Shape from Defocus via Diffusion

Paolo Favaro, *Member, IEEE,* Stefano Soatto, *Member, IEEE,* Martin Burger, and Stanley J. Osher

Abstract— Defocus can be modeled as a diffusion process and represented mathematically using the heat equation, where image blur corresponds to the diffusion of heat. This analogy can be extended to non-planar scenes by allowing a space-varying diffusion coefficient. The inverse problem of reconstructing 3-D structure from blurred images corresponds to an "inverse diffusion" that is notoriously ill-posed. We show how to bypass this problem by using the notion of *relative blur*. Given two images, within each neighborhood, the amount of diffusion necessary to transform the sharper image into the blurrier one depends on the depth of the scene. This can be used to devise a global algorithm to estimate the depth profile of the scene without recovering the deblurred image, using only forward diffusion.

Index Terms— Shape, Reconstruction, Depth cues, Gradient methods, Iterative methods, Partial differential equations, Inverse problems, Sharpening and Deblurring.

I. Introduction

WHEN imaging a scene through a lens, objects at different depths are blurred by different amounts. Indeed, if one knew the shape of the objects then one could predict the exact amount of blur in images. In shape from defocus (SFD) however, one is interested in the inverse problem: Given one or more blurred images, can we reconstruct the shape, or the depth profile, of the scene that generated them?

The answer to this question depends on the scene, and the conditions that allow unique reconstruction are discussed in [1]. Intuitively, the image of a scene depends on its *shape*, that for the case of a static camera can be represented as the depth map along the projection rays, and on its *radiance*, or less formally the appearance of the surfaces, determined by their material and illumination. If the scene is black, or the light is off, there is not much we can say about its shape. However, anyone who has played with an auto-focus camera could guess that by bringing different portions of the scene into focus and reading off the lens setting we ought to be able to say at least something about the depth of the scene.

Accordingly, many algorithms to reconstruct shape from defocus are based on the idea of measuring the "amount of blur" at each location of at least two images obtained with different focus settings. Because blur is not a point property of the image, this approach requires the user to define the spatial regions (windows) where blur is to be computed. Naturally there is a tradeoff between having a window that is as large

as possible, to average out noise, but as small as possible to guarantee that within that window the scene portrayed has constant depth. These local algorithms usually result in artifacts in the reconstruction [2], [3].

Global algorithms for SFD work simultaneously on the entire image, but at the cost of allowing both the radiance and the shape to be (infinite-dimensional) functions of the spatial location. Because details of both the radiance and the shape are lost in the image formation process, formalizing SFD as an optimal inference problem results in an *ill-posed* inverse problem. This problem is typically addressed by means of *regularization*, i.e., by introducing well-posed approximations of the original ill-posed problem. For instance, one could search, among all possible shapes, for the smoothest one. While assuming that the surfaces in space are (at least piece-wise) smooth is somewhat reasonable, assuming the same of the scene radiance is not reasonable. In general, radiances are not smooth and characterized instead by sharp edges, spikes, and very complex and high-frequency patterns. Hence, a more elaborate regularization is required for the radiance.

The approach we propose is novel in several ways, and achieves optimality without resorting to deblurring (backward diffusion), and without imposing overly restrictive assumptions on the radiance of the scene. The key concept is that of *relative blur*: It allows us to eliminate radiance from the image-formation equations, and be left with only depth as the unknown [4]. While one can think of an image as a diffusion of the radiance of the scene, and therefore reconstructing the scene is an inverse diffusion, we think of two or more images as being diffused versions of each other; by appropriately choosing the reference image at each spatial location, such a diffusion is always in the forward direction, and the amount of diffusion depends on the depth of the scene at that location. Such a diffusion is independent of the radiance of the underlying scene, which relieves us from the need of making explicit assumptions about its regularity.

In the next sections we make use of the tools of calculus of variations [5]. The reader who is not familiar with this subject may find Appendix B of [6] useful, in that it contains only the notions essential to optimal SFD.

II. Previous Work

The 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 [7]–[9]. 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 [2], [10]–[19] and those that operate in

P. Favaro is with the School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh, EH14 4AS, UK. E-mail: p.favaro@hw.ac.uk

S. Soatto is with the Computer Science Department, University of California, Los Angeles, CA 90095, USA. E-mail: soatto@ucla.edu

M. Burger is with the Institute for Computational and Applied Mathematics, University of Münster, D 48149 Münster, Germany. E-mail: martin.burger@uni-muenster.de

S. J. Osher is with the Applied Mathematics Department, University of California, Los Angeles, CA 90095, USA. E-mail: sjo@math.ucla.edu

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

IEEE TRANSACTION OF PATTERN RECOGNITION AND MACHINE INTELLIGENCE, VOL. , NO. , MONTH YEAR   2

the frequency domain [20]–[23]. Most of these approaches allow one to easily eliminate undesired unknowns (the appearance, or "radiance"). However, the techniques that are typically employed are local and therefore introduce artifacts in the solution, such as edge bleeding and windowing [2], [3].

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 [15], [24]–[27], others that the radiance can be locally approximated by cubic polynomials [28], or that it can be controlled by using structured light [14], [29], [30]. A more common simplification of the image formation model is the so-called *equifocal assumption*, which consists in assuming that the surface of the scene can be locally approximated by a plane parallel to the image plane [15], [25], [28], [31]–[33]. 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 properly capture a large class of surfaces (non-equifocal surfaces), and does not allow enforcing global regularity on the estimate. Approaches like [7], 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.

In this paper, we present a novel algorithm to optimally recover shape from two defocused images that is computationally efficient. We build on the fact that defocus can be modeled by a diffusion process, which in turn can be modeled by a partial differential equation, the heat equation [34], and on the notion of *relative diffusion* (see section IV) that allows us to avoid estimating the radiance without introducing artifacts in the process.

The literature on diffusion is quite substantial and, therefore, this work relates to a large number of other works. In particular, we highlight connections to the extensive literature in image processing, for instance [35]–[39] and references therein.

III. MODELING DEFOCUS AS A DIFFUSION PROCESS

Consider capturing images of a scene that is static with respect to the camera. If the camera is equipped with a finite aperture lens, the captured images will be subject to a distortion commonly known as *defocus*. Defocus is typically studied via a convolutional image formation model [6], which we briefly review in subsection III-A. However, defocus can also be modeled via partial differential equations. In particular, in the specific case of a scene made of a plane parallel to the image plane, the convolutional model is equivalent to the heat equation (subsection III-B). Since the case of a plane has a rather limited scope, in subsection III-C we generalize the heat equation to handle the case of a scene with a generic (smooth) surface.

A. A Convolutional Model for Defocus

As mentioned above, in this section we will briefly review a convolutional model for defocus. In particular, we will see that in the simple case of uniform blurring, which occurs

whenever we are imaging a plane parallel to the image plane, the convolutional model results to be shift-invariant.

Consider a scene with a smooth *Lambertian*[1] surface. We take images of the scene from the same point of view and assume that scene and illumination are static with respect to the camera . Under these conditions we can represent the surface of the scene with a depth map $s : \mathbb{R}^2 \mapsto [0, \infty)$, and the radiance[2] on s with a function $r : \mathbb{R}^2 \mapsto [0, \infty)$. If we use a real aperture camera, the irradiance (or image intensity) I measured on the image plane with focus setting v of the optics (the distance between the plane containing the lens and the image plane [6]) is a function $I : \mathbb{R}^2 \mapsto [0, \infty)$ that can be approximated via the following equation:

$$I(\mathbf{y}) = \int_{\mathbb{R}^2} h(\mathbf{y}, \mathbf{x}, b) r(\mathbf{x}) d\mathbf{x} \quad \forall \mathbf{y} \in \Omega \subset \mathbb{R}^2 \quad (1)$$

where $h : \Omega \times \mathbb{R}^2 \times [0, \infty) \mapsto [0, \infty)$ is called *point spread function* (PSF). When $\mathbf{y} \notin \Omega \subset \mathbb{R}^2$ we assume that $I(\mathbf{y}) = 0$. The point spread function depends on the *blurring radius* b which in turn depends on the focus setting v and the depth map s. The blurring radius b determines the size of the pattern generated by a point light source at $[\frac{\mathbf{x}^T}{v} \ 1]^T s(\mathbf{x})$ with unit intensity. By employing geometric optics in our analysis, it is easy to find that the blurring radius satisfies

$$b = \frac{Dv}{2} \left| \frac{1}{F} - \frac{1}{v} - \frac{1}{s} \right| \quad (2)$$

where F is the focal length of the lens and D the radius of the lens [6].

An important case that we will consider in the next section is that of a scene made of an *equifocal plane*, i.e., a plane parallel to the image plane. In this case the depth map satisfies $s(\mathbf{x}) = s, \forall \mathbf{x} \in \mathbb{R}^2$, the PSF h is shift-invariant, i.e., $h(\mathbf{x}, \mathbf{y}, b) = h(\mathbf{x} - \mathbf{y}, b)$ and b is a constant. Hence, the image formation model becomes the following simple convolution

$$I = h(\cdot, b) * r. \quad (3)$$

One can argue[3] that h can be well approximated with a Gaussian; such an approximation has been widely used in the literature of depth from defocus [7]. Hence, in our model we will let

$$h(\mathbf{x}, \mathbf{y}, b) = \frac{1}{2\pi\sigma^2} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} \quad (4)$$

with standard deviation $\sigma \doteq \gamma b$ for a certain constant $\gamma > 0$ to be determined via a calibration procedure. Notice that when

---

[1] A Lambertian surface is characterized by having a *bidirectional reflectance distribution function* that is independent of the viewing direction [40].

[2] In the context of *radiometry*, the term *radiance* refers to energy emitted along a certain direction, per solid angle, per foreshortened area and per time interval [40]. However, in our case there is little dependency on direction, and the change in the solid angle is approximately negligible. Hence, a function of the position on the surface of the scene suffices to describe the variability of the radiance.

[3] It has been argued [7], [15], [41] that, when considering diffraction effects, the PSF of a camera can be approximated by a circularly symmetric 2D Gaussian. Alternatively, one can also argue that since we are interested in using the camera as a sensor, we are allowed to modify the optical system. It can be shown that the PSF can be made approximately Gaussian by placing a suitable photographic mask in front of the lens. Notice that this operation does not compromise the shift-invariance of the PSF when imaging an equifocal plane.

the depth map s is not an equifocal plane, the variance σ^2 depends on the $\mathbf{x}$ coordinates, so that h may be not shift-invariant. More in general, one can approximate the PSF with other functions, as long as they satisfy the following *normalization property*:

$$\int h(\mathbf{y}, \mathbf{x}, b)d\mathbf{y} = 1 \quad \forall \mathbf{x} \in \mathbb{R}^2 \tag{5}$$

for any depth map s and focus setting v. The property above corresponds to having a lossless optical system, i.e., an optical system such that all the energy entering the lens is transferred onto the image plane.

B. Equifocal Imaging as Isotropic Diffusion

When the PSF is approximated by a shift-invariant Gaussian function, the imaging model in eq. (1) can be formulated in terms of the *isotropic heat equation*:

$$\begin{cases} \dot{u}(\mathbf{x}, t) = c\triangle u(\mathbf{x}, t) & c \in [0, \infty) \quad t \in (0, \infty) \\ u(\mathbf{x}, 0) = r(\mathbf{x}). \end{cases} \tag{6}$$

The solution $u : \mathbb{R}^2 \times [0, \infty) \mapsto [0, \infty)$ at a time $t = \tau$, plays the role of an image $I(\mathbf{y}) = u(\mathbf{y}, \tau), \forall \mathbf{y} \in \Omega$, captured with a certain focus setting p that is related to τ. The "dot" denotes differentiation in time, i.e., $\dot{u} \doteq \frac{\partial u}{\partial t}$, and the symbol $\triangle$ denotes the Laplacian operator $\sum_{i=1}^{2} \frac{\partial^2}{\partial x_i^2}$ with $\mathbf{x} \doteq [x_1 \; x_2]^T$. The parameter c is called the *diffusion coefficient* and it is nonnegative. It is also easy to verify that the variance σ is related to the diffusion coefficient c via

$$\sigma^2 = 2tc. \tag{7}$$

Notice that there is a scale factor ambiguity between time t and diffusion coefficient c. This will be fixed later by setting t to a prescribed constant.

C. Non-equifocal imaging model

When the depth map s is not an equifocal plane, the corresponding PSF is in general shift-varying. Hence, the equivalence with the isotropic heat equation does not hold. Rather than seeking an approximation for the shift-varying PSF, we propose a model based on a generalization of the isotropic heat equation that satisfies property (5). To take into account the space-varying nature of the non-equifocal case, we propose using the *inhomogeneous diffusion equation*[4], and define a space-varying diffusion coefficient[5] $c \in L^\infty(\mathbb{R}^2)$, and $c(\mathbf{x}) \geq 0 \; \forall \mathbf{x} \in \mathbb{R}^2$. The inhomogeneous diffusion equation is then defined as

$$\begin{cases} \dot{u}(\mathbf{x}, t) = \nabla \cdot (c(\mathbf{x})\nabla u(\mathbf{x}, t)) & t \in (0, \infty) \\ u(\mathbf{x}, 0) = r(\mathbf{x}) \end{cases} \tag{9}$$

[4]Note that the *inhomogeneous diffusion equation* is different from the *nonhomogeneous diffusion equation*, which is characterized by an additional *forcing term* in the heat equation as shown in the example below

$$\begin{cases} \dot{u}(\mathbf{x}, t) = c\triangle u(\mathbf{x}, t) + f(\mathbf{x}, t) & t \in (0, \infty) \\ u(\mathbf{x}, 0) = r(\mathbf{x}) \end{cases} \tag{8}$$

where $f(\mathbf{x}, t)$ is the forcing term. The nomenclature *inhomogeneous* wants to emphasize that the diffusion coefficient is not homogeneous in space [39].
[5]$L^\infty(\mathbb{R}^2)$ is the space of scalar functions bounded almost everywhere in $\mathbb{R}^2$.

where the symbol ∇ is the gradient operator $\left[\frac{\partial}{\partial x_1} \; \frac{\partial}{\partial x_2}\right]^T$ with $\mathbf{x} = [x_1 \; x_2]^T$, and the symbol $\nabla\cdot$ is the divergence operator $\sum_{i=1}^{2} \frac{\partial}{\partial x_i}$. In our analysis, we consider the so-called weak solutions of eq. (9), which are obtained by multiplying the terms of eq. (9) to a space-dependent test function $\varphi : \mathbb{R}^2 \mapsto \mathbb{R}$, and then by integrating the result over $\mathbb{R}^2$. By using Gauss' theorem (integration by parts, cf. [42]), this yields the weak formulation

$$\frac{d}{dt}\int u(\mathbf{x}, t)\varphi(\mathbf{x}) \, d\mathbf{x} + \int c(\mathbf{x})\nabla u(\mathbf{x}, t)\cdot\nabla\varphi(\mathbf{x})d\mathbf{x} = 0, \tag{10}$$

which must hold for a suitable class of test functions φ. This formulation shows that a bounded function c (not necessarily differentiable) is indeed sufficient to guarantee the well-definedness of the solution. Furthermore, it is easy to see that the conservation property (5) can be formulated as

$$\int u(\mathbf{y}, t)d\mathbf{y} = \int u(\mathbf{x}, 0)d\mathbf{x}. \tag{11}$$

In other words, the spatial average of u does not change in time. To see this, let the test function $\varphi \equiv 1$; then, we immediately have that

$$\frac{d}{dt}\int u(\mathbf{x}, t) \, d\mathbf{x} = 0, \tag{12}$$

because $\nabla\varphi \equiv 0$. By integrating eq. (12) in time we obtain the desired conservation property, which we summarize in the following

Proposition 1: The solution of the following inhomogeneous heat equation

$$\begin{cases} \dot{u}(\mathbf{x}, t) = \nabla \cdot (c(\mathbf{x})\nabla u(\mathbf{x}, t)) & c : \mathbb{R}^2 \mapsto [0, \infty), c \in L^\infty \\ u(\mathbf{x}, 0) = r(\mathbf{x}) \end{cases} \tag{13}$$

satisfies property (5).

The solution u satisfies also the following dissipation properties

$$\frac{1}{2}\frac{d}{dt}\int u(\mathbf{x}, t)^2 d\mathbf{x} = -\int c(\mathbf{x})\|\nabla u(\mathbf{x}, t)\|^2 d\mathbf{x} \leq 0 \tag{14}$$

$$\frac{1}{2}\frac{d}{dt}\int c(\mathbf{x})\|\nabla u(\mathbf{x}, t)\|^2 d\mathbf{x} = -\int\left(\frac{\partial u}{\partial t}(\mathbf{x}, t)\right)^2 d\mathbf{x} \leq 0, \tag{15}$$

which imply that the long-time asymptotic limit $u_\infty(\mathbf{x}) \doteq \lim_{t\to\infty} u(\mathbf{x}, t)$ satisfies $c(\mathbf{x})\|\nabla u_\infty(\mathbf{x})\|^2 = 0$, i.e., u_∞ needs to be constant where c is positive.

By assuming that the surface s is smooth, we can relate again the diffusion coefficient c to the space-varying variance σ via:

$$\sigma^2(\mathbf{x}) = 2tc(\mathbf{x}) \tag{16}$$

so that it is immediate to see that c encodes the depth map s of the scene via

$$c(\mathbf{x}) = \frac{\gamma^2 b^2(\mathbf{x})}{2t} = \frac{\gamma^2 D^2 v^2}{8t}\left(\frac{1}{F} - \frac{1}{v} - \frac{1}{s(\mathbf{x})}\right)^2. \tag{17}$$

IV. RELATIVE BLUR AND DIFFUSION

In section III-B we modeled the image $I(\mathbf{y})$ via the diffusion eq. (6) starting from the radiance $r(\mathbf{y})$, which we do not know. Rather than having to estimate it from two or more images, in this section we show how to arrive at a model of the *relative blur* between two images that does not depend on the radiance. Suppose we collect two images $I_1(\mathbf{y}) \doteq I(\mathbf{y}, v_1)$ and $I_2(\mathbf{y}) \doteq I(\mathbf{y}, v_2)$ for two different focus settings v_1 and v_2 corresponding to blurring parameters σ_1 and σ_2. Also, to keep the presentation simple, suppose that $\sigma_1 < \sigma_2$ (i.e., image I_1 is more focused than image I_2 at every pixel); we will remove this assumption later. Then, by substituting the expression of r in terms of image I_1 we can write image I_2 as follows:

$$
\begin{aligned}
I_2(\mathbf{y}) &= \int \frac{1}{2\pi\sigma_2^2} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma_2^2}} r(\mathbf{x})d\mathbf{x} \\
&= \int \frac{1}{2\pi\left(\sigma_2^2-\sigma_1^2\right)} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2(\sigma_2^2-\sigma_1^2)}} \\
&\qquad \int \frac{1}{2\pi\sigma_1^2} e^{-\frac{\|\bar{\mathbf{x}}-\mathbf{x}\|^2}{2\sigma_1^2}} r(\bar{\mathbf{x}})d\bar{\mathbf{x}}d\mathbf{x} \\
&= \int \frac{1}{2\pi\Delta\sigma^2} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\Delta\sigma^2}} I_1(\mathbf{x})d\mathbf{x}
\end{aligned}
\tag{18}
$$

where $\Delta\sigma^2 \doteq \sigma_2^2 - \sigma_1^2$ and $\Delta\sigma$ is called the *relative blurring* between image I_1 and image I_2. Now, I_2 in eq. (18) can also be interpreted as a solution of the heat equation (6), but with I_1, rather than r, as an initial condition: Let t_1 and t_2 be the two time instants computed from eq. (7) with a fixed diffusion coefficient c for blurring parameters σ_1 and σ_2 respectively; then, the solution of eq. (6) satisfies $u(\mathbf{y}, t_1) = I_1(\mathbf{y})$ and $u(\mathbf{y}, t_2) = I_2(\mathbf{y})$, $\forall \mathbf{y} \in \Omega$, and we can write

$$
\left\{
\begin{array}{ll}
\dot{u}(\mathbf{y}, t) = c\triangle u(\mathbf{y}, t) & c \in [0, \infty) \\
u(\mathbf{y}, t_1) = I_1(\mathbf{y}) & \forall \mathbf{y} \in \Omega.
\end{array}
\right.
\tag{19}
$$

If $\sigma_1 > \sigma_2$, one can simply switch I_1 with I_2.

Equation (19) models the *relative diffusion* between image I_1 and image I_2. This allows us to eliminate r as an unknown, and concentrate our resources to recovering only the geometry of the scene.

For simplicity of notation, rather than using the solution $u(\cdot, t)$, we can consider the time-shifted version $u(\cdot, t - t_1)$ (or set $t_1 = 0$) so that

$$
\left\{
\begin{array}{ll}
\dot{u}(\mathbf{y}, t) = c\triangle u(\mathbf{y}, t) & c \in [0, \infty) \\
u(\mathbf{y}, 0) = I_1(\mathbf{y}) & \forall \mathbf{y} \in \Omega
\end{array}
\right.
\tag{20}
$$

with $u(\mathbf{y}, \Delta t) = I_2(\mathbf{y})$, where Δt is defined by the equation

$$
\Delta\sigma^2 = 2(t_2 - t_1)c \doteq 2\Delta t c.
\tag{21}
$$

One can view the time Δt as the variable encoding the global amount of defocus, which we set to $1/2$ in our implementation, and the diffusion coefficient c as the variable encoding the depth map s via

$$
c = \frac{\Delta\sigma^2}{2\Delta t} = \frac{\gamma^2 \left(b_2^2 - b_1^2\right)}{2\Delta t}
\tag{22}
$$

where for $i = 1, 2$

$$
b_i = \frac{Dv_i}{2} \left| \frac{1}{F} - \frac{1}{v_i} - \frac{1}{s} \right|.
\tag{23}
$$

This discussion holds as long as the PSF is shift-invariant. As we have seen in Section III, this corresponds to the entire image being blurred uniformly, or "isotropically," at all locations, which occurs only if the scene is flat and parallel to the image plane (i.e., it is equifocal). Naturally, such scenes are rather uninteresting; therefore, to handle non-flat scenes, we need to extend the diffusion analogy to equations that allow for a space-varying blur.

V. EXTENSION TO SPACE-VARYING RELATIVE DIFFUSION

As we have seen in Section III-C, when the surface s is not an equifocal plane, the corresponding PSF is shift-varying and we cannot use the homogeneous heat equation to model defocused images. Therefore, we have introduced the inhomogeneous diffusion equation (8) by allowing the coefficient c to vary spatially as a function of the location on the image. Similarly, in the context of relative diffusion, we can extend eq. (20) to

$$
\left\{
\begin{array}{ll}
\dot{u}(\mathbf{y}, t) = \nabla \cdot (c(\mathbf{y})\nabla u(\mathbf{y}, t)) & t \in (0, \infty) \\
u(\mathbf{y}, 0) = I_1(\mathbf{y}) & \forall \mathbf{y} \in \Omega \\
c(\mathbf{y})\nabla u(\mathbf{y}, t) \cdot \mathbf{n}(\mathbf{y}) = 0 & \forall \mathbf{y} \in \partial\Omega
\end{array}
\right.
\tag{24}
$$

where $u(\mathbf{y}, \Delta t) = I_2(\mathbf{y})$, $\forall \mathbf{y} \in \Omega$, and $\mathbf{n}$ is the unit vector normal to the boundary $\partial\Omega$. Notice that a third constraint has been introduced in eq. (24). This term is necessary to define the solution uniquely, because the model is now restricted to a domain Ω. In general, c or $\nabla u(y, t) \cdot \mathbf{n}(y)$ may not be null at $\partial\Omega$. However, in the next section we define the above equation on a set Ω such that $c(\mathbf{y}) = 0$ for $\mathbf{y} \in \partial\Omega$, so that the condition is satisfied by construction.

Recall that in Section III-C we required the diffusion coefficient c to be a smooth function $c : \Omega \mapsto \mathbb{R}$ that satisfies a number of properties in order to guarantee that energy is conserved in the imaging process (5). We still assume that I_1 is more focused than I_2, i.e. $\sigma_1 < \sigma_2$, for simplicity.

Remark 1: There are other space-varying models that one could use in place of (24), for instance the oriented-Laplacian [43]. Each has its own advantages, but none is exactly equivalent to the space-varying convolutional model in eq. (1) or to more accurate models derived from diffraction theory [44]. Furthermore, the point spread function of real optical systems can deviate substantially from the simple models that we describe here due to various sources of aberration. Hence, the use of one model or another may be decided based on precise knowledge of the imaging device or the specific application at hand or on computational resources. In this manuscript, for the sake of clarity and simplicity, we will focus only on model (24) as a prototype.

VI. ENFORCING FORWARD DIFFUSION

In the derivation of eq. (24) we made the assumption that I_1 is more focused than I_2. This assumption is necessary in order to guarantee that $c(\mathbf{y}) \geq 0$, $\forall \mathbf{y} \in \Omega$, and hence that (24) involves only numerically stable and well-behaved *forward diffusion*, and not backward diffusion. However, as illustrated in Figure 1, this assumption is in general not valid. The woman in the foreground of image I_1 (left) is more
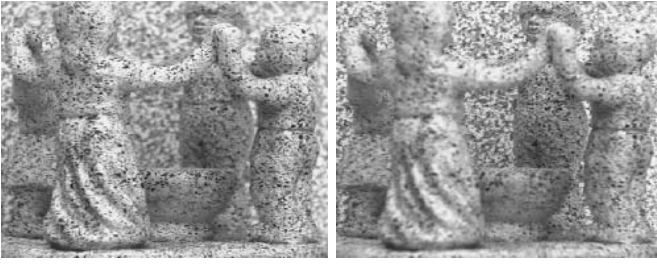
Fig. 1

TWO DEFOCUSED IMAGES I_1 AND I_2. THE WOMAN IN THE FOREGROUND OF THE IMAGE ON THE LEFT (I_1) IS SHARPER THAN THE CORRESPONDING WOMAN IN THE IMAGE ON THE RIGHT (I_2). AT THE SAME TIME, THE BACKGROUND OF THE IMAGE ON THE RIGHT IS SHARPER THAN THE CORRESPONDING BACKGROUND OF THE IMAGE ON THE LEFT.
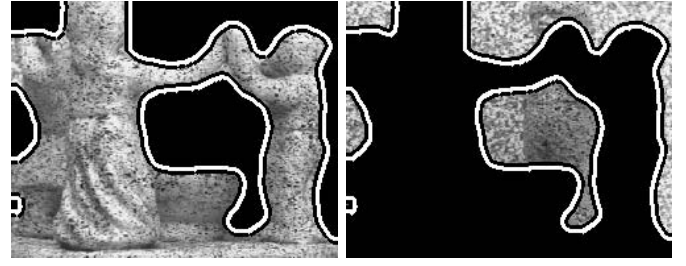

Fig. 2

VISUALIZATION OF THE PARTITION $\{\Omega_+, \Omega_-\}$ WITHIN THE IMAGE DOMAIN Ω OF THE IMAGES IN FIGURE 1. THE SETS Ω_+ (LEFT IMAGE) AND Ω_- (RIGHT IMAGE) ARE VISUALIZED BY SHOWING ONLY THE INTENSITY OF THE ORIGINAL IMAGES IN THE CORRESPONDING PARTITION. THE BOUNDARIES $\partial\Omega_+$ (LEFT IMAGE) AND $\partial\Omega_-$ (RIGHT IMAGE) ARE SHOWN AS A WHITE LINE.

focused than the corresponding woman in image I_2 (right). On the other hand, the background is more focused in image I_2. As a consequence, the relative blurring $\Delta\sigma$ between I_1 and I_2, as well as the corresponding diffusion coefficient c, may be either positive or negative, depending on which location of the image domain Ω they are evaluated at. In this case, model (24) involves both backward diffusion and forward diffusion. Backward diffusion is numerically unstable and has the undesirable effect of amplifying high-frequency content as time increases.

A simple way to avoid backward diffusion in the imaging model (24) is to explicitly define the solution u only on the set of points where the diffusion coefficient is positive

$$\Omega_+ \doteq \{\mathbf{y} \in \mathbb{R}^2 \mid c(\mathbf{y}) > 0\} \quad (25)$$

and $u(\mathbf{y}, \Delta t) = I_2(\mathbf{y}), \forall \mathbf{y} \in \Omega_+$. On the remaining portion

$$\Omega_- \doteq \{\mathbf{y} \in \mathbb{R}^2 \mid c(\mathbf{y}) \leq 0\} \quad (26)$$

the diffusion coefficient c is negative[6] and I_2 is sharper than I_1. Hence, on Ω_- we can employ the following model, that simply switches I_1 and I_2

$$\begin{cases} \dot{u}(\mathbf{y}, t) = \nabla \cdot (-c(\mathbf{y})\nabla u(\mathbf{y}, t)) & t \in (0, \infty) \\ u(\mathbf{y}, 0) = I_2(\mathbf{y}) & \forall \mathbf{y} \in \Omega_- \\ c(\mathbf{y})\nabla u(\mathbf{y}, t) \cdot \mathbf{n}(\mathbf{y}) = 0 & \forall \mathbf{y} \in \partial\Omega_- \end{cases} \quad (27)$$

with $u(\mathbf{y}, \Delta t) = I_1(\mathbf{y}), \forall \mathbf{y} \in \Omega_-$. In Figure 2 we visualize the partition $\{\Omega_+, \Omega_-\}$ obtained from the images in Figure 1. The sets Ω_+ (left image) and Ω_- (right image) are visualized by showing only the intensity of the original images in the corresponding partition. The boundaries $\partial\Omega_+$ and $\partial\Omega_-$ are shown as a white line.

Notice that because of eq. (25) we have defined the partition $\{\Omega_+, \Omega_-\}$ so that the boundary conditions is always satisfied and the imaging models for the pair of defocused images I_1 and I_2 involve only forward diffusion.

[6]We always consider the diffusion coefficient c to be computed via eq. (21), where $\Delta\sigma$ is the relative blur between I_1 and I_2.

VII. DEPTH-MAP ESTIMATION ALGORITHM

In previous sections we have derived an idealized image-formation model in terms of the diffusion eq. (6): First we eliminated the radiance from the model by introducing the notion of relative blur (Section IV); then, we extended the model to capture non-planar scenes (Section V) and, finally, we enforced forward diffusion by partitioning the image domain (Section VI). The result of these steps is an imaging model composed of equations (19) and (27), which we rewrite here in a more complete form, including boundary conditions, as

$$\begin{cases} \dot{u}(\mathbf{y}, t) = \begin{cases} \nabla \cdot (c(\mathbf{y})\nabla u(\mathbf{y}, t)), & \mathbf{y} \in \Omega_+ \\ \nabla \cdot (-c(\mathbf{y})\nabla u(\mathbf{y}, t)), & \mathbf{y} \in \Omega_- \end{cases} \quad t \in (0, \infty) \\ u(\mathbf{y}, 0) = \begin{cases} I_1(\mathbf{y}), & \forall \mathbf{y} \in \Omega_+ \\ I_2(\mathbf{y}), & \forall \mathbf{y} \in \Omega_- \end{cases} \\ 0 = c(\mathbf{y})\nabla u(\mathbf{y}, t) \cdot \mathbf{n}(\mathbf{y}), \quad \forall \mathbf{y} \in \partial\Omega_+ \equiv \partial\Omega_- \\ u(\mathbf{y}, \Delta t) = \begin{cases} I_2(\mathbf{y}), & \forall \mathbf{y} \in \Omega_+ \\ I_1(\mathbf{y}), & \forall \mathbf{y} \in \Omega_-. \end{cases} \end{cases} \quad (28)$$

where $\mathbf{n}$ is the unit vector orthogonal to $\partial\Omega_+$, the boundary of Ω_+, which coincides with the boundary of Ω_-. The boundary conditions of the diffusion equations are satisfied since, by construction, the diffusion coefficient $c(\mathbf{y}) = 0$ for any $\mathbf{y} \in \partial\Omega_+$ and $\mathbf{y} \in \partial\Omega_-$. Also, recall that the diffusion coefficient c depends on the depth map s via eq. (22) and eq. (23), which can be explicitly written as

$$c(\mathbf{y}) = \frac{\gamma^2 D^2}{8\Delta t}\left(v_2^2\left(\frac{1}{F} - \frac{1}{v_2} - \frac{1}{s(\mathbf{y})}\right)^2 \right.$$
$$\left. - v_1^2\left(\frac{1}{F} - \frac{1}{v_1} - \frac{1}{s(\mathbf{y})}\right)^2 \right). \quad (29)$$

Notice that $c(\mathbf{y}) = 0$ when

$$s(\mathbf{y}) = \frac{(v_1 + v_2)F}{v_1 + v_2 - 2F} \quad (30)$$

or

$$s(\mathbf{y}) = F \quad (31)$$

so that

$$\partial\Omega_-, \partial\Omega_+ = \left\{ \mathbf{y} \ \middle| \ s(\mathbf{y}) = \frac{(v_1 + v_2)F}{v_1 + v_2 - 2F} \quad \text{or} \quad s(\mathbf{y}) = F \right\}$$
(32)

and similarly, by direct substitution,

$$
\begin{aligned}
\Omega_+ &= \left\{ \mathbf{y} \ \middle| \ 0 < s(\mathbf{y}) < F \quad \text{or} \quad s(\mathbf{y}) > \frac{(v_1 + v_2)F}{v_1 + v_2 - 2F} \right\} \\
\Omega_- &= \left\{ \mathbf{y} \ \middle| \ F < s(\mathbf{y}) < \frac{(v_1 + v_2)F}{v_1 + v_2 - 2F} \right\}.
\end{aligned}
$$
(33)

Note that the sets Ω_+ and Ω_- depend on the 3-D structure of the scene; the depth s satisfying eq. (30) and eq. (31) corresponds to regions where both images undergo the same amount of defocus. Hence, if we were given a depth map $\tilde{s}$ and the calibration parameters of the camera, we could compute the diffusion coefficient c, the partition $\{\Omega_+, \Omega_-\}$ and then simulate eq. (28). If the depth map $\tilde{s}$ coincides with the depth map s of the scene, then the solution of eq. (28) must satisfy the condition $u(\mathbf{y}, \Delta t) = I_2(\mathbf{y})$, $\forall \mathbf{y} \in \Omega_+$ and $u(\mathbf{y}, \Delta t) = I_1(\mathbf{y})$, $\forall \mathbf{y} \in \Omega_-$.

This naturally suggests an iterative procedure to tackle the inverse problem of reconstructing shape from defocused images. An intuitive description of the iteration is: Starting from an initial estimate of the depth map (e.g., a flat plane such that $c(\mathbf{y}) = 0$) and the corresponding partition, find the amount of diffusion for each image in the region where it is more focused than the other image, so that the two are close. The amount of diffusion required to match the two images encodes information on the depth of the scene. More formally, we could pose the problem as the minimization of the following functional:

$$
\begin{aligned}
\hat{s} = \ \arg\min_s &\int_{\Omega_+} (u(\mathbf{y}, \Delta t) - I_2(\mathbf{y}))^2 d\mathbf{y} \\
&+ \int_{\Omega_-} (u(\mathbf{y}, \Delta t) - I_1(\mathbf{y}))^2 d\mathbf{y}.
\end{aligned}
$$
(34)

where the two terms in the cost functional take into account the discrepancy between the simulated image u and the measured images I_1 and I_2. It is well-known that problems like (34) are ill-posed [45], i.e., the minimizers may not exist and even if they exist they may not be stable with respect to data noise. A common way to regularize the problem is by adding a Tikhonov penalty, that is,

$$
\begin{aligned}
\hat{s} = \ \arg\min_s &\int_{\Omega_+} (u(\mathbf{y}, \Delta t) - I_2(\mathbf{y}))^2 d\mathbf{y} \\
&+ \int_{\Omega_-} (u(\mathbf{y}, \Delta t) - I_1(\mathbf{y}))^2 d\mathbf{y} + \alpha\|\nabla s\|^2 + \alpha\kappa\|s\|^2.
\end{aligned}
$$
(35)

The additional third term imposes a smoothness constraint on the depth map s that is regulated by the parameter $\alpha > 0$. The last term is introduced to guarantee that the estimated depth map is bounded. In practice we use a very small $\kappa > 0$, so that this term has no practical influence on the remaining energy terms. Notice that during the minimization of eq. (35) both the partition and the depth map are estimated simultaneously.

As we shall discuss in the next section, minimizing eq. (35) is not the only way of regularizing eq. (34). Indeed, the energy

eq. (35) does not yield the desired solution for $\alpha > 0$. To illustrate this, notice that the minimum of eq. (35) is reached when the gradients of each term add up to zero at each point $\mathbf{x}$. Such a balance depends on the intensity of the images, so that dark regions are subject to more regularization than bright regions. To counterbalance this dependency, we will not minimize eq. (35), but, rather, we will introduce an alternative method that combines the minima of eq. (34) with those of the regularizer $\|\nabla s\|^2 + \kappa\|s\|^2$.

A. Gradient Flow and Preconditioning

In this section we describe an algorithm to compute regularized solutions of the cost functional (34). The complete derivation of the algorithm is fairly lengthy and is reported in Appendix I. For the benefit of those being familiar with these calculations, and not to disrupt the flow, we will only provide the main ideas in the following.

To simplify the notation, let

$$
\begin{aligned}
E_1(s) &\doteq \int H(c(\mathbf{y}))|u(\mathbf{y}, \Delta t) - I_2(\mathbf{y})|^2 d\mathbf{y} \\
E_2(s) &\doteq \int H(-c(\mathbf{y}))|u(\mathbf{y}, \Delta t) - I_1(\mathbf{y})|^2 d\mathbf{y} \\
E_3(s) &\doteq \alpha\|\nabla s\|^2 + \alpha\kappa\|s\|^2
\end{aligned}
$$
(36)

where H denotes the Heaviside function and $\alpha > 0$, $\kappa > 0$ are arbitrary constants. Also, let

$$E(s) \doteq E_1(s) + E_2(s) + E_3(s)$$
(37)

so that eq. (35) can be rewritten as

$$\hat{s} = \arg\min_s E(s).$$
(38)

For the minimization of the above cost functional, a standard gradient flow approach seems natural. This means that we construct a flow of depth maps s indexed by a pseudo-time variable τ, so that the depth map moves along the direction opposite to the gradient of the cost functional, i.e.,

$$\frac{\partial s}{\partial \tau} \doteq -E'(s).$$
(39)

For practical purposes this flow will be discretized in time. Any suitable forward time integration such as the forward Euler scheme can be applied [46].

Using eq. (37) we can split the computation of E' into the computation of E_1', E_2' and E_3'. In the following we use the notation $\tilde{E}_i$, $i = 1, 2$, for the functional E_i, which is rewritten as a functional of the diffusion coefficient c. By the chain rule, we then have

$$E_i'(s) = \tilde{E}_i'(c(s))c'(s)$$
(40)

for $i = 1, 2$, where

$$c'(s) = \frac{\gamma^2 D^2 (v_2 - v_1)}{4s^2 \Delta t} \left[(v_2 + v_1) \left(\frac{1}{F} - \frac{1}{s} \right) - 1 \right],$$
(41)

and the gradients $\tilde{E}_i'$ are given by

$$
\begin{aligned}
\tilde{E}_1'(c(s))(\mathbf{y}) = &-2H(c(\mathbf{y})) \int_0^{\Delta t} \nabla u(\mathbf{y}, t) \cdot \nabla w_1(\mathbf{y}, \Delta t - t) \ dt \\
&+ \delta(c(\mathbf{y})) \left(u(\mathbf{y}, \Delta t) - I_2(\mathbf{y}) \right)^2
\end{aligned}
$$
(42)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

IEEE TRANSACTION OF PATTERN RECOGNITION AND MACHINE INTELLIGENCE, VOL. , NO. , MONTH YEAR 7

and

$$\tilde{E}_2'(c(s))(\mathbf{y}) = 2H(-c(\mathbf{y}))\int_0^{\Delta t} \nabla u(\mathbf{y},t)\cdot\nabla w_2(\mathbf{y},\Delta t - t)\ dt$$
$$- \delta(c(\mathbf{y}))\left(u(\mathbf{y},\Delta t) - I_1(\mathbf{y})\right)^2. \tag{43}$$

The symbol $\delta(\cdot)$ denotes the Dirac delta. Here $w_1 : \Omega_+ \times [0,\infty) \mapsto \mathbb{R}$ and $w_2 : \Omega_- \times [0,\infty) \mapsto \mathbb{R}$ satisfy the following (adjoint parabolic) equations (see Appendix I)

$$\begin{cases} \dot{w}_1(\mathbf{y},t) = \nabla\cdot(c(\mathbf{y})\nabla w_1(\mathbf{y},t)) & t \in (0,\infty) \\ w_1(\mathbf{y},0) = u(\mathbf{y},\Delta t) - I_2(\mathbf{y}) \\ c(\mathbf{y})\nabla w_1(\mathbf{y},t)\cdot\mathbf{n}(\mathbf{y}) = 0 & \forall\mathbf{y}\in\partial\Omega_+. \end{cases} \tag{44}$$

and

$$\begin{cases} \dot{w}_2(\mathbf{y},t) = \nabla\cdot(-c(\mathbf{y})\nabla w_2(\mathbf{y},t)) & t \in (0,\infty) \\ w_2(\mathbf{y},0) = u(\mathbf{y},\Delta t) - I_1(\mathbf{y}) \\ c(\mathbf{y})\nabla w_2(\mathbf{y},t)\cdot\mathbf{n}(\mathbf{y}) = 0 & \forall\mathbf{y}\in\partial\Omega_-. \end{cases} \tag{45}$$

Finally, the gradient of E_3 can be computed directly as

$$E_3'(s) = -2\alpha\triangle s(\mathbf{y}) + 2\alpha\kappa s(\mathbf{y}). \tag{46}$$

Since the gradient $E'(s)(\mathbf{y})$ depends on the radiance of the scene (see eq. (42) and (43)), which is undesirable, one can modify the above algorithm by *preconditioning*, i.e., by replacing eq. (39) with

$$\frac{\partial s}{\partial\tau} = -M(s)E'(s), \tag{47}$$

with $M(s)$ a positive definite operator. Note that this approach is a metric gradient flow [47] for the same energy E, obtained by weighting the scalar product for s with $M^{-1}(s)$. In particular, we would like to use preconditioning with an operator of the form

$$(M(s)\phi)(\mathbf{y}) = \frac{\phi(\mathbf{y})}{2\left[H(c(\mathbf{y}))I_2(\mathbf{y}) + H(-c(\mathbf{y}))I_1(\mathbf{y})\right]|u'(\mathbf{y},\Delta t)|}, \tag{48}$$

where $u'(\mathbf{y},\Delta T)$ is the functional derivative of u with respect to the shape s. By simplifying eq. (48), we obtain

$$\left(M(s)\tilde{E}_1'(c(s))\right)(\mathbf{y}) = H(c(\mathbf{y}))\left(\frac{u(\mathbf{y},\Delta t)}{I_2(\mathbf{y})} - 1\right)\frac{u'(\mathbf{y},\Delta t)}{|u'(\mathbf{y},\Delta t)|}$$
$$+ \frac{1}{2}\delta(c(\mathbf{y}))\frac{(u(\mathbf{y},\Delta t) - I_2(\mathbf{y}))^2}{I_2(\mathbf{y})|u'(\mathbf{y},\Delta t)|} \tag{49}$$

and, similarly,

$$\left(M(s)\tilde{E}_2'(c(s))\right)(\mathbf{y}) = H(-c(\mathbf{y}))\left(\frac{u(\mathbf{y},\Delta t)}{I_1(\mathbf{y})} - 1\right)\frac{u'(\mathbf{y},\Delta t)}{|u'(\mathbf{y},\Delta t)|}$$
$$- \frac{1}{2}\delta(c(\mathbf{y}))\frac{(u(\mathbf{y},\Delta t) - I_1(\mathbf{y}))^2}{I_1(\mathbf{y})|u'(\mathbf{y},\Delta t)|}. \tag{50}$$

It is now very clear what drove the choice of the preconditioner in eq. (48): The gradients (49) and (50) have been "normalized", as there is no dependency on the magnitude of $u'(\mathbf{y},\Delta t)$ and on $u(\mathbf{y},\Delta t)$. In practice, to avoid division by zero, we also introduce a small positive value at the denominator of eq. (48).

Notice that by preconditioning E' we are also preconditioning the term E_3', that did not need any normalization. This motivates us to consider the flow

$$\frac{\partial s}{\partial\tau} = -M(s)(E_1'(s) + E_2'(s)) - E_3'(s)$$
$$= -M(s)(E_1'(s) + E_2'(s) + M^{-1}(s)E_3'(s)) \tag{51}$$

TABLE I

SUMMARY OF THE DEPTH MAP RECONSTRUCTION ALGORITHM VIA RELATIVE DIFFUSION.

Algorithm (relative diffusion)

1) Given: calibration parameters (from knowledge of the camera) v_1, v_2, F, D, γ, two images I_1, I_2, a chosen threshold ϵ, regularization parameter α and step size β, seek for the depth map s as follows
2) Initialize the depth map with a plane at depth

$$s_0 = \frac{(v_1 + v_2)F}{v_1 + v_2 - 2F};$$

3) Compute the diffusion coefficient c and the partition $\{\Omega_+, \Omega_-\}$ via eq. (29) and eq. (33);
4) Simulate (i.e., numerically integrate) eq. (19) and eq. (27);
5) Using the solutions obtained at the previous step, simulate eq. (44) and eq. (45);
6) Compute the gradient of u and w and evaluate eq. (42), eq. (43), eq. (46), and (48);
7) Update the depth map s by performing a time step of $\frac{\partial s}{\partial\tau} = -M(s)(E_1'(s) + E_2'(s)) - E_3'(s)$, with precomputed right-hand side.
8) Return to Step 3 until norm of $E_1'(s) + E_2'(s) + M(s)^{-1}E_3'(s)$ is below the chosen threshold ϵ.

instead, which we briefly discuss here below and more thoroughly in Appendix II. First, we summarize the algorithm described so far in Table I.

Note that eq. (51) is possibly not a metric gradient flow for some energy, since the last term $M^{-1}(s)E_3'(s)$ needs not be a gradient. Even if it were a gradient, then the corresponding energy functional will be different from E_3 when $M(s)$ is not the identity. Hence, in general, the limit of eq. (51) at infinity may be different from a stationary point or even a minimizer of the energy E. This does not mean that we have lost optimality. As we mentioned in the previous section, the formulation of the energy E is not the only way to introduce the regularization term $\|\nabla s\|^2 + \kappa\|s\|^2$. Our method preserves the location of the minima of each term of the energy E. What changes is the tradeoff between these minima. Indeed, notice that for $\alpha = 0$ we obtain the original metric flow in the data term, and, similarly, if we remove the data term, then we obtain a metric flow that converges to the minima of $\|\nabla s\|^2 + \kappa\|s\|^2$. By leaving the realm of energy minimization, we do not automatically have existence, uniqueness, and stationarity of a solution of our modified gradient flow, but we need to formally prove them. To this purpose we devote Appendix II.

VIII. EXPERIMENTS

To illustrate the algorithm above and validate it empirically, we test it on a number of synthetic and real images. The estimated depth maps are shown together with the ground truth in the case of synthetic images, while in the case of real images validation is performed qualitatively by visual inspection. We provide the performance of the algorithm on a number of examples and for various levels of Gaussian noise. In the case of real data, we visually compare our algorithm to an existing method.

A. Experiments with synthetic images

In this section we test the proposed algorithm to highlight the three contributions in this manuscript. First, we show
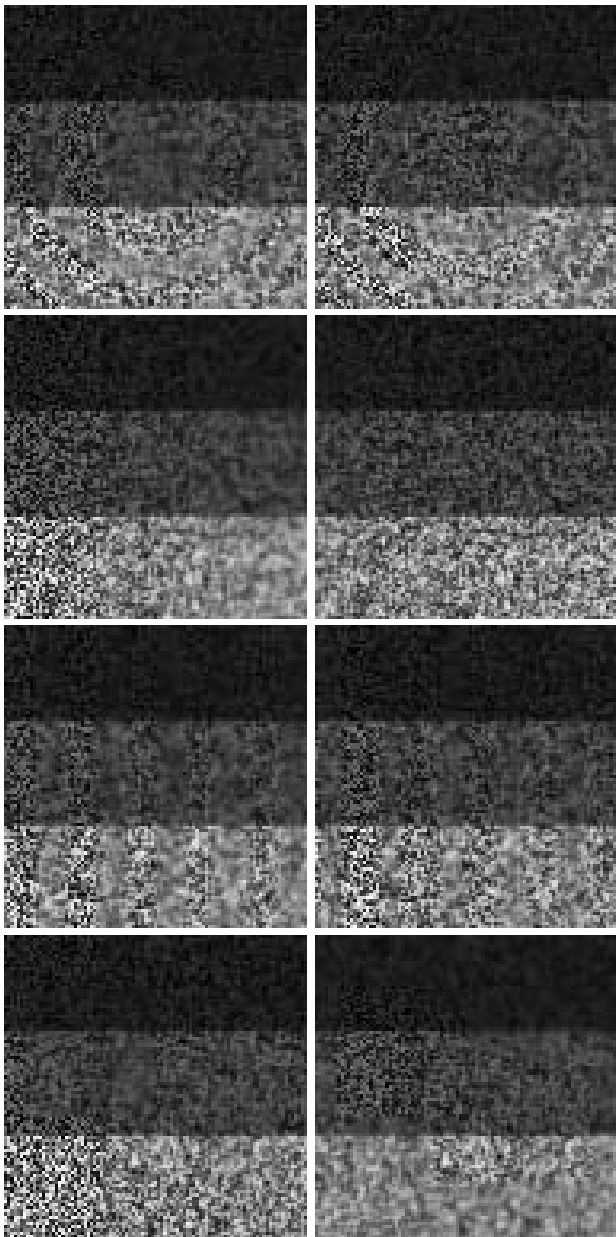
Fig. 3

INPUT IMAGES FROM THE SYNTHETIC DATA SETS. FROM TOP TO BOTTOM: "WAVE" DATA SET, "SLOPE" DATA SET, "SIN" DATA SET, AND "BOX" DATA SET. THE LEFT COLUMN SHOWS THE FAR-FOCUSED ORIGINAL IMAGES AND THE THIRD ROW SHOWS THE NEAR-FOCUSED ORIGINAL IMAGES.

that the method can handle smooth surfaces as well as sharp boundaries; second, we show that using preconditioning in the algorithm is essential to obtain the correct result; third, we show that the algorithm is very robust to noise. In order to do so, we evaluate the algorithm on the following shapes: A wave, a slope, a sinusoid and a scene made of equifocal planes at different depths (called "Box" data set). For each of these shapes we show the results obtained with or without preconditioning, and for 5 levels of additive Gaussian noise:

0%, 1%, 5%, 10%, and 20% of the radiance magnitude. The effect of preconditioning is illustrated by varying the intensity of the radiance and its content in frequency; we choose radiances with 3 different intensity levels along the vertical axis and 3 different texture sharpness levels along the horizontal axis (see Figure 3). In these data sets the camera brings in focus the plane at $520\ mm$ in the near-focused input image and the plane at $850\ mm$ in the far-focused input images, with focal length $12\ mm$, F-number 2 and $\gamma = 1.5 \cdot 10^4\quad pixel^2/mm^2$. In Figure 4 we show experiments conducted on a wave, a slope, a sinusoid and a piecewise constant depth map. For each data set on the leftmost column we show the true depth map in grayscale intensities (light intensities correspond to points close to the camera, while dark intensities correspond to points far from the camera) and a mesh plot of the true depth map. From the second to the fourth column we show results obtained with the following 3 levels of additive noise: 0%, 5%, and 20% of the radiance magnitude. The first and third row of each figure shows the estimated depth map and a mesh of the estimated depth map computed with preconditioning. The second and the fourth rows display the same results as above, but obtained without preconditioning. Notice that in the case without preconditioning there is a strong dependency of the output on the intensity of the input images (see Figure 3). Since we are made available the ground truth, we can also compute the discrepancy between the true and the estimated depth map. We compute two types of discrepancies:

$$
\begin{aligned}
ERR_0 &= \sqrt{E\left[(s - \hat{s})^2\right]} \\
ERR_1 &= \sqrt{E\left[\left(\frac{\hat{s}}{s} - 1\right)^2\right]}
\end{aligned}
\tag{52}
$$

where s is the true depth map and $\hat{s}$ is the estimated depth map. $E[\cdot]$ denotes the average over[7] Ω. The first one, ERR_0, measures the absolute error in meters (m); the second, ERR_1, measures the relative error and it is therefore unitless. In Figure 5 we show the absolute and relative errors of the estimated depth maps of the 4 data sets and for all 5 noise levels. Notice that when preconditioning is not used in the algorithm, the estimated depth map depends on the intensities of the radiance, and, in general, yields a poorer reconstruction. The improvement of the estimate when using preconditioning depends on the level of noise: Without noise, the estimation error is halved; with the highest level of noise (20%), the estimation error is reduced to about $2/3$ of the un-preconditioned error.

B. Experiments with real images

In the case of real data we run 3 experiments. In the experiment with the white sponge and the experiment with the statues we have the same camera settings as in the case of the synthetic data apart from γ that is found to be about $3 \cdot 10^4\quad pixel^2/mm^2$. We used a $35mm$ NIKON NIKKOR

---

[7]In the experiments, the average does not include the boundary of Ω, $\partial\Omega$, which is 3 pixels wide.
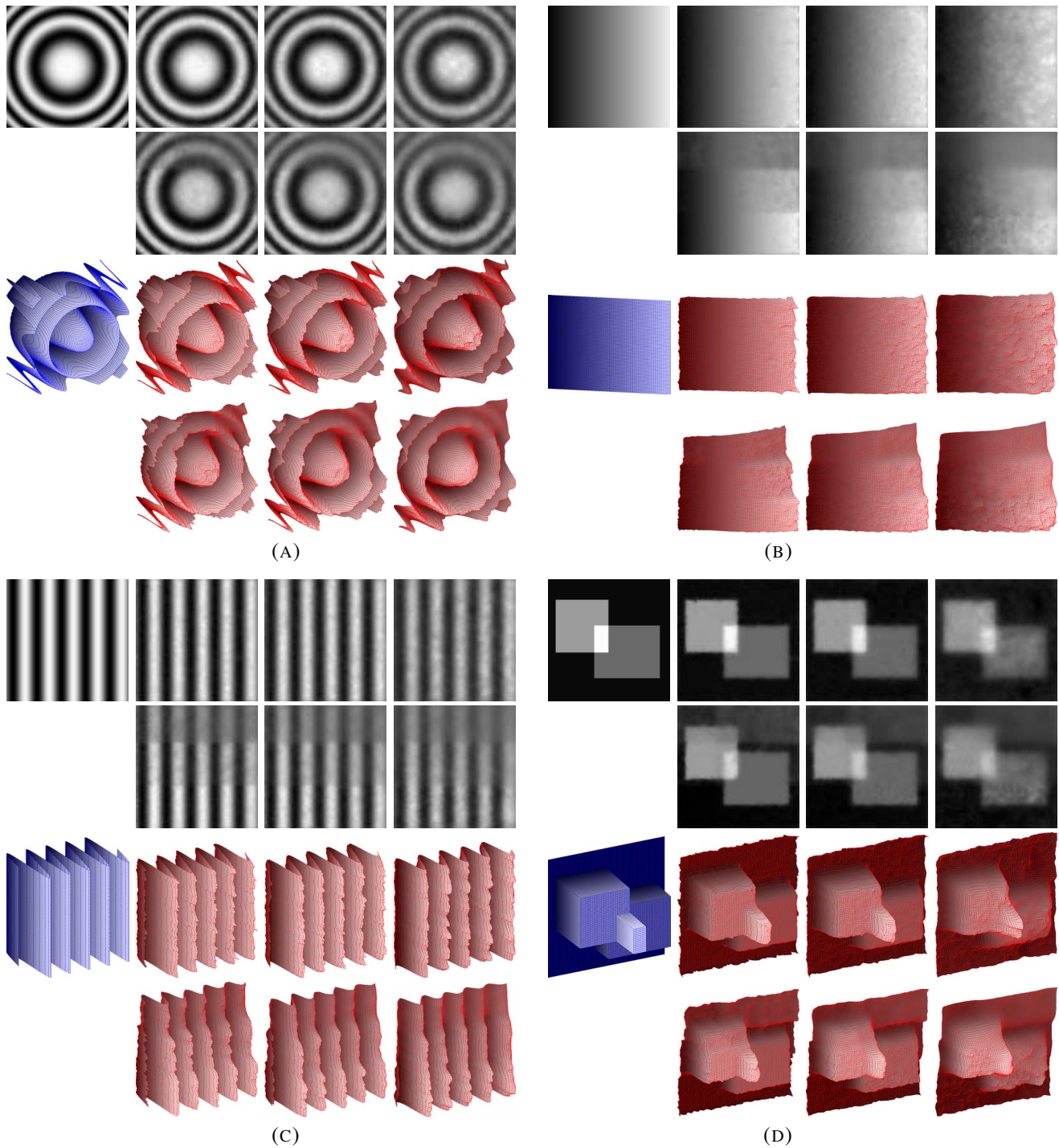
Fig. 4

(A) "WAVE," (B) "SLOPE," (C) "SIN," AND (D) "BOX" DATA SETS. THE SAME DESCRIPTION APPLIES TO EACH DATA SET. LEFTMOST COLUMN: TRUE DEPTH MAP IN GRAYSCALE AND AS A MESH. SECOND TO FOURTH COLUMNS: RESULTS WITH INCREASING NOISE. FIRST AND THIRD ROWS: ESTIMATED DEPTH MAP AND MESH WITH PRECONDITIONING. REMAINING ROWS: AS ABOVE WITHOUT PRECONDITIONING.

lens with F-number 4 and an 8-bit monochromatic camera. In the case of the cup and toast data set, the camera settings are explained in [31]. In the first experiment (first two columns in Figure 6), the scene is composed of a dark paper box at the bottom, a grey cylinder, and a small white sponge on the

top of the paper box (see first two columns of Figure 6); these images have been chosen purposefully to challenge any global optimization method for shape from defocus. Indeed, the three objects have very different graylevel intensities and type of texture. The second experiment is the cup and toast
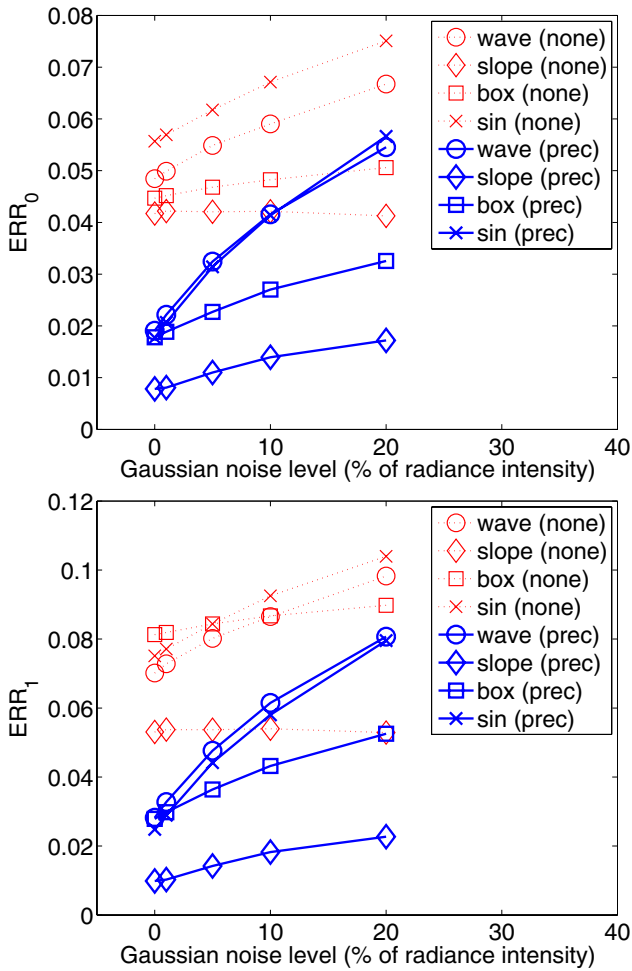
Fig. 5

ERROR PLOTS. ABSOLUTE (ERR_0 - IN METERS) AND RELATIVE (ERR_1) DISCREPANCIES BETWEEN GROUND TRUTH AND ESTIMATED DEPTH MAPS OF THE 4 SYNTHETIC DATA SETS FOR 5 NOISE LEVELS.

data set. This experiment has been done for comparisons with the results of [31]. Notice that we achieve very similar results. However, [31] cannot easily incorporate regularization. The third experiment is the dancing people data set. Notice that since there is no large variation in the intensities of the input images, there is no effect in using preconditioning. In Figure 6, second row, we show the estimated partitions Ω_+. Each pair of images is the result of the algorithm with preconditioning (left image) and without preconditioning (right image). In the third row we show the estimated depth maps in graylevel intensity scale. Each pair of images is the result of the algorithm with preconditioning (left image) and without preconditioning (right image). In the fourth row of the same figure we show mesh plots of the depth maps estimated with preconditioning. Finally, in the bottom row we show the mesh plot of the depth maps estimated without preconditioning. Notice how the non-preconditioned algorithm yields over-smoothed results in regions where the image intensities are dark.

IX. CONCLUSIONS

Using the notion of relative blur we have been able to formulate the problem of shape from defocus as a forward diffusion process, that is numerically stable and yields a global estimate of the depth map of a scene.

Our approach marries the benefits of local approaches (separation of shape from radiance, hence reduced computational complexity) with those of global ones (optimal estimation of shape, reduced sensitivity to noise), while having none of the limitations (windowing effects of local approaches, radiance regularization and computational complexity of global ones).

The resulting algorithms are implemented by solving numerically partial differential equations using stable schemes, and we have provided some analysis on the behavior of the gradient flow. We have provided experiments with real as well as synthetic images, which show the effectiveness of our approach.

APPENDIX I
COMPUTATION OF THE GRADIENTS

To compute the gradients (42) we need to derive the first-order variation [5] of the cost functional (38). For a general cost functional the first-order variation is formulated in terms of a directional derivative (in direction φ), defined via

$$E'(s)\varphi = \lim_{\epsilon \to 0} \frac{1}{\epsilon}(E(s + \epsilon\varphi) - E(s)).$$

In the case of

$$E_3(s) = \int_\Omega \|\nabla s(\mathbf{y})\|^2 + \kappa s(\mathbf{y})^2 d\mathbf{y} \qquad (53)$$

with $\kappa > 0$, the computation amounts to:

$$
\begin{aligned}
E'_3(s)\varphi &= \lim_{\epsilon \to 0} \frac{1}{\epsilon}(E_3(s + \epsilon\varphi) - E_3(s)) \\
&= \lim_{\epsilon \to 0} \frac{1}{\epsilon}\left\{ \int_\Omega \left[\|\nabla(s(\mathbf{y}) + \epsilon\varphi(\mathbf{y}))\|^2 + \kappa(s(\mathbf{y}) \right.\right. \\
&\qquad \left.\left. + \epsilon\varphi(\mathbf{y}))^2\right] - \left[\|\nabla s(\mathbf{y})\|^2 + \kappa s(\mathbf{y})^2\right] d\mathbf{y} \right\} \\
&= \lim_{\epsilon \to 0} \int_\Omega 2\nabla s(\mathbf{y}) \cdot \nabla\varphi(\mathbf{y}) + |\epsilon| \|\nabla\varphi(\mathbf{y})\|^2 \\
&\qquad + 2\kappa s(\mathbf{y})\varphi(\mathbf{y}) + \epsilon\varphi(\mathbf{y})^2 d\mathbf{y} \\
&= 2\int_\Omega \nabla s(\mathbf{y}) \cdot \nabla\varphi(\mathbf{y}) + \kappa s(\mathbf{y})\varphi(\mathbf{y}) d\mathbf{y}
\end{aligned}
$$

for an arbitrary $\varphi \in H^1$, where $H^1 \doteq \{\varphi \in L^2 \mid \nabla\varphi \in L^2\}$, such that $\varphi(\mathbf{y}) = 0, \quad \forall \mathbf{y} \in \partial\Omega$. Integration by parts yields

$$
\begin{aligned}
E'_3(s)\varphi &= 2\int_\Omega \left[-\nabla \cdot \nabla s(\mathbf{y}) + \kappa s(\mathbf{y}) \right]\varphi(\mathbf{y}) d\mathbf{y} \\
&\quad + \int_{\partial\Omega} (\nabla s \cdot \mathbf{n})(\mathbf{y})\varphi(\mathbf{y}) ds(\mathbf{y}),
\end{aligned}
\qquad (54)
$$

where $\mathbf{n}$ is the normal to the boundary $\partial\Omega$; then, because $\varphi(\mathbf{y}) = 0, \forall \mathbf{y} \in \partial\Omega$, the second term vanishes and we are left with

$$E'_3(s)(\mathbf{y}) = -2\triangle s(\mathbf{y}) + 2\kappa s(\mathbf{y}) \qquad (55)$$

for $\mathbf{y} \in \Omega$. The derivatives relative to E_1 and E_2 are similar to each other, so we will derive only the one in E_1. The first
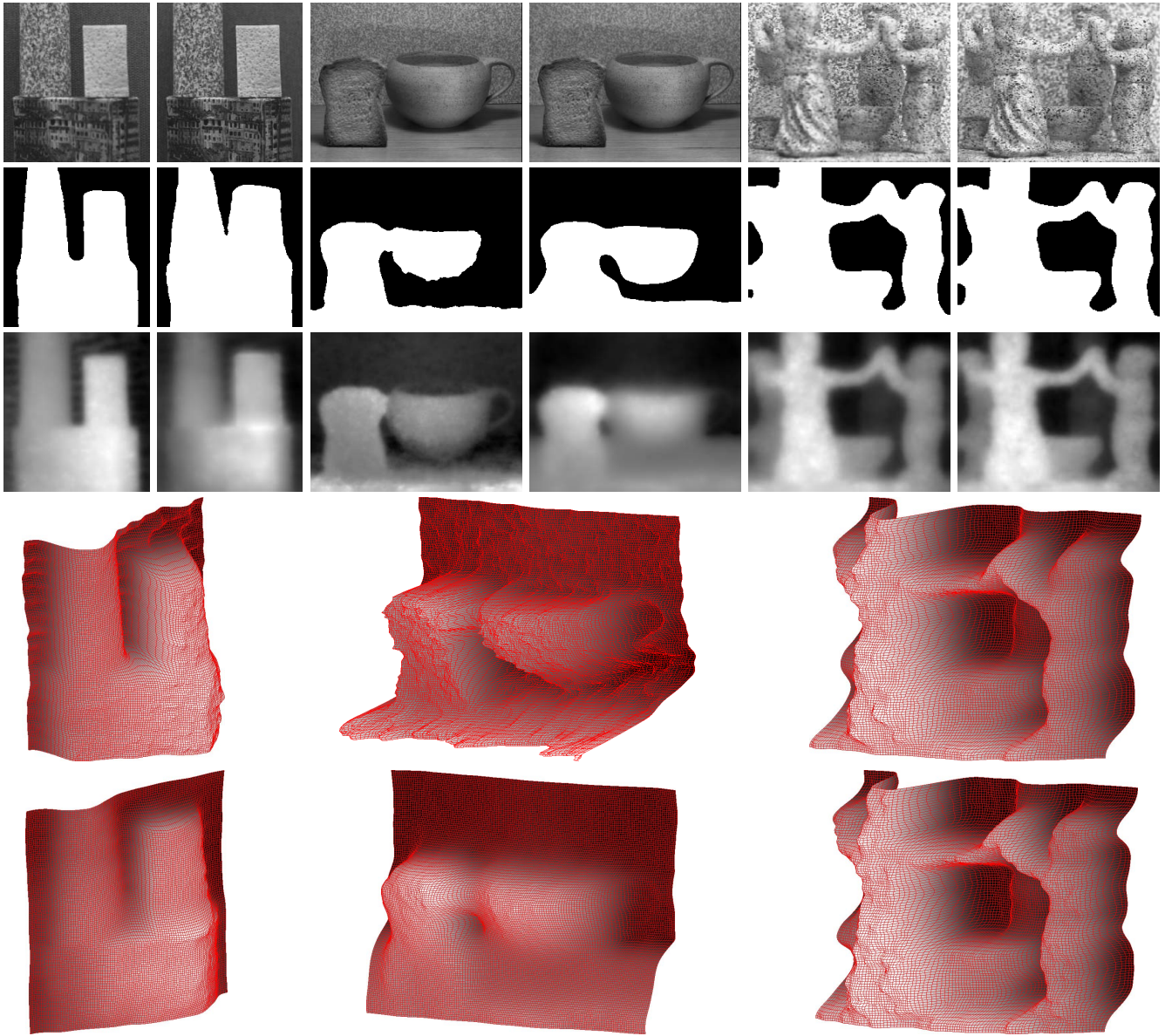
Fig. 6

REAL DATA SETS. TOP ROW: INPUT IMAGES; EACH PAIR IS A NEAR-FOCUSED AND A FAR-FOCUSED IMAGE. SECOND ROW: ESTIMATED PARTITIONS Ω_+.
EACH PAIR OF IMAGES IS THE RESULT OF THE ALGORITHM WITH PRECONDITIONING (LEFT IMAGE) AND WITHOUT PRECONDITIONING (RIGHT IMAGE).
THIRD ROW: ESTIMATED DEPTH MAPS IN GRAY-LEVEL INTENSITY SCALE. EACH PAIR OF IMAGES IS THE RESULT OF THE ALGORITHM WITH
PRECONDITIONING (LEFT IMAGE) AND WITHOUT PRECONDITIONING (RIGHT IMAGE). FOURTH ROW: ESTIMATED DEPTH MAPS WITH PRECONDITIONING
SHOWN AS A MESH PLOT. BOTTOM ROW: ESTIMATED DEPTH MAPS WITHOUT PRECONDITIONING SHOWN AS A MESH PLOT.

difficulty that we meet in computing the functional derivative of E_1 is the fact that the function u is not given in explicit form, but as the solution of a PDE

$$\begin{cases} \dot{u}(\mathbf{y},t) = \nabla \cdot (c(\mathbf{y})\nabla u(\mathbf{y},t)) & t \in [0,\infty) \\ u(\mathbf{y},0) = I_1(\mathbf{y}) & \forall \mathbf{y} \in \Omega_+. \end{cases} \quad (56)$$

The first-order variation of the term

$$\begin{aligned} \tilde{E}_1(c) &\doteq \int_{\Omega_+} (u(\mathbf{y},\Delta t) - I_2(\mathbf{y}))^2 d\mathbf{y} \\ &= \int (u(\mathbf{y},\Delta t) - I_2(\mathbf{y}))^2 H(c(\mathbf{y})) d\mathbf{y} \end{aligned} \quad (57)$$

is

$$\begin{aligned} \tilde{E}_1'(c)\varphi &= 2 \int_{\Omega_+} (u(\mathbf{y},\Delta t) - I_2(\mathbf{y})) u'_\varphi(\mathbf{y},\Delta t) d\mathbf{y} \\ &+ \int \delta(c(\mathbf{y})) \left(u(\mathbf{y},\Delta t) - I_2(\mathbf{y})\right)^2 \varphi(\mathbf{y}) d\mathbf{y} \end{aligned}$$
$$(58)$$

for an arbitrary function $\varphi : \Omega_+ \to \mathbb{R}$ such that $\varphi(\mathbf{y}) = 0$, $\forall \mathbf{y} \in \partial\Omega_+$. The variation

$$u'_\varphi(\mathbf{y},t) \doteq \lim_{\epsilon \to 0} \frac{u(\mathbf{y},t)|_{c+\epsilon\varphi} - u(\mathbf{y},t)|_c}{\epsilon}, \quad (59)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

IEEE TRANSACTION OF PATTERN RECOGNITION AND MACHINE INTELLIGENCE, VOL. , NO. , MONTH YEAR

12

where $u(\mathbf{y},t)\big|_{c+\epsilon\varphi}$ is the solution of eq. (56) with diffusion $c + \epsilon\varphi$ instead of c, is such that u'_φ satisfies

$$\begin{cases} \dot{u}'_\varphi(\mathbf{y},t) = \nabla\cdot\big(c(\mathbf{y})\nabla u'_\varphi(\mathbf{y},t) + \varphi(\mathbf{y})\nabla u(\mathbf{y},t)\big) & t \in (0,\infty) \\ u'_\varphi(\mathbf{y},0) = 0 & \forall \mathbf{y} \in \Omega_+. \end{cases}$$
(60)

Let $z : \Omega \times [0,\infty) \to \mathbb{R}$ be a function such that

$$z(\mathbf{y},\Delta t) = u(\mathbf{y},\Delta t) - I_2(\mathbf{y}).$$
(61)

Then, by substituting z in eq. (58), we obtain

$$\begin{aligned} \tilde{E}'_1(c)\varphi &= 2\int_{\Omega_+} z(\mathbf{y},\Delta t)u'_\varphi(\mathbf{y},\Delta t)d\mathbf{y} \\ &+ \int \delta(c(\mathbf{y}))\left(u(\mathbf{y},\Delta t) - I_2(\mathbf{y})\right)^2 \varphi(\mathbf{y})d\mathbf{y} \\ &= 2\int_{\Omega_+} z(\mathbf{y},0)u'_\varphi(\mathbf{y},0)d\mathbf{y} \\ &+ 2\int_0^{\Delta t}\!\!\int_{\Omega_+} \dot{z}(\mathbf{y},t)u'_\varphi(\mathbf{y},t) + z(\mathbf{y},t)\dot{u}_\varphi(\mathbf{y},t)d\mathbf{y}dt \\ &+ \int \delta(c(\mathbf{y}))\left(u(\mathbf{y},\Delta t) - I_2(\mathbf{y})\right)^2 \varphi(\mathbf{y})d\mathbf{y} \end{aligned}$$
(62)

which becomes

$$\begin{aligned} \tilde{E}'_1(c)\varphi &= 2\int_0^{\Delta t}\!\!\int_{\Omega_+} \Big(\dot{z}(\mathbf{y},t) + \nabla\cdot(c(\mathbf{y})\nabla z(\mathbf{y},t))\Big)u'_\varphi(\mathbf{y},t) \\ &- \varphi(\mathbf{y})\nabla z(\mathbf{y},t)\cdot\nabla u(\mathbf{y},t)d\mathbf{y}dt \\ &+ \int \delta(c(\mathbf{y}))\left(u(\mathbf{y},\Delta t) - I_2(\mathbf{y})\right)^2 \varphi(\mathbf{y})d\mathbf{y} \end{aligned}$$
(63)

after integrating by parts twice and using the fact that both φ and the diffusion coefficient c vanish at the boundary of Ω_+, and noticing that the initial conditions of u'_φ are $u'_\varphi(\mathbf{y},0) = 0$. To simplify the above equation we choose the adjoint solution z in the interval $(0,\Delta t)$ such that

$$\begin{cases} \dot{z}(\mathbf{y},t) = -\nabla\cdot(c(\mathbf{y})\nabla z(\mathbf{y},t)) & t \in (0,\Delta t] \\ z(\mathbf{y},\Delta t) = u(\mathbf{y},\Delta t) - I_2(\mathbf{y}) & \forall \mathbf{y} \in \Omega_+. \end{cases}$$
(64)

The substitution of z in eq. (63) yields

$$\begin{aligned} \tilde{E}'_1(c)\varphi &= -2\int_0^{\Delta t}\!\!\int_{\Omega_+} \varphi(\mathbf{y})\nabla z(\mathbf{y},t)\cdot\nabla u(\mathbf{y},t)d\mathbf{y}dt \\ &+ \int \delta(c(\mathbf{y}))\left(u(\mathbf{y},\Delta t) - I_2(\mathbf{y})\right)^2 \varphi(\mathbf{y})d\mathbf{y}. \end{aligned}$$
(65)

Now, define $w(\mathbf{y},t) = z(\mathbf{y},\Delta t - t)$, then we obtain the *adjoint equation* (also notice that this is the definition of eq. (44) and eq. (45))

$$\begin{cases} \dot{w}(\mathbf{y},t) = \nabla\cdot(c(\mathbf{y})\nabla w(\mathbf{y},t)) & t \in (0,\Delta t] \\ w(\mathbf{y},0) = u(\mathbf{y},\Delta t) - I_2(\mathbf{y}) & \forall \mathbf{y} \in \Omega_+ \end{cases}$$
(66)

which, substituted in the expression of $\tilde{E}'_1$, gives

$$\begin{aligned} \tilde{E}'_1(c)\varphi &= -2\int_0^{\Delta t}\!\!\int_{\Omega_+} \varphi(\mathbf{y})\nabla w(\mathbf{y}\Delta t - t)\cdot\nabla u(\mathbf{y},t)d\mathbf{y}dt \\ &+ \int \delta(c(\mathbf{y}))\left(u(\mathbf{y},\Delta t) - I_2(\mathbf{y})\right)^2 \varphi(\mathbf{y})d\mathbf{y}. \end{aligned}$$
(67)

Now, the gradient of $\tilde{E}_1$ is easily determined as

$$\begin{aligned} \tilde{E}'_1(c) &= -2H(c(\mathbf{y}))\int_0^{\Delta t} \nabla w(\mathbf{y},\Delta t - t)\cdot\nabla u(\mathbf{y},t)dt \\ &+ \delta(c(\mathbf{y}))\left(u(\mathbf{y},\Delta t) - I_2(\mathbf{y})\right)^2. \end{aligned}$$
(68)

Both functions u and w can be computed by *simulating* the respective models, and both models involve only forward diffusions. Finally, the gradient of E_1 can be computed by the chain rule as $E'_1(s) = \tilde{E}'_1(c)c'(s)$.

APPENDIX II
PROPERTIES OF THE MODIFIED GRADIENT FLOW

In the following we prove uniqueness and existence of a stationary solution obtained with the modified gradient flow (51). For the analysis of the flow we assume that H and δ are smoothed (i.e., at least continuously differentiable) approximations of the Heaviside function and Dirac delta, so that all functionals are differentiable. We start by showing the existence and uniqueness of a solution and then prove its stationarity by using Schauder's fixed point theorem [48]. Let $\triangle_\kappa$ be an operator defined by $\triangle_\kappa \doteq \triangle - \kappa I_d$, where I_d is the identity operator and κ was defined in eq. (35); then, $-\triangle_\kappa$ is a positive definite operator on the Sobolev space $H^1 \doteq \{u \in L^2 \mid \nabla u \in L^2\}$. Moreover, let us approximate $c'(s)$ by the following formula

$$c'(s) = \frac{\gamma^2 D^2(v_2 - v_1)}{4\max\{\hat{\epsilon},s\}^2\Delta t}\left[(v_2+v_1)\left(\frac{1}{F} - \frac{1}{\max\{\hat{\epsilon},s\}}\right) - 1\right],$$
(69)

with small $\hat{\epsilon} > 0$, in order to avoid division by zero. In particular, we obtain the uniform bound

$$|c'(s)| \le \frac{\gamma^2 D^2 |v_2 - v_1|}{4\hat{\epsilon}^2\Delta t}\left[(v_2 + v_1)\left(\frac{1}{F} + \frac{1}{\hat{\epsilon}}\right) + 1\right],$$
(70)

We are now ready to verify the existence of the preconditioned flow (51). By defining the operator

$$\mathcal{R}(s) = -M(s)(E'_1(s) + E'_2(s)),$$
(71)

the flow can be rewritten as

$$\frac{\partial s}{\partial \tau} = 2\alpha\triangle_\kappa s + \mathcal{R}(s),$$
(72)

that is, we obtain a heat equation with a nonlocal source term (recall that $\triangle_\kappa \doteq \triangle - \kappa I_d$). One can verify that $\mathcal{R}$ maps L^∞ to L^∞ and is even Lipschitz on these spaces. Hence, by standard results for parabolic equations (cf. [49]) we obtain the existence and uniqueness of a bounded solution, which we summarize in the following

Proposition 2: Let I_1 and I_2 be positive and bounded, and let $s(0) \in L^\infty$ be a given initial value. Then there exists a unique solution s of eq. (51) such that $s(\tau) \in L^\infty$ for almost any $\tau \in \mathbb{R}^+$.

Since eq. (51) is possibly no gradient flow of an energy functional, the existence of a stationary solution (the limit $\tau \to \infty$ of the evolution) of our regularized solution does not follow from standard variational arguments. Therefore, in the following we provide a different argument. The equation that defines a stationary solution is given by $\frac{\partial s}{\partial \tau} = 0$, i.e.,

$$E'_3(s) = -2\alpha\triangle_\kappa s = \mathcal{R}(s) = -M(s)(E'_1(s) + E'_2(s)). \quad (73)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

IEEE TRANSACTION OF PATTERN RECOGNITION AND MACHINE INTELLIGENCE, VOL. , NO. , MONTH YEAR 13

We then prove the existence of a stationary solution by a fixed point argument, i.e., we show that

$$s = \mathcal{F}(s) \doteq -\frac{1}{2\alpha}\triangle_\kappa^{-1}\mathcal{R}(s). \tag{74}$$

satisfies the assumptions of Schauder's fixed point theorem. In order to do so, we prove that $\mathcal{F}$ maps L^∞ into a compact set. First notice that $\mathcal{R}$ is a continuous operator on L^∞. Then, notice that the inverse Laplacian $-\triangle_\kappa^{-1}$ is a continuous and compact operator on L^∞ (which is due to the maximum principle for elliptic differential operators), and one can verify that also the concatenation is continuous on $L^\infty \cap H^1$. Moreover, we have

$$\left| \left(M(s)\tilde{E}_1'(c(s)) \right)(\mathbf{y}) \right| = \left| H(c(\mathbf{y})) \frac{(u(\mathbf{y},\Delta t)-I_2(\mathbf{y}))u'(\mathbf{y},\Delta t)}{\frac{\epsilon}{2}+I_2(\mathbf{y})|u'(\mathbf{y},\Delta t)|} \right.$$
$$\left. + \frac{1}{2}\delta(c(\mathbf{y})) \frac{(u(\mathbf{y},\Delta t)-I_2(\mathbf{y}))^2}{\frac{\epsilon}{2}+I_2(\mathbf{y})|u'(\mathbf{y},\Delta t)|} \right|$$
$$\leq \eta_H \left(\frac{|u(\mathbf{y},\Delta t)|}{\epsilon}+1 \right)$$
$$+ \eta_\delta \frac{(u(\mathbf{y},\Delta t)-I_2(\mathbf{y}))^2}{\epsilon}, \tag{75}$$

where the constants η_H and η_δ are the maximal values of the smoothed Heaviside function and Dirac delta, respectively, and ϵ is a small positive value. Now finally, let η_I be an upper bound for I_1, I_2, and u_D. Then, a maximum principle for parabolic equations (cf. [50]) implies $|u(\mathbf{y},\Delta t)| \leq \eta_I$ for all $y \in \Omega$. Hence, we obtain

$$\sup_{\mathbf{y}\in\Omega} \left| \left(M(s)\tilde{E}_1'(c(s)) \right)(\mathbf{y}) \right| \leq \eta_H \left(\frac{\eta_I}{\epsilon}+1 \right) + \eta_\delta \frac{4\eta_I^2}{\epsilon}, \tag{76}$$

i.e., an estimate of the supremum norm of $M(s)\tilde{E}_1'(c(s))$ independent of s. In an analogous way, we can estimate the norm of $M(s)\tilde{E}_2'(c(s))$. Moreover, since we also have obtained a uniform bound for $c'(s)$ above, we may conclude that the operator

$$\mathcal{R}(s) = M(s)\left[\tilde{E}_1'(c(s))c'(s) + \tilde{E}_2'(c(s))c'(s) \right] \tag{77}$$

maps into a bounded set in L^∞. Finally, since $\triangle_\kappa^{-1}$ is compact, we can conclude that the operator $\mathcal{F}$ maps into a compact set in L^∞ (and in particular this compact set into itself). Thus, by Schauder's fixed point theorem (cf. [48]) we conclude the existence of a solution of eq. (74), which is equivalent to eq. (73). We summarize this result in the following Proposition:

Proposition 3: Let I_1 and I_2 be positive and bounded, and let $\kappa > 0$. Then there exists a stationary solution $s \in L^\infty \cap H^1$ of eq. (73).

Proposition 3 guarantees the existence of a bounded stationary solution for the preconditioned flow. Although the stationary problem does not correspond to the minimization of some energy, we have shown that the preconditioned approach is a regularization of the original problem (34).

Acknowledgment

References

[1] P. Favaro, A. Mennucci, and S. Soatto, "Observing shape from defocused images," *International Journal of Computer Vision*, vol. 52, no. 1, pp. 25–43, 2003.

[2] J. Ens and P. Lawrence, "An investigation of methods for determining depth from focus," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 97–108, 1993.

[3] N. Nair and C. Stewart, "Robust focus ranging," *Proc. IEEE Computer Vision and Pattern Recognition*, pp. 309–314, 1992.

[4] P. Favaro, S. Soatto, L. Vese, and S. Osher, "3-d shape from anisotropic diffusion," *Proc. IEEE Computer Vision and Pattern Recognition*, vol. I, pp. 179–186, 2003.

[5] D. Luenberger, *Optimization by vector space methods.* Wiley, 1968.

[6] P. Favaro and S. Soatto, *3-d shape reconstruction and image restoration: exploiting defocus and motion-blur.* Springer-Verlag, 2006.

[7] S. Chaudhuri and A. Rajagopalan, *Depth from defocus: a real aperture imaging approach.* Springer Verlag, 1999.

[8] A. Rajagopalan and S. Chaudhuri, "Optimal selection of camera parameters for recovery of depth from defocused images," *Proc. IEEE Computer Vision and Pattern Recognition*, pp. 219–224, 1997.

[9] ——, "Optimal recovery of depth from defocused images using an mrf model," *Proc. IEEE International Conference on Computer Vision*, pp. 1047–1052, 1998.

[10] P. Favaro, A. Mennucci, and S. Soatto, "Observing shape from defocused images," *International Journal of Computer Vision*, vol. 52, no. 1, pp. 25–43, 2003.

[11] P. Favaro and S. Soatto, "Learning shape from defocus," *Proc. European Conference on Computer Vision*, vol. 2, pp. 735–745, 2002.

[12] ——, "Shape and radiance estimation from the information divergence of blurred images," *Proc. IEEE Computer Vision and Pattern Recognition*, pp. 755–68, 2000.

[13] S. Nayar and Y. Nakagawa, "Shape from focus: An effective approach for rough surfaces," *IEEE International Conference on Robotics and Automation*, pp. 218–225, 1990.

[14] M. Noguchi and S. Nayar, "Microscopic shape from focus using active illumination," *International Conference on Pattern Recognition*, pp. 147–152, 1994.

[15] A. Pentland, "A new sense for depth of field," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 523–531, 1987.

[16] A. Pentland, S. Scherock, T. Darrell, and B. Girod, "Simple range cameras based on focal error," *Journal of the Optical Society of America A*, vol. 11, no. 11, pp. 2925–2934, 1994.

[17] S. Soatto and P. Favaro, "A geometric approach to blind deconvolution with application to shape from defocus," *Proc. IEEE Computer Vision and Pattern Recognition*, vol. 2, pp. 10–17, 2000.

[18] M. Watanabe and S. Nayar, "Minimal operator set for passive depth from defocus," *Proc. IEEE Computer Vision and Pattern Recognition*, pp. 431–438, 1996.

[19] A. Rajagopalan, S. Chaudhuri, and U. Mudenagudi, "Depth estimation and image restoration using defocused stereo pairs." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1521–1525, 2004.

[20] M. Bertero and P. Boccacci, *Introduction to inverse problems in imaging.* Institute of Physics Publications, 1998.

[21] M. Gokstorp, "Computing depth from out-of-focus blur using a local frequency representation," *International Conference on Pattern Recognition*, pp. 153–158, 1994.

[22] Y. Schechner and N. Kiryati, "The optimal axial interval in estimating depth from defocus," *Proc. of the International Conference of Computer Vision*, pp. 843–848, 1993.

[23] Y. Xiong and S. Shafer, "Moment filters for high precision computation of focus and stereo," *Proc. of International Conference on Intelligent Robots and Systems*, pp. 108–113, 1995.

[24] N. Asada, H. Fujiwara, and T. Matsuyama, "Edge and depth from focus," *International Journal of Computer Vision*, vol. 26, no. 2, pp. 153–163, 1998.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

IEEE TRANSACTION OF PATTERN RECOGNITION AND MACHINE INTELLIGENCE, VOL. , NO. , MONTH YEAR   14

[25] J. Marshall, C. Burbeck, and D. Ariely, "Occlusion edge blur: a cue to relative visual depth," *Journal of the Optical Society of America A*, vol. 13, pp. 681–688, 1996.

[26] M. Subbarao and N. Gurumoorthy, "Depth recovery from blurred edges," *Proc. IEEE Computer Vision and Pattern Recognition*, pp. 498–503, 1988.

[27] G. Schneider, B. Heit, J. Honig, and J. Bremont, "Monocular depth perception by evaluation of the blur in defocused images," *Proc. IEEE International Conference on Image Processing*, vol. 2, pp. 116–119, 1994.

[28] M. Subbarao and G. Surya, "Depth from defocus: a spatial domain approach," *International Journal of Computer Vision*, vol. 13, pp. 271–294, 1994.

[29] B. Girod and S. Scherock, "Depth from focus of structured light," in *Proc. SPIE: Optics, Illumination, and Image Sensing for Machine Vision*, vol. 1194, 1989, pp. 209–15.

[30] S. Nayar, M. Watanabe, and M. Noguchi, "Real-time focus range sensor," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1186–1198, 1996.

[31] M. Watanabe and S. Nayar, "Rational filters for passive depth from defocus," *International Journal of Computer Vision*, vol. 27, no. 3, 1998.

[32] Y. Xiong and S. Shafer, "Depth from focusing and defocusing," *Proc. IEEE Computer Vision and Pattern Recognition*, pp. 68–73, 1993.

[33] D. Ziou and F. Deschenes, "Depth from defocus estimation in spatial domain," *Computer Vision and Image Understanding*, vol. 81, no. 2, pp. 143–165, 2001.

[34] J. Koenderink, "The structure of images," *Biological Cybernetics*, vol. 50, pp. 363–370, 1984.

[35] B. Fischl and E. Schwartz, "Learning an integral equation approximation to nonlinear anisotropic diffusion in image processing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 342–52, 1997.

[36] S. Osher and S. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulation," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1998.

[37] M. Black, G. Sapiro, D. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 421–32, 1998.

[38] P. Perona and J. Malik, "Scale space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–39, 1990.

[39] J. Weickert, *Anisotropic diffusion in image processing*. B.G.Teubner Stuttgart, 1998.

[40] D. Forsyth and J. Ponce, *Computer Vision: a modern approach*. Prentice Hall, 2002.

[41] M. Subbarao, "Parallel depth recovery by changing camera parameters," *Proc. IEEE International Conference on Computer Vision*, pp. 149–155, 1988.

[42] G. Gilardi, "Analisi due," in *McGraw-Hill*, 1992.

[43] D. Tschumperle and R. Deriche, "Vector-valued image regularization with pde's: A common framework for different applications," *Proc. IEEE Computer Vision and Pattern Recognition*, pp. 651–656, 2003.

[44] H. Hopkins, "The frequency response of a defocused optical system," *Proc. Royal Society London Series A*, vol. 231, pp. 91–103, 1955.

[45] R. Lagnado and S. Osher, "A technique for calibrating derivative security pricing models: numerical solution of an inverse problem," *Journal of Computational Finance*, vol. 1, no. 1, pp. 13–26, 1997.

[46] G. Aubert and P. Kornprobst, *Mathematical problems in image processing: partial differential equations and the calculus of variations*. Springer Verlag, Applied Mathematical Sciences, 2001.

[47] L. Ambrosio, N. Gigli, and G. Savare, *Gradient flows in metric spaces and in the space of probability measures*. Birkhaeuser, Basel, 2005.

[48] E. Zeidler, *Nonlinear functional analysis and its applications. part 1: fixed-point theorems*. Springer New York, 1998.

[49] H. Amann, *Linear and quasilinear parabolic problems, volume I: abstract linear theory*. Birkhaeuser, 1995.

[50] M. Protter and H. Weinberger, *Maximum principles in differential equations*. Prentice-Hall, 1967.

Paolo Favaro Paolo Favaro received the D.Ing. degree from Università di Padova, Italy in 1999, and the M.Sc. and Ph.D. degree in electrical engineering from Washington University in St. Louis in 2002 and 2003 respectively. He was a postdoctoral researcher in the computer science department of the University of California, Los Angeles and subsequently in Cambridge University, UK. Dr. Favaro is now assistant professor in Heriot-Watt University, UK. His research interests are in computer vision, signal and image processing, estimation theory, sensor-based control, stochastic optimization and variational techniques. Dr. Favaro is a member of the IEEE Society.

Stefano Soatto Stefano Soatto received his Ph.D. in Control and Dynamical Systems from the California Institute of Technology in 1996; he joined UCLA in 2000 after being Assistant and then Associate Professor of Electrical and Biomedical Engineering at Washington University, and 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 his D.Ing. degree (highest honors) from the University of Padova- Italy in 1992. Dr. Soatto is the recipient of the David Marr Prize (with Y. Ma, J. Kosecka and S. Sastry of U.C. Berkeley) for work on Euclidean reconstruction and reprojection up to subgroups. He also received the Siemens Prize with the Outstanding Paper Award from the IEEE Computer Society for his work on optimal structure from motion (with R. Brockett of Harvard). He received the National Science Foundation Career Award and the Okawa Foundation Grant. He is Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) and a Member of the Editorial Board of the International Journal of Computer Vision (IJCV) and Foundations and Trends in Computer Graphics and Vision.

Martin Burger Martin Burger was born in Wels, Austria, in 1976. He studied mathematics in Linz and Milano, Italy, and received his PhD from Johannes Kepler University (JKU), Linz, Austria, in 2000. He has worked as a pre- and post-doc researcher as well as assistant professor at JKU, and was CAM assistant professor at the Department of Mathematics, University of California, Los Angeles. Since 2006 he is full professor at the Institute for Computational and Applied Mathematics, University of Münster, Germany. His research interests are image restoration and processing, inverse problems, as well as PDE models in life sciences and material sciences applications.

Stanley J. Osher Stanley Osher received the B.S. degree in mathematics from Brooklyn College in 1962, the M.S. and Ph.D. degrees in mathematics from New York University in 1964 and 1966 respectively. He has been with the University of California, Los Angeles, since 1977. He is currently a Professor in the Department of Mathematics and the Director of Special Projects in the Institute for Pure and Applied Mathematics. Dr. Osher is the founder and CEO of Level Set Systems, Inc. and has co-founded two other companies. He was an invited speaker at the 1994 International Congress of Mathematics and an ISI Original Highly Cited Researcher. He received the 2002 Computational Mechanics Award from Japan Society of Mechanical Engineers, the 2003 ICIAM Pioneer Prize and the 2005 SIAM Kleinman Prize. He was elected to the National Academy of Sciences in 2005.