Learning to Autofocus

Charles Herrmann[1] Richard Strong Bowen[1] Neal Wadhwa[2] Rahul Garg[2]
Qiurui He[2] Jonathan T. Barron[2] Ramin Zabih[1,2]
[1]Cornell Tech [2]Google Research
{cih, rsb, rdz}@cs.cornell.edu {nealw, rahulgarg, qiurui, barron, raminz}@google.com

Abstract

Autofocus is an important task for digital cameras, yet current approaches often exhibit poor performance. We propose a learning-based approach to this problem, and provide a realistic dataset of sufficient size for effective learning. Our dataset is labeled with per-pixel depths obtained from multi-view stereo, following [9]. Using this dataset, we apply modern deep classification models and an ordinal regression loss to obtain an efficient learning-based autofocus technique. We demonstrate that our approach provides a significant improvement compared with previous learned and non-learned methods: our model reduces the mean absolute error by a factor of 3.6 over the best comparable baseline algorithm. Our dataset and code are publicly available.

1. Introduction

In a scene with variable depth, any camera lens with a finite-size aperture can only focus at one scene depth (the focus distance), and the rest of the scene will contain blur. This blur is difficult to remove via post-processing, and so selecting an appropriate focus distance is crucial for image quality.

There are two main, independent tasks that a camera must address when focusing. First, the camera must determine the salient region that should be in focus. The user may choose such a region explicitly, e.g., by tapping on the screen of a smartphone, or it may be detected automatically by, for example, a face detector. Second, given a salient region (which camera manufacturers often refer to as "autofocus points") and one or more possibly out-of-focus observations, the camera must predict the most suitable focus distance for the lens that brings that particular region into focus. This second task is called *autofocus*.

Conventional autofocus algorithms generally fall into two major categories: contrast-based and phase-based methods. Contrast-based methods define a sharpness metric, and identify the ideal focus distance by maximizing the sharpness metric across a range of focus distances. Such methods are necessarily slow in practice, as they must make a large number of observations, each of which requires physical lens movement. In addition, they suffer from a few important weaknesses, which we discuss in Section 4.

Modern phase-based methods leverage disparity from the dual-pixel sensors that are increasingly available on smartphones and DSLR cameras. These sensors are essentially two-view plenoptic cameras [26] with left and right sub-images that receive light from the two halves of the aperture. These methods operate under the assumption that in-focus objects will produce similar left and right sub-images, whereas out-of-focus objects will produce sub-images with a displacement or disparity that is proportional to the degree of defocus. Naively, one could search for the focus distance that minimizes the left/right mismatch, like the contrast-based methods. Alternatively, some methods use calibration to model the relationship between disparity and depth, and make a prediction with just one input. However, accurate estimation of disparity between the dual-pixel sub-images is challenging due to the small effective baseline. Further, it is difficult to characterize the relationship between disparity and depth accurately due to optical effects that are hard to model, resulting in errors [9].

In this paper, we introduce a novel learning-based approach to autofocus: a ConvNet that takes as input raw sensor data, optionally including the dual-pixel data, and predicts the ideal focus distance. Deep learning is well-suited to this task, as modern ConvNets are able to utilize subtle defocus clues (such as irregularly-shaped point spread functions) in the data that often mislead heuristic contrast-based autofocus methods. Unlike phase-based methods, a learned model can also directly estimate the position the lens should be moved to, instead of determining it from disparity using a hand-crafted model and calibration—a strategy which may be prone to errors.

In order to train and evaluate our network, we also introduce a large and realistic dataset captured using a smartphone camera and labeled with per-pixel depth computed using multi-view stereo. The dataset consists of *focal* *stacks*: a sequence of image patches of the same scene, varying only in focus distance. We will formulate the autofocus problem precisely in section 3, but note that the output of autofocus is a focal index which specifies one of the patches in the focal stack. Both regular and dual-pixel raw image data are included, allowing evaluation of both contrast- and phase-based methods. Our dataset is larger than most previous efforts [4, 14, 23], and contains a wider range of realistic scenes. Notably, we include outdoors scenes (which are particularly difficult to capture with a depth sensor like Kinect) as well as scenes with different levels of illumination.

We show that our models achieve a significant improvement in accuracy on all versions of the autofocus problem, especially on challenging imagery. On our test set, the best baseline algorithm that takes one frame as input produces a mean absolute error of 11.3 (out of 49 possible focal indices). Our model with the same input has an error of 3.1, and thus reduces the mean absolute error by a factor of 3.6.

2. Related Work

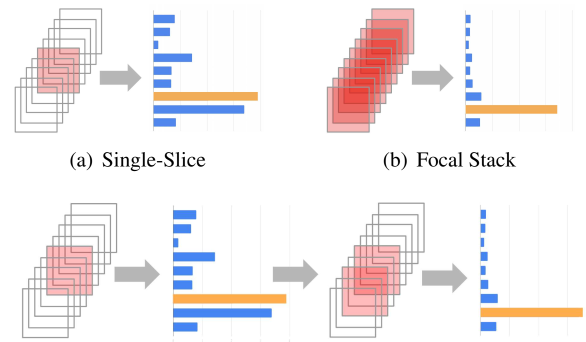
There has been surprisingly little work in the computer vision community on autofocus algorithms. There are a number of non-learning techniques in the image processing literature [5, 19, 20, 45, 46], but the only learning approach [23] uses classical instead of deep learning.

A natural way to use computer vision techniques for autofocus would be to first compute metric depth. Within the vast body of literature on depth estimation, the most closely related work of course relies on focus.

Most monocular depth techniques that use focus take a complete focal stack as input and then estimate depth by scoring each focal slice according to some measure of sharpness [16, 25, 40]. Though acquiring a complete focal stack of a static scene with a static camera is onerous, these techniques can be made tractable by accounting for parallax [38]. More recently, deep learning-based methods [14] have yielded improved results with a full focal stack approach.

Instead of using a full focal stack, some early work attempted to use the focal cues in just one or two images to estimate depth at each pixel, by relating the apparent blur of the image to its disparity [10, 28], though these techniques are necessarily limited in their accuracy compared to those with access to a complete focal stack. Both energy minimization [39] and deep learning [4, 35] have also been applied to single-image approaches for estimating depth from focus, with significantly improved accuracy. Similarly, much progress has been made in the more general problem of using learning for monocular depth estimation using depth cues besides focus [8, 33], including dual-pixel cues [9, 42].

In this work, we address the related problem of autofocus



(a) Single-Slice (b) Focal Stack

(c) Two-Step

Figure 1. Three different autofocus subproblems; in each, the goal is to estimate the in-focus slice, by taking the argmax (orange) of a set of scores produced for each possible focal slice (blue). In the single-slice problem (a), the algorithm is given a single observed slice (red). In the focal stack problem (b), the algorithm is given the entire stack. In the multi-step problem (here shown with just two steps) (c), the problem is solved in stages; Given an initial lens position and image, we decide where to focus next, obtain a new observation, and then make a final estimate of the in-focus slice using both observed images.

by applying deep learning. A key aspect of the autofocus problem is that commodity focus modules require a single focus estimate to guide them, that may have a tenuous connection with predicted depth map due to hardware issues (see Section 4). Many algorithms predict non-metric depth maps, making the task harder, e.g., scale invariant monocular depth prediction [8] or affine invariant depth prediction using dual-pixel data [9]. Hence, instead of predicting a dense depth map, we directly predict a single estimate of focal depth that can be used to guide the focus module. This prediction is done end to end with deep learning.

3. Problem Formulation

In the natural formulation of the autofocus problem, the lens can move continuously, producing an infinite set of possible focus distances corresponding to different focal planes. We discretize the continuous lens positions into n focus distances, and from each position we extract an image patch $I_k, k \in \{1, \ldots, n\}$ corresponding to the region of interest. We assume the location of the patch I_k has been determined by a user or some external saliency algorithm, and so we consider this image patch to be "the image" and will refer to it as such throughout the paper. Further, the image can either contain the dual-pixel subimages as two channels or it can contain just the green channel based on the type of input being considered. We refer to the set of images obtained at different focus distances $\{I_k\}$ as a *focal stack*, an individual image I_k as a *focal slice*, and k as the *focal index*. We assume each focal stack has exactly one focal index whose slice is in focus.

Standard autofocus algorithms can be naturally partitioned according to the number of focal slices they require as input. For example, contrast-based methods often require the entire focal stack (or a large subset), whereas phase-based or depth-from-defocus algorithms can estimate a focus distance given just a single focal slice. Motivated by the differences in input space among standard autofocus algorithms, we define three representative sub-problems (visualized in Figure 1), which all try to predict the correct focal index but vary based primarily on their input.

Focal Stack:

$$f : \{I_k \mid k = 1, \ldots, n\} \mapsto k^* \tag{1}$$

This is the simplest formulation where the algorithm is given a completely observed focal stack. Algorithms for this type typically define a sharpness or contrast metric and pick the focal index which maximizes the chosen metric.

Single Slice:

$$f : I_k \mapsto k^*, \ \forall k \in \{1, \ldots, n\} \tag{2}$$

This is the most challenging formulation, as the algorithm is given only a single, random focal slice, which can be thought of as the starting position of the lens. In this formulation, algorithms generally try to estimate blur size or use geometric cues to estimate a measure of depth that is then translated to a focal index.

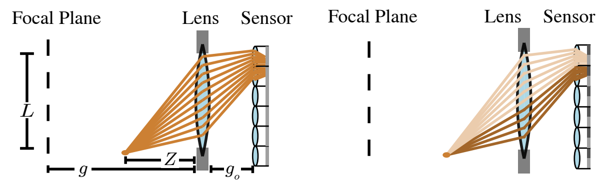
Multi-Step:

$$\begin{aligned} f_1 &: I_{k_0} \mapsto k_1 \\ f_2 &: I_{k_0}, I_{k_1} \mapsto k_2 \\ &\cdots \\ f_m &: I_{k_0}, \ldots, I_{k_{m-1}} \mapsto k_m \end{aligned} \tag{3}$$

where $k_0 \in \{1, \ldots, n\}$, and m is a predetermined constant controlling the total number of steps. The multi-step problem is a mix between the previous two problems. The algorithm is given an initial focal index, acquires and analyzes the image at that focus distance, and then is permitted to move to an additional focal index of its choice, repeating the process at most m times. This formulation approximates the online problem of moving the lens to the correct position with as few attempts as possible. This multi-step formulation resembles the "hybrid" autofocus algorithms that are often used by camera manufacturers, in which a coarse focus estimate is produced by some phase-based system (or a direct depth sensor if available) which is then refined by a contrast-based solution that uses a constrained and abbreviated focal stack as input.

4. Autofocus Challenges

We now describe the challenges in real cameras that make the autofocus problem hard in practice. With the thin-lens and paraxial approximations, the amount of defocus blur is specified by



(a) Ordinary Sensor (b) Dual-Pixel Sensor

Figure 2. Cameras (a) focus by moving the sensor or lens, and only produce sharp images at a single depth (g in this case). Dual-pixel sensors (b) split each pixel into two halves that each collect light from the two halves of the lens, which aides autofocus.

$$\frac{Lf}{1 - f/g}\left(\left|\frac{1}{g} - \frac{1}{Z}\right|\right) \tag{4}$$

where L is the aperture size, f the focal length, Z the depth of a scene-point and g the focus distance (Figure. 2(a)). g is related to the distance g_o between the lens and the sensor by the thin-lens equation. This implies that if the depth Z is known, one can focus, i.e, reduce the defocus blur to zero by choosing an appropriate g, which can be achieved by physically adjusting the distance between the lens and the sensor g_o. This suggests that recovering depth (Z) is sufficient to focus. Dual-pixel sensors can aid in the task of finding Z as they produce two images, each of which sees a slightly different viewpoint of the scene (Figure 2(b)). The disparity d between these viewpoints [9] is

$$d = \alpha \frac{Lf}{1 - f/g}\left(\frac{1}{g} - \frac{1}{Z}\right) \tag{5}$$

where α is a constant of proportionality.

This theoretical model is often used in the academic pursuit of autofocus (or more often, depth-from-defocus) algorithms. However, the paraxial and thin lens approximations are significant simplifications of camera hardware design and of the physics of image formation. Here we detail some of the issues ignored by this model and existing approaches, and explain how they are of critical importance in the design of an effective, practical autofocus algorithm.

Unrealistic PSF Models. One core assumption underlying contrast-based algorithms is that, as the subject being imaged moves further out of focus, the high-frequency image content corresponding to the subject is reduced. The assumption that in-focus content results in sharp edges while out-of-focus content results in blurry edges has only been shown to be true for Gaussian point spread functions (PSF) [22, 48]. However, this assumption can be broken by real-world PSFs, which may be disc- or hexagon-shaped with the goal of producing an aesthetically pleasing "bokeh". Or

(a) Im, $\|\nabla\| = 1.22$ (b) Blur, $\|\nabla\| = 0.62$ (c) Disc, $\|\nabla\| = 2.45$

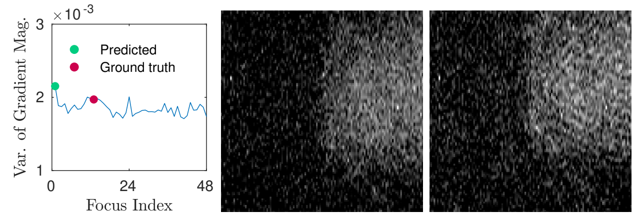
Figure 3. Many contrast-based autofocus algorithms return the focus distance that maximizes image sharpness, measured here as the norm of the image gradient $\|\nabla\|$. This works well for some camera PSFs, as a sharp image (such as the saturated delta function in (a)) will likely have more gradient energy than the same image seen out of focus under a Gaussian PSF (such as in (b)). But actual cameras tend to have irregular PSFs that more closely resemble discs than Gaussians, and as a result an out-of-focus image may have a *higher* gradient energy than an in-focus image (such as the delta function convolved with a disc filter in (c)). This is one reason why simple contrast-based autofocus algorithms often fail in practice.

they may be some irregular shape that defies characterization as a side effect of hardware and cost constraints of modern smartphone camera construction. In the case of a disc-shaped PSF, for example, an out-of-focus delta function may actually have *more* gradient energy than an in-focus delta function, especially when pixels are saturated (See Figure 3).

Noise in Low Light Environments. Images taken in dim environments often contain significant noise, a problem that is exacerbated by the small aperture sizes and small pixel pitch of consumer cameras [13]. Prior work in low-light imaging has noted that conventional autofocus algorithms systematically break in such conditions [21]. This appears to be due to the gradient energy resulting from sensor noise randomly happening to exceed that of the actual structure in the image, which causes contrast-based autofocus algorithms (which seek to maximize contrast) to be misled. See Figure 4 for a visualization of this issue.

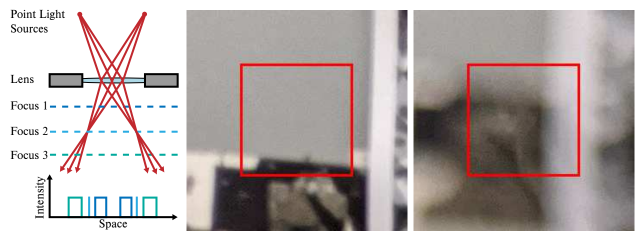
Focal Breathing. A camera's field of view depends on its focus distance, a phenomenon called focal breathing.[1] This occurs because conventional cameras focus by changing the distance between the image plane and the lens, which induces a zoom-like effect as shown in Figure 5. This effect can be problematic for contrast-based autofocus algorithms, as edges and gradients can leave or enter the field of view of the camera over the course of a focal sweep, even when the camera and scene are stationary. While it is possible to calibrate for focal breathing by modeling it as a zoom and crop, applying such a calibration increases latency, may be inaccurate due to unknown radial distortion, and may introduce resampling artifacts that interfere with contrast-based metrics.

[1] Sometimes also referred to as focus breathing or lens breathing.



(a) Contrast metric (b) Predicted (c) Ground truth

Figure 4. Image noise misleads contrast-based focus measures, making it difficult to focus in low-light. There is no obvious peak in a contrast measure (a) applied to the noisy patches in (b) and (c). As a result, the argmax index results in patch (b) that is out of focus, instead of the in-focus ground-truth patch (c), which contains subtle high-frequency texture.



(a) Optics (b) Focused, $\|\nabla\|$=0.88 (c) Unfocused, $\|\nabla\|$=1.02

Figure 5. The optics of image formation mean that modifying the focus of a lens causes "focal breathing": a change of the camera's field of view. Consider light from two points that is being imaged at three different focus distances, as in the top of (a). Because the light is spreading away from the center of the sensor, focusing therefore causes the positions of the points on the imaging plane to shift inwards as the distance between the imaging plane and lens (i.e., focus distance) decreases. This occurs in real image patches and can mislead contrast-based metrics: the in-focus image patch (b) has less gradient energy than the out-of-focus image patch (c) because edges move in and out of the patch when focusing. (Gradient energy is only computed within the red rectangles of (b) and (c).)

Hardware Support. Nearly all smartphone cameras use voice coil motors (VCMs) to focus: the lens sits within a barrel, where it is attached to a coil spring and positioned near an electromagnet, and the electromagnet's voltage is adjusted to move the camera along the 1D axis of the spring and barrel, thereby changing the focus distance of the camera. Though VCMs are inexpensive and ubiquitous, they pose a number of issues for the design of an autofocus or depth-from-defocus algorithm. 1) Most VCM autofocus modules are "open loop": a voltage can be specified, but it is not possible to determine the actual metric focus distance that is then induced by this voltage. 2) Due to variation in temperature, the orientation of the lens relative to gravity, cross talk with other components (e.g., the coils and magnets in optical image stabilization (OIS) module), and simple wear-and-tear on the VCM's spring, the mapping from a specified voltage to its resulting metric focus distance be grossly inaccurate. 3) The lens may move "off-axis" (perpendicular to the spring) during autofocus due to OIS, changing both the lens's focus distance and its principal point.

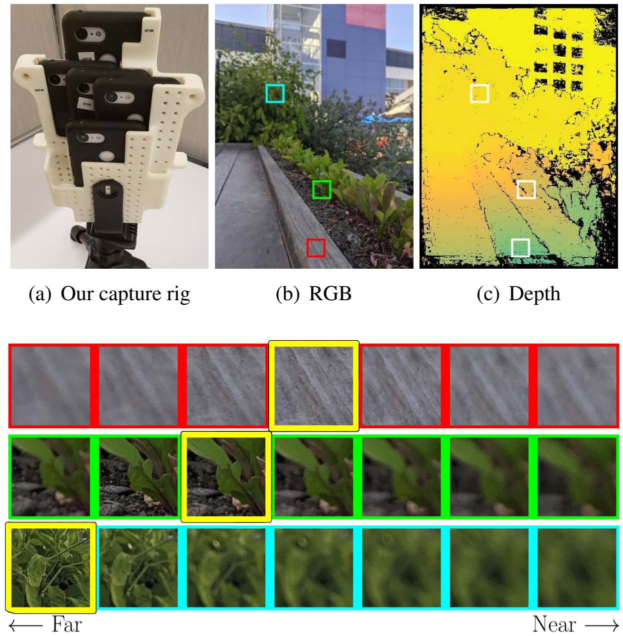
Unknown and uncalibrated PSFs, noise, focal breathing, and the large uncertainty in how the VCM behaves make it difficult to manually engineer a reliable solution to the autofocus problem. This suggests a learning-based approach using a modern neural network.

5. Dataset

Our data capture procedure generally follows the approach of [9], with the main difference being that we capture and process focal stacks instead of individual in-focus captures. Specifically, we use the smartphone camera synchronization system of [1] to synchronize captures from five Google Pixel 3 devices arranged in a cross pattern (Figure 6(a)). We capture a static scene with all five cameras at 49 focal depths sampled uniformly in inverse depth space from 0.102 meters to 3.91 meters. We jointly estimate intrinsics and extrinsics of all cameras using structure from motion [12], and then compute depth (Figure 6(c)) for each image using a modified form of the multi-view stereo pipeline of [9]. We sample 128×128 patches with a stride of 40 from the central camera capture yielding focal stacks of dimensions $128 \times 128 \times 49$. We then calculate the ground-truth index for each stack by taking the median of the corresponding stack in the associated depth maps and finding the focal index with the closest focus distance in inverse-depth space. The median is robust to errors in depth and a reasonable proxy for other sources of ground truth that might require more effort, e.g., manual annotation. We then filter these patches by the median confidence of the depth maps. Please see the supplemental material for more details.

Our dataset has 51 scenes, with 10 stacks per scene containing different compositions, for a total of 443,800 patches. These devices capture both RGB and dual-pixel data. Since autofocus is usually performed on raw sensor data (and not a demosaiced RGB image), we use only the raw dual-pixel data and their sum, which is equivalent to the raw green channel. To generate a train and test set, we randomly selected 5 scenes out of the 51 to be the test set; as such, our train set contains 460 focal stacks (387,000 patches) and our test set contains 50 (56,800 patches).

Our portable capture rig allows us to capture a semantically diverse dataset with focal stacks from both indoor and outdoor scenes using a consumer camera (Figure 6), making the dataset one of the first of its kind. Compared to other datasets primarily intended for autofocus [4,23], our dataset is substantially larger, a key requirement for deep learning techniques. Our dataset is comparable in size to [14], which uses a Lytro for lightfield capture and a Kinect for metric depth. However, we have significantly more scenes (51 vs 12) and use a standard phone camera instead of a plenoptic camera. The latter has a lower resolution (383×552 for the Lytro used in [14] vs 1512×2016 for our dual-pixel data) and "focal stacks" generated by algorithmic refocusing do not exhibit issues such as focal breathing, hardware control, noise, PSFs, etc, which are present upon focusing a standard camera. These issues are some of the core challenges of autofocus, as described above in Section 4.



(a) Our capture rig (b) RGB (c) Depth

(d) Example focal stacks

Figure 6. Our portable rig (a) with 5 synchronized cameras similar to the one in [9] allows us to capture outdoor scenes (b) and compute ground truth depth (c) using multi-view stereo. In (d) we show 7 of the 49 slices from three focal stacks at different depths corresponding to the patches marked in (b). The ground truth patches (the in-focus patches according to our estimated depth) are marked in yellow.

6. Our Model

We build our model upon the MobileNetV2 architecture [31], which has been designed to take as input a conventional 3-channel RGB image. In our use case, we need to represent a complete focal stack, which contains 49 images. We encode each slice of the focal stack as a separate channel, so the model can reason about each image in the focal stack. In our experiments where we give the model access to dual pixel data, each image in the focal stack is a 2-channel image where the channels correspond to the left and right dual-pixel images respectively. In our ablations where the model is deprived of dual-pixel data, each image in the focal stack is a 1-channel image that contains the sum of the left and right views (which is equivalent to the green channel of the raw RGB image). To accommodate this much “wider” number of channels in the input to our network, we increase the number of channels by 4 times the original amount (width multiplier of 4) to prevent a contraction in the number of channels between the input and the first layer. In practice, the network runs quickly: 32.5 ms on a flagship smartphone.

In the setup where the full focal stack is available as input, the model is given a $128 \times 128 \times 98$ tensor for dual-pixel data, and a $128 \times 128 \times 49$ tensor for traditional green-channel sensor data. In the task where only one focal slice is observable, we use one-hot encoding along the channel dimension as input: the input is a 98-channel tensor (or 49 for green-channel only input) where the channels that correspond to unobserved slices in the focal stack are all zeros. We use this same encoding in the first step of our multi-step model, but we add an additional one-hot encoding for each subsequent step of the model, thereby giving the model access to all previously-observed images in the focal stack. We train this network by taking a completed single-slice network and evaluate it on all possible focal stacks and input indices. We then feed a new network this one-hot encoding, so the new network sees the first input index and the prediction of the single-slice network.

We model autofocus as an ordinal regression problem: we treat each focal index as its own discrete distinct class, but we assume that there is an ordinal relationship between the class labels corresponding to each focal index (e.g., index 6 is closer to index 7 than it is to index 15). The output of all versions of our network is 49 logits. We train our model by minimizing the ordinal regression loss of [7], which is similar to the cross-entropy used by traditional logistic regression against unordered labels, but where instead of calculating cross-entropy with respect to a Kronecker delta function representing the ground-truth label, that delta function is convolved with a Laplacian distribution. This encourages the model to make predictions that are as close as possible to the ground-truth, while using traditional cross-entropy would incorrectly model any prediction other than the ground-truth (even those immediately adjacent) as being equally costly.

For training, we use Adam [17] with default parameters (initial lr $= 1e-3$, beta1 $= 0.5$, beta2 $= 0.999$), with a batchsize of 128 and for 20k global steps. For the ordinal regression loss, we use L2 cost metric of [7] with a coefficient of 1.

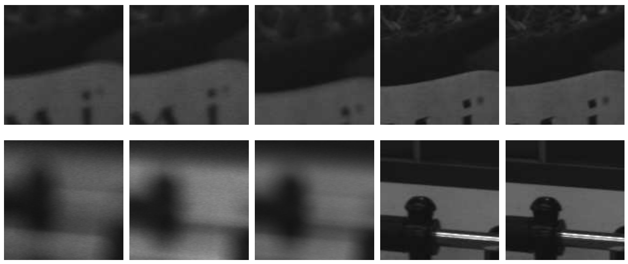
7. Results

	Algorithm	higher is better $= 0$	≤ 1	≤ 2	≤ 4	lower is better MAE	RMSE
I*	DCT Reduced Energy Ratio [20]	0.034	0.082	0.122	0.186	18.673	22.855
I*	Total Variation (L1) [24,30]	0.048	0.136	0.208	0.316	15.817	21.013
I*	Histogram Entropy [18]	0.087	0.230	0.326	0.432	14.013	20.223
I*	Modified DCT [19]	0.033	0.091	0.142	0.235	15.713	20.197
I*	Gradient Count ($t = 3$) [18]	0.109	0.312	0.453	0.612	9.543	16.448
I*	Gradient Count ($t = 10$) [18]	0.126	0.347	0.493	0.645	9.103	16.218
I*	DCT Energy Ratio [6]	0.110	0.286	0.410	0.554	9.556	15.286
I*	Eigenvalue Trace [43]	0.116	0.303	0.434	0.580	8.827	14.594
I*	Intensity Variance [18]	0.116	0.303	0.434	0.580	8.825	14.593
I*	Intensity Coefficient of Variation	0.125	0.327	0.469	0.624	8.068	13.808
I*	Percentile Range ($p = 3$) [32]	0.110	0.293	0.422	0.570	8.404	13.761
I*	Percentile Range ($p = 1$) [32]	0.123	0.326	0.470	0.633	7.126	12.312
I*	Percentile Range ($p = 0.3$) [32]	0.134	0.347	0.502	0.672	6.372	11.456
I*	Total Variation (L2) [30]	0.167	0.442	0.611	0.770	5.488	11.409
I*	Sum of Modified Laplacian [25]	0.209	0.524	0.706	0.852	4.169	9.781
I*	Diagonal Laplacian [41]	0.210	0.528	0.709	0.857	4.006	9.467
I*	Laplacian Energy [37]	0.208	0.520	0.701	0.852	3.917	9.062
I*	Laplacian Variance [27]	0.195	0.496	0.672	0.832	3.795	8.239
I*	Mean Local Log-Ratio ($\sigma = 1$)	0.220	0.559	0.751	0.906	2.652	6.396
I*	Mean Local Ratio ($\sigma = 1$) [15]	0.220	0.559	0.751	0.906	2.645	6.374
I*	Mean Local Norm-Dist-Sq ($\sigma = 1$)	0.219	0.562	0.752	0.907	2.526	5.924
I*	Wavelet Sum ($\ell = 2$) [47]	0.210	0.547	0.752	0.918	2.392	5.650
I*	Mean Gradient Magnitude [40]	0.210	0.545	0.747	0.915	2.359	5.284
I*	Wavelet Variance ($\ell = 2$) [47]	0.198	0.522	0.731	0.906	2.398	5.105
I*	Gradient Magnitude Variance [27]	0.205	0.536	0.739	0.909	2.374	5.103
I*	Wavelet Variance ($\ell = 3$) [47]	0.162	0.429	0.636	0.854	2.761	5.006
I*	Wavelet Ratio ($\ell = 3$) [44]	0.161	0.430	0.640	0.862	2.706	4.856
I*	Mean Wavelet Log-Ratio ($\ell = 2$)	0.208	0.544	0.753	0.927	2.191	4.843
I*	Mean Local Ratio ($\sigma = 2$) [15]	0.221	0.570	0.772	0.931	2.072	4.569
I*	Wavelet Ratio ($\ell = 2$) [44]	0.199	0.527	0.734	0.911	2.265	4.559
I*	Mean Local Log-Ratio ($\sigma = 2$)	0.221	0.571	0.772	0.931	2.067	4.554
I*	Wavelet Sum ($\ell = 3$) [47]	0.170	0.458	0.672	0.888	2.446	4.531
I*	Mean Local Norm-Dist-Sq ($\sigma = 2$)	0.221	0.572	0.770	0.929	2.056	4.395
I*	Mean Local Ratio ($\sigma = 4$) [15]	0.210	0.550	0.755	0.927	2.085	4.309
I*	Mean Local Log-Ratio ($\sigma = 4$)	0.211	0.551	0.755	0.927	2.083	4.305
I*	Mean Wavelet Log-Ratio ($\ell = 3$)	0.169	0.458	0.672	0.891	2.358	4.174
I*	Mean Local Norm-Dist-Sq ($\sigma = 4$)	0.212	0.555	0.760	0.928	2.059	4.164
I*	**Our Model**	**0.233**	**0.600**	**0.798**	**0.957**	**1.600**	**2.446**
D*	Normalized SAD [11]	0.166	0.443	0.636	0.819	4.280	8.981
D*	Ternary Census (L1, $\epsilon = 30$) [36]	0.171	0.450	0.633	0.802	4.347	8.794
D*	Normalized Cross-Correlation [2,11]	0.168	0.446	0.639	0.824	4.149	8.740
D*	Rank Transform (L1) [49]	0.172	0.451	0.633	0.811	4.138	8.558
D*	Census Transform (Hamming) [49]	0.179	0.473	0.663	0.842	3.737	8.126
D*	Ternary Census (L1, $\epsilon = 10$) [36]	0.178	0.472	0.664	0.841	3.645	7.804
D*	Normalized Envelope (L2) [3]	0.155	0.432	0.633	0.856	2.945	5.665
D*	Normalized Envelope (L1) [3]	0.165	0.448	0.653	0.870	2.731	5.218
D*	**Our Model**	**0.241**	**0.606**	**0.807**	**0.955**	**1.611**	**2.674**
D1	ZNCC Disparity with Calibration	0.064	0.181	0.286	0.448	8.879	12.911
D1	SSD Disparity† [42]	0.097	0.262	0.393	0.547	7.537	11.374
D1	Learned Depth† [9]	0.108	0.289	0.428	0.586	7.176	11.351
D1	**Our Model**	**0.164**	**0.455**	**0.653**	**0.885**	**2.235**	**3.112**
I1	**Our Model**	0.115	0.318	0.597	0.691	4.321	6.737

Table 1. Results of our model and baselines on the test set for four different versions of the autofocus problem. The leftmost column indicates problem type with I* meaning the full focal stack of green-channel images is passed to the algorithm. In D*, the full focal stack of dual-pixel data is passed to the algorithm. In D1, a randomly chosen dual-pixel focal slice is passed to the algorithm and in I1, a randomly chosen green-channel slice is passed. Results are sorted by RMSE independently for each input type. The top three techniques for each metric are highlighted with single slice techniques clubbed together. A † indicates that the results were computed on patches inside a 1.5x crop of the entire image.

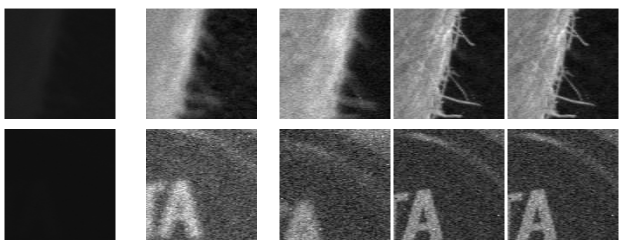
We demonstrate that our approach is better than numerous baselines on several variants of the autofocus problem.

We use similar error metrics as the Middlebury stereo dataset [34]: the fraction of patches whose predicted focal



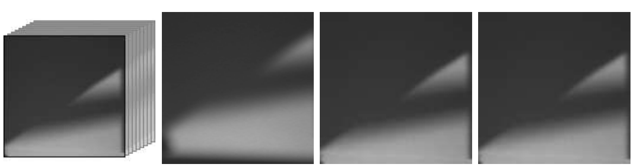
(a) Dual-pixel input (b) Baseline (c) Ours (d) GT

Figure 7. Qualitative results using Learned Depth† [9] and our D1 model. Given a defocused dual-pixel patch (a), the baseline predicts out-of-focus slices (b); our model predicts in-focus slices (c) that are similar to the ground truth (d).



(a) Input original (b) Input brightened (c) Baseline (d) Ours (e) GT

Figure 8. Qualitative results on low-light examples using ZNCC disparity as baseline and our D1 model on an example patch for a dark scene. The images have been brightened for visualization.



(a) Input stack I* (b) Baseline (c) Ours (d) GT

Figure 9. Qualitative result on an example patch (a) for I* . All 49 images are passed as input. The output (b) from the I* baseline Mean Local Norm-Dist-Sq ($\sigma = 4$) is out of focus. There is less dark image content in the output due to focal breathing which fools the contrast-based baseline. The output (c) from our I* model is the same as the ground truth (d).

indices have an error of no more than 0, 1, 2, or 4, as well as the mean absolute error (MAE) and root mean square error (RMSE). For the focal stack problem, all algorithms are run on all elements of the test set and aggregated. For the single-slice problem, an algorithm will be run on I_k for all k. For the multi-step problem, each patch in the test set will be evaluated 49 different times, with different focal indices acting as the starting position.

We compare our model's performance against a wide range of baselines. For the baselines labeled as I* , we take all 49 images (*i.e.*, the sum of the two dual-pixel images) from the input focal stack, evaluate a sharpness metric for each image, and then take the top-scoring image as the predicted focal depth for the stack. This is basically contrast-based depth-from-defocus. We take the top performing techniques from a recent survey paper [29].

The baselines labeled as D* use dual-pixel images as input. Instead of maximizing contrast, they instead attempt to identify which slice in the dual-pixel focal stack has the most similar-looking left and right sub-images, under the assumption that the two sub-images of an in-focus image are identical. Because there is little prior work on dual-pixel autofocus or depth-from-focus using the entire focus stack, we use classical techniques in stereo image-matching to produce a similarity metric between the left and right images that we maximize.

Finally, the D1 baselines try to predict the in-focus index given only one dual-pixel image pair. These baselines compute a disparity between the left and right views. As these baselines lack the global knowledge of the entire focal stack, they require calibration mapping this disparity to focus distances in the physical world. This calibration is spatially-varying and typically less accurate in the periphery of the field-of-view [42]. Two of the baselines based on prior work only work in the center 1.5x crop of the image. We evaluate these baselines only in the crop region. This only helps those baselines, as issues like focal breathing and irregular PSFs are worse at the periphery. Please see the supplemental material for a description of the baselines.

7.1. Performance

Table 1 presents our model's performance for the full-focal green (I*), full-focal dual pixel (D*), single-slice green (I1), and single-slice dual pixel (D1) problems. Our D1 model significantly out-performs other single-slice algorithms, with a RMSE of 3.11 compared to the closest baseline value of 11.351, and MAE of 2.235 compared to 7.176. In other words, baselines were wrong on average by 14.6% of the focal sweep, whereas our learned model was wrong by only 4.5%. We also demonstrate improved performance for the full-focal sweep problem, with a MAE of 1.60 compared to 2.06 of Mean Local Norm-Dist. Our D* model also outperforms the baselines in its category but performs about the same as our I* model; despite having better within-0, within-1, and within-2 scores, it has slightly lower MAE and MSQE. In a visual comparison, we observed that both of our full-focal models produced patches which were visually very similar to the ground truth and were rarely blatantly incorrect. This suggests that both I* and D* have enough information to make an accurate prediction; as such, the additional information in D* does not provide a significant advantage.

7.2. Multi-step

Table 2 presents the results for the multi-step problem. Two D1 baselines were extended into multi-step algorithms by re-evaluating them on the results of the previous run's output. Both improve substantially from the additional step. In particular, these algorithms are more accurate on indices with less defocus blur (indices close to the ground truth). The first step serves to move the algorithm from a high blur slice to a lower blur slice and the second step then fine-tunes. We see similar behavior from our I1 model, which also improves substantially in the second step. We attribute this gain to the model solving the focus-blur ambiguity which we discuss more in Section 7.4. Our D1 model improves but by a smaller amount than other techniques, likely because it already has high performance in the first step. It also gains much less information from the second slice than the I1 model since there is no ambiguity to resolve.

	Algorithm	# of steps	higher is better $=0$	≤ 1	≤ 2	≤ 4	lower is better MAE	RMSE
D1	ZNCC Disparity with Calibration	1	0.064	0.181	0.286	0.448	8.879	12.911
		2	0.100	0.278	0.426	0.617	6.662	10.993
D1	Learned Depth[†] [9]	1	0.108	0.289	0.428	0.586	7.176	11.351
		2	0.172	0.433	0.618	0.802	3.876	7.410
D1	Our model	1	0.164	0.455	0.653	0.885	2.235	3.112
		2	**0.201**	**0.519**	**0.723**	**0.916**	**1.931**	**2.772**
I1	Our model	1	0.115	0.318	0.597	0.691	4.321	6.737
		2	0.138	0.377	0.567	0.807	2.855	4.088

Table 2. Multi-step problem. Note that the D1 Learned Depth model uses a 1.5x center crop on the images it evaluates; it evaluates on a subset of the test set which has generally fewer artifacts (eg. focal breathing, radial distortion, etc.).

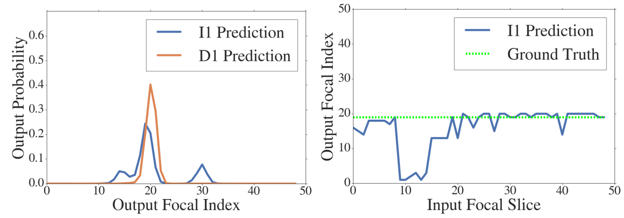
7.3. Performance with Registration

As stated in Section 4, focal breathing can cause errors in contrast-based techniques. Here, we estimate the magnitude of this problem by registering the focal stack to compensate for focal breathing and then re-evaluating the algorithms on the registered focal stack.

	Algorithm	higher is better $=0$	≤ 1	≤ 2	≤ 4	lower is better MAE	RMSE
I*	Mean Local Ratio ($\sigma = 2$) [15]	0.222	0.578	0.776	0.932	2.181	5.184
I*	Mean Local Log-Ratio ($\sigma = 2$)	0.222	0.579	0.776	0.932	2.176	5.178
I*	Mean Local Norm-Dist-Sq ($\sigma = 2$)	0.221	0.576	0.773	0.928	2.202	5.097
I*	Mean Local Ratio ($\sigma = 4$) [15]	0.212	0.565	0.773	0.940	1.923	3.920
I*	Mean Local Log-Ratio ($\sigma = 4$)	0.213	0.566	0.774	0.941	1.916	3.917
I*	Wavelet Sum ($\ell = 3$) [47]	0.194	0.520	0.731	0.922	2.019	3.558
I*	Mean Wavelet Log-Ratio ($\ell = 3$)	0.185	0.504	0.718	0.922	2.003	3.239
I*	**Our Model**	**0.251**	**0.610**	**0.809**	**0.957**	**1.570**	**2.529**

Table 3. Ablation study with regards to registrations. Existing techniques perform better when the focal stack has undergone a simple registration. However, our model trained on the registered data still performs better than the baselines.

Theoretically, the change in FoV due to focal breathing can be removed using a zoom-and-crop registration calibrated by the camera's focal distance. However, in practice, this registration is far from perfect and can introduce artifacts into the scene. Additionally, any noise in the measurement of focal distance means that a calibration-based registration may be imperfect. To evaluate this approach, we tested two different registrations: a zoom-and-crop registration calibrated by reported focal distance, and a grid search over zoom-and-crop registration parameters to minimize the L2 difference between the images. We note that both of these techniques led to registrations that eliminated some but not all of the change in FOV.



(a) (b)

Figure 10. (a) I1 and D1 model predictions for a patch given focal slice 25 as input. I1 model outputs a bimodal distribution as it struggles to disambiguate between the focal indices in-front of and behind the current slice that can generate the same focus-blur. D1 distribution is unimodal as dual-pixel data helps disambiguate between the two. For the same patch, I1 model's prediction for different input slices is visualized in (b). For focal slices that are towards the near or the far end, the model predicts correctly as one of the two candidate indices lie outside the range while the ambiguity is problematic for input slices in the middle. Inside this problematic range, the model tends to predict focal indices corresponding to depths which, while on the wrong side of the in-focus plane, would produce the same size circle of confusion.

Table 3 shows the performance of a model we trained and the best contrast techniques on the registered data. Most of the contrast algorithms improved when run on the registered focal stack, gaining approximately 0.1 MAE. This suggests that focal breathing affects their performance. In addition, our model trained and evaluated on registered data outperforms our model trained and evaluated on non-registered data.

7.4. Single-slice Focus-blur Ambiguity

In the single-slice problem, an algorithm given only the green-channel faces a fundamental ambiguity: out-of-focus image content may be on either side of the in-focus plane, due to the absolute value in equation 4. On the other hand, the model with dual-pixel data can resolve this ambiguity since dual-pixel disparity is signed (Equation 5). This can be seen from I1 vs D1 results in Table 2 where I1 single step results are significantly worse than single step D1 results, but the difference narrows down for the two step case where the ambiguity can be resolved by looking at two slices.

The ambiguity is also visualized in Figure 10(a) for a particular patch where the I1 model outputs a bimodal distribution while the D1 model's output probability is unimodal. Interestingly, this ambiguity is only problematic for focal-slices where both the candidate indices are plausible, i.e., lie between 0 and 49, as shown in Figure 10(b).

References

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

[2] Daniel I. Barnea and Harvey F. Silverman. A class of algorithms for fast digital image registration. *Transactions on Computers*, 1972.

[3] Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *TPAMI*, 1998.

[4] Marcela Carvalho, Bertrand Le Saux, Pauline Trouvé-Peloux, Andrés Almansa, and Frédéric Champagnat. Deep depth from defocus: how can defocus blur improve 3d estimation using dense neural networks? *ECCV*, 2018.

[5] C. Chan, S. Huang, and H. H. Chen. Enhancement of phase detection for autofocus. *ICIP*, pages 41–45, Sep. 2017.

[6] Chun-Hung Shen and H. H. Chen. Robust focus measure for low-contrast images. *International Conference on Consumer Electronics*, 2006.

[7] Raul Diaz and Amit Marathe. Soft labels for ordinal regression. *CVPR*, 2019.

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

[9] Rahul Garg, Neal Wadhwa, Sameer Ansari, and Jonathan T. Barron. Learning single camera depth estimation using dual-pixels. *ICCV*, 2019.

[10] Paul Grossmann. Depth from focus. *Pattern recognition letters*, 1987.

[11] Marsha Jo Hannah. *Computer Matching of Areas in Stereo Images.* PhD thesis, 1974.

[12] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[13] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *SIGGRAPH Asia*, 2016.

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

[15] Franz Helmli and Stefan Scherer. Adaptive shape from focus with an error estimation in light microscopy. *International Symposium on Image and Signal Processing and Analysis*, 2001.

[16] Berthold K.P. Horn. Focusing. 1968.

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

[18] Eric Krotkov. Focusing. *IJCV*, 1988.

[19] Sang Yong Lee, Yogendera Kumar, Ji Man Cho, Sang Won Lee, and Soo-Won Kim. Enhanced autofocus algorithm using robust focus measure and fuzzy reasoning. *Transactions on Circuits and Systems for Video Technology*, 2008.

[20] Sang Yong Lee, Jae Tack Yoo, and Soo-Won Kim. Reduced energy-ratio measure for robust autofocusing in digital camera. *Signal Processing Letters*, 2009.

[21] Orly Liba, Kiran Murthy, Yun-Ta Tsai, Tim Brooks, Tianfan Xue, Nikhil Karnad, Qiurui He, Jonathan T. Barron, Dillon Sharlet, Ryan Geiss, Samuel W. Hasinoff, Yael Pritch, and Marc Levoy. Handheld mobile photography in very low light. *SIGGRAPH Asia*, 2019.

[22] Tony Lindeberg. Scale-space for discrete signals. *TPAMI*, 1990.

[23] Hashim Mir, Peter Xu, Rudi Chen, and Peter van Beek. An autofocus heuristic for digital cameras based on supervised machine learning. *Journal of Heuristics*, 2015.

[24] Harsh Nanda and Ross Cutler. Practical calibrations for a real-time digital omnidirectional camera. Technical report, In Technical Sketches, Computer Vision and Pattern Recognition, 2001.

[25] Shree K. Nayar and Yasuo Nakagawa. Shape from focus. *TPAMI*, 1994.

[26] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Eden Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. Technical report, Stanford University, 2005.

[27] José Luis Pech-Pacheco, Gabriel Cristóbal, Jesús Chamorro-Martínez, and Joaquín Fernández-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. *ICPR*, 2000.

[28] Alex Paul Pentland. A new sense for depth of field. *TPAMI*, 1987.

[29] Said Pertuz, Domenec Puig, and Miguel García. Analysis of focus measure operators in shape-from-focus. *Pattern Recognition*, 2012.

[30] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 1992.

[31] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. *CVPR*, 2018.

[32] Andrés Benavides Santos, Carlos Ortiz de Solorzano, Juan J. Vaquero, Javier Márquez Peña, Norberto Malpica, and Francisco del Pozo. Evaluation of autofocus functions in molecular cytogenetic analysis. *Journal of microscopy*, 1997.

[33] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *TPAMI*, 2008.

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

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

[36] Fridtjof Stein. Efficient computation of optical flow using the census transform. *Pattern Recognition*, 2004.

[37] Murali Subbarao, Tae-Sun Choi, and Arman Nikzad. Focusing techniques. *Optical Engineering*, 1993.

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

[39] Huixuan Tang, Scott Cohen, Brian Price, Stephen Schiller, and Kiriakos N. Kutulakos. Depth from defocus in the wild.

In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[40] Jay Martin Tenenbaum. *Accommodation in Computer Vision*. PhD thesis, Stanford University, 1971.

[41] Andrea Thelen, Susanne Frey, Sven Hirsch, and Peter Hering. Improvements in shape-from-focus for holographic reconstructions with regard to focus operators, neighborhood-size, and height value interpolation. *TIP*, 2009.

[42] 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.

[43] Chong-Yaw Wee and Raveendran Paramesran. Measure of image sharpness using eigenvalues. *Information Sciences*, 2007.

[44] Hui Xie, Weibin Rong, and Lining Sun. Wavelet-based focus measure and 3-d surface reconstruction method for microscopy images. *IROS*, 2006.

[45] C. Yang and H. H. Chen. Gaussian noise approximation for disparity-based autofocus. *ICIP*, Sep. 2016.

[46] C. Yang, S. Huang, K. Shih, and H. H. Chen. Analysis of disparity error for stereo autofocus. *IEEE Transactions on Image Processing*, 2018.

[47] Ge Yang and B. J. Nelson. Wavelet-based autofocusing and unsupervised segmentation of microscopic images. *IROS*, 2003.

[48] Alan L Yuille and Tomaso A Poggio. Scaling theorems for zero crossings. *TPAMI*, 1986.

[49] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. *ECCV*, 1994.