		.0490	.0631	.549	.0492	.0630	.558	.1196
RGB + DP Input								
DPNet	None	.0581	.0735	.827	.0587	.0742	.834	.1530
	Scale	.0202	.0295	.162	.0213	.0322	.178	.0477
	Affine	**.0175**	**.0264**	**.139**	**.0190**	**.0298**	**.156**	**.0422**
VGG	None	.0370	.0492	.350	.0383	.0513	.360	.0876
	Scale	.0224	.0321	.181	.0242	.0356	.208	.0535
	Affine	**.0186**	**.0275**	**.149**	**.0202**	**.0308**	**.166**	**.0446**
Godard *et al*. [21] (ResNet50)	None†	.0562	.0714	.738	.0568	.0720	.745	.1442
	Scale†	.0260	.0367	.227	.0270	.0383	.239	.0613
	Affine†	**.0251**	**.0356**	**.222**	**.0257**	**.0366**	**.232**	**.0592**
Garg *et al*. [19] (ResNet50)	None	.0571	.0722	.761	.0577	.0728	.772	.1472
	Scale†	.0261	.0369	.228	.0267	.0382	.237	.0613
	Affine†	**.0248**	**.0352**	**.216**	**.0255**	**.0365**	**.227**	**.0584**
Wadhwa *et al*. [50]		.0270	.0375	.236	.0276	.0388	.245	.0630

Table 2. Accuracy of different models and approaches evaluated on our depth and COLMAP depth with the right-most column containing the geometric mean of all the metrics. For models trained with different degrees of invariance, the best-performing invariance's score is bolded. The overall best-performing technique is highlighted in yellow. A † indicates that we use only DP images as input to a model, which we do if it produces better results compared to using RGB+DP input.

of the following experiments.

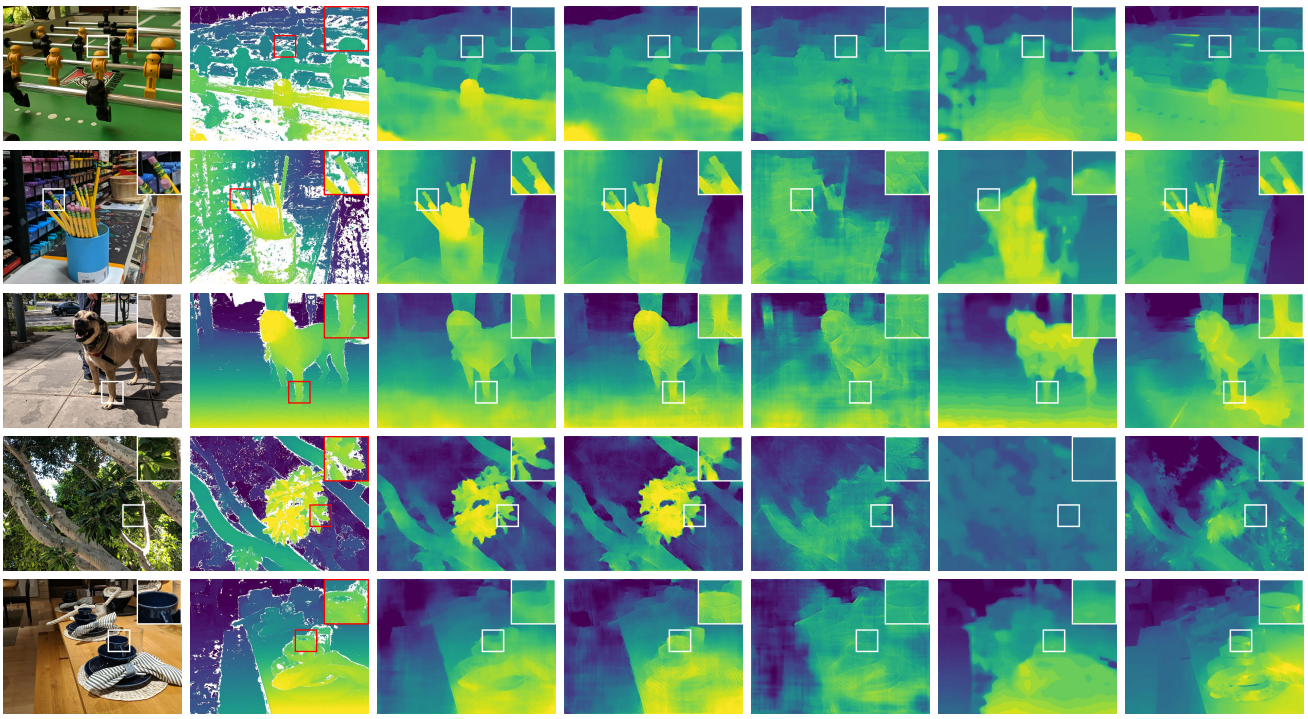
6.5. Comparison to Other Methods

We show that our models trained with affine invariant loss have higher accuracy than those trained with conventional losses. Our loss also improves the accuracy of existing view-supervision based monocular depth estimation methods when applied to dual-pixel data. As benchmarks, we compare against Fu *et al*. [18], the current top performing monocular depth estimation algorithm on the KITTI [39] and ScanNet [13] benchmarks, which has been trained on large pre-existing external RGBD datasets, and Wadhwa *et al*. [50], that applies classical stereo methods to recover depth from dual-pixels.

We evaluate our affine invariant loss against two baseline strategies: a scale invariant loss, and no invariance. Scale-invariance is motivated by the well-understood inherent scale ambiguity in monocular depth estimation, as used by Eigen *et al*. [16]. No-invariance is motivated by view-supervised monocular depth estimation techniques that directly predict disparity [19, 21]. We implement scale invariance by fixing $b = 0$ in Eqn. 6.

Our affine-invariant loss can also be used to enable view-supervised monocular depth estimation techniques [19, 21] to use dual-pixel data. Since they require stereo data for training, we used images from the center and top cameras of our rig as the left and right images in a stereo pair. The technique of Godard *et al*. [21] expects rectified stereo images as input, which is problematic because our images are not rectified, and rectifying them would require a resampling operation that would act as a lowpass filter and thereby remove much of the depth signal in our dual-pixel data. We circumvent this issue by replacing the one dimensional bilinear warp used by [21] during view supervision with a two dimensional warp based on each camera's intrinsics and extrinsics. We also remove the scaled sigmoid activation function used when computing disparities, which improved performance due to our disparities being significantly larger than those of the datasets used in [21]. We also decreased the weight of the left-right consistency loss by 0.15 times to compensate for our larger disparities. We use the ResNet50 [26] version of [21]'s model as they report it provides the highest quality depth prediction. We also used the codebase of [21] to implement a version of [19] by removing the left-right consistency loss and the requirement that the right disparity map be predicted from the left image.

We show quantitative results in Table 2, and visualizations of depth maps in Fig. 5 (see the supplement for additional images). Using our affine-invariant loss instead of scale-invariant or not-invariant loss improves performance



(a) Input Image | (b) GT Depth | (c) DPNet (Affine) RGB + DP | (d) DPNet (Scale) RGB + DP | (e) DPNet (Affine) RGB | (f) DORN [18] RGB | (g) Wadhwa *et al.* RGB + DP [50]

Figure 5. Input images (a) from the test set of our dataset, their ground-truth depth (b), the output of DPNet with RGB + DP input trained with affine invariance (c) and scale invariance (d), output of DPNet with RGB input trained with affine invariance (e), and as baselines, the output of [18] trained on the NYUDv2 dataset [46] (f) and the output of [50] (g). An affine transform has been applied to all visualizations to best fit the ground truth. Results from [50] exhibit fine details due to the use of bilateral smoothing [11] as a post process but otherwise show many depth errors, e.g., the shadow of the dog on the ground. DORN [18] lacks fine details and fails to generalize to the variety of scenes in our dataset. For DPNet, results with RGB + DP input are better than results with RGB input. RGB + DP input with affine invariance yields better results than scale invariance, e.g., the space between the pencils in the second image. Best seen zoomed-in in an electronic version.

for all models where different degrees of invariance are investigated. In particular, while VGG is more accurate than DPNet when using no invariance, affine invariance allows the small DPNet model to achieve the best results. In comparing the performance of DPNet with and without DP input, we see that taking advantage of dual-pixel data produces significantly improved performance over using just RGB input. While [18] trained on external RGBD datasets performs well on this task when compared to our model trained on just RGB data, its accuracy is significantly lower than our model and many baseline models trained on dual-pixel data with our affine-invariant loss, thereby demonstrating the value of DP imagery.

7. Conclusion

In summary, we have presented the first learning based approach for estimating depth from dual-pixel cues. We have identified a fundamental affine ambiguity regarding depth as it relates to dual-pixel cues, and with this observation we have developed a technique that allows neural networks to estimate depth from dual-pixel imagery despite this ambiguity. To enable learning and experimentation for this dual-pixel depth estimation task, we have constructed large dataset of 5-view in-the-wild RGB images paired with dual-pixel data. We have demonstrated that our learning technique enables our model (and previously published view-supervision-based depth estimation models) to produce accurate, high-quality depth maps from dual-pixel imagery.

Acknowledgements

We thank photographers Michael Milne and Andrew Radin for collecting data. We also thank Yael Pritch and Marc Levoy for technical advice and helpful feedback.

References

[1] COLMAP. `https://colmap.github.io/`.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[3] Edward H Adelson and James R Bergen. The plenoptic function and the elements of early vision. *Computational Models of Visual Processing*, 1991.

[4] Edward H Adelson and John YA Wang. Single lens stereo with a plenoptic camera. *TPAMI*, 1992.

[5] Sameer Ansari, Neal Wadhwa, Rahul Garg, and Jiawen Chen. Wireless software synchronization of multiple distributed cameras. *ICCP*, 2019.

[6] Sam Y. Bae, Ronald J. Korniski, Michael Shearn, Harish M. Manohara, and Hrayr Shahinian. 4-mm-diameter three-dimensional imaging endoscope with steerable camera for minimally invasive surgery (3-D-MARVEL). *Neurophotonics*, 2016.

[7] Ruzena Bajcsy and Lawrence Lieberman. Texture gradient as a depth cue. *CGIP*, 1976.

[8] Jonathan T. Barron. A general and adaptive robust loss function. *CVPR*, 2019.

[9] Jonathan T. Barron, Andrew Adams, YiChang Shih, and Carlos Hernández. Fast bilateral-space stereo for synthetic defocus. *CVPR*, 2015.

[10] Jonathan T. Barron and Jitendra Malik. Shape, albedo, and illumination from a single image of an unknown object. *CVPR*, 2012.

[11] Jonathan T. Barron and Ben Poole. The fast bilateral solver. *ECCV*, 2016.

[12] Michael Brady and Alan Yuille. An extremum principle for shape from contour. *TPAMI*, 1984.

[13] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CVPR*, 2017.

[14] Stephen DiVerdi and Jonathan T. Barron. Geometric calibration for mobile, stereo, autofocus cameras. *WACV*, 2016.

[15] Alexey Dosovitskiy, Philipp Fischery, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *ICCV*, 2015.

[16] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *NIPS*, 2014.

[17] John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. Deepstereo: Learning to predict new views from the world's imagery. *CVPR*, 2016.

[18] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. *CVPR*, 2018.

[19] Ravi Garg, BG Vijay Kumar, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. *ECCV*, 2016.

[20] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. *AISTATS*, 2010.

[21] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CVPR*, 2017.

[22] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Digging into self-supervised monocular depth estimation. *CoRR*, 2018.

[23] P. Grossmann. Depth from focus. *Pattern Recognition Letters*, 1987.

[24] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[25] Caner Hazrba, Sebastian Georg Soyer, Maximilian Christian Staab, Laura Leal-Taix, and Daniel Cremers. Deep depth from focus. *ACCV*, 2018.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016.

[27] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Automatic photo pop-up. *SIGGRAPH*, 2005.

[28] Derek Hoiem, Alexei A. Efros, and Martial Hebert. Putting objects in perspective. *CVPR*, 2006.

[29] Berthold KP Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. 1970.

[30] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *NIPS*, 2015.

[31] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T. Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3-d object dataset: Putting the kinect to work. *ICCV Workshops*, 2011.

[32] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and Inso Kweon. Accurate depth map estimation from a lenslet light field camera. *CVPR*, 2015.

[33] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon. Depth from a light field image with learning-based matching costs. *PAMI*, 41(2):297–310, 2019.

[34] Huaizu Jiang, Erik G. Learned-Miller, Gustav Larsson, Michael Maire, and Greg Shakhnarovich. Self-supervised depth learning for urban scene understanding. *ECCV*, 2018.

[35] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[36] Anat Levin, Rob Fergus, Frédo Durand, and William T Freeman. Image and depth from a conventional camera with a coded aperture. *SIGGRAPH*, 2007.

[37] Marc Levoy and Pat Hanrahan. Light field rendering. *SIGGRAPH*, 1996.

[38] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. *CVPR*, 2018.

[39] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. *CVPR*, 2015.

[40] Ren Ng. Fourier slice photography. *SIGGRAPH*, 2005.

[41] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. *Stanford University: Computer Science Technical Report*, 2005.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *MICCAI*, 2015.

[43] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. Learning depth from single monocular images. *NIPS*, 2006.

[44] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2002.

[45] Johannes L. Schönberger, Enliang Zheng, Jan Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. *ECCV*, 2016.

[46] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. *ECCV*, 2012.

[47] Pratul P. Srinivasan, Rahul Garg, Neal Wadhwa, Ren Ng, and Jonathan T. Barron. Aperture supervision for monocular depth estimation. *CVPR*, 2018.

[48] Supasorn Suwajanakorn, Carlos Hernandez, and Steven M. Seitz. Depth from focus with your mobile phone. *CVPR*, 2015.

[49] Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. *ICCV*, 2013.

[50] Neal Wadhwa, Rahul Garg, David E. Jacobs, Bryan E. Feldman, Nori Kanazawa, Robert Carroll, Yair Movshovitz-Attias, Jonathan T. Barron, Yael Pritch, and Marc Levoy. Synthetic depth-of-field with a single-camera mobile phone. *SIGGRAPH*, 2018.

[51] Jure Žbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. *CVPR*, 2015.

[52] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. *CVPR*, 2017.