Discriminative Filters for Depth from Defocus

Fahim Mannan and Michael S. Langer
School of Computer Science, McGill University
Montreal, Quebec H3A 0E9, Canada.
{fmannan, langer}@cim.mcgill.ca

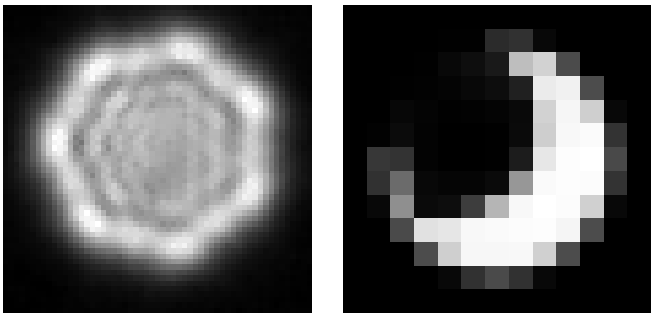
Abstract

Depth from defocus (DFD) requires estimating the depth dependent defocus blur at every pixel. Several approaches for accomplishing this have been proposed over the years. For a pair of images this is done by modeling the defocus relationship between the two differently defocused images and for single defocused images by relying on the the properties of the point spread function and the characteristics of the latent sharp image. We propose depth discriminative filters for DFD that can represent many of the widely used models such as the relative blur, Blur Equalization Technique, deconvolution based depth estimation, and subspace projection methods. We show that by optimizing the parameters of this general model we can obtain state-of-the-art result on synthetic and real defocused images with single or multiple defocused images with different apertures.

1. Introduction

Finite aperture cameras can only focus at a single distance or a focal plane at a time. Scene points outside that focal plane are defocused. When the defocus blur size is larger than a pixel, it becomes visible in an image. Blur size can be modeled using geometric optics and the thin-lens model in terms of the focal length, focus distance, aperture size, and scene depth. In real optics there are other factors such as lens aberrations, diffraction, and shape of the aperture that add to the characteristics of the blur pattern (Fig. 1). In depth from defocus, we are primarily concerned with the depth dependent variation in the blur or the point-spread function (PSF). If we can measure the blur from a given defocused image or an image pair with known camera parameters, then we can infer the depth of the corresponding scene points.

Many different methods have been proposed over the years to solve this problem. The common principle in most of these methods is to apply a depth dependent transformation and evaluate the quality or the reconstruction error due to the transformation. Transformations that produce small (ideally zero) error are indicative of depth. We refer to these different transformations as different models.



(a) Real PSF (57×57) [15] (b) Coded Aperture (13×13) [31]

Figure 1: Examples of PSFs from our dataset.

There are two main contributions of this work. First, we present a depth discriminative filter based model and show that a large set of existing models are special cases of our model. Second, we propose to learn the parameters of this model under a uniqueness constraint. We experimentally show that our model can work with fewer parameters than comparable learning based approaches and is more flexible and accurate.

The paper is organized as follows: Sec 2 reviews some of the works related to modeling depth from defocus. Sec. 3 shows that many of the existing depth from defocus models can be represented using a unified model. Sec. 4 proposes finding an optimal model using a data-driven approach. Depth discriminative filters obtained from our method are evaluated in Sec. 5 with synthetic and real images.

2. Related Work

The earliest and most widely used model for DFD is the relative blur model. For a pair of defocused images obtained using two different camera parameters, relative blur is the amount by which the sharper image is blurred to obtain the blurrier image. The model is motivated by the observation

that depth depends on the ratio of the Fourier transform of the two defocused images. This model is the basis of several works that estimated depth in the frequency domain [18, 24]. Later Ens and Lawrence [2] showed that relative blur estimation in the spatial domain performs better for noisy inputs. Most of these methods assume that the PSF is either a Gaussian or a pillbox. Watanabe and Nayar in [27] proposed modeling the normalized ratio or the ratio between the difference and sum of the images rather than the unnormalized versions of relative blur. Subbarao et al. [25] proposed a spatial domain approach for modeling the relative blur by considering a polynomial model for the image. Later Xian and Subbarao [29] extended the idea and used the commutative property of convolution with the Blur Equalization Technique (BET). In addition to these there is also the non-blind deconvolution based approach used for coded apertures [13, 31]. It should be noted that many of the later works that use relative blur and deconvolution include some type of smoothness prior for improving depth estimation accuracy e.g. [3, 5, 6, 14, 17, 20, 23].

Models such as relative blur, BET and deconvolution are primarily motivated by the blur formation process and aim to explain the defocus blur size by finding an equalizing transformation. On the other hand subspace projection methods such as [4, 16] try to learn the transformation from the data. They model the problem as projection of the defocused images to a depth dependent subspace. Favaro and Soatto [4] consider the null space of defocused images while Martinello and Favaro [16] consider the rank space of defocused image. However these methods do not try to ensure that the subspaces for different depths are independent from one another. Motivated by this, Wu et al. [28] proposed finding the independent orthogonal subspaces. Their work is closest to ours. All these methods require a large number of basis for the orthogonal subspaces. In contrast to these approaches, we only consider subspaces that are depth discriminative.

Filter based approaches such as rational filters [27, 19] or active DFD [8] are also similar to ours, in that their goal is to find optimal depth discriminative filters. However their filters are designed manually based on assumptions about the PSF or on the projected texture in the case of active DFD. Furthermore, they may have restriction on camera parameters (e.g. only variable focus) or range of depth (e.g. only between the two focal planes). Compared to these methods we learn the optimal filters from defocus data and do not make assumptions about the PSF.

Other learning based methods dealing with image blur, e.g. [21, 22, 26, 30], do not estimate depth and are about deblurring an entire image. Some of these methods use convolutional neural networks (CNN) [7, 11] where the input image is convolved with a set of filters. However, those filters are different from ours in that they are used to reduce the number of parameters and find shift-invariant features that are useful for later stages. Our filters are depth discriminative and only classify patches (i.e. input is the size as the filters) rather than an entire image.

3. Depth Discriminative Filter Model

Convolution as Matrix-Vector Product From the linear space-invariant model of blurred image formation, if the blur kernel corresponding to a scene point at depth d is h_d, and the latent sharp image is i_0, then the observed blurred image is:

$$i_d = h_d * i_0. \tag{1}$$

In matrix notation the above convolution can be written as,

$$x_d = H_d x_0$$

where H_d is the convolution matrix corresponding to the PSF h_d, and x_0, x_d are vectorized versions of latent and observed images i_0 and i_d respectively. The vectorized image is formed by stacking all the image columns into a single column vector. The convolution matrix H_d can have different structures based on the boundary condition. But in general, each row is a vectorized version of the spatial PSF (h_d) with the elements of the PSF organized in such a way that the dot product with the image results in the final pixel value. More details on these convolution matrices can be found in [9].

Patch-centric View of Convolution In this work, we consider image patches rather than an entire image. This results in the convolution being represented as a dot product between the PSF and an image patch, i.e.,

$$x_d(p) = h_d^T x_0^p.$$

Here x_0^p is an image patch centered at pixel p (e.g. the front defocused image at the top of Fig. 2), and $x_d(p)$ is the value of the pixel in the observed image (i.e. the dotted pixel in the figure). To correctly represent convolution, h corresponds to a 180° rotated (i.e. flipped up-down and left-right) version of the PSF.

Depth Discriminative Filter Representation We now generalize this idea of dot product between a patch and a kernel to the problem of depth estimation. From here on we will use x to denote a vectorized patch or a concatenation of patches centered around some pixel (as shown on the top-right of the figure). We consider the problem of depth estimation to be applying a set of depth discriminative filters to x, and choosing depth based on the filter response. For instance, to estimate depth at a pixel of a single image, we take a patch centered at that pixel and compute the

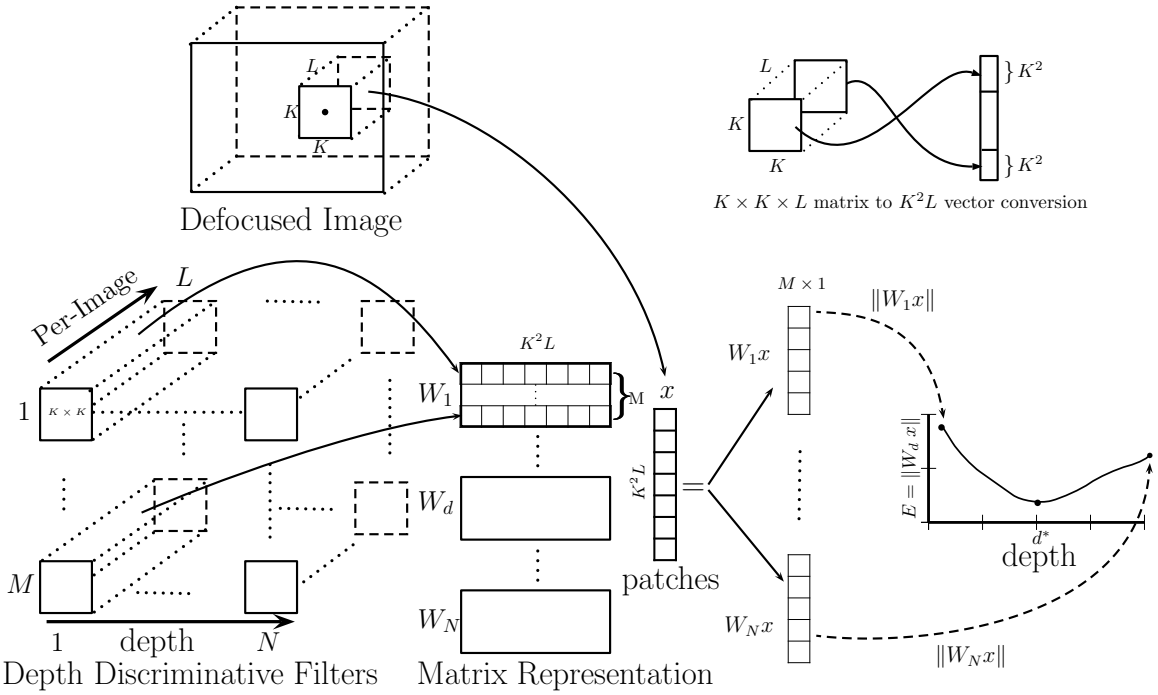


Figure 2: Depth from defocus can be considered as applying a set of depth discriminative filters (bottom-left) to a set of defocused image patches (top). These filters can either be analytically derived, or estimated from calibrated PSFs, or learned from defocused images. The filters can be represented using a set of matrices $W = \{W_1, \ldots, W_N\}$ of size $\mathbb{R}^{M \times K^2 L}$, where M is the number of filters of size $K \times K$ for L patches with depth $d \in [1, N]$. The estimated depth is d^*, if for any other depth d, $\|W_{d^*} x\| < \|W_d x\|$.

magnitude of the response of a set of filters at every depth. The depth of the pixel is chosen to be the depth of the filter bank that gives the lowest value. For multiple differently defocused images of the same scene at a given depth (i.e. $L > 1$), we take the inner product of three-dimensional filters of size $K \times K \times L$ with the patch-volume centered at the pixel. For depth d and vectorized patch-volume x, this operation can be represented as $W_d x$, where each row of W_d corresponds to L filters of size $K \times K$ as shown in Fig. 2. In general, we consider M filter volumes, or a mapping from K^2L-dimensional defocused patch space to M-dimension space. Therefore, for a set of defocused image patches $x \in \mathbb{R}^{K^2 L}$ and a set of depth discriminative filters $W_d \in \mathbb{R}^{M \times K^2 L}$, the following problem is solved for each pixel:

$$\underset{d}{\operatorname{argmin}} \|W_d x\|^2. \tag{2}$$

DFD and Orthogonality In the rest of this section, we represent some of the widely used DFD models using the above general formulation. These models are built on the assumption that in the absence of noise and modeling errors, for every depth d there exists W_d, such that $\|W_d x_d\|^2 = 0$, where x_d[1] is any vectorized patch at depth d.

3.1. Relative Blur

The relative blur model assumes that the sharper (x_S) and blurrier images (x_B) are related by a depth dependent relative blur, i.e., in terms of the convolution matrix notation,

$$x_B = H_R x_S + \mathcal{N}(0, \sigma_N^2). \tag{3}$$

Therefore, the relative blur estimation problem can be written as the following optimization problem:

$$\underset{d}{\operatorname{argmin}} \left\| \begin{bmatrix} I, & -H_R(d) \end{bmatrix} \begin{bmatrix} x_B \\ x_S \end{bmatrix} \right\|^2. \tag{4}$$

In the patch-level representation $w_d^T = [e_c^T, -h_R^T(d)]$, where c is the center-pixel's index. In terms of Fig. 2, at every depth $M = 1$ and $L = 2$.

[1]Notation: We use x to denote some patch at an unknown depth, subscript of x e.g. x_d to denote any patch at depth d, x^t to denote t-th patch in a training set, and x_1, x_2, x_S, x_B to denote 1st, 2nd, sharper and blurrier images or patches depending on the context.

In the absence of noise and Gaussian PSFs, we have, $w_d^T x_d = 0$. When the PSFs are not Gaussian we can formulate the relative blur estimation problem as finding the closest orthogonal vector to the observed defocused images. That is,

$$\underset{h_R}{\operatorname{argmin}} \left\| [e_c^T, -h_R^T] \begin{bmatrix} x_B \\ x_S \end{bmatrix} \right\|^2 \quad \text{subject to } \|h_R\|_1 = 1, \ h_R \geq 0. \tag{5}$$

3.2. Blur Equalization Technique

Blur equalization relies on the commutative property of convolution which in the matrix form can be expressed as:

$$H_2 x_1 = H_1 x_2 + \mathcal{N}(0, \sigma_N^2). \tag{6}$$

This results in the following depth estimation problem,

$$\underset{d}{\operatorname{argmin}} \left\| \begin{bmatrix} H_2(d) & -H_1(d) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right\|^2. \tag{7}$$

In terms of the patch-level representation $w_d^T = [h_2^T(d), h_1^T(d)]$ and $M = 1, L = 2$.

In the absence of noise and PSF modeling error, w_d is orthogonal to the observed defocused images at depth d, i.e., $w_d^T x_d = 0$.

3.3. Deconvolution

In the deconvolution based depth estimation approaches, the first step is to obtain a deconvolved image patch for different depth hypotheses. Then quality of the reconstruction is evaluated. The depth hypothesis that results in the highest quality sharp patch (e.g. sharp and no ringing artifacts) is considered to be the depth label. For known PSFs H_d for depth d, the deconvolution based formulation for DFD is:

$$\underset{x_0, d}{\operatorname{argmin}} \|H_d x_0 - x\|^2 + \lambda ||C x_0||^2. \tag{8}$$

Here x_0 is the latent sharp patch and H_d is a convolution matrix or a concatenation of matrices for a given depth d. Similarly x can be a single patch or L patches. In the last term, C is a matrix representing a prior for natural images. The goal is to jointly find the latent patch and the depth estimate that explains the observed blurred patches.

A typical way to solve this problem is to first find a deconvolved patch for a given blur PSF H_d for the depth hypothesis d. The solution to this problem is:

$$\hat{x}_0(d) = H_d^+ x \tag{9}$$

where H_d^+ is an inversion matrix for depth d. Solving Eq. 8 for x_0 gives the inversion matrix to be

$$H_d^+ = (H_d^T H_d + \lambda C^T C)^{-1} H_d^T. \tag{10}$$

But there are also other ways of obtaining an inversion matrix, such as the Truncated Singular Value Decomposition (TSVD), where the prior on patches imply discarding the singular vectors corresponding to singular values below some threshold. In this case, the inversion matrix is:

$$H^+ = V_k \Sigma_k^{-1} U_k^T \tag{11}$$

where $H = U \Sigma V^T$, and U_k, Σ_k, and V_k correspond to the first k singular vectors. Hence H_d^+ is the truncated inverse of the convolution matrix for depth d. The truncation threshold can be determined from spectral analysis of the PSF [9].

The above finds the latent sharp patch. For depth estimation we find which depth dependent PSF produces the correct sharp patch. This can be found from the reconstruction error of the observed patch, i.e.,

$$\underset{d}{\operatorname{argmin}} \|(I - H_d H_d^+) x\|^2. \tag{12}$$

Therefore, we have $W_d = I - H_d H_d^+$. It can be shown that W_d is symmetric (i.e. $W_d^T = W_d$) and idempotent (i.e. $W_d^T W_d = W_d$) and therefore W_d is an orthogonal projection operator. Furthermore, in the absence of noise and modeling error, $\|W_d x_d\|^2 = 0$. In terms of our discriminative filter representation, $M = K^2 L$, where the number of observed patches $L = 1$ or 2.

3.4. Subspace Projection

Based on the orthogonal projection matrix of W_d in Eq. 12, [4] proposes to find orthogonal projectors from known PSFs or learn them from defocused images. An orthogonal projector is a matrix $H^\perp$ with the property that $H^\perp = UU^T$, where U is an orthonormal matrix. For known PSFs, these orthogonal projectors are derived from the PSF, but for unknown PSFs they are learned from a training set of defocused images using the singular value decomposition. If columns of U_N correspond to the smallest singular values then $H^\perp = U_N U_N^T$.

Depth estimation can be expressed as matrix vector multiplication between the left null space vectors and the input images, i.e. $W_d = U_N$,

$$\underset{d}{\operatorname{argmin}} \|U_N^T(d) x\|^2. \tag{13}$$

Martinello and Favaro proposed an equivalent formulation in [16] where the images are projected onto the rank space U_R, i.e., $W_d = U_R$, or

$$\underset{d}{\operatorname{argmin}} -\|U_R^T(d) x\|^2. \tag{14}$$

In both cases, we need to choose the number of vectors (i.e. M in Fig. 2) that form the basis of the subspace. Usually

they are in the order of hundreds. As the number of vectors and the number of depth level grows the computational cost grows significantly.

Another limitation is that the subspace projection methods compute the subspaces of each depth independently from other depths. This does not enforce the subspaces to be maximally independent from one another. Based on this observation, Wu et al. [28] proposed an extension to subspace projection where they learn subspaces that are discriminative. For any two depths i and j, they enforce $\|W_i x_i\|^2 < \|W_j x_i\|^2 \ \forall j \neq i$. For a patch of size $K \times K$, the discriminative metric W_d is a matrix of size $K^2 \times K^2$. Therefore, for L defocused images, $M = K^2 L$.

To summarize, we have shown that many of DFD models can be expressed into a common representation of $\|W_d x\|$. More specifically, for different models, we show how to construct W_d. We have also shown under what conditions they satisfy the orthogonality assumption, i.e. $\|W_d x_d\|^2 = 0$.

4. Learning Depth Discriminative Filters

In the last section, we expressed the problem of depth from defocus as solving Eq. 2. The argmin operator has a unique solution d^* when

$$\|W_{d^*} x_{d^*}\| < \|W_d x_{d^*}\| \ \forall d \neq d^* \tag{15}$$

Different models result in different forms of W_d. The underlying assumption in all of them is that when W_d is orthogonal to defocused images x_d at depth d, i.e., $\|W_d x_d\| = 0$, then Eq. 15 is satisfied. In this section, we first discuss the limitations of the existing models and their assumptions, then propose our approach and discuss its advantages and disadvantages.

Limitations of Existing Models The existing models, except for the learning based approaches, are based on certain assumptions about the image formation process. They perform well when those assumptions hold. In general, these accuracy of the modeling assumptions can change with camera parameters and depth. For instance, relative blur may perform better when the aperture is smaller because in that case the PSFs can be modeled by a Gaussian. For larger aperture (i.e. less diffraction) with complex shape, BET or deconvolution would perform better than relative blur. Also as the size of the PSF changes with depth, different models may perform differently.

The subspace projection based methods learn from data and do not have the above limitations. However, they may not necessarily satisfy Eq. 15. While the discriminative metric learning approach enforces Eq. 15, it requires learning a large matrix and also requires large number of convolution operations during evaluation.

Our Proposed Approach We propose to directly learn the filters that satisfy Eq. 15 from a set of fronto-parallel defocused images collected at different depths. This leads to the following constrained optimization problem.

$$\begin{aligned} &\underset{W}{\text{argmin}} \ \rho(W) \\ &\text{subject to } \|W_{y_t} x^t\| < \|W_j x^t\| \ \forall t, j \ j \neq y_t \end{aligned} \tag{16}$$

Here x^t is the t-th training image patches and y_t is the associated depth. $\rho(W)$ is a regularization function on the filters which in our case is the squared Frobenius norm.

The above optimization problem can be solved in many ways. In this work, we transform it into the following unconstrained optimization problem using the Hinge-Loss [1, 12]:

$$\begin{aligned} \underset{W}{\text{argmin}} \ & \lambda_1 \rho(W) \\ & + \frac{\lambda_2}{N} \sum_{t,j} \rho_l(y_t, j) \max(0, E_{y_t} - E_j + m) \end{aligned} \tag{17}$$

Here, $E_{y_t} = \|W_{y_t} x^t\|$ and $E_j = \|W_j x^t\|$ and m is the margin between them. The regularization function ρ_l is an additional weight as a function of true label of the t-th example y_t and any label j. This ensures that wrong labels that are farther away from the correct label are penalized more than those that are closer to the correct label. As a result this weighting function maximizes the curvature of the cost function at the true depth and in turn minimizes the variance of the estimate. In our experiments we use $E_d = \log(\|W_d x\|)$, margin $m = 1$, and $\rho_l(y_t, j) = (a|y_t - j|)^k$, with $k \in [0, 2]$ and $a \in (0, 1]$.

Advantages Our approach allows for the filters to be orthogonal to defocused images. For instance, the individual rows can be relative blur or BET as long as the constraint in Eq. 16 is satisfied. In fact our solution allows for both models to co-exist even though they are not orthogonal to each other which is a requirement for [4, 16, 28]. Our approach is different from Wu et al. [28] in that we do not require W_d to be a projection operator and as a result we solve a smaller optimization problem. In our approach, we choose the kernel size K and the number of filters M per depth, where $M \ll K^2 L$. This way we are forcing the optimizer to find the best possible depth discriminative M filters per depth.

Disadvantages We share some of the disadvantages of other learning based approaches. For instance, we need to the learn the filters for different camera configurations. Similar to [28], our approach requires jointly learning all the filters. This is a computationally expensive process. But unlike Wu et al, we optimize fewer variables, i.e., MK^2LN instead of K^4L^2N.

Existence of Solution For certain camera configurations and aperture shapes, Eq. 15 may not be satisfiable. For instance, if the aperture is circularly symmetric and two images are taken with two different apertures with focus at the center of the scene, then scene points equidistant from the focal plane and on opposite sides will produce the same PSFs. In such cases, depth from defocus cannot be performed.

5. Experiments

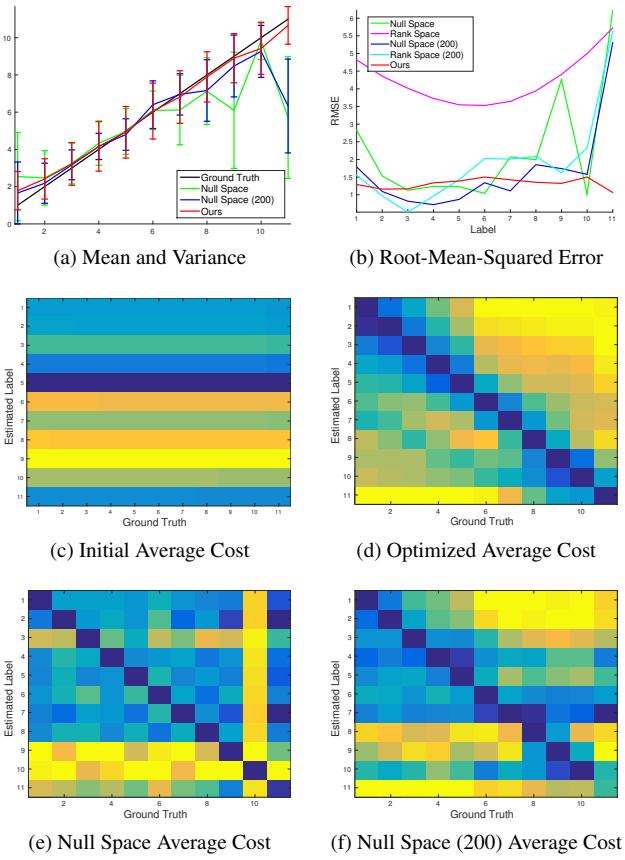
In this section we look at results for both synthetic and real defocus blur experiments. We show that the optimization process reduces the mean and variance of the depth estimates. Our experiments use both single images and a pair of images.

General Setup In all the experiments training and test images are formed from different sets of images. The training set is formed by taking the training images at different depths, partitioning them into blocks of size equal to the kernel size, and only considering patches that have variance above some threshold. We choose the number of filters, their size, and initialize them either randomly or using the left null space vectors corresponding to the lowest singular values. Starting with the left null space vectors usually results in faster convergence. Eq. 17 is optimized using batch gradient descent with the Adam update rule [10].

Image Dataset Synthetic defocused images are generated by taking a set of sharp textures and then blurring them with differently scaled PSFs. For real images, we collected a set of defocused images at 27 depths uniformly spaced in inverse depth between 0.61 m and 1.5 m, with 7 different aperture and 9 different focus distances. We present results for a subset of these configurations. For comparison with other methods such as the relative blur, BET and deconvolution, we first calibrate the PSFs for all the configurations using [15]. For the relative blur method, the relative blur kernels for the relevant configurations are also estimated using [15] which is a version of Eq. 5. In all the real image based experiments, the filters are trained on the $1/f$ (Fig. 4a) and Flower images (4b) and tested on the Bark images (Fig. 4c).

5.1. Depth from a Single Defocused Image

Synthetically Defocused with a Coded Aperture Fig. 3 shows the results for estimating depth from a single coded aperture image. Both the training and test images are synthetically generated using 11 PSF kernels of size ranging from 13×13 (Fig. 1b) to a single pixel. For each blur level, 10 Filters of size 21×21 were learned from random initialization. Fig. 3c shows the average cost i.e. average of $\|W_d x\|$ over many different x for the initial filters and Fig. 3d for the final optimized filters. Figs. 3e and 3f show results for the null space filters with different rank. It can be seen that our 10 filters per depth achieve equivalent accuracy and in certain cases are more accurate than using a large number (i.e. 200) of filters. For instance, when the PSF is a delta function then orthogonal projection based methods can have large error in the estimates, i.e. the rightmost column of Figs. 3e and 3f have off-diagonal minima.



(a) Mean and Variance (b) Root-Mean-Squared Error

(c) Initial Average Cost (d) Optimized Average Cost

(e) Null Space Average Cost (f) Null Space (200) Average Cost

Figure 3: Experiments using synthetically defocused images with a single coded aperture from Zhou et al. [31]. All plots correspond to evaluation on the test data. We used 11 equally spaced blurs with blur radii (in pixels) ranging from -5 (index 1) to 0 (index 11). For each blur level, 10 filters of size 21×21 were learned. Comparisons are shown for equivalent number (i.e. 10) and a larger number (i.e. 200) of null space (Eq. 13) and rank space (Eq. 14) filters. Rank space filter results are only shown in (b).

Real Defocused Images with a Conventional Aperture Fig. 5 shows the mean and standard deviation of inverse

(a) $1/f$ noise (b) Brodatz - Flower (c) Brodatz - Bark

Figure 4: Textures rendered on the display as captured by the camera. For all these images the object to sensor distance is 1.5 m, focus distance 1.22 m, and $f/11$ with 0.5 s exposure. The captured images are of size 4288×2848 pixels but we only use the center part for our experiments.

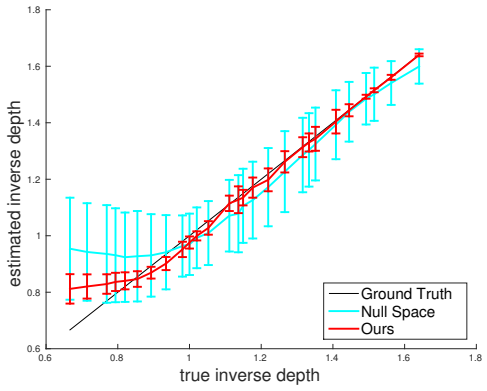


Figure 5: Mean and standard deviation of inverse depth estimates from a single real defocused image of a tree bark texture (Fig. 4c) taken with a conventional aperture with $f/22$ and focus at 0.61 m ($1.64D$ or diopters).

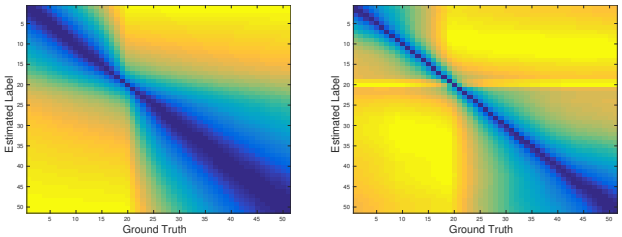
depth for a single defocused image with $f/22$ and focus at 0.61 m. For each depth 10 filters of size 21×21 are used. As the blur size gets larger, it gets more difficult to discriminate depth. However compared to the null space based filters (i.e. rank 10), our optimized filters perform significantly better. For our optimized filters, the average depth only deviates from the correct depth when the blur gets very large and for smaller blur sizes the variance is smaller.

5.2. Depth from Defocused Pair

Trained using Synthetically Defocused Images Fig. 6 shows result for Gaussian PSFs. The training set is synthetically generated assuming a scene within 52.9 cm to 86.9 cm and camera with focal length 25 mm and $f/8.3$. The largest blur radius is ≈ 2.3 pixels. The blur scale is divided into 51 depth levels and natural image textures are synthetically blurred to generate the training and test sets. In all cases, the training and test sets contain different image textures.

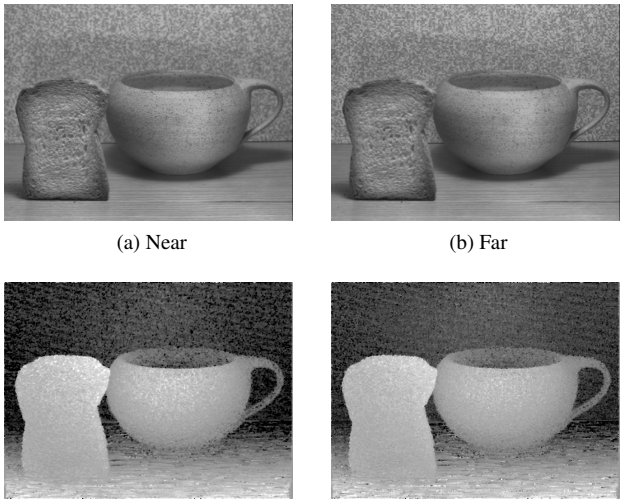
The strength of our approach is illustrated by Figs. 6a and 6b, which show the average cost from the null space filters and our optimized filters respectively. Each column corresponds to a depth, and each row, the response of a filter bank for the corresponding depth. Each cell is the average response of a few thousand patches at a given depth. Therefore, each column is the response of different filters for a given depth or the cost function for a given input. The columns are normalized to highlight the shape of the cost function. Ideally, we want to have a strong diagonal. From the plots, we can see that our approach produces a stronger diagonal than the same number of null space filters.

From the null space filters we can see that near the scene boundary the cost function gets flatter. Therefore we would expect high variance in the depth estimates in those regions. This is visible in the depthmaps estimated from real defocused images obtained in [27] (see Fig. 7). The top edge of the cup is noisier than our optimized filters. We used 3 filters of size 3×3 which is significantly smaller than the 7×7 filters used by [27].



(a) Null Space Average Cost (b) Our Average Cost

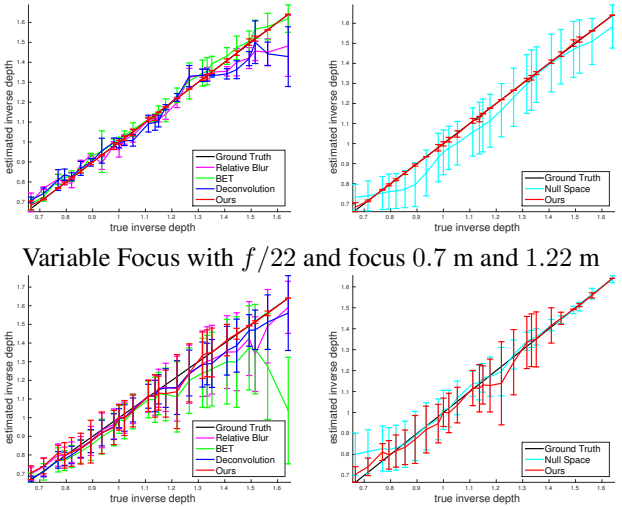
Figure 6: Average cost for Gaussian defocus pair. Each column is an average of a few thousand image patches. Blue indicates 0 and bright yellow 1. Our method forces the minimum to be on the diagonal.



(a) Near (b) Far

(c) Null Space (d) Optimized Filters

Figure 7: Top: The Breakfast dataset from Watanabe et al. [27]. Bottom: Results using $M = 3$ and $K = 3$ Null space filters and our depth discriminative filters post-processed using a 3×3 median filter. Ours have lower noise for the cup.



Variable Focus with $f/22$ and focus 0.7 m and 1.22 m

Variable Aperture with $f/22$ and $f/11$, and focus 0.61 m.

Figure 8: Mean and standard deviation of inverse depth estimates from real image based experiments. Top row corresponds to variable focus setting and bottom row to variable aperture setting. The left column compares our estimated filters with relative blur, BET, and deconvolution, and the right column with equivalent null space filters (Eq. 13).

Trained using Real Defocused Images Two different camera settings are used for the real image-pair based experiments shown in Fig. 8. The first setting is variable focus where the aperture is fixed to $f/22$ and the two images are taken at focus distances 0.7 m and 1.22 m. The second setting is variable aperture where the focus is fixed at 0.61 m and two images are taken with apertures $f/22$ and $f/11$. In both cases, we compare our proposed method to relative blur, BET, deconvolution (left column of the figure) and filters from the left null space (right column).

The variable focus case (top row of Fig. 8) uses 10 filters per depth image-pair of size 13×13. The variable aperture case (bottom row of Fig. 8) uses 10 filters of size 21×21. In all cases, our method works as well as or better than the other methods. In the experiments, most of the methods worked better for the variable focus case because the maximum blur size is smaller for variable focus than for variable aperture.

6. Conclusion

We have shown that DFD can be represented using a general depth discriminative filter based model. Our model specifies a matrix W_d for every depth. To demonstrate its generalization ability we expressed a large subset of existing models using our representation. These different models have different forms for W_d. Using a single representation also allows us to see the similarities and differences between these models. In particular, we show that most of the models prefer $\|W_d x_d\|^2 = 0$ under ideal conditions – e.g. noise-free and no modeling error. We argue that the main requirement for finding unique depth estimates is for the W_ds to satisfy Eq. 15 but the condition $\|W_d x_d\|^2 = 0$ is not sufficient for ensuring Eq. 15 holds. This led us to propose a learning based approach where we learn W_ds that satisfy Eq. 15.

We experimentally compared our learning based approach with other models using both synthetic and real images with different types of apertures and number of images (i.e. single or pair). For single image based experiments we found our approach to work better than subspace projection based methods with equivalent number of filters. For a pair of images, we have shown that our approach works as well as the other models and in some cases even better. By looking at the average of $\|W_d x\|$ over a large number of images (i.e. Fig. 3 and 6), we can see that our approach on average gives the correct solution (i.e. on the diagonal) with low variance (concentrated near the diagonal).

To conclude, we proposed a novel generalized formulation of the DFD problem and showed that we can learn the parameters of our model by enforcing a uniqueness constraint. We have experimentally verified that our proposed approach works well on a range of dataset.

References

[1] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, 2:265–292, Mar. 2002. 5

[2] J. Ens and P. Lawrence. An investigation of methods for determining depth from focus. *PAMI*, 15(2):97–108, 1993. 2

[3] P. Favaro. Recovering thin structures via nonlocal-means regularization with application to depth from defocus. In *CVPR*, pages 1133–1140, June 2010. 2

[4] P. Favaro and S. Soatto. A geometric approach to shape from defocus. *PAMI*, 27(3):406 –417, march 2005. 2, 4, 5

[5] P. Favaro and S. Soatto. *3-D Shape Estimation and Image Restoration - Exploiting Defocus and Motion Blur*. Springer, 2007. 2

[6] P. Favaro, S. Soatto, M. Burger, and S. Osher. Shape from defocus via diffusion. *PAMI*, 30(3):518–531, March 2008. 2

[7] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980. 2

[8] O. Ghita, P. F. Whelan, and J. Mallon. Computational approach for depth from defocus. *Journal of Electronic Imaging*, 14(2):023021–023021–8, 2005. 2

[9] P. Hansen, J. Nagy, and D. O'Leary. *Deblurring Images*. Society for Industrial and Applied Mathematics, 2006. 2, 4

[10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 6

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. 2

[12] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 1:0, 2006. 5

[13] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3):70, 2007. 2

[14] F. Li, J. Sun, J. Wang, and J. Yu. Dual-focus stereo imaging. *Journal of Electronic Imaging*, 19:043009, 2010. 2

[15] F. Mannan and M. S. Langer. Blur calibration for depth from defocus. In *CRV*, 2016. 1, 6

[16] M. Martinello and P. Favaro. *Video Processing and Computational Video: International Seminar, Dagstuhl Castle, Germany, October 10-15, 2010. Revised Papers*, chapter Single Image Blind Deconvolution with Higher-Order Texture Statistics, pages 124–151. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. 2, 4, 5

[17] V. Namboodiri, S. Chaudhuri, and S. Hadap. Regularized depth from defocus. In *ICIP*, pages 1520 –1523, oct. 2008. 2

[18] A. P. Pentland. A new sense for depth of field. *PAMI*, 9:523–531, July 1987. 2

[19] A. N. J. Raj and R. C. Staunton. Rational filter design for depth from defocus. *Pattern Recognition*, 45(1):198 – 207, 2012. 2

[20] A. Rajagopalan and S. Chaudhuri. An MRF model-based approach to simultaneous recovery of depth and restoration from defocused images. *PAMI*, 21(7):577 –589, jul 1999. 2

[21] U. Schmidt, C. Rother, S. Nowozin, J. Jancsary, and S. Roth. Discriminative non-blind deblurring. In *CVPR*, pages 604–611, June 2013. 2

[22] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schlkopf. Learning to deblur. *TPAMI*, 38(7):1439–1451, July 2016. 2

[23] S. M. Seitz and S. Baker. Filter flow. In *ICCV*, pages 143 –150, 29 2009-oct. 2 2009. 2

[24] M. Subbarao. Parallel depth recovery by changing camera parameters. In *ICCV*, pages 149–155, dec 1988. 2

[25] M. Subbarao and G. Surya. Depth from defocus: A spatial domain approach. *IJCV*, 13(3):271–294, 1994. 2

[26] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, pages 769–777, June 2015. 2

[27] M. Watanabe and S. K. Nayar. Rational filters for passive depth from defocus. *IJCV*, 27:203–225, May 1998. 2, 7, 8

[28] Q. Wu, K. Wang, W. Zuo, and Y. Chen. *Neural Information Processing: 18th International Conference, ICONIP 2011, Shanghai, China, November 13-17, 2011, Proceedings, Part III*, chapter Depth from Defocus via Discriminative Metric Learning, pages 676–683. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. 2, 5

[29] T. Xian and M. Subbarao. Depth-from-defocus: Blur equalization technique. *SPIE*, 6382, 2006. 2

[30] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, pages 1790–1798, 2014. 2

[31] C. Zhou, S. Lin, and S. Nayar. Coded Aperture Pairs for Depth from Defocus and Defocus Deblurring. *IJCV*, 93(1):53, May 2011. 1, 2, 6