

Learning Shape from Defocus^{*}

Paolo Favaro[†] and Stefano Soatto[‡]

[†] Department of Electrical Engineering, Washington University, St. Louis - MO 63130, USA
fava@ee.wustl.edu

[‡] Department of Computer Science, University of California, Los Angeles - CA 90095, USA
soatto@ucla.edu

Abstract. We present a novel method for inferring three-dimensional shape from a collection of defocused images. It is based on the observation that defocused images are the null-space of certain linear operators that depend on the three-dimensional shape of the scene as well as on the optics of the camera. Unlike most current work based on inverting the imaging model to recover the “deblurred” image and the shape of the scene, we approach the problem from a new angle by collecting a number of deblurred images, and estimating the operator that spans their left null space directly. This is done using a singular value decomposition. Since the operator depends on the depth of the scene, we repeat the procedure for a number of different depths. Once this is done, depth can be recovered in real time: the new image is projected onto each null-space, and the depth that results in the smallest residual is chosen. The most salient feature of this algorithm is its robustness: not only can one learn the operators with one camera and then use them to successfully retrieve depth from images taken with another camera, but one can even learn the operators from simulated images, and use them to retrieve depth from real images. Thus we train the system with synthetic patterns, and then use it on real data without knowledge of the optics of the camera. Another attractive feature is that the algorithm does not rely on a discretization or an approximation of the radiance of the scene (the “deblurred” image). In fact, the operator we recover is finite-dimensional, but it arises as the orthogonal projector of a semi-infinite operator that maps square-integrable radiance distributions onto images. Thus, the radiance is never approximated or represented via a finite set of filters. Instead, the rank of the operator learned from real data provides an estimate of the intrinsic dimensionality of the radiance distribution of real images. The algorithm is optimal in the sense of \mathcal{L}^2 and can be implemented in real time.

1 Introduction

A real imaging system involves a map from a 3-D environment to a 2-D surface which is function of the optical settings (e.g. focal length, lens diameter etc.) and the shape of the scene. The intensity at each pixel in an image also depends upon the radiance distribution¹ “glued” to the observed surface.

^{*} This research is sponsored by ARO grant DAAD19-99-1-0139 and Intel grant 8029.

¹ We do not distinguish between the radiance and the reflectance of a scene since both camera and lights are not moving.

Estimating shape from defocus consists in retrieving the depth information of a scene exploiting the blurring variation of a number of images captured at different focus settings. Together with depth, one can also recover the radiance distribution. Thus, the whole system is not only useful as a modality to retrieve depth with a single camera, but can also be used, for example, to enhance pictorial recognition systems by providing shape cues.

Defocused images are typically modeled as the result of a linear operator with a kernel that depends upon the optics of the camera as well as the three-dimensional shape of the scene, acting on an “ideal” deblurred image. The process can be expressed as:

$$I_p(x, y) = \int h_p^s(x, y) dR \quad (1)$$

where I_p denotes the image generated with optics parameters p , (x, y) are coordinates lying on a compact discrete lattice (defined on the image plane), h_p^s is a family of kernels that depends on the scene shape s and the parameters p , and R is the radiance distribution defined on the continuous 3-D space.

If we collect a number K of images with different parameters p , and organize them as $I = [I_{p_1} \ I_{p_2} \ \dots \ I_{p_K}]$ and we do the same for the corresponding kernels $h^s = [h_{p_1}^s \ h_{p_2}^s \ \dots \ h_{p_K}^s]$, then the equations can be rewritten more compactly as:

$$I(x, y) = \int h^s(x, y) dR \quad (2)$$

and we can pose the estimation of shape and radiance from blurred images as the following optimization problem:

$$\hat{s}, \hat{R} \doteq \arg \min_{s, R} \left\| I - \int h^s dR \right\|^2 \quad (3)$$

where $\| \cdot \|$ is the Euclidean norm. Typically, the problem is addressed by inverting the imaging model to retrieve both the radiance distribution (the “deblurred” image) and the shape of the scene (the kernel parameters). This is a severely ill-posed inverse problem.

1.1 Relation to previous work

This work falls within the general category of passive methods for depth from defocus. That is, no active illumination is used, and no active control on the focus setting is performed (shape from focus [7, 9]). A common assumption in passive shape from defocus is that shape can be approximated locally with planes parallel to the image plane. This implies that the family of kernels describing the imaging process is shift-invariant and common techniques can be applied such as Fourier transforms, moment filters, estimating relative blurring between images, or approximating images in the spatial domain through polynomials or simple discretizations [4, 5, 10–14]. Some work has been done also when kernels are shift-varying [8] and Markov random fields have been proposed as models for both depth and focused image [1, 3]. Our approach differs from most current approaches in that we bypass both the modeling of the optics and the choice

of a finite-dimensional representations for the radiance. Instead, working in a learning framework, we automatically learn the defocusing process through the properties of linear operators. Our approach could potentially be applied to solve blind deconvolution problems in different fields.

When modeling the optics, one is faced with the problem of choosing the kernel family that best approximates the real imaging process. Kernel families (or *point spread functions*, PSF) adopted by most of the algorithms in the literature include *pillbox* functions and Gaussians. Also, during the shape reconstruction process, many algorithms require modeling the radiance using an ad-hoc choice of a basis of filters, or a discretization. Our approach does not entail any modeling of the PSF, nor a choice of a finite representation for the radiance distribution. We construct a set of linear operators (matrices) which are learned from blurred images. In order to do this, we generate a *training set* of images of a certain shape², by changing the radiance distribution defined on its surface. Then, we capture the surface parameters common to the training set using the singular value decomposition (SVD), and build an orthogonal projector operator such that novel images generated by the same shape, but with any radiance, belong to its null space. We repeat this procedure for a number of shapes and store the resulting set of operators. Then, we compute the residual for each stored operator. Depth is estimated at each pixel by searching for the operator leading to the minimum residual. These operations involve only a finite set of matrix-vector computations (of the size of the set of the pre-computed orthogonal operators) which can be performed independently at each pixel on the images, allowing for high parallelism. Hence, the whole system can be implemented in real-time. Notice that, since the same set of operators is used for any novel set of images, shape reconstruction is independent of the radiance distribution, and no approximations or finite representations is required.

2 Learning defocus

In this section we will explain in detail how to *learn* shape from defocused images through orthogonal operators, and how to use them to retrieve shape from novel images.

2.1 Formalization of the problem

Under mild assumptions on the reflectance of the scene, the integral in Eq. (2) can be interpreted in the Riemannian sense (see [2]). If we choose a suitable parameterization, we can rewrite Eq. (2) as $\int_{\mathbb{R}^2} h^s(x, y, \hat{x}, \hat{y})r(\hat{x}, \hat{y})d\hat{x}d\hat{y}$, where r is the radiant density. r belongs to the Hilbert space $\mathcal{L}^2(\mathbb{R}^2)$ with the inner product $\langle \cdot, \cdot \rangle : \mathcal{L}^2(\mathbb{R}^2) \times \mathcal{L}^2(\mathbb{R}^2) \mapsto \mathbb{R}$ defined by:

$$(f, g) \mapsto \langle f, g \rangle \doteq \int f(\hat{x}, \hat{y})g(\hat{x}, \hat{y})d\hat{x}d\hat{y}. \quad (4)$$

Since images are defined on a compact lattice $C \doteq \mathbb{R}^{M \times N}$ (the CCD array grid), where images have size $M \times N$ pixels, it is also useful to recall the finite dimensional

² In our current implementation we choose planes parallel to the image plane, but the method can be extended to other shapes as well.

Hilbert space \mathbb{R}^L of vectors, where $L = M \times N \times K$, with inner product $\ll \cdot, \cdot \gg$: $\mathbb{R}^L \times \mathbb{R}^L \mapsto \mathbb{R}$ defined as:

$$(V, W) \mapsto \ll V, W \gg \doteq \sum_{i=1}^L V_i W_i. \quad (5)$$

We define the linear operator $H_s : \mathcal{L}^2(\mathbb{R}^2) \mapsto \mathbb{R}^L$ such that $r \mapsto \langle h^s(x, y), r \rangle$. Using this notation, we can rewrite our imaging model as

$$I(x, y) = (H_s r)(x, y). \quad (6)$$

The problem can then be stated as

$$s, r = \arg \min_{s \in D, r \in \mathcal{L}^2(\mathbb{R}^2)} \|I - H_s r\|^2 \quad (7)$$

where D is a suitable compact set (the space of shapes), and the norm $\|\cdot\|$ is naturally induced by the inner product relative to the same Hilbert space, i.e. $\|V\|^2 = \ll V, V \gg$.

2.2 Learning null spaces

In order to compute explicitly our interest operators, we need some definitions. Since H_s is a linear bounded operator, it admits a unique adjoint $H_s^* : \mathbb{R}^L \mapsto \mathcal{L}^2(\mathbb{R}^2)$, mapping $I \mapsto \ll \hat{h}^s(\hat{x}, \hat{y}), I \gg$, where $\hat{h}^s(\hat{x}, \hat{y}, x, y) = h^s(x, y, \hat{x}, \hat{y})$, and such that

$$\ll H_s r, I \gg = \langle r, H_s^* I \rangle \quad (8)$$

for any $r \in \mathcal{L}^2(\mathbb{R}^2)$ and $I \in \mathbb{R}^L$. Now we are ready to define the orthogonal projector, which is a matrix $H_s^\perp : \mathbb{R}^L \mapsto \mathbb{R}^L$ with $I \mapsto H_s^\perp I = I - (H_s H_s^\dagger) I$, where H_s^\dagger is the (Moore-Penrose) pseudo-inverse. The pseudo-inverse, when it exists, is defined as the operator $H_s^\dagger : \mathbb{R}^L \mapsto \mathcal{L}^2(\mathbb{R}^2)$ such that $r = H_s^\dagger I$ satisfies $H_s^*(H_s r) = H_s^* I$.

The orthogonal operator H_s^\perp is useful in that the original (infinite dimensional) minimization in Eq. (7) can be transformed into a finite-dimensional one. This is guaranteed by the fact that the functionals

$$\Phi(s, r) = \|I - H_s r\|^2 \quad (9)$$

$$\Psi(s) = \|H_s^\perp I\|^2 \quad \text{with } r = H_s^\dagger I \quad (10)$$

have the same set of local extrema, assuming that there exists a non-null operator H_s^\perp and an operator H_s^\dagger , as discussed in [6, 11].

Rather than obtaining a closed form solution for H_s^\perp , we take a different point of view, and choose to compute it numerically exploiting its properties. H_s^\perp is a symmetric matrix (i.e. such that $H_s^\perp = (H_s^\perp)^T$) which is also idempotent (i.e. such that $H_s^\perp = (H_s^\perp)^2$). According to the first property, we can write H_s^\perp as the product of a matrix A of dimensions $m \times n$, $m \geq n$, with its transpose; for the second property we have that the columns of A must be orthonormal, and thus H_s^\perp can uniquely be written as:

$$H_s^\perp = A A^T \quad (11)$$

where $A \in V_{n,m}$ and $V_{n,m}$ is a Stiefel manifold³. Furthermore, from Eq. (10), the null space of H_s^\perp is all of the range of H_s . That is, given any set of T images $\{I_i\}_{i=1\dots T}$ generated by the same shape s_1 , for fixed focal settings, but changing radiance densities, we have $\|H_{s_1}^\perp I_i\|^2 = 0 \quad \forall i = 1 \dots T$, where $H_{s_1}^\perp$ has been trained with shape s_1 .

Now we are ready to list the 4 steps to *learn* H_s^\perp from data, once a specific class of shapes has been chosen:

1. Generate a number T of training images $I_1 \dots I_T$ of the shape s changing the optical settings; for example, see Figure 3. On the left two images have been generated. One horizontal stripe from each of the images corresponds to the same equifocal plane⁴ in the scene. Hence, we can split each of the two stripes in T square windows. Then, we couple corresponding windows from the two stripes and collect them into I_i . Every image I_i is generated by a different radiance density r_i . Then, rearrange I_i in a column vector of length L for each $i = 1 \dots T$;
2. Collect all images into a matrix $P = [I_1 \ I_2 \ \dots \ I_T]$ of dimensions $L \times T$. Apply the singular value decomposition (SVD) to P such that $P = USV^T$. Figure 1 shows the resulting P for a single depth level and two optical settings;
3. Determine the rank q of P (for example by imposing a threshold on the singular values);
4. Decompose U as $U = [U_1 \ U_2]$, where U_1 contains the first q columns of U and U_2 the remaining columns; then build H_s^\perp as:

$$H_s^\perp = U_2 U_2^T. \quad (12)$$

The number T of training images has to be at least bigger or equal than L (in our experiments we set $T = 2L$). Otherwise, the rank of P will not be determined by the defocusing process, but by the lack of data.

Remark 1. It is interesting to note that the rank of H_s^\perp relates inversely with the intrinsic dimensionality of the observed radiance. For example, if in the training phase we employ “simple” radiance densities, for example piecewise constant intensities, the rank of P will be low, and, as a consequence, the rank of the resulting operator H_s^\perp will be high. Intuitively, as the rank of H_s^\perp approaches L (the maximum value allowed), the bigger the set of possible orthogonal operators will be, and the bigger the space of shapes that can be distinguished.

This gives insight into how to estimate shape from defocus using linear operators, which is the case of most of the algorithms in the literature. There is a fundamental tradeoff between the precision with which we can determine shape and the precision with which we can model the radiance dimensionality: *the finer the shape estimation,*

³ A Stiefel manifold $V_{n,m}$ is a space where each point is a set of n orthonormal vectors $v \in \mathbb{R}^m$ and $n \leq m$: $V_{n,m} = \{X(m \times n) : X^T X = I_d(n)\}$, where $I_d(n)$ is the $n \times n$ identity matrix.

⁴ An equifocal plane is a plane parallel to both the image plane and the lens plane. By the thin lens law, such a plane contains points with the same focal.

and the more we require “simple” radiance densities. The more we model high dimensionality radiance densities, and the rougher the shape estimation.

2.3 Estimating shape

Certainly, it would not be feasible to apply the previous procedure to “recognize” any kind of shape. The set of orthogonal operators we can build is always with finite dimension, and hence we are only able to reconstruct shapes (which are defined in the continuum) up to a class of equivalence. Even restricting the system to recognize a small set of shapes would not be the best answer to the problem, since that would reduce the range of applications. Therefore, we choose to simplify the problem by assuming that locally, at each point on the image plane, the corresponding surface can be approximated with an equifocal plane. Thus, restricting our attention to a small window around each pixel, we estimate the corresponding surface through one single parameter, its depth in the camera frame. Notice that the image model (2) does not hold exactly when restricted to a small window of the input image. In fact, it does not take into account additional terms coming from outside the window. However, these terms are automatically modeled during the training phase (see Section (2.2)), since we obtain the operators H_s^\perp using images with such artifacts. The overall effect is a “weighting” of the operators, where components relative to the window boundary will have smaller values than the others (see Figure 2).

Also, to achieve real-time performance, we propose to pre-compute a finite set of operators for chosen depths. Then, the estimation process consists in computing the norm (10) for each operator and searching for the one leading to the smallest residual. The procedure can be summarized as follows:

1. Choose a finite set of depths \mathcal{S} (i.e. our depth resolution) and a window size $W \times W$ such that the equifocal assumption holds (in our experiments we choose windows of 7×7 pixels). Then pre-compute the corresponding set of orthogonal operators using the algorithm described in Section (2.2)
2. At each pixel (i, j) of the novel L input images, collect the surrounding window of size $W \times W$ and rearrange the resulting set of windows into a column vector I (of length LW^2)
3. Compute the cost function $\|H_d^\perp I\|^2$ for each depth value $d \in \mathcal{S}$ and search for the minimum. The depth value \hat{d} leading to the minimum is set to be the reconstructed depth of the surface at (i, j) .

This algorithm enjoys a number of properties that make it suitable for real-time implementation. First, the only operations involved are matrix-vector multiplications, which can be easily implemented in hardware. Second, the process is carried out at each pixel independently, enabling for high parallelism computations. It would be possible, for example, to have CCD arrays where each pixel is a computing unit returning the depth relative to it.

Also, as to overcome the limitations of choosing a finite set of depth levels, one can further refine the estimates by interpolating the computed cost function. The search process can be accelerated, using well-known descent methods (i.e. gradient descent, Newton-Raphson, tangents etc.), or using a dicotomic search, or exploiting the previous estimates as a starting point (this relies on smoothness assumptions on the shape). All these improvements are possible since the computed cost functions are smooth, as it can be seen in Figure 5.

3 Experiments

We test our algorithm on both synthetic and real images. In generating a synthetic set of images, we use a *pillbox* kernel, for simplicity. We use such synthetic images to train the system, and then test the reconstruction on real images captured using a camera kindly lent to us by S. K. Nayar. It has two CCDs mounted on carts with one degree of freedom along the optical axis. These carts can be shifted by means of micrometers, which allows for a fine tuning of the focal settings.

3.1 Experiments with synthetic data

In Figure 1 we show part of the training set used to compute the orthogonal operators. We choose a set of 51 depth levels and optical settings such that the maximum blurring radius is of 1.7 pixels. The first and last depth levels correspond to the first and second focal planes. We generate random radiance densities and for each of them compute the images as if they were generated by equifocal planes placed at all the chosen depth levels. Based on these images we compute the 51 orthogonal operators H_s^\perp , 6 of which are shown in Figure 2. Notice that the operators are symmetric and that as we go from the smallest value of the parameter s (relative to the first depth level) to the biggest one (relative to the last depth level), the distribution of “weights” moves from one side of the matrix to the other, corresponding to the image more in focus.

To estimate the performance of the algorithm we generate a set of two novel images of 51×2601 pixels that are segmented in horizontal stripes of 51×51 pixels (see Figure 3). Every stripe has been generated by the the same random radiance but with equifocal planes at decreasing depths as we move from the top to the bottom of the image. We generate a plane for each of the 51 depths, starting from 0.52m and ending at 0.85m. In Figure 4 we evaluate numerically the shape estimation performance. Both mean and standard deviation (solid lines) of the estimated depth are plotted over the ideal characteristic curve (dotted). Notice that even when no post-filtering is applied, the mean error (RMS) is remarkably small (3.778mm). In the same figure we also show the characteristic curve when a 3×3 median filter is applied to the resulting shape. While there is a visible improvement in the standard deviation of the shape estimation, the mean error (RMS) presents almost no changes (3.774mm).

In Figure 5 we show a typical minimization curve. As it can be seen, it is smooth and allows for using descent methods (to improve the speed of the minimum search) and interpolation (to improve precision in determining the minimum).

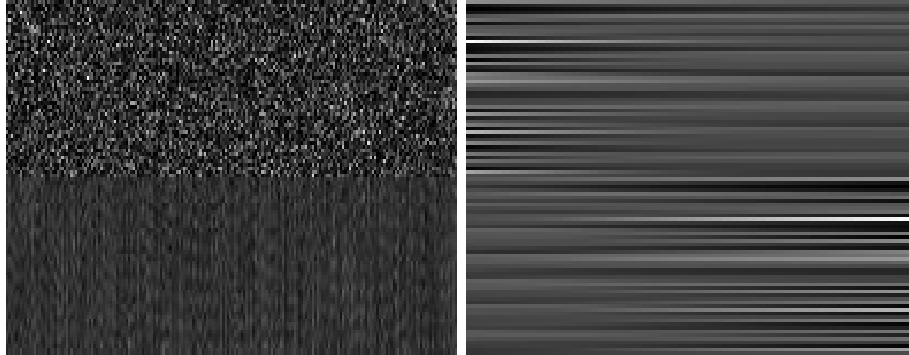


Fig. 1. Left: the training set for the first depth level (corresponding to blurring radii 0 and 1.7 pixels for the first and second image respectively). The resulting matrix has size 98×196 pixels. Right: the training set produced with a single radiance for all the 51 depth levels (the matrix has size 98×51 pixels).

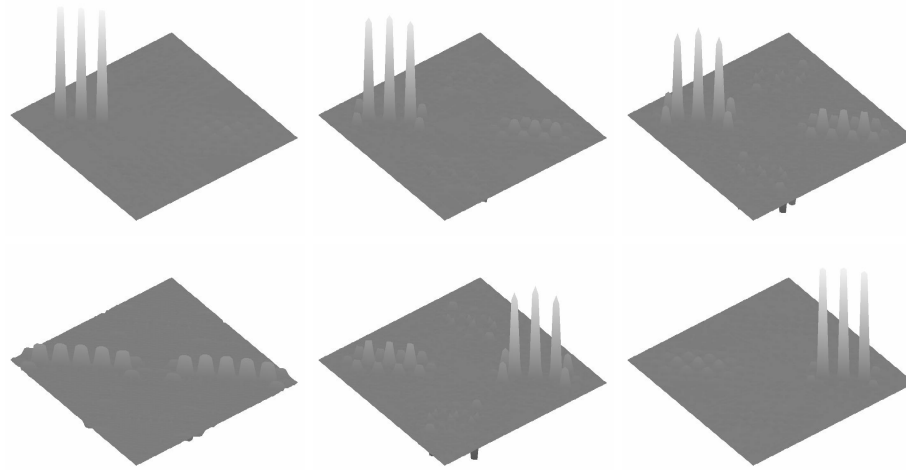


Fig. 2. The operator H_s^\perp as it appears for few depth levels. Top: from left to right, depth levels are 1, 10 and 15, corresponding to blurring radii of (0, 1.7) pixels, (0.45, 1.25) pixels and (0.66, 1.04) pixels respectively, where with (b_1, b_2) we mean the blurring radius b_1 is for the near focused image and the blurring radius b_2 for the far focused one. Bottom: from left to right, depth levels are 20, 35 and 40, corresponding to blurring radii of (0.85, 0.84) pixels, (1.3, 0.38) pixels and (1.45, 0.25) pixels. Notice that as we go from left to right and from top to bottom, the weights of the operators move from one side to the other, corresponding to which of the images is more focused.

3.2 Experiments with real data

In order to test the robustness of the proposed algorithm, we train it on synthetic images as described above, and then use it on real images obtained with two different

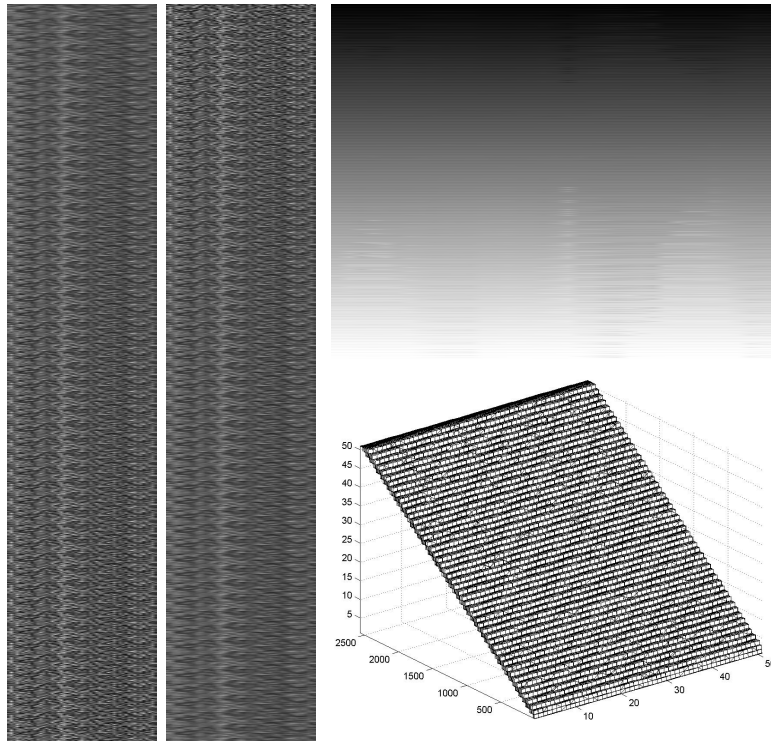


Fig. 3. Test with simulated data. Left and Middle: two artificially generated images are shown. The surface in the scene is a piecewise constant function (a stair) such that each horizontal stripe of 51×51 pixels corresponds to an equifocal plane. Depth levels decrease moving from the top to the bottom of the images. Top-Right: gray-coded plot of the estimated depths. Darker intensities mean large depths values, while lighter intensities mean small depth values. Bottom-Right: mesh of the reconstructed depths after a 3×3 pixels median filtering (re-scaling has been used for ease of visualization).

cameras. In the first data set we capture images using an 8 bits camera made of two independently moving CCDs, with focal length of 35mm and lens aperture of $F/8$. Figure 6 shows the two input images and the relative depth reconstruction. The far focused plane is at 0.77m while the near focused is at 0.70m. Windows of 7×7 pixels have been chosen. Figure 7 shows an experiment with the second data set (provided to us by S. K. Nayar) of a scene captured with a different camera. The focal length is of 25mm and the lens aperture of $F/8.3$. Even though no ground truth is available to compare to the reconstructed surface, the shape has been qualitatively captured in both cases.

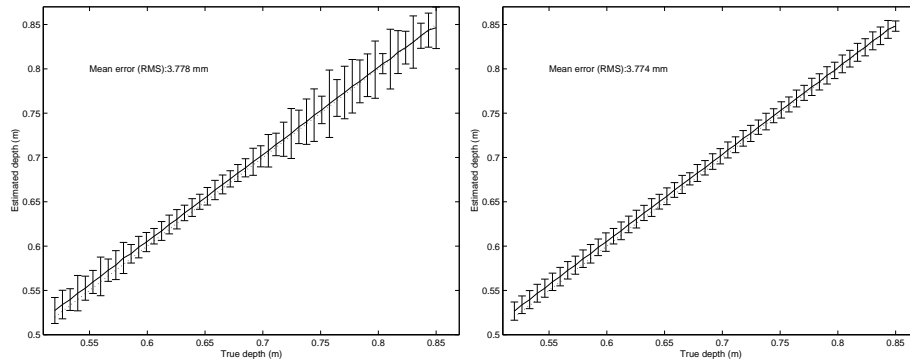


Fig. 4. Performance test. Left: evaluation of mean and standard deviation of the depth reconstruction. Right: evaluation of mean and standard deviation of the depth reconstruction after a 3×3 pixels median filtering. Notice that the mean error (RMS) over all depths both with or without pre-filtering is substantially low (3.778mm and 3.774mm respectively).

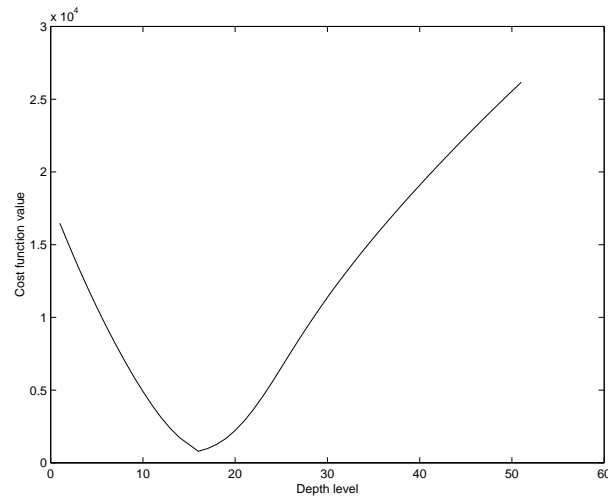


Fig. 5. One minimization curve. The curve is smooth and allows for fast searching methods. Also, interpolation can be employed to achieve higher precision in depth estimation.

4 Conclusions

We have presented a novel technique to infer 3-D shape from defocus which is optimal in the sense of \mathcal{L}^2 . We construct a set of linear operators H_s^\perp (matrices) learning the left null space of blurred images through singular value decomposition. This set of orthogonal operators is used to estimate shape in a second stage, computing the cost function $\|H_s^\perp I\|$ for each H_s^\perp and novel input images I . Then, depth is determined

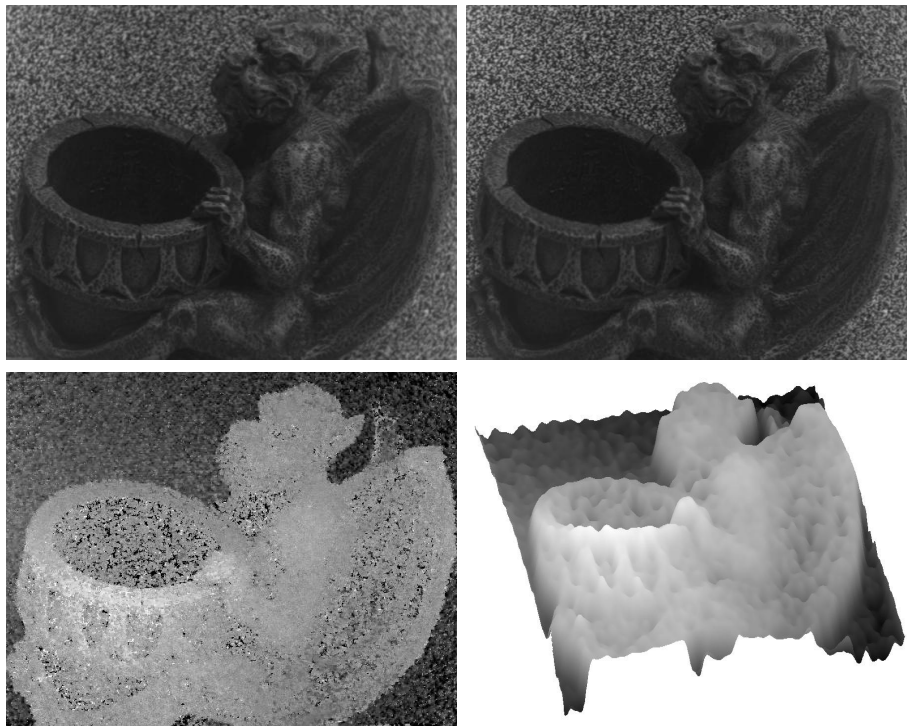


Fig. 6. First set of real images. Top: the two input images (of size 640×480 pixels). The left image is near focused (at 0.70m), while the right one is far focused (at 0.77m). Bottom-Left: gray-coded plot of the estimated depths. Dark intensities map to large depths, while light intensities map to small depths. Notice that where the radiance is too poor (for example inside the vase), depth recovery is not reliable. Bottom-Right: smoothed 3-D mesh of the reconstructed surface.

performing a simple minimization of the residual value. The algorithm is robust, since operators learned via simulated data can be used effectively on real data, and it does not entail any choice of basis or discretization of the radiance distribution. Furthermore, the rank of the orthogonal operators provides an estimate of the intrinsic dimensionality of the radiance distribution in the scene.

References

1. A. N. Rajagopalan and S. Chaudhuri. A variational approach to recovering depth from defocused images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, (no.10):1158–64, October 1997.
2. W. Boothby. *Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1986.
3. S. Chaudhuri and A. N. Rajagopalan. *Depth from defocus: a real aperture imaging approach*. Springer Verlag, 1999.

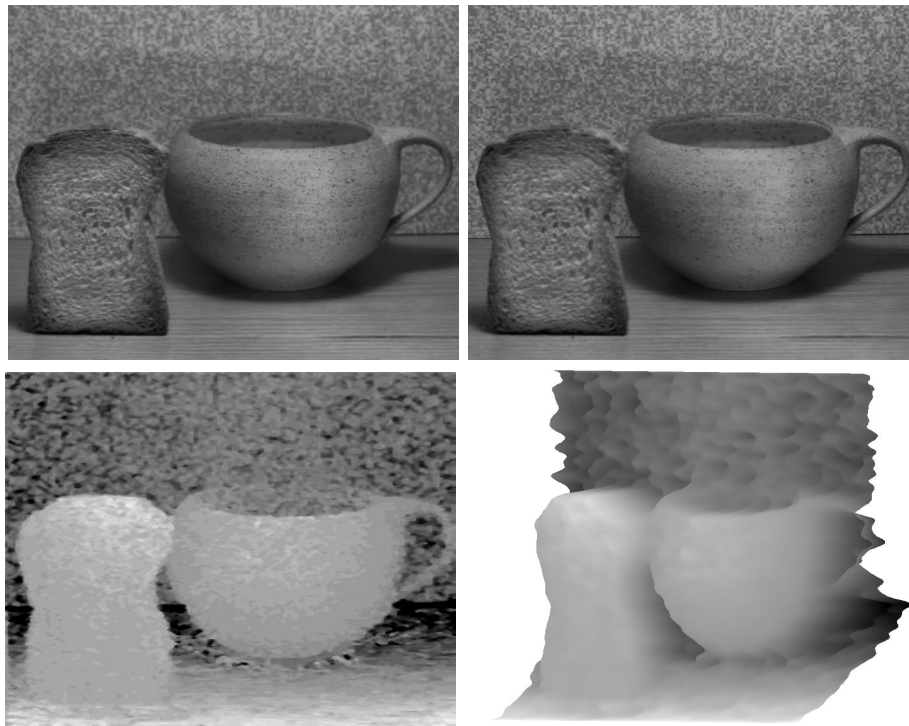


Fig. 7. Second set of real images (provided to us by S. K. Nayar). Top: the two input images (of size 320×480 pixels). The left image is near focused (at 0.53m), while the right one is far focused (at 0.87m). Bottom-Left: gray-coded plot of the estimated depths. Notice that where the radiance is too poor and the surface is far from being an equifocal plane (bottom of the table), depth recovery is not reliable. Bottom-Right: smoothed 3-D mesh of the reconstructed surface.

4. J. Ens and P. Lawrence. An investigation of methods for determining depth from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:97–108, 1993.
5. P. Favaro and S. Soatto. Shape and reflectance estimation from the information divergence of blurred images. In *European Conference on Computer Vision*, pages 755–768, June 2000.
6. G. Golub and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least-squares problems whose variables separate. *SIAM J. Numer. Anal.*, 10(2):413–532, 1973.
7. R. A. Jarvis. A perspective on range finding techniques for computer vision. In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, volume 2, page 122:139, March 1983.
8. H. Jin and P. Favaro. A variational approach to shape from defocus. In *Proc. of the European Conference on Computer Vision*, volume in press, May 2002.
9. S. Nayar and Y. Nakagawa. Shape from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):824–831, 1994.
10. A. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9:523–531, 1987.

11. S. Soatto and P. Favaro. A geometric approach to blind deconvolution with application to shape from defocus. In *Intl. Conf. on Computer Vision and Pattern Recognition*, pages 10–17, June 2000.
12. M. Subbarao and G. Surya. Depth from defocus: a spatial domain approach. *Intl. J. of Computer Vision*, 13:271–294, 1994.
13. M. Watanabe and S. K. Nayar. Rational filters for passive depth from defocus. *Intl. J. of Comp. Vision*, 27(3):203–225, 1998.
14. Y. Xiong and S. Shafer. Depth from focusing and defocusing. In *Proc. of the Intl. Conf. of Comp. Vision and Pat. Recogn.*, pages 68–73, 1993.