A Geometric Approach to Shape from Defocus

Paolo Favaro, *Member*, *IEEE*, and Stefano Soatto, *Member*, *IEEE*

Abstract—We introduce a novel approach to *shape from defocus*, i.e., the problem of inferring the three-dimensional (3D) geometry of a scene from a collection of defocused images. Typically, in shape from defocus, the task of extracting geometry also requires deblurring the given images. A common approach to bypass this task relies on approximating the scene locally by a plane parallel to the image (the so-called *equifocal* assumption). We show that this approximation is indeed not necessary, as one can estimate 3D geometry while avoiding deblurring without strong assumptions on the scene. Solving the problem of shape from defocus requires modeling how light interacts with the optics before reaching the imaging surface. This interaction is described by the so-called *point spread function* (PSF). When the form of the PSF is known, we propose an optimal method to infer 3D geometry from defocused images that involves computing orthogonal operators which are regularized via functional singular value decomposition. When the form of the PSF is unknown, we propose a simple and efficient method that first *learns* a set of projection operators from blurred images and then uses these operators to estimate the 3D geometry of the scene from novel blurred images. Our experiments on both real and synthetic images show that the performance of the algorithm is relatively insensitive to the form of the PSF. Our general approach is to minimize the Euclidean norm of the difference between the estimated images and the observed images. The method is geometric in that we reduce the minimization to performing projections onto linear subspaces, by using inner product structures on both infinite and finite-dimensional Hilbert spaces. Both proposed algorithms involve only simple matrix-vector multiplications which can be implemented in real-time.

Index Terms—Shape from defocus, depth from defocus, blind deconvolution, image processing, deblurring, shape, 3D reconstruction, shape estimation, image restoration, learning subspaces.

1 Introduction

We are interested in reconstructing the three-dimensional (3D) geometry of a scene from a collection of images. In computer vision such a task is called *shape-from-X*, where X denotes the cue used to infer shape. For example, one can capture images from different vantage points as in *stereo* and *motion* [18], [8], [4]. In this paper, instead, we consider images that are captured with different *optical settings* of the imaging device, which leads to the problem of *shape from defocus*.

In shape from defocus one can, for instance, take photographs of a scene while changing the relative position of the lens with respect to the CCD sensor. Notice that, when bringing a certain object into focus, objects that are away from it appear blurred and the amount of blur increases with the relative distance (see Fig. 1). This suggests that defocus and geometry are related and, therefore, it may be possible to estimate the geometry of a scene by measuring the amount of defocus in an image. However, one defocused image is not sufficient to obtain a unique reconstruction of the scene unless additional information is available. For example, one cannot distinguish between the sharp image of an object with blurred texture and the blurred image of an object with sharp texture. To cope with this ambiguity, one can analyze two or more defocused images obtained with different focus settings, as shown in Fig. 1.

- *P. Favaro is with the Electrical Engineering Department, University of Cambridge, Cambridge, UK. E-mail: favaro@cs.ucla.edu, pf255@cam.ac.uk.*
- *S. Soatto is with the Computer Science Department, University of California, Los Angeles, CA 90095. E-mail: soatto@ucla.edu.*

Manuscript received 30 Jan. 2003; revised 17 Feb. 2004; accepted 23 July 2004; published online 14 Jan. 2005.
Recommended for acceptance by R.C. Nelson.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 118205.

2 Relation to Previous Work

The general problem of shape from defocus has been addressed in a variety of contexts: Earlier approaches adopted Markov random fields to model both shape and appearance [6], [29], [30]. This approach has been shown to be effective for surface reconstruction from defocused images, but at the price of a high computational cost. Among deterministic approaches, we distinguish between those that maintain a spatial representation of the imaging model [7], [9], [10], [12], [21], [24], [25], [26], [27], [33], [37] and those that operate in the frequency domain [2], [15], [28], [31], [39]. In particular, most of the latter approaches are appealing since they allow one to formally eliminate undesired unknowns (the appearance, or "radiance"). However, the assumptions required in order to do so introduce artifacts in the solution due, for example, to noise and windowing [7], [20].

Another way to classify approaches to shape from defocus is based on simplifications of the image formation model. For example, some assume that the scene contains "sharp edges," i.e., discontinuities in the scene radiance [1], [19], [25], [34], [32], others that the radiance can be locally approximated by cubic polynomials [35], or that it can be controlled by using structured light [14], [22], [24]. A more common simplification of the image formation model is the so-called *equifocal assumption*, which consists of assuming that the surface of the scene can be locally approximated by a plane parallel to the image plane [19], [35], [25], [36], [38], [41]. One advantage of such an assumption is that it allows one to avoid reconstructing the appearance of the scene while recovering its geometry. However, it also fails to

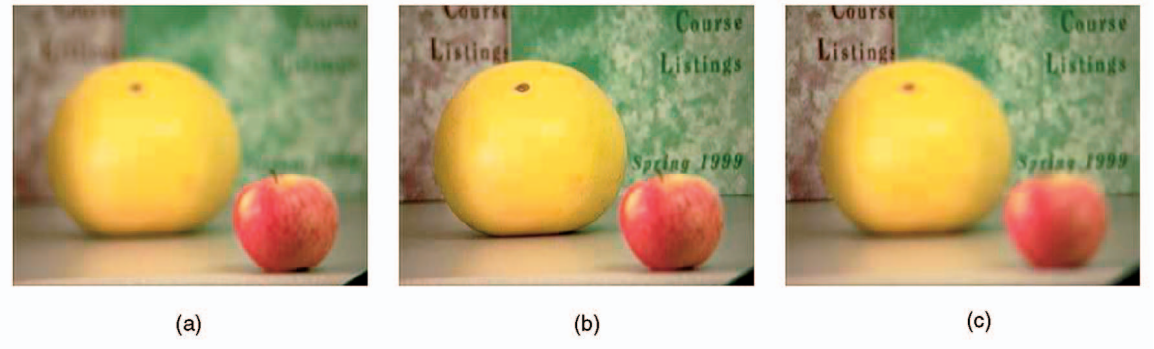


Fig. 1. Three images of the same scene taken with different camera settings. (a) The apple is brought into focus. (b) The grapefruit is brought into focus. (c) The background is brought into focus. When the background is brought into focus, both the grapefruit and the apple are defocused. In addition, the apple is more blurred than the grapefruit since it is farther from the background than the grapefruit.

properly capture a large class of surfaces (nonequifocal surfaces) and does not allow enforcing global regularity on the estimate. Approaches that do not make this assumption yield accurate estimates of geometry, but are computationally challenging because they require estimating the radiance of the scene along with geometry [6], [40].

We propose a novel approach that is not based on the above assumptions. We show that it is not necessary to simplify the imaging model to avoid estimating the appearance of the scene while recovering its geometry (Section 4.2). Based on this result, we propose two methods: one that can be used when an explicit characterization of the camera is available (Section 5.1) and one when it is not available (Section 5.2). Furthermore, the implementation of both methods does not require an explicit discretization or choice of basis as has been done in one way or another in most of the algorithms in the literature. In our approach, the size of the measurement array naturally imposes regularity in the solution, which is obtained in infinite-dimensional space using a functional SVD (singular value decomposition). We exploit the geometry of Hilbert spaces, which makes the analysis simple and intuitive.

For simplicity, in our current implementation, we restrict our attention to the simplest class of shapes, namely, local equifocal planes. However, we would like to stress the fact that the general algorithms we propose are not restricted to any specific model class: One could choose a class of slanted planes or curved surfaces, or avoid choosing classes altogether, by implementing a variational minimization, as suggested in Section 6. In this paper we are only interested in the simplest implementation of the proposed algorithms, for the purpose of showing the effectiveness of our approach. Our implementation only requires a finite set of matrix-vector computations (of the size of the set of the precomputed orthogonal operators) which can be performed independently at each pixel, allowing for a high level of parallelism. Our current experimental evaluation shows that these implementations have potential for real-time performance on current personal computers.

3 Problem Statement

In this section we introduce the problem of shape from defocus and the notation we will use later on. In particular, we will briefly introduce the image formation model for defocused images and outline the assumptions we make in deriving it.

A rigorous image formation model would require us to consider light interactions based on Maxwell's equations [5]. A simpler way of proceeding is, instead, to resort to notions of *optical geometry* (see, for example, [13]). We shall be content with a reasonable approximation of light interaction based on assuming that the surface is Lambertian,[1] that there are no occlusions or self-occlusions (see [3], [11] for these cases), that we are in a vacuum (see [23] when this hypothesis is removed), and that optics and surfaces are not dependent on light wavelength. These hypotheses allow us to describe the geometry of the scene with a function $s : \mathbb{R}^2 \mapsto [0, \infty)$ that we call *surface* and the appearance of the scene with a function $r : \mathbb{R}^2 \mapsto [0, \infty)$ that we call *radiance* with an abuse of terminology.[2] The function s maps points $\mathbf{x} \in \mathbb{R}^2$ on the lens plane to the distance of the surface of the scene from the lens.

We model the image plane as a finite-dimensional lattice (the CCD grid) $\Gamma = \mathbb{R}^M \times \mathbb{R}^N$, with coordinates $\mathbf{y} \in \Gamma$. A defocused image is a function $I : \Gamma \mapsto [0, \infty)$ that maps points $\mathbf{y}$ on the image plane to intensity values. Since we capture multiple images by changing the optics settings, we denote each image with I_i, where $i = 1, \ldots, K$ and K is the number of captured images.

Given the assumptions above, a defocused image I_i can be expressed as the result of a linear operator $h_i^s : \mathbb{R}^2 \times \Gamma \mapsto [0, \infty)$ that depends upon the optics of the camera as well as the three-dimensional shape of the scene, acting on the radiance r [25], [6]:

$$I_i(\mathbf{y}) = \int h_i^s(\mathbf{x}, \mathbf{y}) r(\mathbf{x}) d\mathbf{x}. \quad (1)$$

The operator h_i^s is called the *point spread function* (PSF).

1. A less stringent assumption is indeed sufficient to our purposes, as pointed out by an anonymous reviewer. We only require that all the points on the lens receive the same amount of energy from the same point-source on the scene. In typical situations, the dimension of the lens is relatively small in comparison to the distance of the scene from the camera. This means that we assume that the reflectance of the scene does not change within a small angle of viewing directions. In practice, this assumption is always satisfied if the scene is not made of highly specular surfaces.

2. In the context of *radiometry*, the term *radiance* refers to a more complex object that describes energy emitted along a certain direction, per solid angle, per foreshortened area and per unit time [13]. However, in our case, there is no dependency on direction and the change in the solid angle is negligible. Hence, a function of the position on the surface of the scene suffices to describe the variability of the radiance.

TABLE 1
Summary of the Operators Introduced in Sections 4.1 and 4.2 with Their Respective Domain, Codomain, and Functionality

operator	domain	codomain	functionality
H	continuum $(L^2(\mathbb{R}^2))$	discrete $(\mathbb{R}^P)$	blurs
H^*	discrete $(\mathbb{R}^P)$	continuum $(L^2(\mathbb{R}^2))$	blurs
$H^\dagger$	discrete $(\mathbb{R}^P)$	continuum $(L^2(\mathbb{R}^2))$	sharpens
$H^\perp$	discrete $(\mathbb{R}^P)$	discrete $(\mathbb{R}^P)$	maps range of H to 0

We are interested in inverting (1) by finding a radiance r and a surface s that verify the equation when I_i is measured on a CCD sensor. This problem is well-known to be ill-posed and, therefore, we will look for solutions that minimize a suitable *optimization criterion*, for instance, a regularized norm $\|\cdot\|$:

$$\hat{s}, \hat{r} \doteq \arg \min_{s,r} \sum_{i=1}^{K} \|n_i\| \quad \text{subject to} \tag{2}$$

$$I_i(\mathbf{y}) = \int h_i^s(\mathbf{x}, \mathbf{y}) r(\mathbf{x}) d\mathbf{x} + n_i(\mathbf{y}) \; \forall \, \mathbf{y} \in \Gamma \quad i = 1, \ldots, K. \tag{3}$$

4 A Least-Squares Solution

In this section, we introduce the core of our algorithm. We work in function space and use the geometry of operators between Hilbert spaces. For basic results on operators between finite and infinite-dimensional Hilbert spaces, see, for instance, [17].

4.1 Notation and Formalization of the Problem

For ease of notation we rearrange an image I in a column vector of dimension MN. Also, we collect a number K of images for different optics settings and organize them in a column vector by stacking each image on top of each other so that $\mathbf{I} = [I_1; \, I_2; \, \ldots; \, I_K] \in \mathbb{R}^P$, where $P = MNK$. If we do the same for the corresponding kernels $\mathbf{h}^s = [h_1^s; \, h_2^s; \, \ldots; \, h_K^s]$, then the equations (1) can be rewritten more compactly as:

$$\mathbf{I}(\mathbf{y}) = \int \mathbf{h}^s(\mathbf{x}, \mathbf{y}) r(\mathbf{x}) d\mathbf{x}. \tag{4}$$

We now want to write the above equation in a more concise form. To this end, consider the Hilbert space $L^2(\mathbb{R}^2)$ with the inner product $\langle \cdot, \cdot \rangle : L^2(\mathbb{R}^2) \times L^2(\mathbb{R}^2) \mapsto \mathbb{R}$ defined by:

$$(f, g) \mapsto \langle f, g \rangle \doteq \int f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}. \tag{5}$$

Since images are defined on a lattice Γ of size $M \times N$ pixels, it is also useful to recall the finite-dimensional Hilbert space $\mathbb{R}^P$ of vectors with inner product $\langle\langle \cdot, \cdot \rangle\rangle : \mathbb{R}^P \times \mathbb{R}^P \mapsto [0, \infty)$ defined as:

$$(V, W) \mapsto \langle\langle V, W \rangle\rangle \doteq \sum_{i=1}^{P} V_i W_i. \tag{6}$$

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

$$\mathbf{I}(\mathbf{y}) = (H_s r)(\mathbf{y}) \tag{7}$$

and impose that r belongs to the Hilbert space $L^2(\mathbb{R}^2)$. The problem (2)-(3) can then be stated as

$$\hat{s}, \hat{r} = \arg \min_{s,r} \|\mathbf{I} - H_s r\|^2, \tag{8}$$

where the norm $\|\cdot\|$ is naturally induced by the inner product relative to the same Hilbert space, i.e., $\|V\|^2 = \langle\langle V, V \rangle\rangle$.

4.2 Adjoints and Orthogonal Operators

By assuming H_s to be a linear bounded operator,[3] there exists a unique adjoint $H_s^* : \mathbb{R}^P \mapsto L^2(\mathbb{R}^2)$, mapping $\mathbf{I} \mapsto H_s^* \mathbf{I} \doteq \langle\langle \mathbf{h}^s(\mathbf{x}, \cdot), \mathbf{I} \rangle\rangle$ such that

$$\langle\langle H_s r, \mathbf{I} \rangle\rangle = \langle r, H_s^* \mathbf{I} \rangle \tag{9}$$

for any $r \in L^2(\mathbb{R}^2)$ and $\mathbf{I} \in \mathbb{R}^P$. The (Moore-Penrose) pseudoinverse $H_s^\dagger : \mathbb{R}^P \mapsto L^2(\mathbb{R}^2)$ is defined as the operator such that $\hat{r} = H_s^\dagger \mathbf{I}$ satisfies the equation

$$H_s^*(H_s \hat{r}) = H_s^* \mathbf{I}. \tag{10}$$

The orthogonal projection operator $H_s^\perp : \mathbb{R}^P \mapsto \mathbb{R}^P$ is defined such that

$$\mathbf{I} \mapsto H_s^\perp \mathbf{I} \doteq \mathbf{I} - H_s H_s^\dagger \mathbf{I}, \tag{11}$$

where $H_s^\dagger$ is the pseudoinverse. Note that the orthogonal projection operator is finite-dimensional and, therefore, represented by a matrix. A summary of all the operators introduced so far with their domain, codomain, and functionality is shown in Table 1.

The next proposition introduces the main result of this paper. We show that, when solving shape from defocus, it is possible to avoid reconstructing the radiance without introducing restrictions on the point spread function $\mathbf{h}^s$ (e.g., by imposing shift-invariance).

Proposition. *Let $\hat{s}, \hat{r}$ be local extrema of the functional*

$$\phi(s, r) \doteq \|\mathbf{I} - H_s r\|^2 \tag{12}$$

and let $\tilde{s}$ be a local extremum of the function

$$\psi(s) \doteq \|H_s^\perp \mathbf{I}\|^2. \tag{13}$$

Furthermore, let $\tilde{r}$ be obtained from $\tilde{s}$ by $\tilde{r} \doteq \chi(\tilde{s})$, where χ is defined as

$$\chi(s) \doteq H_s^\dagger \mathbf{I}. \tag{14}$$

Then, $\hat{s}$ is also a local extremum of $\psi(s)$ and $\tilde{s}, \tilde{r}$ are also local extrema of $\phi(s, r)$.

3. Note that this assumption imposes constraints on the scene/optics combination. Alternatively, it can be thought of as a regularizing condition.

Proof. The proof extends the results of Golub and Pereyra [16]. For more details, see the Appendix. □

Remark 1. The significance of the proposition above is that (12) and (13) have the same minima in the 3D structure of the surface s, but, while (12) is an optimization problem in two unknowns, (13) is an optimization problem in a single unknown, s. Furthermore, if we constrain the surface s to belong to a finite-dimensional set, while the problem in (12) is still infinite-dimensional, the problem in (13) becomes finite-dimensional. Indeed, in the implementations we consider in the experimental section it is a one-dimensional space (depth). Note also that the proposition is nontrivial: In fact, (13) is obtained by multiplying on the left (7) by the (singular) matrix $H_s^\perp$. This can add spurious solutions to the problem, as we know by solving linear systems of equations.[4] The proposition shows that, in this specific case, this does not happen.

Notice that, if $H_{\hat{s}}$ is surjective[5] for a given $\hat{s}$, the orthogonal operator $H_{\hat{s}}^\perp$ is the null map (see (11)). In this case, (13) is trivially satisfied for any measured image $\mathbf{I}$ and, therefore, $\hat{s}$ is always a minimizer. Hence, a necessary condition to avoid this scenario is to impose that $H_{\hat{s}}$ maps functions in $L^2(\mathbb{R}^2)$ to a subspace of $\mathbb{R}^P$ of dimension less than P. In the next section, we will do so by truncating the singular value decomposition of either the operator H_s or the operator $H_s^\perp$.

As we have seen in the proposition, rather than solving the original problem in (12), we can solve the simpler problem in (13). Then, the minimization of (12) boils down to computing the orthogonal operators $H_s^\perp$. As we show in the next section, $H_s^\perp$ can be computed in different ways.

5 Computation of the Orthogonal Operators

When the complete characterization of the PSF of the camera is known, one can directly compute the orthogonal operators $H_s^\perp$ in closed form, at least for simple classes of PSFs (Section 5.1). More in general, one can express H_s via the *functional singular value decomposition* [2]. When the PSF is not known, one can compute $H_s^\perp$ directly from blurred images, as we explain in Subsection 5.2. The advantage of this second solution is its simplicity: To compute $H_s^\perp$, one only needs to collect a training set of controlled blurred images and then express the training set via the singular value decomposition.

5.1 Regularization via Functional Singular Value Decomposition

Assuming that we have a model of the PSF, we can express the operator H_s using its functional singular value decomposition. Let $\{\lambda_k\}$, $k = 1, \ldots, P$, be a sequence of nonnegative scalars sorted in decreasing order, $\{v_k\}$ an orthonormal set of vectors in $\mathbb{R}^P$, and $\{u_k\}$ an orthonormal set of functions in $L^2(\mathbb{R}^2)$. We now look for the particular choice of such sets that allows us to express H_s as

$$H_s = \sum_{k=1}^{P} \lambda_k u_k v_k, \tag{15}$$

where H_s maps $L^2(\mathbb{R}^2)$ on $\mathbb{R}^P$ as follows:

$$r \mapsto H_s r \doteq \sum_{k=1}^{P} \lambda_k \langle r, u_k \rangle v_k. \tag{16}$$

Using the same sequences of u_k and v_k, we can obtain an expression for the adjoint H_s^*; in particular, $H_s^*\mathbf{I}$ is defined via

$$H_s^*\mathbf{I} \doteq \sum_{k=1}^{P} \lambda_k \langle\langle \mathbf{I}, v_k \rangle\rangle u_k. \tag{17}$$

If there exists a suitable integer ρ (the rank of the operator) such that $\lambda_k > 0$ for $1 \leq k \leq \rho$ and $\lambda_k = 0$ for $\rho < k \leq P$, it is easy to verify by substitution that the pseudoinverse is given by[6]

$$H_s^\dagger = \sum_{k=1}^{\rho} \lambda_k^{-1} u_k v_k^t. \tag{18}$$

Then, the orthogonal projection operator is

$$H_s^\perp = \mathbf{1}_P - \sum_{k=1}^{\rho} v_k v_k^t, \tag{19}$$

where $\mathbf{1}_P$ is the $P \times P$ identity matrix. In order for the orthogonal projection operator to be nontrivial, we need to assume that $\rho < P$. This is equivalent to assuming that H_s maps to a finite-dimensional subspace of $L^2(\mathbb{R}^2)$, which imposes a lower bound on the dimensionality of the data to be acquired, i.e., the minimum number of blurred images and their size.

The sequences $\{\lambda_k\}$, $\{u_k\}$, and $\{v_k\}$ are found by solving the *normal equations*:

$$\begin{cases} H_s^* H_s u_k = \lambda_k^2 u_k \\ H_s H_s^* v_k = \lambda_k^2 v_k \end{cases} \quad k = 1 \ldots \rho \tag{20}$$

or, making the notation explicit,

$$\begin{cases} \sum_{\mathbf{y}} \mathbf{h}^s(\tilde{\mathbf{x}}, \mathbf{y}) \left(\int \mathbf{h}^s(\mathbf{x}, \mathbf{y}) u_k(\mathbf{x}) d\mathbf{x} \right) = \lambda_k^2 u_k(\tilde{\mathbf{x}}) \\ \int \mathbf{h}^s(\mathbf{x}, \tilde{\mathbf{y}}) \left(\sum_{\mathbf{y}} \mathbf{h}^s(\mathbf{x}, \mathbf{y}) v_k(\mathbf{y}) \right) d\mathbf{x} = \lambda_k^2 v_k(\tilde{\mathbf{y}}) \end{cases} \quad k = 1 \ldots \rho. \tag{21}$$

The second of the normal equations (21) can be written as

$$\mathcal{M} v_k = \lambda_k^2 v_k \quad k = 1 \ldots \rho, \tag{22}$$

where $\mathcal{M}$ is the P-dimensional symmetric matrix $\langle \mathbf{h}^s(\cdot, \tilde{\mathbf{y}}), \mathbf{h}^s(\cdot, \mathbf{y}) \rangle$. Since this is a (finite-dimensional) symmetric eigenvalue problem, there exists a unique decomposition of $\mathcal{M}$ of the form

4. For instance, the solution of $Ax = 0$ is $\{x \in Null(A)\}$, while the solution to $BAx = 0$ is $\{x \in Null(a)\} \cup \{x \mid Ax \in Null(B)\}$.

5. Recall that a function is surjective when its range is the whole codomain. In our case, $H_{\hat{s}}$ is surjective when for each image $I \in \mathbb{R}^P$ there exists a radiance r such that $H_{\hat{s}} r = I$.

6. The symbol $(\cdot)^t$ denotes matrix transposition.

$$\mathcal{M} = V\Lambda^2 V^t, \tag{23}$$

with $V^tV = \mathbf{1}_\rho$, $\Lambda^2 = \text{diag}\{\lambda_1^2 \dots \lambda_\rho^2\}$, and $V = [v_1, \dots, v_\rho]$.

We are now left with the first equation in (21) in order to retrieve $u_k(\mathbf{x})$. However, instead of solving that directly, we use the adjoint operator H_s^* to map the basis of $\mathbb{R}^P$ onto a basis of a ρ-dimensional subspace of $L^2(\mathbb{R}^2)$ via $H_s^* v_k = \lambda_k u_k$. Making the notation explicit, we have

$$u_k(\mathbf{x}) = \lambda_k^{-1} \sum_{\mathbf{y}} \mathbf{h}^s(\mathbf{x}, \mathbf{y}) v_k(\mathbf{y}) \quad k = 1 \dots \rho. \tag{24}$$

Remark 2 (Regularization). In the computation of $H_s^\perp$ (which we will see more in detail in Section 7), the sum is effectively truncated at $k = \rho < P$, where the dimension P depends upon the amount of data acquired. As a consequence of the properties of the SVD, the solution obtained enjoys a number of regularity properties. Note that the solution is *not* the one that we would have obtained by first writing r using a truncated orthonormal expansion in $L^2(\mathbb{R}^2)$, then expanding the kernel $\mathbf{h}^s$ in (7) in series, and then applying the finite-dimensional version of the orthogonal projection theorem.

5.2 Learning Null Spaces from Defocus

When the model of the PSF $\mathbf{h}^s$ is not known, we cannot use the method described in the previous section to compute the orthogonal operators. Here, we show that complete knowledge of the point spread function is indeed not necessary. To compute the orthogonal operators, one only needs the finite-dimensional range of the PSF, which can also be obtained directly from a collection of blurred images.

Recall that a defocused image $\mathbf{I}$ can be written as $H_{\bar{s}} r$ for some surface $\bar{s}$ and a radiance r (see (7)). By definition, if we multiply the orthogonal operator $H_{\bar{s}}^\perp$ by $\mathbf{I}$ on the right, we obtain

$$H_{\bar{s}}^\perp \mathbf{I} = H_{\bar{s}}^\perp H_{\bar{s}} r = 0. \tag{25}$$

Notice that this equation is satisfied for any radiance r. Hence, if we collect a set of T images[7] $\{\mathbf{I}\}_{i=1\dots T}$ by letting the radiance vary $r = \{r_1, \dots, r_T\}$, we obtain

$$H_{\bar{s}}^\perp [\mathbf{I}_1 \ \dots \ \mathbf{I}_T] = H_{\bar{s}}^\perp H_{\bar{s}} [r_1 \ \dots \ r_T] = [0 \ \dots \ 0] \doteq \mathbf{0} \tag{26}$$

as long as the surface $\bar{s}$ of the scene remains the same. We can therefore find $H_{\bar{s}}^\perp$ by simply solving the following system of linear equations:

$$H_{\bar{s}}^\perp [\mathbf{I}_1 \ \mathbf{I}_2 \ \dots \ \mathbf{I}_T] = \mathbf{0}. \tag{27}$$

Notice, however, that $H_{\bar{s}}^\perp$ is not a generic matrix but, rather, has some important structure that must be exploited in solving the system of equations above. In particular, $H_s^\perp$ is a *symmetric* matrix (i.e., $H_s^\perp = (H_s^\perp)^t$) which is also *idempotent* (i.e., $H_s^\perp = (H_s^\perp)^2$). 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; as for the second property, we have that the columns of A must be orthonormal and, thus, $H_s^\perp$ can be written uniquely as:

$$H_s^\perp = AA^t, \tag{28}$$

where $A \in V_{n,m}$ and $V_{n,m}$ is the space of $n \times m$ rectangular matrices with orthonormal columns.

Let $\mathcal{I} = [\mathbf{I}_1 \ \mathbf{I}_2 \ \dots \ \mathbf{I}_T] \in \mathbb{R}^{P \times T}$, then the solution of (27) can be obtained via the singular value decomposition of $\mathcal{I}$

$$\mathcal{I} = UBW^t, \tag{29}$$

where $U \in V_{P,P}$, $W \in V_{T,T}$, and $B \in \mathbb{R}^{P \times T}$ is a diagonal matrix whose values are nonnegative, by defining

$$H_{\bar{s}}^\perp = U_2 U_2^t, \tag{30}$$

where $U = [U_1 \ U_2]$ and U_2 are the orthonormal vectors corresponding to the null singular values of B. In other words, given a surface $\bar{s}$ of the scene, we can *learn* the corresponding orthogonal operator $H_{\bar{s}}^\perp$ by applying the SVD to a matrix whose column vectors are defocused images.

In the presence of deviations from the ideal model (1), this yields the least-squares estimate of $H_{\bar{s}}^\perp$, which can be thought of as a *learning* procedure, summarized in Table 2.

Remark 3 (Excitation). The computation of the orthogonal operator $H_s^\perp$ depends strongly on the training sequence of defocused images that we use (see, (27)). In order to be able to learn a nontrivial orthogonal operator, we expect the training set to span a subspace of dimension less than P. However, there are two factors that determine the rank of the training set: One is the intrinsic structure of the PSF, which is what we want to characterize, the other is the rank of the chosen set of radiances $r_1, \dots, r_T$. For example, if we employ radiances (or "scene textures") that are linearly dependent and span a subspace of dimension $\rho < P$, the corresponding defocused images are also linearly dependent (due to the linear relation between radiances and defocused images) and span a subspace of dimension less than or equal to ρ (some radiances may be mapped to the same blurred image). If the rank due to the intrinsic structure of the PSF is larger than ρ, by using these images we do not reconstruct the correct $H_s^\perp$.

To determine the correct structure of $H_s^\perp$, we need to guarantee that only the first factor is lowering the rank. In order to do that, we need to choose a set or radiances that is large enough, i.e., $T \geq P$, spans a subspace of dimension larger than or equal to P, and does not belong to the null space of H_s. We call such radiances *sufficiently exciting* for the training sequence.

6 Shape Estimation Algorithm

So far, we have shown that the original problem in (12) can be reduced to the problem in (13), which involves the computation of orthogonal projectors $H_s^\perp$ (Section 4.2). Then, we have shown two methods to compute the projectors (Section 5.1 and Section 5.2). Now that we have derived all the necessary components, we will introduce our algorithm to reconstruct the geometry of a scene from defocused images.

Given a collection of K defocused images $\mathbf{I} = [I_1; I_2; \dots; I_K]$, we want to estimate the 3D structure of the scene by minimizing the following cost functional:

7. Recall that each image $\mathbf{I}_i \in \mathbb{R}^P$, $P = M \times N \times K$, and $\mathbf{I}_i$ is a column vector collecting K defocused images $[I_1, \dots, I_K]$, $I_j \in \mathbb{R}^{M \times N}$ $\forall j = 1, \dots, K$, captured for K different optics settings. As we will discuss in Section 7.2, $[\mathbf{I}_1, \dots, \mathbf{I}_T]$ will be small image patches carved out of one single image.

TABLE 2
Summary of the Procedure to "Learn" the Orthogonal Operator $H_s^\perp$ from Defocused Images

Algorithm (learning approach)

1. Choose a surface s, either a physical surface or a synthetic one.
2. Generate (in the case of synthetic data) or collect (in the case of real data) a number T of training images $[\mathbf{I}_1\ \mathbf{I}_2\ \ldots\ \mathbf{I}_T] \in \mathbb{R}^{P\times T}$ from a scene with surface s and corresponding radiances $r_1, r_2, \ldots, r_T$. To make the terms explicit, notice that for each radiance r_j, $j = 1, \ldots, T$ we have K defocused images $I_{i,j} \in \mathbb{R}^{M\times N}$, $i = 1, \ldots, K$ taken with K focus settings. The images $I_{i,j}$ are rearranged as a column vector and then stacked on top of each other, so as to form a column vector $\mathbf{I}_j \in \mathbb{R}^P$, with $P = MNK$. Finally, $[\mathbf{I}_1\ \mathbf{I}_2\ \ldots\ \mathbf{I}_T]$ is a matrix in $\mathbb{R}^{P\times T}$;
3. Collect all rearranged column vectors into the matrix $\mathcal{I} = [\mathbf{I}_1\ \mathbf{I}_2\ \ldots\ \mathbf{I}_T]$ of dimensions $P \times T$. Apply the SVD to $\mathcal{I}$ such that $\mathcal{I} = UBW^t$;
4. Determine the rank q of $\mathcal{I}$ (for example by imposing a threshold on the singular values);
5. 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:

$$H_s^\perp = U_2 U_2^t. \tag{31}$$

$$\tilde{s} = \arg\min_s \|H_s^\perp \mathbf{I}\|^2. \tag{32}$$

If one is also interested in reconstructing the radiance (i.e., deblurring the defocused images $\mathbf{I}$), the solution $\tilde{s}$ of (32) can be used to compute:

$$\tilde{r} = H_{\tilde{s}}^\dagger \mathbf{I}. \tag{33}$$

In principle, the above minimization can be carried out by using the tools of calculus of variations. One can estimate $\tilde{s}$ by implementing the following gradient descent flow:

$$\frac{\partial s}{\partial \tau} = -\nabla_s E, \tag{33}$$

where τ is the iteration index, $E(s) = \|H_s^\perp \mathbf{I}\|^2$ is the cost functional, and $\nabla_s E$ is the functional gradient of E with respect to the surface s.

In this paper, however, we are interested only in the simplest implementation. Hence, rather than implementing computationally expensive gradient flows, we solve (32) locally around patches of the defocused images and, for each patch, we assume that the corresponding surface belongs to a finite dimensional set of admissible surfaces $\mathcal{S}$. In particular, in the simplest case, we assume that the scene that generates a local patch can be approximated with a small planar patch parallel to the image plane. In other words, we solve

$$\tilde{s}(\mathbf{x}) = \arg\min_{s\in\mathcal{S}} \|H_s^\perp \mathbf{I}(\mathbf{y})\|^2 \quad \forall\, \mathbf{y} \in \mathcal{W}, \tag{35}$$

where $\mathcal{W}$ is a patch centered in $\mathbf{x}$ and $\mathcal{S}$ is a discrete set of depths. This simplification allows us to precompute the orthogonal projectors $H_s^\perp$, one for each depth level, and to minimize the cost functional by a simple one-dimensional exhaustive search.

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, thus enabling highly parallel implementations. It would be possible, for example, to have CCD arrays where each pixel neighborhood maps to a computing unit returning the depth relative to it.

Also, one could compensate for the coarseness of choosing a finite set of depth levels by interpolating the computed cost function. The search process can be accelerated by using well-known descent methods (i.e., gradient descent, Newton-Raphson, tangents, etc.) or by using a dichotomic search.

Notice that, despite its simplicity, the proposed algorithm is very general and the choice of working locally at patches is not crucial to the feasibility of the algorithm.

7 Experiments

In this section, we present a set of experiments both on synthetic data (unaffected by noise) and on real images (affected by sensor noise). We have verified experimentally that the performance of the proposed algorithms in the case of a Gaussian or a "Pillbox" point spread function is very similar. Hence, for simplicity, when computing the orthogonal operators, we consider only the Gaussian family. This is not a crucial choice, however, since the algorithm can be carried out for any other family of PSFs at the cost of an increased computational burden.

For a general description and characterization of the Gaussian and Pillbox families, we refer the reader to the book by Chaudhuri and Rajagopalan [6]. Here, we only want to address the specific issue of modeling the *pixel discretization* effect of the CCD. We approximate this effect by integrating the energy hitting the surface of a pixel with a Gaussian kernel, i.e.,

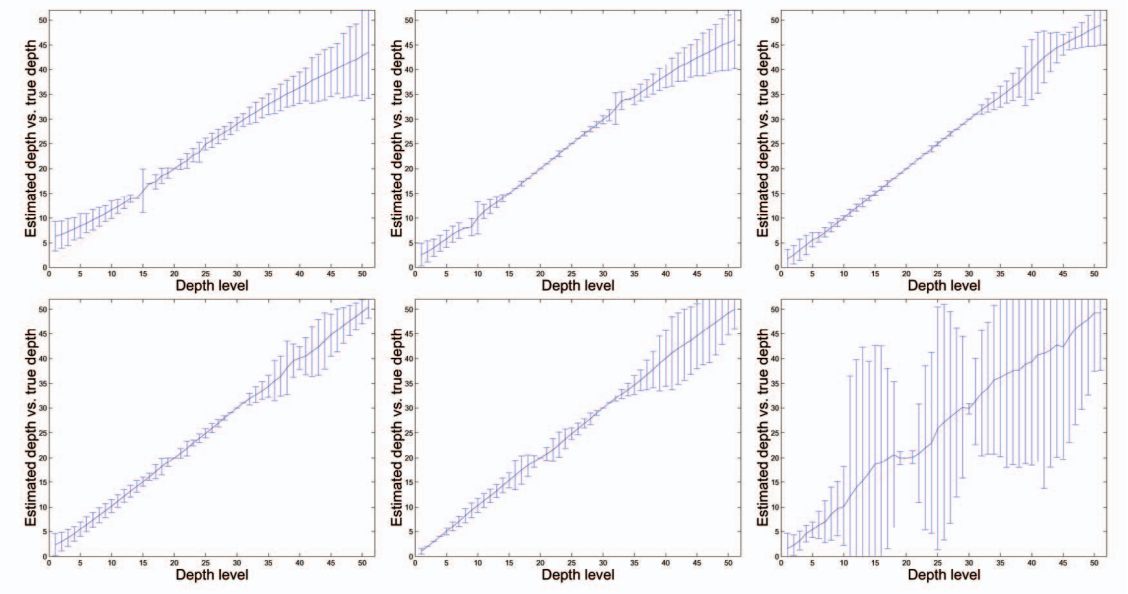


Fig. 2. Performance test for different ranks of the orthogonal operators computed in closed form in the case of a Gaussian PSF. From left to right and from top to bottom, the ranks are 40, 55, 65, 70, 75, and 95. Both mean and standard deviation (in the graphs, we show three times the computed standard deviation) of the estimated depth (solid line) are plotted over the ideal characteristic curve (dotted line) for 50 experiments.

$$\tilde{I}(\mathbf{y}) = \int \frac{1}{2\pi\sigma^2} e^{-\frac{\|\mathbf{y}-\tilde{\mathbf{y}}\|^2}{2\sigma^2}} \left(\int h^s(\mathbf{x}, \tilde{\mathbf{y}}) r(\mathbf{x}) d\mathbf{x} \right) d\tilde{\mathbf{y}}, \qquad (36)$$

where $\sigma = 1/4$ pixels. By taking the discretization into account, we obtain a point spread function that is never a *Dirac delta*, but a regular smooth function, even when the scene is brought into focus.

In all the experiments we compute orthogonal operators locally on windows of 7×7 or 9×9 pixels. Since we always collect two defocused images only, while changing the focus settings, the orthogonal operators are matrices of size $2 \cdot 7^2 \times 2 \cdot 7^2 = 98 \times 98$ pixels or $2 \cdot 9^2 \times 2 \cdot 9^2 = 162 \times 162$ pixels. As we will see, a qualitative comparison of the reconstruction of shape from real images does not reveal a noticeable difference among the orthogonal operators computed via the method in Section 7.1 or the method in Section 5.2. In other words, we can "learn" the orthogonal projector $H_s^\perp$ corresponding to "virtual" or synthetic cameras and then use it in Algorithm 2 to infer the shape of scenes captured with real cameras. The results are comparable to those obtained by inferring the orthogonal projector through a careful calibration procedure, as we have described in Section 7.1. This speaks of the remarkable flexibility of such a simple algorithm for shape from defocus.

For a better evaluation of the performance of this algorithm, we are in the process of making its implementation in Matlab code available online on our Web page http://www.cs.ucla.edu/~favaro.

7.1 Experiments with Known PSF

Following the procedure presented in Section 5.1, we compute a set of orthogonal operators $H_s^\perp$ in the case of a Gaussian kernel for patches of 7×7 pixels. We simulate a scene made of 51 equifocal planes placed equidistantly in the range between $520mm$ and $850mm$ in front of a camera with a $35mm$ lens and F-number 4. We capture two defocused images. One is obtained by bringing the plane at $520mm$ into focus. The other is obtained by bringing the plane at $850mm$ into focus. Each of the 51 equifocal planes corresponds to one orthogonal operator. We would like to stress that the orthogonal operators do not need to be computed for equifocal planes, but can be computed for any other set of surfaces (Section 6).

Once the orthogonal projectors are computed, we apply them on both synthetically generated and real images. In the synthetic case, we simulate 50 defocused images for each of the 51 equifocal planes used to generate the orthogonal operators. Each of the 50 simulations is obtained by employing a radiance of the scene that is generated randomly. At each depth level, and for each of these experiments, we estimate a depth level. Fig. 2 shows the depth estimation performance when we use the computed orthogonal operators with ranks 40, 55, 65, 70, 75, and 95. Both mean and standard deviation (in the graphs we show three times the computed standard deviation) of the estimated depth (solid line) are plotted over the ideal characteristic curve (the diagonal dotted line). Clearly, when the chosen rank does not correspond to the true rank of the operators, the performance rapidly degenerates. In this case, the correct rank is 70. For this choice, the average estimation error[8] is $31mm$. We also test the performance of this algorithm on the real images shown in Fig. 3 and Fig. 6 by working on patches of 9×9 pixels. In Fig. 3, the scene is composed of objects placed between $640mm$ and $750mm$ in front of the lens. From the bottom to the top, we have: a box, a slanted plane, and two cylinders. We capture images using an 8-bit camera containing two independently moving CCDs (kindly provided to us by Professor S.K. Nayar of Columbia University). The lens is a $35mm$ Nikon NIKKOR with F-number 4. In Fig. 4, we show the estimated depth map as a gray-level image, where light intensities correspond to points that are close to the cameras and dark intensities to points that are far from the cameras. In Fig. 5,

8. We compute the average estimation error as the mean of the absolute value of the difference between the estimated depth of the scene and the ground truth.

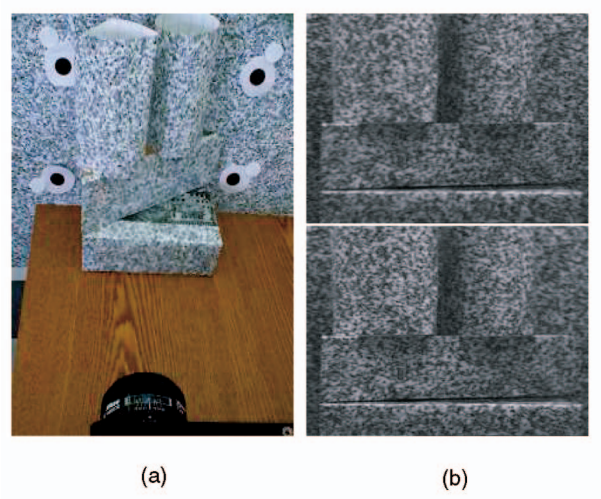


Fig. 3. (a) Setup of the real scene. From the bottom to the top we have: a box (parallel to the image plane), a slanted plane, and two cylinders. (b) Two images captured with different focal settings.

we show the estimation results as texture mapped surfaces, after smoothing.

For comparison, we can test the same data (Fig. 6) on an algorithm using the rational filter approach described in [36]. In Fig. 7, we show the estimated depth map in the case of the simple least-squares algorithm (a) with the depth map estimated with the more elaborate algorithm of [36] (b). The quality of the estimates is very similar.

7.2 Experiments with Unknown PSF

In this section, we evaluate the performance of the proposed depth estimation algorithm when the orthogonal operators are computed via the procedure described in Section 5.2. As in the previous section, we perform experiments on both real and synthetic data. We use operators computed from synthetic data on both real and synthetic imagery and operators computed from real data on real imagery obtained from the same camera. We divide the range between $520mm$ and $850mm$ in front of the camera into 51 intervals and compute 51 orthogonal projectors each corresponding to a plane parallel to the image plane placed at one of the intervals. Each operator is computed by capturing only two defocused images of 640×480 pixels.

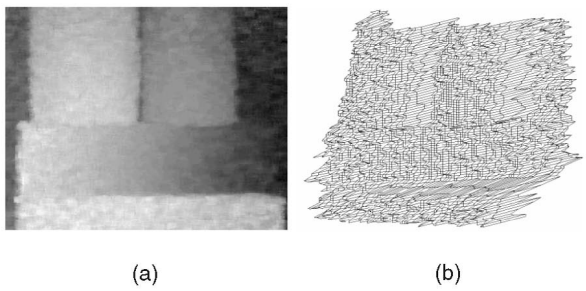


Fig. 4. Estimated depth of the real images in Fig. 3 when using the Gaussian PSF and the closed form solution for the orthogonal operators. (a) The estimated depth in gray-level intensities, where light corresponds to small depths, and dark to large depths. (b) Visualization of the estimated depth as a mesh.

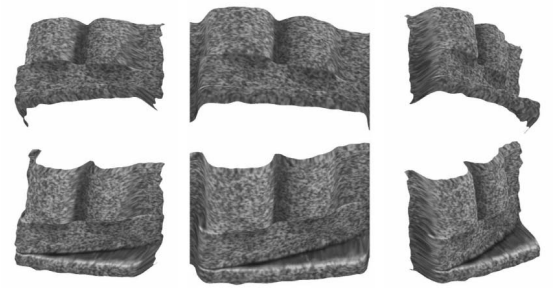


Fig. 5. Novel views of the estimated shape (after smoothing) and with texture mapping. Shape has been reconstructed with orthogonal operators computed in closed form in the case of a Gaussian PSF.

We collect 200 patches of 7×7 pixels or 9×9 pixels from these images and use them to estimate the orthogonal operator. We test that the radiances collected from each patch are *sufficiently exciting* (see Remark 3) by making sure they are not linearly dependent. The whole procedure can be easily performed both on synthetic and real data.

Experiments on Synthetic Data. We simulate the same scene and camera settings as in the previous section. Fig. 8 shows the depth estimation performance when we use the orthogonal operators learned from synthetic data, with ranks 40, 55, 65, 70, 75, and 95 on synthetic images. Both mean and standard deviation (in the graphs, we show three times the computed standard deviation) of the estimated depth (solid line) are plotted over the ideal characteristic curve (the diagonal dotted line). Clearly, when the chosen rank does not correspond to the correct rank of the operators, the performance degrades rapidly. In this case, the correct rank is again 70. For this choice, the average estimation error is $27mm$.

Experiments on Real Data. We perform three experiments with real data. In two experiments, we use operators learned from synthetic images, by assuming the PSF to be Gaussian. The operators are computed as described in the previous section, applied to the pair of real images shown in Fig. 3, and return the depth map estimate in Fig. 9. As one can observe, the estimated depth map is very similar to the estimate obtained when the operators are computed in closed form, as prescribed in Section 5.1 (compare to Fig. 4). We also apply these operators (learned from synthetic images) to the pair of real images shown in Fig. 6, so as to compare the estimated depth map to the one obtained by a more elaborate algorithm [36]. The estimated depth map and the one computed by [36] are shown in Fig. 10. In the third experiment, we learn the orthogonal projection operators from a collection of real images and then apply these operators to novel real images obtained from the same camera. We test these operators on the images in Fig. 3 and obtain the depth map shown in Fig. 11. As one can see, the estimated depth map is very similar to the estimates obtained in the previous experiments.

8 Conclusions

We have presented a novel and optimal (in the L^2 sense) technique to infer 3D shape from defocus. Rather than solving

Fig. 6. Two real images captured with different focus settings. For more details on the scene and camera settings, please refer to [36].

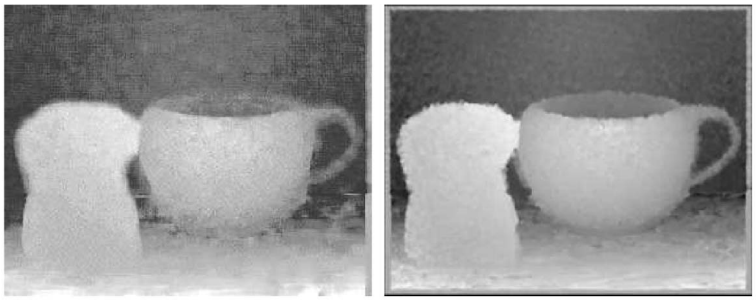


Fig. 7. Estimated depth maps from the two input images in Fig. 6. Both depth maps are not postprocessed. On the left, we show the depth map estimated with the simple algorithm described in this manuscript with known PSF. On the right, we show the depth map estimated with the algorithm described in [36].

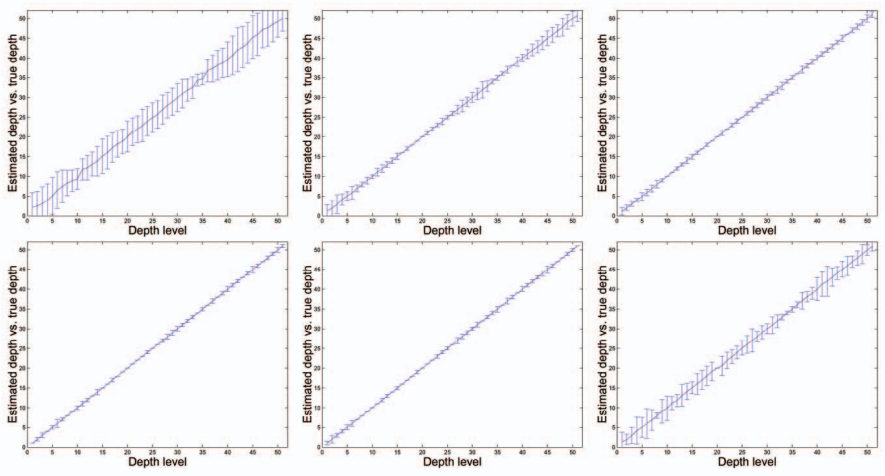


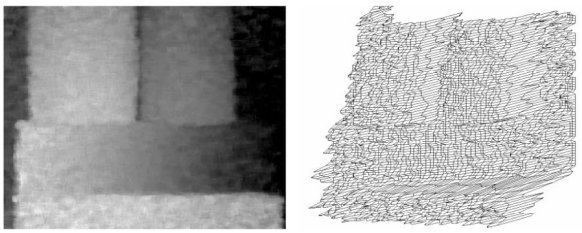
Fig. 8. Performance test for different ranks of the orthogonal operators learned from synthetic data in the case of a Gaussian PSF. From left to right and top to bottom, the ranks are 40, 55, 65, 70, 75, and 95. Both mean and standard deviation (in the graphs, we show three times the computed standard deviation) of the estimated depth (solid line) are plotted over the ideal characteristic curve (dotted line) for 50 experiments.

the problem of matching data to measurements, our approach consists of solving an equivalent but simpler problem where a set of interest operators are applied to the given blurred images. The solution is then obtained by searching for the operator whose output has the lowest energy.

We propose two approaches to compute the interest operators. In one approach, we describe a regularized closed form solution via the functional SVD of the imaging kernels. In the other, we avoid modeling the optics and construct the operators by *learning* the left null space of blurred images through singular value decomposition. This second method is so flexible that we can learn a characterization of the imaging kernel even from synthetic images and then use it on images obtained from a real camera.

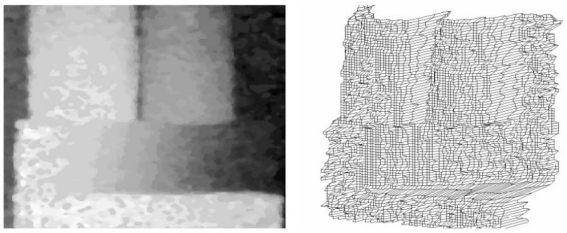
In our approach, the size of the measurement array naturally imposes regularity in the solution, which is obtained in infinite-dimensional space using a functional singular value decomposition. We use the structure of maps between (finite and infinite-dimensional) Hilbert spaces, which makes the analysis simple and intuitive.

The algorithms are robust to noise and can be used effectively to estimate depth, as we showed in the experiments. Furthermore, the proposed algorithms can be implemented in real time and are suitable for highly parallel computational schemes.



(a) (b)

Fig. 9. Estimated depth of the scene shown in Fig. 3 when using the Gaussian PSF and the learning method for the orthogonal operators on synthetic images. (a) The estimated depth in gray-level intensities, where light corresponds to small depths and dark to large depths. (b) Visualization of the estimated depth as a mesh.



(a) (b)

Fig. 11. Estimated depth for the scene in Fig. 3 when using the orthogonal operators learned from real images. (a) The estimated depth in gray-level intensities, where light corresponds to small depths and dark to large depths. (b) Visualization of the estimated depth as a mesh.

APPENDIX

The proof of the proposition in Section 4.2 closely relates to the results of Golub and Pereyra [16]. Before proceeding with the proof, we need to introduce some additional notation. The extrema of the functional ϕ and ψ are defined via their Fréchet functional derivatives. They result in the following coupled equations:

$$\begin{cases} \nabla_s \phi(\hat{s}, \hat{r}) = 0 \\ \nabla_r \phi(\hat{s}, \hat{r}) = 0 \end{cases} \tag{37}$$

and

$$\begin{cases} \nabla_s \psi(\tilde{s}) = 0 \\ r \doteq \chi(\tilde{s}), \end{cases} \tag{38}$$

where $\nabla_s \phi$ and $\nabla_r \phi$ stand for the gradients of ϕ with respect to s and r, respectively, and $\nabla_s \psi$ stands for the gradient of ψ with respect to s [17]. For simplicity, where possible, we indicate with $\dot{A}$ the derivative of A with respect to s instead of using the equivalent but bulkier notation $\nabla_s A$.

For ease of reading, we simplify the proof of the proposition by gathering some of the results in the following lemma:

Lemma 1. *Let $P_{H_s} \doteq H_s H_s^\dagger$ be the projection operator onto the range of H_s and recall that $H_s^\dagger$ verifies $H_s H_s^\dagger H_s = H_s$ and $(H_s H_s^\dagger)^* = H_s H_s^\dagger$, then*

$$\dot{P}_{H_s} = H_s^\perp \dot{H}_s H_s^\dagger + \left(H_s^\perp \dot{H}_s H_s^\dagger\right)^*. \tag{39}$$

Proof. Since $P_{H_s} H_s = H_s$, then:

$$\nabla_s (P_{H_s} H_s) = \dot{P}_{H_s} H_s + P_{H_s} \dot{H}_s = \dot{H}_s \tag{40}$$

and

$$\dot{P}_{H_s} H_s = \dot{H}_s - P_{H_s} \dot{H}_s = H_s^\perp \dot{H}_s. \tag{41}$$

Also,

$$\dot{P}_{H_s} P_{H_s} = \dot{P}_{H_s} H_s H_s^\dagger = H_s^\perp \dot{H}_s H_s^\dagger. \tag{42}$$

Since $\left(\dot{P}_{H_s} P_{H_s}\right)^* = P_{H_s} \dot{P}_{H_s}$, then

$$\begin{aligned} \dot{P}_{H_s} &= \nabla_s (P_{H_s} P_{H_s}) = \dot{P}_{H_s} P_{H_s} + P_{H_s} \dot{P}_{H_s} \\ &= H_s^\perp \dot{H}_s H_s^\dagger + \left(H_s^\perp \dot{H}_s H_s^\dagger\right)^*, \end{aligned} \tag{43}$$

which completes the proof. □

We now use the results of Lemma 1, together with the fact that $\dot{H}_s^\perp = -\dot{P}_{H_s}$, to prove the proposition:

Proof of the Proposition. We have that

$$\frac{1}{2} \nabla_s \phi(\hat{s}, \hat{r}) = (H_{\hat{s}} \hat{r})^T \dot{H}_{\hat{s}} \hat{r} - I^T \dot{H}_{\hat{s}} \hat{r} = 0, \tag{44}$$

while

$$\frac{1}{2} \nabla_r \phi(\hat{s}, \hat{r}) = H_{\hat{s}}^* H_{\hat{s}} \hat{r} - H_{\hat{s}}^* I = 0 \tag{45}$$

leads to

$$H_{\hat{s}}^* H_{\hat{s}} \hat{r} = H_{\hat{s}}^* I. \tag{46}$$

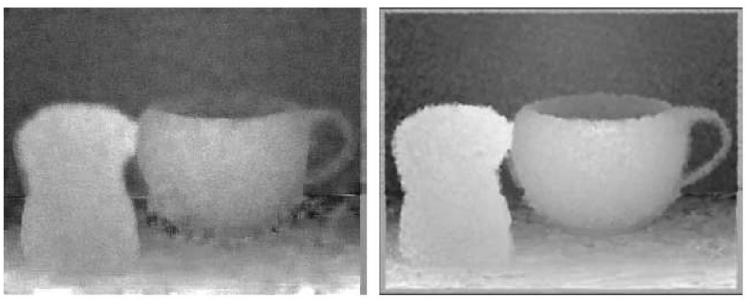


Fig. 10. Estimated depth maps from the two input images in Fig. 6. Both depth maps are not postprocessed. On the left, we show the depth map estimated with the simple least-squares algorithm with unknown PSF. On the right, we show the depth map estimated with the algorithm described in [36].

Now, the last equation is what defines the pseudoinverse $H_{\hat{s}}^{\dagger}$ (see (10)) and, therefore, it is satisfied, by construction, when

$$\hat{r} = H_{\hat{s}}^{\dagger} I = \chi(\hat{s}). \tag{47}$$

This shows that, if $\hat{s}$ is a stationary point of ϕ, its corresponding $\hat{r}$ must be of the form $\chi(\hat{s})$. The computation of (38) returns

$$\frac{1}{2}\nabla_s \psi(\tilde{s}) = I^T H_{\tilde{s}}^{\perp} \dot{H}_{\tilde{s}}^{\perp} I = 0. \tag{48}$$

$\Longleftarrow$) Let us now assume that $\nabla_s \psi(\tilde{s}) = 0$ and let $\tilde{r} = \chi(\tilde{s})$. We want to show that $\nabla_s \phi(\tilde{r}, \tilde{s}) = 0$, that is, (44) is satisfied with $\hat{s} = \tilde{s}$ (that (46) is satisfied follows directly from our choice of $\tilde{r}$ from (47)). To this end, notice that

$$\begin{aligned} H_s^{\perp}\left(H_s^{\dagger}\right)^* &= \left(H_s^{\dagger}\right)^* - H_s H_s^{\dagger}\left(H_s^{\dagger}\right)^* \\ &= \left(H_s^{\dagger}\right)^* - \left(H_s H_s^{\dagger}\right)^*\left(H_s^{\dagger}\right)^* \\ &= 0 \end{aligned} \tag{49}$$

and, therefore, substituting the expression of $\dot{H}_s^{\perp}$ (obtained in Lemma 1) and the expression for $\tilde{r} = \chi(\tilde{s})$ in (48), we obtain

$$\begin{aligned} 0 = \frac{1}{2}\nabla_s \psi(\tilde{s}) &= I^T H_{\tilde{s}}^{\perp} \dot{H}_{\tilde{s}}^{\perp} I = -I^T H_{\tilde{s}}^{\perp} \dot{H}_{\tilde{s}} H_{\tilde{s}}^{\dagger} I \\ &= (H_{\tilde{s}} H_{\tilde{s}}^{\dagger} H_{\tilde{s}} \tilde{r})^T \dot{H}_{\tilde{s}} \tilde{r} - I^T \dot{H}_{\tilde{s}} \tilde{r} \\ &= \frac{1}{2}\nabla_s \phi(\tilde{r}, \tilde{s}). \end{aligned} \tag{50}$$

$\Longrightarrow$) Now, let (44) and (46) hold for $\hat{s}, \hat{r}$. All we need to show is that $\nabla_s \psi(\hat{s}) = 0$. Since $\hat{r}$ satisfies (47) because of (46), we can read (50) backward and have

$$\begin{aligned} 0 = \frac{1}{2}\nabla_s \phi(\hat{r}, \hat{s}) &= (H_{\hat{s}}\hat{r})^T \dot{H}_{\hat{s}} \hat{r} - I^T \dot{H}_{\hat{s}} \hat{r} \\ &= \left(H_{\hat{s}} H_{\hat{s}}^{\dagger} I\right)^T \dot{H}_{\hat{s}} H_{\hat{s}}^{\dagger} I - I^T \dot{H}_{\hat{s}} H_{\hat{s}}^{\dagger} I \\ &= -I^T H_{\hat{s}}^{\perp} \dot{H}_{\hat{s}} H_{\hat{s}}^{\dagger} I \\ &= I^T H_{\hat{s}}^{\perp} \dot{H}_{\hat{s}}^{\perp} I \\ &= \frac{1}{2}\nabla_s \psi(\hat{s}). \end{aligned} \tag{51}$$

which concludes the proof. □

ACKNOWLEDGMENTS

The authors wish to thank Professor Shree Nayar and the anonymous reviewers for numerous constructive comments and suggestions. This research is sponsored by US Air Force Office of Scientific Research F49620-03-1-0095/E-16-V91-G2 and US Office of Naval Research N00014-03-1-0850:P0001/N00014-02-1-0720.

REFERENCES

[1] N. Asada, H. Fujiwara, and T. Matsuyama, "Edge and Depth from Focus," *Int'l J. Computer Vision*, vol. 26, no. 2, pp. 153-163, 1998.
[2] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. Inst. of Physics Publications, 1998.
[3] S.S. Bhasin and S. Chaudhuri, "Depth from Defocus in Presence of Partial Self Occlusion," *Proc. Int'l Conf. Computer Vision*, vol. 1, no. 2, pp. 488-93, 2001.
[4] A. Blake and A. Zisserman, *Visual Reconstruction*. MIT Press, 1987.
[5] M. Born and E. Wolf, *Principles of Optics*. Pergamon Press, 1980.
[6] S. Chaudhuri and A. Rajagopalan, *Depth from Defocus: A Real Aperture Imaging Approach*. Springer Verlag, 1999.
[7] J. Ens and P. Lawrence, "An Investigation of Methods for Determining Depth from Focus," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 97-108, 1993.
[8] O. Faugeras, *Three Dimensional Vision, a Geometric Viewpoint*. MIT Press, 1993.
[9] P. Favaro, A. Mennucci, and S. Soatto, "Observing Shape from Defocused Images," *Int'l J. Computer Vision*, vol. 52, no. 1, pp. 25-43, Apr. 2003.
[10] P. Favaro and S. Soatto, "Learning Shape from Defocus," *Proc. European Conf. Computer Vision*, vol. 2, pp. 735-745, 2002.
[11] P. Favaro and S. Soatto, "Seeing beyond Occlusions (and Other Marvels of a Finite Lens Aperture)," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '03)*, pp. II: 579-586, 2003.
[12] P. Favaro and S. Soatto, "Shape and Radiance Estimation from the Information Divergence of Blurred Images," *Proc. European Conf. Computer Vision*, pp. 755-768, June 2000.
[13] D.A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall, 2002.
[14] B. Girod and S. Scherock, "Depth from Focus of Structured Light," *Proc. SPIE*, pp. 209-15, 1989.
[15] M. Gokstorp, "Computing Depth from Out-of-Focus Blur Using a Local Frequency Representation," *Proc. Int'l Conf. Pattern Recognition*, pp. A:153-158, 1994.
[16] G. Golub and V. Pereyra, "The Differentiation of Pseudo-Inverses and Nonlinear Least-Squares Problems Whose Variables Separate," *SIAM J. Numerical Analysis*, vol. 10, no. 2, pp. 413-532, 1973.
[17] D. Luenberger, *Optimization by Vector Space Methods*. Wiley, 1968.
[18] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3D Vision, from Images to Models*. Springer Verlag, 2003.
[19] J. Marshall, C. Burbeck, and D. Ariely, "Occlusion Edge Blur: A Cue to Relative Visual Depth," *Intl J. Optical Soc. Am. A*, vol. 13, pp. 681-688, 1996.
[20] H.N. Nair and C.V. Stewart, "Robust Focus Ranging," *Computer Vision and Pattern Recognition*, pp. 309-314, 1992.
[21] S.K. Nayar and Y. Nakagawa, "Shape from Focus: An Effective Approach for Rough Surfaces," *IEEE Int'l Conf. Robotics and Automation*, pp. 218-225, 1990.
[22] S.K. Nayar, M. Watanabe, and M. Noguchi, "Real-Time Focus Range Sensor," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1186-1198, Dec. 1996.
[23] S.K. Nayar and S.G. Narasimhan, "Vision in Bad Weather," *Proc. Int'l Conf. Computer Vision (ICCV '99)*, pp. 820-827, 1999.
[24] M. Noguchi and S.K. Nayar, "Microscopic Shape from Focus Using Active Illumination," *Proc. Int'l Conf. Pattern Recognition*, pp. 147-152, 1994.
[25] A. Pentland, "A New Sense for Depth of Field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, pp. 523-531, 1987.
[26] A. Pentland, T. Darrell, M. Turk, and W. Huang, "A Simple, Real-Time Range Camera," *Computer Vision and Pattern Recognition*, pp. 256-261, 1989.
[27] A. Pentland, S. Scherock, T. Darrell, and B. Girod, "Simple Range Cameras Based on Focal Error," *J. Optical Soc. Am. A*, vol. 11, no. 11, pp. 2925-2934, Nov. 1994.
[28] A.N. Rajagopalan and S. Chaudhuri, "A Block Shift-Variant Blur Model for Recovering Depth from Defocused Images," *Proc. Int'l Conf. Image Processing*, pp. 636-39, 1995.
[29] A.N. Rajagopalan and S. Chaudhuri, "Optimal Selection of Camera Parameters for Recovery of Depth from Defocused Images," *Computer Vision and Pattern Recognition*, pp. 219-224, 1997.
[30] A.N. Rajagopalan and S. Chaudhuri, "Optimal Recovery of Depth from Defocused Images Using an MRF Model," *Proc. Int'l Conf. Computer Vision*, pp. 1047-1052, 1998.
[31] Y. Schechner and N. Kiryati, "The Optimal Axial Interval in Estimating Depth from Defocus," *Proc. Int'l Conf. Computer Vision*, pp. 843-848, 1993.
[32] G. Schneider, B. Heit, J. Honig, and J. Bremont, "Monocular Depth Perception by Evaluation of the Blur in Defocused Images," *Proc. Int'l Conf. Image Processing*, vol. 2, pp. 116-9, 1994.

[33] S. Soatto and P. Favaro, "A Geometric Approach to Blind Deconvolution with Application to Shape from Defocus," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 10-17, June 2000.
[34] M. Subbarao and N. Gurumoorthy, "Depth Recovery from Blurred Edges," *Computer Vision and Pattern Recognition,* pp. 498-503, 1988.
[35] M. Subbarao and G. Surya, "Depth from Defocus: A Spatial Domain Approach," *Int'l J. Computer Vision,* vol. 13, pp. 271-294, 1994.
[36] M. Watanabe and S. Nayar, "Rational Filters for Passive Depth from Defocus," *Int'l J. Computer Vision,* vol. 27, no. 3, pp. 203-225, 1998.
[37] M. Watanabe and S.K. Nayar, "Minimal Operator Set for Passive Depth from Defocus," *Computer Vision and Pattern Recognition,* pp. 431-438, 1996.
[38] Y. Xiong and S. Shafer, "Depth from Focusing and Defocusing," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 68-73, 1993.
[39] Y. Xiong and S.A. Shafer, "Moment Filters for High Precision Computation of Focus and Stereo," *Proc. Int'l Conf. Intelligent Robots and Systems,* pp. 108-113, Aug. 1995.
[40] L. Yen-Fu, "A Unified Approach to Image Focus and Defocus Analysis," Dept. of Electrical and Computer Eng., State Univ. of New York at Stony Brook, 1998.
[41] D. Ziou and F. Deschenes, "Depth from Defocus Estimation in Spatial Domain," *Computer Vision and Image Understanding,* vol. 81, no. 2, pp. 143-165, Feb. 2001.

Paolo Favaro received the DIng degree from the University of Padova, Italy, in 1999 and the MSc and PhD degrees in electrical engineering from Washington University in 2002 and 2003, respectively. He is currently working as a postdoctoral researcher in the Computer Science Department at the University of California, Los Angeles. His research interests are in computer vision, signal and image processing, estimation theory, sensor-based control, optimization, and variational techniques. He is the recipient of a best poster award at CVPR 2004. He is a member of the IEEE.

Stefano Soatto received the PhD degree in control and dynamical systems from the California Institute of Technology in 1996; he joined the University of California at Los Angeles (UCLA) in 2000 after being an assistant and then associate professor of electrical and biomedical engineering at Washington University and a research associate in applied sciences at Harvard University. Between 1995 and 1998, he was also Ricercatore in the Department of Mathematics and Computer Science at the University of Udine, Italy. He received the DIng degree from the University of Padova, Italy, in 1992. His general research interests are in computer vision and nonlinear estimation and control theory. In particular, he is interested in ways for computers to use sensory information (e.g., vision, sound, touch) to interact with humans and the environment. He is the recipient of the 1999 David Marr Prize (with Y. Ma, J. Kosecka, and S. Sastry) for work on Euclidean reconstruction and reprojection up to subgroups. He also received the 1998 Siemens Prize with the Outstanding Paper Award from the IEEE Computer Society for his work on optimal structure from motion (with R. Brockett). He received the US National Science Foundation Career Award and the Okawa Foundation Grant. He is an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*TPAMI*) and a member of the editorial board of the *International Journal of Computer Vision*. He is a member of the IEEE and the IEEE Computer Society.