Deep Depth from Focus with Differential Focus Volume

Fengting Yang   Xiaolei Huang   Zihan Zhou

The Pennsylvania State University

{fuy34, suh972, zuz22}@psu.edu

Abstract

Depth-from-focus (DFF) is a technique that infers depth using the focus change of a camera. In this work, we propose a convolutional neural network (CNN) to find the best-focused pixels in a focal stack and infer depth from the focus estimation. The key innovation of the network is the novel deep differential focus volume (DFV). By computing the first-order derivative with the stacked features over different focal distances, DFV is able to capture both the focus and context information for focus analysis. Besides, we also introduce a probability regression mechanism for focus estimation to handle sparsely sampled focal stacks and provide uncertainty estimation to the final prediction. Comprehensive experiments demonstrate that the proposed model achieves state-of-the-art performance on multiple datasets with good generalizability and fast speed.

1. Introduction

Recovering depth using a single RGB camera is a critical problem in 3D vision. Many applications can benefit from this technique, including 3D reconstruction, virtual and augmented reality, and image editing. In the literature, various cues have been explored to tackle the problem, such as focus [33], ego-motion [40], and structured-light patterns [3]. But cues like ego-motion and structured-light patterns require either extra motion or additional devices, which limit their applications on smart devices and hand-held cameras. In contrast, the focus (or defocus) cue is what we can gain almost for "free." To capture a well-focused image, many digital cameras rapidly sweep a focal plane in its focus range, resulting in a series of images (*i.e.*, a focal stack) with different focal distances. A pioneering work [45] has shown that depth can be inferred from focal stacks taken by a mobile phone.

Image context is another "free" cue we can get from pictures. Based on it, a series of works [9, 11, 35, 36] have shown the ability to infer depth from a single image. But it is still a challenge to generalize these single-view methods to unknown scenes. Plus, due to the scale ambiguity,

one cannot estimate the absolute depth from a single image without scene priors, even if the camera is well-calibrated.

In this work, we utilize both the focus and context cues and develop a deep depth-from-focus (DFF) network for focus analysis and depth inference. We prefer DFF over the other focus-based technique, namely depth-from-defocus (DFD), because of its generalizability. For DFD, a mathematical model between the depth and the defocus pattern needs to be established. These methods commonly assume the object point is centralized at a small plane and the point spread function (PSF) follows a certain distribution [25]. But such assumptions may not hold in real-world scenarios. DFF only assumes there is one and only one best-focused frame for a pixel in a focal stack, which is guaranteed in theory [39] for thin lens cameras if the sampling focal distances are dense enough. The focal distance of the best-focused frame can serve as the pixel depth estimation.

Several major challenges still exist in DFF. The first one is the focus measure design. A large number of focus measures have been proposed, but none of them is perfect in the wild environment [34]. The second one is regarding textureless regions. As the responses to focus measures stay low in textureless regions, the context information has to be used to infer the focus status [4, 10]. The third one is the high sampling frequency requirement. In theory, for an object point falls into the depth of field of a camera, there must be one sharpest pixel in the focal sweep. But it may not be visible if the sampling frequency is low. Thus, many traditional DFF methods [29, 44] take tens of frames as input, which limits their running speed.

To overcome these challenges, we propose to learn a deep focus measure with the convolutional neural network (CNN). The global information embedded in the deep features extracted by the CNN can help focus analysis in textureless regions. Further, the focus status is represented as a probability distribution of the best-focused pixels, which can be learned from a limited number of frames. Note that earlier works [6, 13, 47] have also tried to tackle the DFF problem with CNN. But their networks are adopted from either the general dense prediction tasks [6, 13] or the video representation task [47], not specifically designed for DFF.

Our network is inspired by the close relationship between stereo matching and DFF [22, 39] and the recent success of deep stereo matching [5, 26, 50]. The network employs a 2D CNN to learn the deep focus and context representations, stacks them into 4D focus volumes (FVs), and computes the differences in their frame dimension to build deep differential focus volumes (DFVs). The DFVs are later processed by a 3D CNN to predict the best-focused probability of pixels. The probabilistic prediction helps locate the best-focused frames for pixels even if they are not visible in the focal stack, thus decreasing the required focal stack size. Finally, the depth estimation is obtained by probability regression. Our pipeline is analogous to traditional FV-based DFF approaches [10, 24, 44], which first compute the in-focus score using hand-crafted measures and then aggregate the score volume to obtain the final focus analysis for depth estimation. Besides depth prediction, our model also provides an uncertainty estimation that shows the reliability of the prediction. To the best of our knowledge, this is the first work that introduces the deep 4D focus volume to DFF.

We have conducted comprehensive experiments to evaluate our method on both synthetic and real-world datasets. Our method achieves state-of-the-art performance while consuming a small number of frames (*e.g.*, five) per focal stack, outperforming all DFF and DFD baselines. Our method also generalizes well to unknown scenes without fine-tuning. Further, the model runs at about 18.2 ms/stack with 256×256 input resolution and about 33.3 ms/stack with 383×552 input resolution on an NVIDIA 1080Ti GPU, making it suitable for potential real-time applications.

2. Related Works

Depth-from-Focus (DFF). DFF is also known as shape-from-focus. Due to physical constraints, most cameras can only capture a clear image of objects in a certain range called depth of field (DoF). A point outside the range appears to be blurry and forms a circle of confusion (CoC). Researchers have been utilizing this phenomenon to infer depth for a long time [30, 32, 34, 42]. According to the thin lens model [39], given a densely sampled focal stack, there must be one and only one best-focused (sharpest) frame for a pixel. The focal distance of the frame can serve as the depth estimation of the pixel with the uncertainty of DoF.

In practice, however, if the sample rate of the focal stack is low, such a frame may not always be visible. Therefore, traditional DFF methods usually take tens of frames in a focal stack as input and focus on finding good focus measures to identify the sharpest pixels. A number of focus measures have been proposed, including gradient-based measures [30], Laplacian-based measures [1, 31], frequency transformation-based measures [48], and statistic-based measures [21]. We refer readers to [34] for a comprehensive review. More recently, [44] proposes a ring differ-

ence filter measure, [37] combines multiple measures into a composite, and [29] attempts to solve the noisy prediction with a variational framework. Deep DFF methods are proposed in [6, 13, 47]. However, [13] and [6] adopt their network from general dense prediction tasks, and the network of [47] is from a video representation work [2]. None of them fully considers the specialty of DFF.

Focus Volume and Cost Volume. In DFF, focus volume (FV) is commonly used to store the "in-focus" score computed by focus measures. Once the FV is built, a naive approach can take the focal distance of the pixels with the highest scores as the depth estimation. But such an estimation is likely to be noisy due to the imperfect focus measurements. Thus, most works take the initial FV as features, and aggregate it with the average filter [16, 31], bilateral filter [41], and guided filter [18] for better depth estimation.

It is worth noting that a similar concept called "cost volume" is popular in traditional matching tasks, *e.g.*, stereo matching [38, 51] and optical flow [49], where the matching score rather than the in-focus score is stored and later aggregated. [44] shows that the traditional cost aggregation method proposed for stereo matching [51] can be adapted to aggregate FV in DFF. With the advance in deep learning, many methods [5, 7, 8, 17, 20, 26, 43, 50] build deep cost volume to incorporate context information and perform cost aggregation with 2D or 3D CNN to improve the estimation accuracy. In this work, we show that the core idea of deep cost volume and deep cost aggregation is also suitable for DFF. Moreover, we propose a deep differential focus volume by considering the special characteristics of DFF.

3. Methodology

Figure 1 presents the overall pipeline of our method. As with [13, 47], we assume the input focal stack is pre-aligned by optical flow [45], homography [18], or any other methods. The network first extracts image features using a shared 2D CNN, and then builds differential focus volumes by computing the first-order derivative on the stacked features. This volume is further processed by a 3D CNN to predict the best-focused probabilities. Finally, multi-scale depth predictions are obtained by probability regression.

In the following, we first briefly review the deep cost volume (Sec. 3.1) and describe the proposed deep focus volume (Sec. 3.2) and deep differential focus volume (Sec. 3.3) for the DFF task. Then, we introduce our probability regression and uncertainty estimation approach (Sec. 3.4) and discuss the implementation details (Sec. 3.5).

3.1. A Brief Review of Deep Cost Volume

Deep cost volume is widely used in stereo matching [5, 26, 28, 50] and optical flow [8, 43, 46] nowadays. Although the exact implementation varies, there are two main types
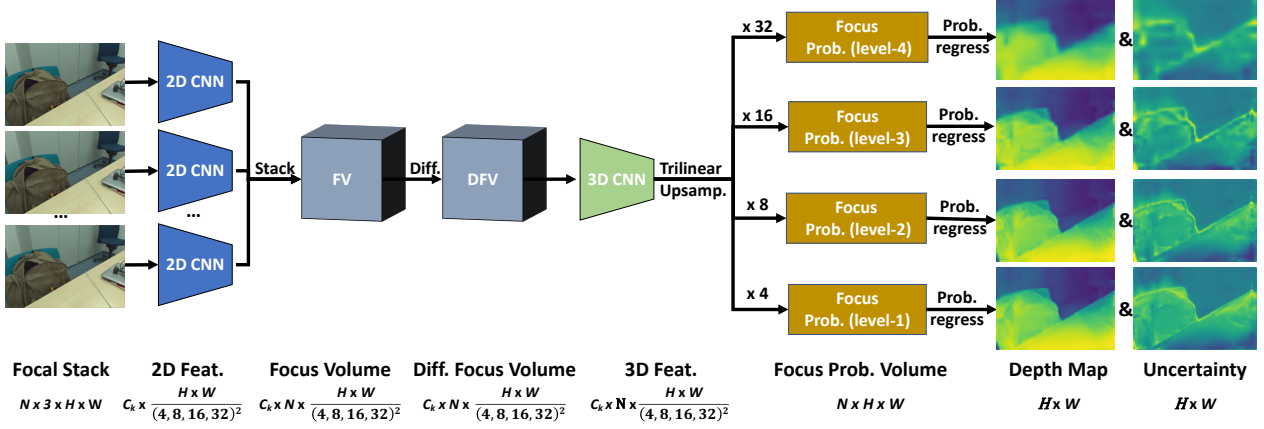
Figure 1. Our multi-scale DFF network. Given an N-frame focal stack with $H \times W$ resolution, a shared 2D CNN first processes the frames to build the 4D FVs, where each layer corresponds to the features from one frame. The volume is then differentiated to obtain the DFVs, and sent to 3D CNN to predict the focus probabilities. The depth is estimated by the weighted sum in the probability regression, and the standard deviation is calculated to indicate the prediction uncertainty. The denominator in the feature dimension indicates the resolution scale of the corresponding level.

of volume designs: 3D cost volume and 4D cost volume.

In both designs, input frames are first processed by a 2D CNN to extract deep features (feature channel × height × width). The main difference is, for 3D cost volume [8, 28], a hand-crafted similarity measure (*e.g.*, cross-correlation) is applied to the deep features to generate the cost volume, whereas for 4D cost volume [5, 26, 50], the deep features are stacked in a new "disparity" dimension, resulting in a 4D representation (feature channel × disparity × height × width). Every index in the disparity dimension represents a disparity proposal. Another difference is that the 3D cost volume is aggregated by 2D CNN to directly predict disparity values, but the 4D volume is followed by 3D CNN to produce a probability for each proposal. Although the 3D cost volume involves less computation, the 4D cost volume has been reported as a better implementation to integrate the context information and improve the model accuracy.

3.2. Deep Focus Volume

As discussed in Sec. 2, there is a long history of using focus volume (FV) in DFF. Compared to cost volume in stereo matching and optical flow, the only difference is that the FV is used to find the best-focused (sharpest) pixels rather than the best-matched pairs. Given the two deep cost volume designs described in Sec. 3.1, we can naturally develop two ways to build the deep FV in a DFF network. On the one hand, we can construct a 3D FV using hand-crafted focus measures and process it with 2D CNN to predict best-focused frames. On the other hand, we can construct a 4D FV by stacking the features in a new "frame" dimension and use 3D CNN to produce a probability for each best-focused frame proposal. As we wish to eliminate the use of hand-crafted measures, we adopt the latter approach.
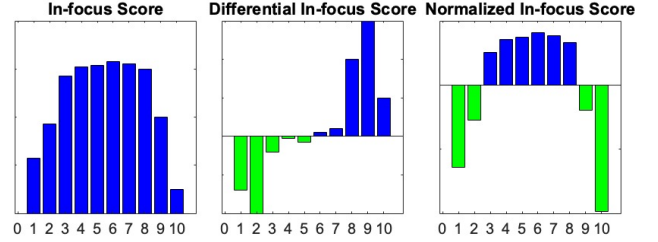

Figure 2. Illustration of differential features and normalized features. The blue and green colors indicate positive and negative values, respectively.

With this, our network would be the same as the one in Figure 1, except that the DFVs are excluded. In Sec. 4, we demonstrate that this simple approach already achieves state-of-the-art performance on multiple datasets.

3.3. Deep Differential Focus Volume

Despite the excellent performance of deep FV, we argue that such a design does not fully utilize the characteristics of DFF. In theory, in textured regions, DFF does not suffer from matching ambiguity if the comparing patch is larger than the widest CoC [32, 39]. This means that if we have a good focus measure and operate it on a sufficiently large patch, there is always a single extremum on the pixel's in-focus curve, indicating the best-focused status. This is different from stereo matching or optical flow, where a reference patch may be matched with multiple targets due to repeated patterns.

We utilize this single extremum and propose the deep differential focus volume. It is well-known that the gradient is a good indicator for the single extremum. This is illustrated in Figure 2. For a single pixel in a 10-frame fo-

cal stack, we pick the positive in-focus score as the feature. Sometimes, due to weak texture, the feature magnitude of the sharpest pixel may not be salient - see the scores of frames 4 to 8 in the left figure. Consequently, it could be hard for the network to locate the sharpest pixel. However, if we take the difference in the feature, the sharpest pixel will correspond to the zero-crossing, as the middle figure shows. This is a more significant pattern to distinguish. We also compare the effect between differential and normalization (the right figure). While normalization can enlarge the relative score gaps, it does not make the sharpest pixel as prominent as the differential one does.

Therefore, we differentiate the focus volume Q along the frame dimension to build the differential focus volume:

$$V^i = \begin{cases} Q^i - Q^{i+1}, & i = 1, \ldots, N-1 \\ Q^i, & i = N \end{cases} \quad (1)$$

where i is the index of the frame dimension, and N is the total number of frames. Note that the first $N-1$ dimensions of V contain the first-order feature derivatives of adjacent frames, whereas the last dimension is equal to the features of frame N, containing the original context information. Thus, the 3D CNN has access to both differential features and context features for focus analysis. This is important, especially for textureless regions. Arguably, because of defocus, some parts of the last frame may be blurry. But it can still serve as "context." As the purpose of the context is to capture global relationships among image regions, the lack of certain fine details will not impair its effectiveness. Nevertheless, better ways to gather the context information can be studied in future work. For example, we may perform an average pooling or convolution over Q along the frame dimension to learn context from the entire stack.

According to the universal approximation theorem [15], the network might be able to eventually learn a similar representation by itself. But we believe explicitly introducing the known prior to the network will help the learning process, just as the deep cost volume for the matching tasks.

3.4. Depth Regression and Uncertainty Estimation

Besides textureless regions, the required number of frames is another issue in classic DFF. Because classic DFF methods infer the depth by locating the sharpest pixels in a focal stack, they usually need tens of frames as input. Otherwise, the sharpest pixels may be invisible, and the depth inference will be inaccurate. In this work, we locate the sharpest pixel in sub-frame accuracy by learning a probability distribution.

The final output of our network is a focus probability volume P, where p_j^i indicates the probability that the pixel x_j in the ith frame is the best-focused. The sum of the probability of pixel x_j is constrained to 1 by softmax activation. Thus, the best-focused frame ID of pixel x_j is

$\hat{i}_j = \sum_{i=1}^N p_i^j \cdot i$. Note, $\hat{i}_j$ can be a float value that indicates the best-focused frame locates between $\lfloor \hat{i}_j \rfloor$ and $\lceil \hat{i}_j \rceil$. Similarly, we can obtain the depth $\hat{d}_j$ as

$$\hat{d}_j = \sum_{i=1}^N p_j^i \cdot l^i, \quad (2)$$

where l^i is the focal distance of the ith frame. If the camera is calibrated, we take the actual value of l^i. Otherwise, we assume the frames are sorted with ascending focal distances and set the distance from 0 to 1. In practice, the sorting can be achieved by first estimating the up-to-scale focal distances with the method proposed in [45].

The prediction confidence of the network can be revealed as the uncertainty Φ. We compute the uncertainty of pixel x_j using the weighted standard deviation:

$$\phi_j = \sqrt{\sum_{i=1}^N p_j^i \cdot (l^i - \hat{d}_j)^2}. \quad (3)$$

Besides serving as a confidence indicator, this uncertainty measure may also be used in scenarios such as multi-task learning [19] and measurement fusion [52].

3.5. Implementation Details

Our network design is adopted from [50] and optimized for the DFF task. For the 2D CNN, we employ a ResNet-18-FPN [23] which is pre-trained on ImageNet. A spatial pyramid pooling (SPP) [53] module is inserted between the encoder and decoder to better capture local and global information. We take the last 4-level features and build four DFVs at corresponding resolutions. For each DFV, we use two 3D-ResNet blocks [12] followed by a 3D-SPP for aggregation. A 2-layer 3D convolution is used to predict the focus probability. The aggregated DFV is also sent to the next scale level after upsampling and a 3D convolution. Please refer to our supplementary material for more details.

We implement our model using Pytorch and optimize it with Adam ($\beta_1 = 0.9$ and $\beta_2 = 0.999$) on one NVIDIA 1080Ti GPU for 700 epochs. The batch size is 20, and the learning rate is 1×10^{-4}. Given a training focal stack, we first randomly select 5 frames and crop them into 224×224 resolution with random-flipping afterward. All the frames are organized with the ascending focal distance order.

At training time, we take all 4-level outputs and compare the predicted pixel depth $\hat{d}_j^s$ with the ground truth depth d_j^s, using a multi-scale smooth L1 loss, L. At test time, we only output the largest scale (level-1 in Figure 1) focus probability volume for depth regression.

$$L = \sum_{s=1}^4 \alpha_s \left(\frac{1}{M} \sum_{j=1}^M smooth_{L_1}(d_j^s - \hat{d}_j^s) \right), \quad (4)$$

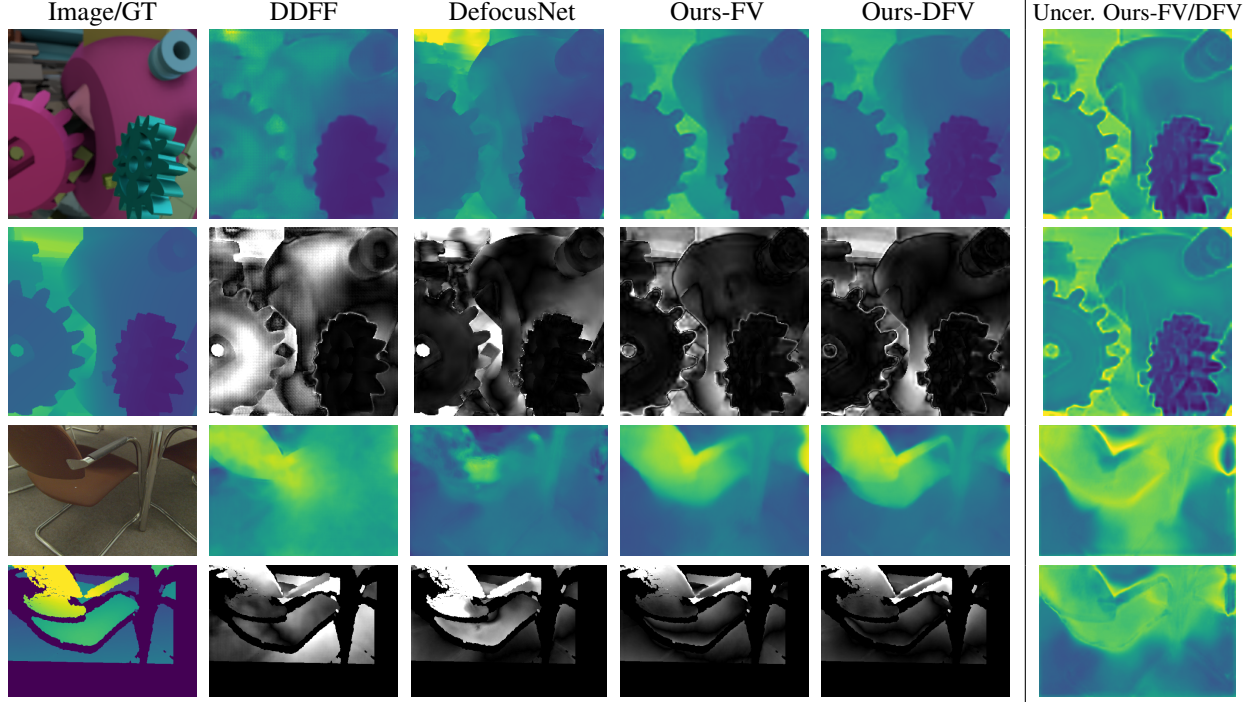| Image/GT | DDFF | DefocusNet | Ours-FV | Ours-DFV | Uncer. Ours-FV/DFV |

Figure 3. Qualitative results on FoD500 (first two rows) and DDFF-12 (last two rows). The first column shows the first image in the input focal stack and the corresponding ground truth. The next 4 columns show depth (Row 1) and disparity (Row 3) predictions, and the corresponding error maps (Rows 2 and 4). The last column shows the uncertainty maps of Ours-FV (Rows 1 and 3) and Ours-DFV (Rows 2 and 4). The warmer or brighter the color, the higher the value.

where M is the total number of pixels, and we set $\alpha_s = \left\{ \frac{8}{15}, \frac{4}{15}, \frac{2}{15}, \frac{1}{15} \right\}$ for all experiments.[1]

4. Experiments

We conduct comprehensive experiments to study the model performance. First, we use two annotated datasets, FoD500 [27] and DDFF-12 [13], for quantitative and qualitative comparison. Next, we conduct ablation studies on the model's sensitivity to the focal stack size and the multi-scale architecture. Finally, we test the model generalizability on an unlabeled dataset, Mobile depth [45].

4.1. Datasets

FoD500 [27] is a synthetic DFD dataset[2], containing 400 training samples and 100 test samples. Every sample has a 5-frame focal stack with known focal distances and a ground truth depth map. The image resolution is 256×256. As the dataset is originally designed for DFD, the focal distance range of a sample does not always cover its ground truth depth range. We mask pixels whose depth values are out of the focal distance range at both training and test time.

DDFF-12 [13] is a real-world DFF dataset captured by a light-field camera from 12 different scenes. Six scenes,

glassroom, kitchen, office41, seminaroom, socialcorner, studentlab, are selected as the training set, each containing 100 samples. The other six scenes, *cafeteria, library, lockeroom, magistrale, office44, spencerlab*, are chosen as the test set with 20 samples per scene. Each sample contains a 10-frame focal stack with known focal disparities and a ground truth disparity map. The image resolution is 383×552. We further divide the original training set into 4 training scenes (*kitchen, seminaroom, socialcorner, studentlab*) and 2 validation scenes (*glassroom, office41*) by random selection. Following [13], we evaluate the disparity accuracy on this dataset.

Mobile depth [45] is a real-world DFF dataset captured by a mobile phone. The dataset consists of 11 aligned focal stacks and 2 unaligned stacks from 11 scenes. The image resolution varies between 360×640 and 518×774, and the number of frames ranges from 14 to 33 per stack. Neither ground truth depth nor the focal distance is provided. As [45] only releases their results on the 11 aligned focal stacks and no code is available, we only evaluate our method on the aligned scenes for qualitative comparison.

4.2. Comparison to the State-of-the-Art

To evaluate the model performance, we compare our differential focus volume network, denoted as "Ours-

---

[1]Code is available at https://github.com/fuy34/DFV

[2]https://github.com/dvl-tum/defocus-net

Method	MSE ↓	RMS↓	log RMS ↓	Abs. rel.↓	Sqr. rel.↓	δ ↑	δ^2 ↑	δ^3 ↑	Bump.↓	avgUnc.↓	Time(ms)↓
VDFF [29]	$29.66e^{-2}$	$5.05e^{-1}$	0.87	1.18	$85.62e^{-2}$	17.92	32.66	50.31	1.12	–	–
RDF [18]	$11.15e^{-2}$	$3.22e^{-1}$	0.71	0.46	$23.95e^{-2}$	39.48	64.65	76.13	1.54	–	–
DDFF [13]	$3.34e^{-2}$	$1.67e^{-1}$	0.27	0.17	$3.56e^{-2}$	72.82	89.96	96.26	1.74	–	50.6
DefocusNet [27]	$2.18e^{-2}$	$1.34e^{-1}$	0.24	0.15	$3.59e^{-2}$	81.14	93.31	96.62	2.52	–	24.7
Ours-FV	$\mathbf{1.88e^{-2}}$	$\mathbf{1.25e^{-1}}$	**0.21**	0.14	$2.43e^{-2}$	81.16	**94.97**	**98.08**	1.45	0.24	**18.1**
Ours-DFV	$2.05e^{-2}$	$1.29e^{-1}$	0.21	**0.13**	$\mathbf{2.39e^{-2}}$	**81.90**	94.68	98.05	**1.43**	**0.17**	18.2

Table 1. Evaluation results on FoD500 test set.

Method	MSE ↓	RMS↓	log RMS ↓	Abs. rel.↓	Sqr. rel.↓	δ ↑	δ^2 ↑	δ^3 ↑	Bump.↓	avgUnc.↓	Time(ms)↓
VDFF [29]	$156.55e^{-4}$	12.14^{-2}	0.98	1.38	$241.2e^{-3}$	15.26	29.46	44.89	0.43	–	–
RDF [18]	$91.81e^{-4}$	$9.41e^{-2}$	0.91	1.00	$139.4e^{-3}$	15.65	33.08	47.48	1.33	–	–
DDFF [13]	$8.97e^{-4}$	$2.76e^{-2}$	0.28	0.24	$9.47e^{-3}$	61.26	88.70	96.49	0.52	–	191.7
DefocusNet [27]	$8.61e^{-4}$	$2.55e^{-2}$	0.23	0.17	$\mathbf{6.00e^{-3}}$	72.56	94.15	97.92	0.46	–	34.3
Ours-FV	$6.49e^{-4}$	$2.28e^{-2}$	0.23	0.18	$7.10e^{-3}$	71.93	92.80	97.86	0.42	$5.20e^{-2}$	**33.2**
Ours-DFV	$\mathbf{5.70e^{-4}}$	$\mathbf{2.13e^{-2}}$	**0.21**	**0.17**	$6.26e^{-3}$	**76.74**	**94.23**	**98.14**	**0.42**	$\mathbf{4.99e^{-2}}$	33.3

Table 2. Evaluation results on DDFF-12 validation set.

DFV," with five existing methods, VDFF [29], RDF [18], DDFF [13], DefocusNet [27], and AiFDepthNet [47]. We also compare to a variant of our method called "Ours-FV," which uses the deep focus volume as stated in Sec. 3.2.

VDFF and RDF are classic DFF methods. VDFF uses a Laplacian focus measure and a variational framework to generate smooth depth estimation. RDF uses the ring difference filter as the focus measure and aggregates the focus volume with guided filter. We use the code provided by the authors and run it on the test sets with default parameters.

DDFF and DefocusNet are two deep learning methods for DFF and DFD, respectively. We train the models with the original code from scratch in two different ways: (1) train on the mixed dataset as our methods do, and (2) first train on FoD500 dataset and then fine-tune it on DDFF-12 dataset as DefocusNet did. For both ways, we train it until the validation loss converges and pick the one with better performance on the DDFF-12 validation set for testing. As our model learns to predict the best-focused probability instead of the depth or disparity, in the mixed dataset training, there is no need to unify the ground truth from disparity to depth, or vice versa. But we find that, for DDFF and DefocusNet, such a unification improves their performance. Therefore, we convert the ground truth of the two datasets to the methods' original settings for training and convert their predictions back for evaluation. That is, at training time, we convert the ground truth disparity in DDFF-12 to depth for DefocusNets and convert the ground truth depth in FoD500 to disparity for DDFF. As a result, DDFF favors the first training approach, and DefocusNet prefers the second.

AiFDepthNet is a recent deep learning method trained on a large amount of data, including DDFF-12, FoD500, FlyingThings3D [28], and 4D Light Field [14]. The network supervision can be either the ground truth depth or the corresponding all-in-focus image. Because it does not release the training code, we cannot reproduce their results under our experiment settings. Thus, we only include AiFDepthNet in the comparison on the DDFF-12 test set, where its results are directly copied from the AiFDepthNet paper [47].

For a fair comparison, all methods take five frames from the input focal stack by random sampling at training time and by equidistant sampling at test time. The frame with the smallest or the largest focal distance/disparity will always be sampled at test time. We employ the same metrics used in [13] and introduce a new metric called average uncertainty (avgUnc.): $\frac{1}{M}\sum_{j=1}^{M}\phi_j$ to compare the prediction confidence of Ours-FV and Ours-DFV.

Result on FoD500. Table 1 presents the quantitative results on FoD500. Because of the limited sample frequency (5 frame/stack), both classic methods fail to deliver competitive results. In contrast, all deep methods work well. Compared to DDFF and DefocusNet, our methods provide more accurate and smooth results, which can also be seen from the qualitative results in the first two rows in Figure 3. Our methods better preserve the object boundaries, such as the cog and the central hole of the left gear, and show more smooth depth inference on the object surfaces. This observation verifies the power of the deep 4D FV. The effectiveness of the DFV is not significant in terms of the accuracy (first eight) and smoothness (Bump.) metrics, due to the dataset simplicity. But the avgUnc. of Ours-DFV is lower than Ours-FV, showing the higher confidence of Ours-DFV.

Result on DDFF-12. DDFF-12 is a more challenging dataset because of the large textureless regions in real-world scenarios. To reach high accuracy on this dataset, methods have to capture the weak defocus signal and utilize the context information to make a reasonable estimation. As DDFF-12 leaderboard only allows participants to submit their own methods, we first perform the comparison on

Method	MSE ↓	RMS↓	log RMS ↓	Abs. Rel.↓	Sqr. rel.↓	δ ↑	δ^2 ↑	δ^3 ↑	Bump.↓
DDFF [13]	$9.68e^{-4}$	$9.01e^{-2}$	0.32	0.29	0.01	61.95	85.14	92.98	0.59
AiFDepthNet [47]	$8.6e^{-4}$	–	0.29	0.25	0.01	68.33	87.40	93.96	0.63
Ours-FV	$6.54e^{-4}$	$7.55e^{-2}$	0.25	0.20	0.01	68.58	91.26	97.36	0.58
Ours-DFV	$\mathbf{5.58}e^{-4}$	$\mathbf{6.87}e^{-2}$	**0.23**	**0.19**	**0.01**	**74.26**	**92.38**	**97.39**	**0.57**

Table 3. Evaluation results on DDFF-12 test set. All values are from DDFF-12 leaderboard except that AiFDepthNet's results are from [47].

the validation set. Table 2 presents the quantitative results. As with the FoD500 results, deep learning methods outperform the classic approaches with large margins. Both of our methods consistently outperform the other deep learning methods. Thanks to the DFV module, Ours-DFV is able to better identify the weak defocus signal, and is more accurate and more confident than Ours-FV. For example, the MSE of Ours-DFV is 36.4% lower than DDFF, 33.8% lower than DefocusNet, and 12.2% lower than Ours-FV. The qualitative results are presented in the last two rows in Figure 3. More examples are available in our supplementary material. In the material, we also illustrate the focus probability distribution of Ours-DFV, which empirically shows the network does learn to locate the best-focused pixels.

Finally, we train Ours-FV and Ours-DFV models on the training and validation sets with the same scheme and submit the test result to the leaderboard[3]. Table 3 shows the results, where all values are from the leaderboard, except the one of AiFDepthNet is from their paper [47]. At the time of submission, Ours-DFV ranks 1st on the leaderboard.

Runtime. We compare the running speed among the deep learning methods. All models are tested with 5-frame focal stack inputs on an NVIDIA 1080Ti GPU. The results are shown in Table 1 and Table 2 with unit ms. Our methods are slightly faster than the DefocusNet and about three to six times faster than DDFF. In addition, Ours-FV and Ours-DFV have almost the same running time (± 0.1ms), which shows the efficiency of the DFV module.

4.3. Ablation Study

Focal Stack Size. In DFF, the input focal stack size is a key variable to the method performance. We evaluate our model's sensitivity to the stack size by training Ours-DFV with different sizes, $N = 2, \ldots, 10$, on DDFF-12 training set with the same training scheme and testing on the validation set. We exclude FoD500 in the experiment, because its focal stacks only contain 5 frames. Figure 4 shows the MSE and avgUnc. changes w.r.t. the input frame number. The full metrics table is available in the supplementary material. We can see the model starts to deliver fairly accurate results in terms of MSE with only 3 frames/stack. The avgUnc. keeps decreasing with increasing stack size, showing the model's growing confidence with more input

---
[3] https://competitions.codalab.org/competitions/17807#results

Method	MSE ↓	RMS↓	Abs.rel.↓	Time(ms) ↓
DFV-L1	$6.65e^{-4}$	$2.30e^{-2}$	0.18	**24.1**
DFV-L2	$6.16e^{-4}$	$2.20e^{-2}$	0.17	28.0
DFV-L3	$5.94e^{-4}$	$2.16e^{-2}$	0.17	30.9
Ours-DFV	$\mathbf{5.70}e^{-4}$	$\mathbf{2.13}e^{-2}$	**0.17**	33.3

Table 4. Performance of different multi-scale variants.

frames. However, we notice that the impact of additional frames is diminishing as the frame number increases. The authors of [13] also report that their network performance on DDFF-12 dataset stops improving after the frame number reaches 10 (which is the stack size they finally released to the public). Further studies with a new dataset containing more frames per stack are needed to find the actual reason.
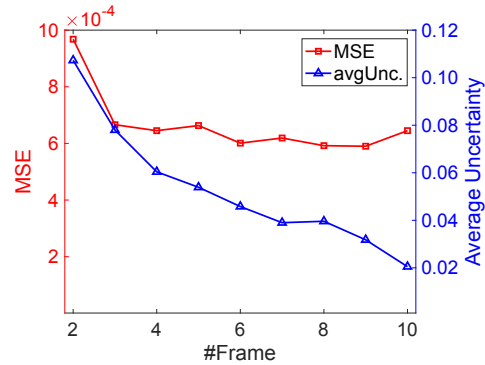

Figure 4. Model performance w.r.t. focal stack sizes.

Multi-scale Architecture. Our network uses a multi-scale architecture, and deep supervision (Eq. 4) is applied at training time. To investigate the impact of this design, we train three variants of Ours-DFV model named DFV-Lk, where $k = 1, 2, 3$. In each variant, we only use the k largest scale DFVs and remove the rest DFVs and the associated 3D CNN modules from the network. For example, for DFV-L2, only the 2 largest-scale DFVs ($\frac{1}{4}$ and $\frac{1}{8}$) and their associated 3D CNN modules are used. The training scheme is the same as Ours-DFV. The evaluation results on the DDFF-12 validation set are shown in Table 4. The depth accuracy increases as more DFV modules are used, verifying the effectiveness of the multi-scale design. Compared with Table 2, even with a single DFV module (DFV-L1), the model still outperforms existing DFF and DFD methods by large margins. This demonstrates the effectiveness of DFV.
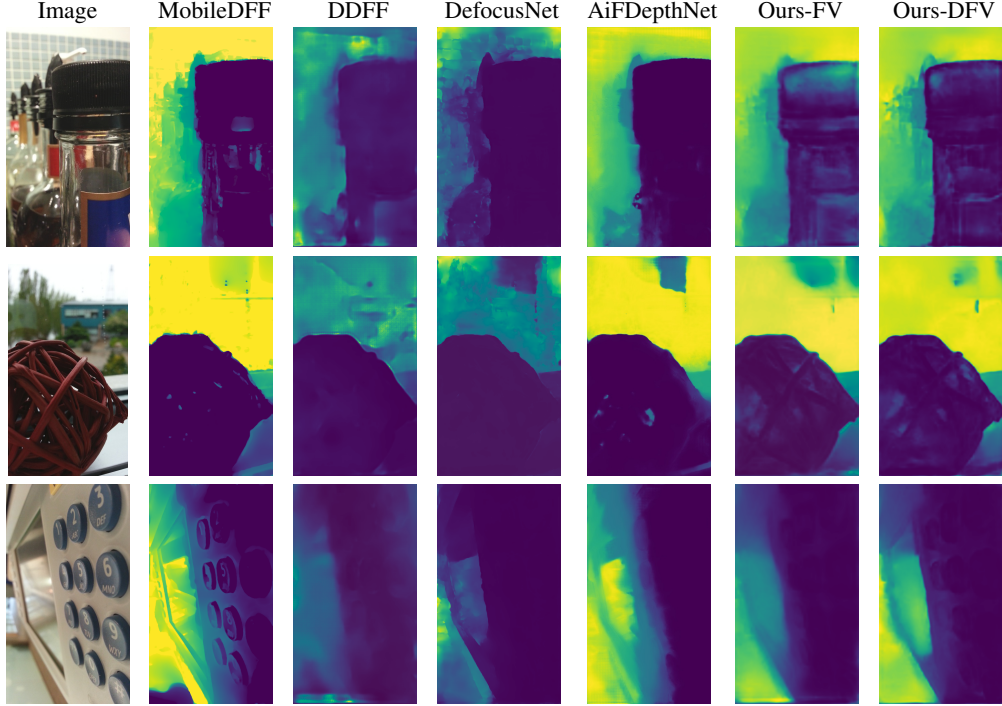
Image	MobileDFF	DDFF	DefocusNet	AiFDepthNet	Ours-FV	Ours-DFV

Figure 5. Qualitative results on Mobile depth dataset. The warmer the color, the larger the depth value.

4.4. Generalization to Unknown Scenes

Model generalizability to unknown scenes is important for deep DFF methods. To evaluate our model's generalizability, we directly test the pre-trained DDFF [13], Defocus-Net [27], Ours-FV, and Ours-DFV models, from Sec. 4.2, on the Moblie depth dataset, and compare them with MobileDFF [45] and AiFDepthNet [47]. MobileDFF is a traditional method that combines the merits of DFF and DFD. The results are directly from the authors. For AiFDepth-Net, as the authors only release their pre-trained models, we apply their depth supervised Mobile depth model for the experiment. Because of the lack of ground truth depth, we normalize all the results and compare them visually.

Part of the results are shown in Figure 5, and the rest are available in the supplementary material. All the deep methods provide reasonable depth estimation. Compared to the other deep methods, Ours-DFV better preserves object boundaries, e.g., the front bottle in the first row and the ball in the second row. Ours-DFV is also less sensitive to background patterns, such as the tile in the first row. However, all methods (except DDFF) are sensitive to mirror reflection (the screen in the third row). One possible reason is that the effective distance between the camera and the object shown on the screen is equal to the sum of the distance between the camera and the screen and the distance between the screen and the object. The CoC becomes larger, so all five methods infer it as an object at a farther distance. The DDFF network is adopted from the general dense prediction task, which may focus more on the context information, and thus gives a better estimation in such a region.

Among all the methods, MobileDFF appears to recover the most details. This is partially because MobileDFF takes the whole focal stack (ranging from 14 to 33 frame/scene) as input, whereas the deep models (including ours) only use 5 frames. Besides, MobileDFF is an optimization method which takes several minutes to process one scene [45]. Meanwhile, our methods run in tens of milliseconds.

5. Conclusion

In this work, we developed a novel CNN architecture for the DFF task. To some extent, our network design is analogous to traditional DFF methods and recent deep stereo matching methods. The proposed deep differential focus volume module is able to combine both the focus and the context information for focus estimation. Experiments on various datasets demonstrated the superior performance of our models in terms of accuracy, efficiency, and generalizability. Currently, our models can only work on well-aligned focal stacks in static scenes. In the future, it is interesting to explore specific data augmentation methods for imperfectly aligned stacks. We also plan to integrate the optical flow [45] or homography [18] alignment mechanism into the model and extend it to dynamic environments.

References

[1] Muhammad Bilal Ahmad and Tae Sun Choi. Application of three dimensional shape from image focus in lcd/tft displays manufacturing. *IEEE Trans. Consum. Electron.*, 53(1):1–4, 2007. 2

[2] Jean-Baptiste Alayrac, Joao Carreira, and Andrew Zisserman. The visual centrifuge: Model-free layered video representations. In *CVPR*, pages 2457–2466, 2019. 2

[3] Chadi Albitar, Pierre Graebling, and Christophe Doignon. Robust structured light coding for 3d reconstruction. In *ICCV*, pages 1–6. IEEE, 2007. 1

[4] Usman Ali, Ik Hyun Lee, and Muhammad Tariq Mahmood. Guided image filtering in shape-from-focus: A comparative analysis. *Pattern Recognition*, 111:107670, 2021. 1

[5] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, pages 5410–5418, 2018. 2, 3

[6] Zhang Chen, Xinqing Guo, Siyuan Li, Yang Yang, and Jingyi Yu. Deep eyes: Joint depth inference using monocular and binocular cues. *Neurocomputing*, 453:812–824, 2021. 1, 2

[7] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *PAMI*, 42(10):2361–2379, 2019. 2

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, pages 2758–2766, 2015. 2, 3

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014. 1

[10] Tiantian Fan and Hongbin Yu. A novel shape from focus method based on 3d steerable filters for improved performance on treating textureless region. *Optics Communications*, 410:254–261, 2018. 1, 2

[11] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *ICCV*, pages 3828–3838, 2019. 1

[12] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *ICCVW*, pages 3154–3160, 2017. 4

[13] Caner Hazirbas, Sebastian Georg Soyer, Maximilian Christian Staab, Laura Leal-Taixé, and Daniel Cremers. Deep depth from focus. In *ACCV*, pages 525–541, 2018. 1, 2, 5, 6, 7, 8

[14] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *ACCV*, pages 19–34, 2016. 6

[15] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5), 1989. 4

[16] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *PAMI*, 35(2):504–511, 2012. 2

[17] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, pages 2462–2470, 2017. 2

[18] Hae-Gon Jeon, Jaeheung Surh, Sunghoon Im, and In So Kweon. Ring difference filter for fast and noise robust depth from focus. *TIP*, 29:1045–1060, 2019. 2, 6, 8

[19] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, pages 7482–7491, 2018. 4

[20] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, pages 66–75, 2017. 2

[21] Eric Krotkov. Focusing. *IJCV*, 1(3):223–237, 1988. 2

[22] Jae Young Lee and Rae-Hong Park. Complex-valued disparity: Unified depth model of depth from stereo, depth from focus, and depth from defocus based on the light field gradient. *PAMI*, 2019. 2

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 4

[24] Muhammad Tariq Mahmood and Tae-Sun Choi. Nonlinear approach for enhancement of image focus volume in shape from focus. *TIP*, 21(5):2866–2873, 2012. 2

[25] Fahim Mannan and Michael S Langer. What is a good model for depth from defocus? In *CRV*, pages 273–280, 2016. 1

[26] Yamin Mao, Zhihua Liu, Weiming Li, Yuchao Dai, Qiang Wang, Yun-Tae Kim, and Hong-Seok Lee. Uasnet: Uncertainty adaptive sampling network for deep stereo matching. In *ICCV*, pages 6311–6319, 2021. 2, 3

[27] Maxim Maximov, Kevin Galim, and Laura Leal-Taixé. Focus on defocus: bridging the synthetic to real domain gap for depth estimation. In *CVPR*, pages 1071–1080, 2020. 5, 6, 8

[28] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, pages 4040–4048, 2016. 2, 3, 6

[29] Michael Moeller, Martin Benning, Carola Schönlieb, and Daniel Cremers. Variational depth from focus reconstruction. *TIP*, 24(12):5369–5378, 2015. 1, 2, 6

[30] Hari N Nair and Charles V Stewart. Robust focus ranging. In *CVPR*, pages 309–310, 1992. 2

[31] Shree K Nayar and Yasuo Nakagawa. Shape from focus. *PAMI*, 16(8):824–831, 1994. 2

[32] Illah R Nourbakhsh, David Andre, Carlo Tomasi, and Michael R Genesereth. Obstacle avoidance via depth from focus. In *IUW*, pages 1339–1344, 1996. 2, 3

[33] Alex Paul Pentland. A new sense for depth of field. *PAMI*, (4):523–531, 1987. 1

[34] Said Pertuz, Domenec Puig, and Miguel Angel Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415–1432, 2013. 1, 2

[35] Michael Ramamonjisoa, Michael Firman, Jamie Watson, Vincent Lepetit, and Daniyar Turmukhambetov. Single im-

age depth prediction with wavelet decomposition. In *CVPR*, 2021. 1

[36] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *PAMI*, 2020. 1

[37] Parikshit Sakurikar and PJ Narayanan. Composite focus measure for high quality depth maps. In *ICCV*, pages 1614–1622, 2017. 2

[38] Daniel Scharstein and Richard Szeliski. Stereo matching with nonlinear diffusion. *IJCV*, 28(2):155–174, 1998. 2

[39] Yoav Y Schechner and Nahum Kiryati. Depth from defocus vs. stereo: How different really are they? *IJCV*, 39(2):141–162, 2000. 1, 2, 3

[40] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, pages 4104–4113, 2016. 1

[41] H Shoji, K Shirai, and Masaaki Ikehara. Shape from focus using color segmentation and bilateral filter. In *2006 IEEE 12th Digital Signal Processing Workshop & 4th IEEE Signal Processing Education Workshop*, pages 566–571, 2006. 2

[42] Muralidhara Subbarao and J-K Tyan. Selecting the optimal focus measure for autofocusing and depth-from-focus. *PAMI*, 20(8):864–870, 1998. 2

[43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018. 2

[44] Jaeheung Surh, Hae-Gon Jeon, Yunwon Park, Sunghoon Im, Hyowon Ha, and In So Kweon. Noise robust depth from focus using a ring difference filter. In *CVPR*, pages 6328–6337, 2017. 1, 2

[45] Supasorn Suwajanakorn, Carlos Hernandez, and Steven M Seitz. Depth from focus with your mobile phone. In *CVPR*, pages 3497–3506, 2015. 1, 2, 4, 5, 8

[46] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 2

[47] Ning-Hsu Wang, Ren Wang, Yu-Lun Liu, Yu-Hao Huang, Yu-Lin Chang, Chia-Ping Chen, and Kevin Jou. Bridging unsupervised and supervised depth from focus via all-in-focus supervision. In *ICCV*, pages 12621–12631, 2021. 1, 2, 6, 7, 8

[48] Hui Xie, Weibin Rong, and Lining Sun. Wavelet-based focus measure and 3-d surface reconstruction method for microscopy images. In *IROS*, pages 229–234, 2006. 2

[49] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *CVPR*, pages 1289–1297, 2017. 2

[50] Gengshan Yang, Joshua Manela, Michael Happold, and Deva Ramanan. Hierarchical deep stereo matching on high-resolution images. In *CVPR*, 2019. 2, 3, 4

[51] Qingxiong Yang. A non-local cost aggregation method for stereo matching. In *CVPR*, pages 1402–1409, 2012. 2

[52] Teresa Yeo, Oğuzhan Fatih Kar, and Amir Zamir. Robustness via cross-domain ensembles. *ICCV*, 2021. 4

[53] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 4