Shape from Defocus via Diffusion

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

Manuscript received ; revised .

P. Favaro is with Heriot-Watt University, United Kingdom.

S. Soatto and S. J. Osher are with the University of California, Los Angeles, USA.

M. Burger is with the University of Münster, Germany.

Abstract

Defocus can be modeled as a diffusion process, and represented mathematically using the heat equation, where image blur corresponds to 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 it takes to transform the sharper image into the blurrier depends on the depth of the scene. This can be used to devise global algorithms that estimate the depth profile of the scene without recovering the de-blurred image, using only forward diffusion.

Index Terms

Shape from Defocus, Depth from Defocus, Blurring, 3D Shape Estimation, 3D Reconstruction, Diffusion, Heat Equation, Gradient Flow, Preconditioning.

I. Introduction

When imaging a scene through a lens, objects at different depths are blurred by different amounts. The wider the aperture of the lens, the smaller the range of depths where objects appear "in focus;" the farther from such a region, the more blurred objects appear. In shape from defocus (SFD), 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 some analysis on the condition 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 achieve a reliable estimate of the amount of blur despite noise, but as small as possible to guarantee that within that window the scene portrayed has constant depth. This usually results in artifacts in the reconstruction.

Global algorithms for SFD exploit the entire image, but at the cost of allowing the shape to be an (infinite-dimensional) function of the spatial location. Because details on the radiance are lost in the image formation process, formalizing SFD as an optimal inference problem results in an *ill-posed* inverse problem, that typically involves numerically unstable backward-diffusion. This problem is typically addressed by means of *regularization*, that is, by introducing well-posed approximations of the original ill-posed problem. For instance, one could search, among all possible shapes and radiances, for the smoothest ones. While assuming that the surfaces in space are (at least piecewise) smooth is somewhat reasonable, the scene radiance is definitely not smooth (i.e. blurry), and instead is characterized by sharp edges, spikes, and very complex and high-frequency patterns.

The approach we propose is novel in several ways, and achieves optimality without resorting to 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 [2]. 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, that relieves us from the need of making explicit assumptions about its regularity.

The reader who is not familiar with the calculus of variations may find Appendix B of [3] useful, in that it contains only the notions essential to the understanding of optimal SFD.

July 25, 2006

II. 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 [4]–[6].

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]–[18] and those that operate in the frequency domain [19]–[22]. Most of these approaches allow one to easily eliminate undesired unknowns (the appearance, or "radiance"). However, the techniques that are typically employed introduce artifacts in the solution, such as edge bleeding and windowing [7], [23].

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 [13], [24]–[27], others that the radiance can be locally approximated by cubic polynomials [28], or that it can be controlled by using structured light [12], [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 [13], [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 [4], 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 instead, 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 [3], which we quickly 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 [3]) is a function $I : \Omega \subset \mathbb{R}^2 \mapsto [0, \infty)$ that can be approximated via the following equation:

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

where $h : \Omega \times \mathbb{R}^2 \times [0, \infty) \mapsto [0, \infty)$ is called *point spread function* (PSF). The point spread function depends on the *blurring radius* b which in turn depends on the focus setting v and the

[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 instant [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.

depth map s. The blurring radius b determines the size of the pattern generated by a point light source at $[\frac{x^T}{v}\ 1]^T s(x)$ with unit intensity. By employing geometric optics in our analysis, it is easy to find that the blurring radius satisfies

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

where F is the focal length of the lens and D the radius of the lens [3].

An important case 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 $s(x) = s$, $\forall x \in \mathbb{R}^2$, the PSF h is shift-invariant, i.e. $h(x, y, b) = h(x - y, b)$ and b is a constant. Hence, the image formation model becomes the following simple convolution

$$I = h(\cdot, b) * r. \tag{3}$$

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 [4]. Hence, in our model we will let

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

with standard deviation $\sigma \doteq \gamma b$ for a certain constant $\gamma > 0$ to be determined via a calibration procedure. Notice that when the depth map s is not an equifocal plane, the variance σ^2 depends on the 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*:

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

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 emitted by a point in the scene is transferred to the image plane.

[3]It has been argued [4], [13], [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 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.

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*:

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

The solution $u : \mathbb{R}^2 \times [0,\infty) \mapsto [0,\infty)$ at a time $t = \tau$, plays the role of an image $I(y) = u(y,\tau)$, $\forall y \in \Omega$, 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 $x \doteq [x_1 \; x_2]^T$. 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] $c \in C^1(\mathbb{R}^2)$, and $c(x) \geq 0 \; \forall x \in \mathbb{R}^2$. Let the open set $\mathcal{O} \doteq \{x : c(x) > 0\} \subset \Omega$, and assume that $\mathcal{O}$ is such that $\mathcal{O} \in C^1$ (i.e. $\mathcal{O}$ is bounded and the

[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

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

where $f(x,t)$ is the forcing term. The nomenclature *inhomogeneous* wants to emphasize that the diffusion coefficient is not homogeneous in space.

[5]$C^1(\mathbb{R}^2)$ is the space of scalar functions with continuous partial derivatives in $\mathbb{R}^2$.

boundary of $\mathcal{O}$ can be locally mapped by functions in $C^1(\mathbb{R})$). The inhomogeneous diffusion equation is then defined as

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

where the symbol ∇ is the gradient operator $\begin{bmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} \end{bmatrix}^T$ with $x = [x_1\ x_2]^T$, and the symbol $\nabla\cdot$ is the divergence operator $\sum_{i=1}^2 \frac{\partial}{\partial x_i}$. It is easy to verify that eq. (9) satisfies property (5) by using the divergence theorem [42] and the assumptions on the diffusion coefficient. Notice that property (5) can also be stated as follows:

$$\int u(y,t)dy = \iint G(y,x,t)u(x,0)dxdy = \int u(x,0)dx \tag{10}$$

where $u(x,0)$ is the initial condition of eq. (9), and G is Green's function[6] relative to the inhomogeneous heat eq. (9). In other words, the spatial average of u does not change in time.

The next proposition shows that the inhomogeneous heat equation satisfies this principle under some constraints on the diffusion coefficient c.

Proposition 1: The solution of the following inhomogeneous heat equation

$$\begin{cases} \dot{u}(x,t) &=& \nabla \cdot (c(x)\nabla u(x,t)) \qquad c : \mathbb{R}^2 \mapsto [0,\infty), c \in C^1 \\ u(x,0) &=& r(x) \end{cases} \tag{12}$$

satisfies property (5) if the open set $\mathcal{O} = \{x : c(x) > 0\}$ is such that $\mathcal{O} \in C^1$.

Proof: See Appendix. ■

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

$$\sigma^2(x) = 2tc(x) \tag{13}$$

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

$$c(x) = \frac{\gamma^2 b^2(x)}{2t} = \frac{\gamma^2 D^2 v^2}{8t}\left(\frac{1}{F} - \frac{1}{v} - \frac{1}{s}\right)^2. \tag{14}$$

[6]Green's function $G : \Omega \times \mathbb{R}^2 \times [0,\infty) \mapsto [0,\infty)$ is defined as the *impulse response* of eq. (9), i.e. $G(y,x,t) = u(y,t)$ where u is the solution of eq. (9) with initial conditions $u(y,0) = \delta(x-y)$. Hence, the function G satisfies the equation

$$u(y,t) = \int G(y,x,t)u(x,0)dx = \int G(y,x,t)r(x)dx. \tag{11}$$

IV. RELATIVE BLUR AND DIFFUSION

In section III-B we modeled the image $I(y)$ via the diffusion eq. (6) starting from the radiance $r(y)$, 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 $I_1(y) \doteq I(y, v_1)$ and $I_2(y) \doteq I(y, v_2)$ 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, we can write image I_2 by substituting the expression of r in terms of image I_1 as follows:

$$\begin{aligned} I_2(y) &= \int \frac{1}{2\pi\sigma_2^2} e^{-\frac{\|x-y\|^2}{2\sigma_2^2}} r(x)dx \\ &= \int \frac{1}{2\pi\left(\sigma_2^2 - \sigma_1^2\right)} e^{-\frac{\|x-y\|^2}{2\left(\sigma_2^2-\sigma_1^2\right)}} \int \frac{1}{2\pi\sigma_1^2} e^{-\frac{\|\bar{x}-x\|^2}{2\sigma_1^2}} r(\bar{x})d\bar{x}dx \\ &= \int \frac{1}{2\pi\Delta\sigma^2} e^{-\frac{\|x-y\|^2}{2\Delta\sigma^2}} I_1(x)dx \end{aligned} \tag{15}$$

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. (15) 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 $u(y, t_1) = I_1(y)$ and $u(y, t_2) = I_2(y)$, $\forall y \in \Omega$, and we can write

$$\begin{cases} \dot{u}(y,t) = c\triangle u(y,t) & c \in [0, \infty) \\ u(y, t_1) = I_1(y) & \forall y \in \Omega. \end{cases} \tag{16}$$

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

Equation (16) 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

$$\begin{cases} \dot{u}(y,t) = c\triangle u(y,t) & c \in [0, \infty) \\ u(y, 0) = I_1(y) & \forall y \in \Omega \end{cases} \tag{17}$$

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

$$\Delta\sigma^2 = 2(t_2 - t_1)c \doteq 2\Delta t c. \tag{18}$$

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

$$c = \frac{\Delta\sigma^2}{2\Delta t} = \frac{\gamma^2\left(b_2^2 - b_1^2\right)}{2\Delta t} \tag{19}$$

where for $i = 1, 2$

$$b_i = \frac{Dv_i}{2}\left|\frac{1}{F} - \frac{1}{v_i} - \frac{1}{s}\right|. \tag{20}$$

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. (17) to

$$\begin{cases} \dot{u}(y,t) = \nabla \cdot (c(y)\nabla u(y,t)) & t \in (0,\infty) \\ u(y,0) = I_1(y) & \forall y \in \Omega \\ c(y)\nabla u(y,t) \cdot n(y) = 0 & \forall y \in \partial\Omega \end{cases} \tag{21}$$

where $u(y,\Delta t) = I_2(y)$, $\forall y \in \Omega$, and n is the unit vector normal to the boundary $\partial\Omega$. 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 (21), for instance the oriented-Laplacian [43]. Each has its own advantages, but none is exactly equivalent

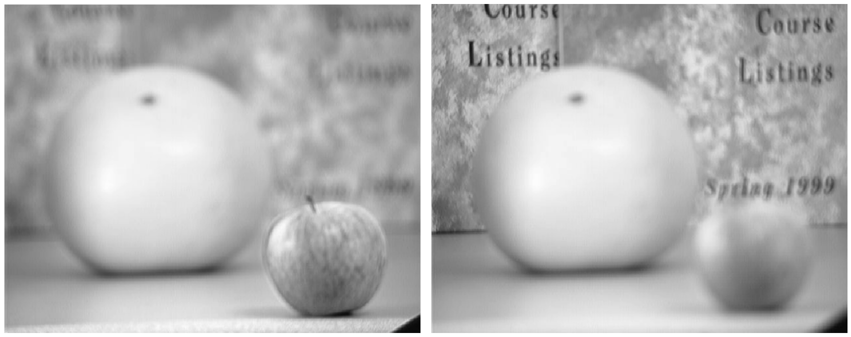

Fig. 1

TWO DEFOCUSED IMAGES I_1 AND I_2. THE APPLE IN THE IMAGE ON THE LEFT (I_1) IS SHARPER THAN THE APPLE 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.

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, including parting from linearity. 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 (21) as a prototype.

VI. ENFORCING FORWARD DIFFUSION

In the derivation of eq. (21) we made the assumption that I_1 is more focused than I_2. This assumption is necessary in order to guarantee that $c(y) \geq 0$, $\forall y \in \Omega$, and hence that (21) 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 apple in image I_1 (left) is more focused than the corresponding apple 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 (21) involves both *backward diffusion* and forward diffusion. Backward diffusion

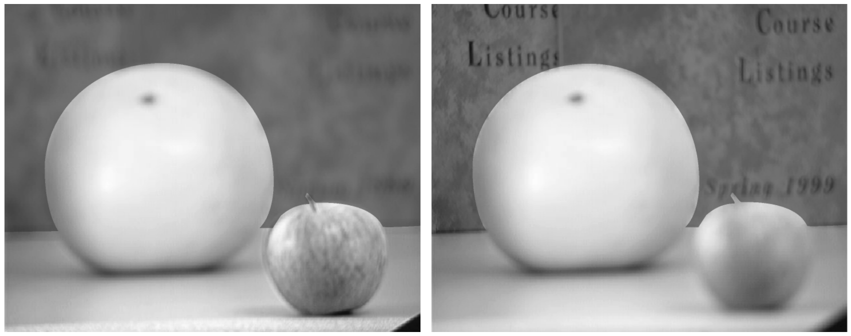

Fig. 2

VISUALIZATION OF THE PARTITION $\{\Omega_+, \Omega_-\}$ OBTAINED FROM THE IMAGES IN FIGURE 1. THE SET Ω_+ IS VISUALIZED BY BRIGHTENING THE INTENSITY OF THE ORIGINAL IMAGES, WHILE THE SET Ω_- IS VISUALIZED BY DARKENING THE INTENSITY OF THE ORIGINAL IMAGES.

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 (21) is to restrict the solution u to the set Ω_+ where the diffusion coefficient is positive, that is

$$\Omega_+ \doteq \{y \in \Omega \mid c(y) > 0\} \tag{22}$$

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

$$\Omega_- \doteq \Omega \setminus \Omega_+ \doteq \{y \in \Omega \mid c(y) \leq 0\} \tag{23}$$

the diffusion coefficient c is negative[7] 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

$$\begin{cases} \dot{u}(y,t) = \nabla \cdot (-c(y)\nabla u(y,t)) & t \in (0, \infty) \\ u(y,0) = I_2(y) & \forall y \in \Omega_- \\ c(y)\nabla u(y,t) \cdot n(y) = 0 & \forall y \in \partial\Omega_- \end{cases} \tag{24}$$

[7]We always consider the diffusion coefficient c to be computed via eq. (18), where $\Delta\sigma$ is the relative blur between I_1 and I_2.

with $u(y,\Delta t) = I_1(y)$, $\forall y \in \Omega_-$. In Figure 2 we visualize the partition $\{\Omega_+, \Omega_-\}$ obtained from the images in Figure 1. The set Ω_+ is visualized by brightening the original images, while the set Ω_- is visualized by darkening it.

By using the partition $\{\Omega_+, \Omega_-\}$ we obtain an imaging model for pairs of defocused images I_1 and I_2 that involves only forward diffusion. Since eq. (22) tells us that the partition depends on the diffusion c which, in turn, depends on the unknown depth map s via equations (19) and (20), by recovering the depth map we also recover the partition.

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 (16) and (24), which we rewrite here in a more complete form, including boundary conditions, as:

$$\left\{\begin{array}{rcl} \dot{u}(y,t) & = & \begin{cases} \nabla\cdot\left(c(y)\nabla u(y,t)\right), & y\in\Omega_+ \\ \nabla\cdot\left(-c(y)\nabla u(y,t)\right), & y\in\Omega_- \end{cases} \quad t\in(0,\infty) \\ u(y,0) & = & \begin{cases} I_1(y), & \forall y\in\Omega_+ \\ I_2(y), & \forall y\in\Omega_- \end{cases} \\ 0 & = & c(y)\nabla u(y,t)\cdot n(y), \quad \forall y\in\partial\Omega_+\equiv\partial\Omega_+ \\ u(y,\Delta t) & = & \begin{cases} I_2(y), & \forall y\in\Omega_+ \\ I_1(y), & \forall y\in\Omega_-. \end{cases} \end{array}\right. \tag{25}$$

where 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 $c(y) = 0$ for any $y \in \partial\Omega_+$ and $y \in \partial\Omega_-$. Also, recall that the diffusion coefficient c depends on the depth map s via eq. (19) and eq. (20), which can be explicitly written as

$$c(y) = \frac{\gamma^2 D^2}{8\Delta t}\left(v_2^2\left(\frac{1}{F}-\frac{1}{v_2}-\frac{1}{s(y)}\right)^2 - v_1^2\left(\frac{1}{F}-\frac{1}{v_1}-\frac{1}{s(y)}\right)^2\right). \tag{26}$$

Notice that $c(y) = 0$ when

$$s(y) = \frac{(v_1 + v_2)F}{v_1 + v_2 - 2F} \tag{27}$$

or

$$s(y) = F \tag{28}$$

so that

$$\partial\Omega_-, \partial\Omega_+ = \left\{ y \;\middle|\; s(y) = \frac{(v_1 + v_2)F}{v_1 + v_2 - 2F} \quad \text{or} \quad s(y) = F \right\} \tag{29}$$

and similarly, by direct substitution,

$$\begin{aligned} \Omega_+ &= \left\{ y \;\middle|\; 0 < s(y) < F \quad \text{or} \quad s(y) > \frac{(v_1 + v_2)F}{v_1 + v_2 - 2F} \right\} \\ \Omega_- &= \left\{ y \;\middle|\; F < s(y) < \frac{(v_1 + v_2)F}{v_1 + v_2 - 2F} \right\}. \end{aligned} \tag{30}$$

Note that the sets Ω_+ and Ω_- depend on the 3-D structure of the scene; the depth s satisfying eq. (27) and eq. (28) 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. (25). If the depth map $\tilde{s}$ coincides with the depth map s of the scene, then the solution of eq. (25) must satisfy the condition $u(y, \Delta t) = I_2(y)$, $\forall y \in \Omega_+$ and $u(y, \Delta t) = I_1(y)$, $\forall y \in \Omega_-$.

This naturally suggests an iterative procedure to tackle the inverse problem of reconstructing shape from defocused images: Starting from an initial estimate of the depth map and the resulting partition (e.g. a flat plane), diffuse each image in the region where it is more focused than the other image, until 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:

$$\hat{s} = \arg\min_s \int_{\Omega_+} (u(y, \Delta t) - I_2(y))^2 dy + \int_{\Omega_-} (u(y, \Delta t) - I_1(y))^2 dy. \tag{31}$$

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 (31) 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, i.e.

$$\hat{s} = \arg\min_s \int_{\Omega_+} (u(y, \Delta t) - I_2(y))^2 dy + \int_{\Omega_-} (u(y, \Delta t) - I_1(y))^2 dy + \alpha\|\nabla s\|^2. \tag{32}$$

The additional third term imposes a smoothness constraint on the depth map s that is regulated by the parameter $\alpha > 0$. Notice that during the minimization of eq. (32) both the partition and the depth map are estimated simultaneously.

As we shall discuss in the next section, minimizing eq. (32) is not the only way of regularizing eq. (31). Indeed, one may also require that the flow be insensitive to variations in the brightness and the contrast of the radiance. For instance, the convergence of the depth map s to the true 3D shape should not depend on whether an object is bright or dark. However, eq. (31) yields a gradient that is proportional to the brightness of the input images. To counterbalance this dependency, we will not minimize eq. (32), but rather we will introduce an alternative method that combines the minima of eq. (31) with those of the regularizer $\|\nabla s\|^2$ in a (possibly non-variational) regularization scheme.

A. Gradient Flow and Preconditioning

In this section we describe an algorithm to compute regularized solutions of the cost functional (31). The complete derivation of the algorithm is fairly lengthy and is reported in the Appendix. For the benefit of those being familiar with these calculations, and to not disrupt the flow, we will only provide the main ideas in the following.

To simplify the notation, let

$$\begin{aligned} E_1(s) &\doteq \int H(c(y))|u(y,\Delta t) - I_2(y)|^2 dy \\ E_2(s) &\doteq \int H(-c(y))|u(y,\Delta t) - I_1(y)|^2 dy \\ E_3(s) &\doteq \alpha\|\nabla s\|^2 \end{aligned} \tag{33}$$

where H denotes the Heaviside function. Also, let

$$E(s) \doteq E_1(s) + E_2(s) + E_3(s) \tag{34}$$

so that eq. (32) can be rewritten as

$$\hat{s} = \arg\min_s E(s). \tag{35}$$

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.,

$$\frac{\partial s}{\partial \tau} \doteq -E'(s). \tag{36}$$

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. (34) 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

$$E_i'(s) = \tilde{E}_i'(c(s))c'(s) \tag{37}$$

for $i = 1, 2$, where

$$c'(s) = \frac{\gamma^2 D^2 (v_2 - v_1)}{4 s^2 \Delta t} \left[(v_2 + v_1) \left(\frac{1}{F} - \frac{1}{s} \right) - 1 \right], \tag{38}$$

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

$$\begin{aligned} \tilde{E}_1'(c(s))(y) \;&=\; -2H(c(y)) \int_0^{\Delta t} \nabla u(y,t) \cdot \nabla w_1(y, \Delta t - t)\; dt \\ &\quad + \delta(c(y)) \left(u(y, \Delta t) - I_2(y) \right)^2 \end{aligned} \tag{39}$$

and

$$\begin{aligned} \tilde{E}_2'(c(s))(y) \;&=\; 2H(-c(y)) \int_0^{\Delta t} \nabla u(y,t) \cdot \nabla w_2(y, \Delta t - t)\; dt \\ &\quad - \delta(c(y)) \left(u(y, \Delta t) - I_1(y) \right)^2 . \end{aligned} \tag{40}$$

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):

$$\begin{cases} \dot{w}_1(y,t) = \nabla \cdot (c(y) \nabla w_1(y,t)) & t \in (0, \infty) \\ w_1(y, 0) = u(y, \Delta t) - I_2(y) & \\ c(y) \nabla w_1(y,t) \cdot n(y) = 0 & \forall y \in \partial\Omega_+. \end{cases} \tag{41}$$

and

$$\begin{cases} \dot{w}_2(y,t) = \nabla \cdot (-c(y) \nabla w_2(y,t)) & t \in (0, \infty) \\ w_2(y, 0) = u(y, \Delta t) - I_1(y) & \\ c(y) \nabla w_2(y,t) \cdot n(y) = 0 & \forall y \in \partial\Omega_-. \end{cases} \tag{42}$$

Finally, the gradient of E_3 can be computed directly as

$$E_3'(s) = -\alpha \triangle s(y). \tag{43}$$

Since the gradient $E'(s)(y)$ depends on the radiance of the scene (see eq. (39) and (40)), which is undesirable, one can modify the above algorithm by *preconditioning*, i.e., by replacing eq. (36) with

$$\frac{\partial s}{\partial \tau} = -M(s)E'(s), \tag{44}$$

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

$$(M(s)\phi)(y) = \frac{\phi(y)}{\epsilon + 2\big(H(c(y))I_2(y) + H(-c(y))I_1(y)\big)|u'(y,\Delta t)|}, \tag{45}$$

where ϵ is a small positive number to avoid division by zero, and $u'(y, \Delta T)$ is the functional derivative of u with respect to the shape s. By using eq. (45) with $\epsilon = 0$, we obtain that

$$\begin{aligned} \left(M(s)\tilde{E}_1'(c(s))\right)(y) &= H(c(y))\left(\frac{u(y,\Delta t)}{I_2(y)} - 1\right)\frac{u'(y,\Delta t)}{|u'(y,\Delta t)|} \\ &+ \frac{1}{2}\delta(c(y))\frac{\big(u(y,\Delta t) - I_2(y)\big)^2}{I_2(y)|u'(y,\Delta t)|} \end{aligned} \tag{46}$$

and, similarly,

$$\begin{aligned} \left(M(s)\tilde{E}_2'(c(s))\right)(y) &= H(-c(y))\left(\frac{u(y,\Delta t)}{I_1(y)} - 1\right)\frac{u'(y,\Delta t)}{|u'(y,\Delta t)|} \\ &- \frac{1}{2}\delta(c(y))\frac{\big(u(y,\Delta t) - I_1(y)\big)^2}{I_1(y)|u'(y,\Delta t)|}. \end{aligned} \tag{47}$$

However, by preconditioning E', we are also preconditioning the term E_3', that did not need any normalization. Thus, we consider instead the flow

$$\frac{\partial s}{\partial \tau} = -M(s)(E_1'(s) + E_2'(s)) - E_3'(s) = -M(s)(E_1'(s) + E_2'(s) + M^{-1}(s)E_3'(s)) \tag{48}$$

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

Note that eq. (48) 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. (48) at infinity will not be a stationary point or even a minimizer of the energy E. However, as we mentioned in the previous section, the formulation of the energy E is not the only way to introduce the regularization term $\|\nabla s\|^2$. To clarify that eq. (48) is an alternative to the original flow of E, notice that for $\alpha = 0$ we obtain the original metric flow in the data term, and,

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. (26) and eq. (30);
4) Simulate (i.e. numerically integrate) eq. (16) and eq. (24);
5) Using the solutions obtained at the previous step, simulate eq. (41) and eq. (42);
6) Compute the gradient of u and w and evaluate eq. (39), eq. (40), eq. (43), and (45);
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.

similarly, if we remove the data term, then we obtain a metric flow that converges to the minima of $\|\nabla s\|^2$. More detailed arguments on the mathematics behind this approach are given in the Appendix.

VIII. ON THE EXTENSION TO MULTIPLE IMAGES

Extending the algorithm above to handle multiple images while avoiding restoration of the radiance results in a combinatorial problem of factorial complexity: In practice, we have to consider the diffusion equations between all possible orderings of the images. To get a more concrete understanding of what happens with $K > 2$ images, consider the problem of "sorting" the images in increasing order of blur at each pixel, so that we are always applying forward diffusions to the input images. This leads to $K!$ possible orderings. In the case of two images, this amounts to simply determining 2 possible orderings, that can be easily tackled as we suggested in the previous section. However, to compute the solution in the general case, we need to simulate $K!$ different PDEs, one for each ordering (again, notice that in the case $K = 2$ we only need to simulate $K! = 2$ PDEs). Hence, both the number of equations we need to solve and the complexity of the algorithm depend on $K!$, which grows rather quickly as $K > 2$. One can, however, use a different approach. For instance, one can solve a *segmentation background-*

foreground between pairs of images such as in [2]. Then, the resulting segmentation can be used to build a *layered* representation of the depth map. For more details see [2].

IX. 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. Although we provide the performance of the algorithm on a number of examples, we do not directly compare our algorithm to existing methods. This is due to the lack of a common public dataset that can be used to benchmark the different methods developed in shape from defocus.

A. Experiments with synthetic images

In the synthetic data set we evaluate the algorithm on a number of different shapes: A wave, a slope, and a scene made of equifocal planes at different depths. In addition, to illustrate the insensitivity of the algorithm to the magnitude of the radiance and its content in frequency, we choose radiances with 3 different intensity levels along the y axis and 3 different textures along the x axis (see for example Figure 3). In this data set the camera is set to bring in focus first the plane at $520\ mm$ and then the plane at $850\ mm$, with focal length $12\ mm$, F-number 2 and $\gamma = 1.5 \cdot 10^4 pixel^2/mm^2$. In Figures 3, 5 and 7 we show experiments conducted on a wave, a slope and a piecewise constant depth map respectively. In each figure we show the input images on the top row, and the true region Ω_+ on the bottom-left image; on the bottom-right we show the estimated region Ω_+. In Figures 4, 6 and 8 we display the true depth on the top-left image as a grayscale intensity map (light intensities correspond to points close to the camera, while dark intensities correspond to points far from the camera) and on the bottom-left image as a mesh. Similarly, the estimated depth is displayed as a grayscale intensity map on the top-right image and as a mesh on the bottom-right image. Notice that the estimated depth map converges evenly towards the true depth map in all cases, despite the different intensity (top to bottom) and the different level of sharpness (left to right) of the input images (see, for instance, the images on the top-row of Figure 3).

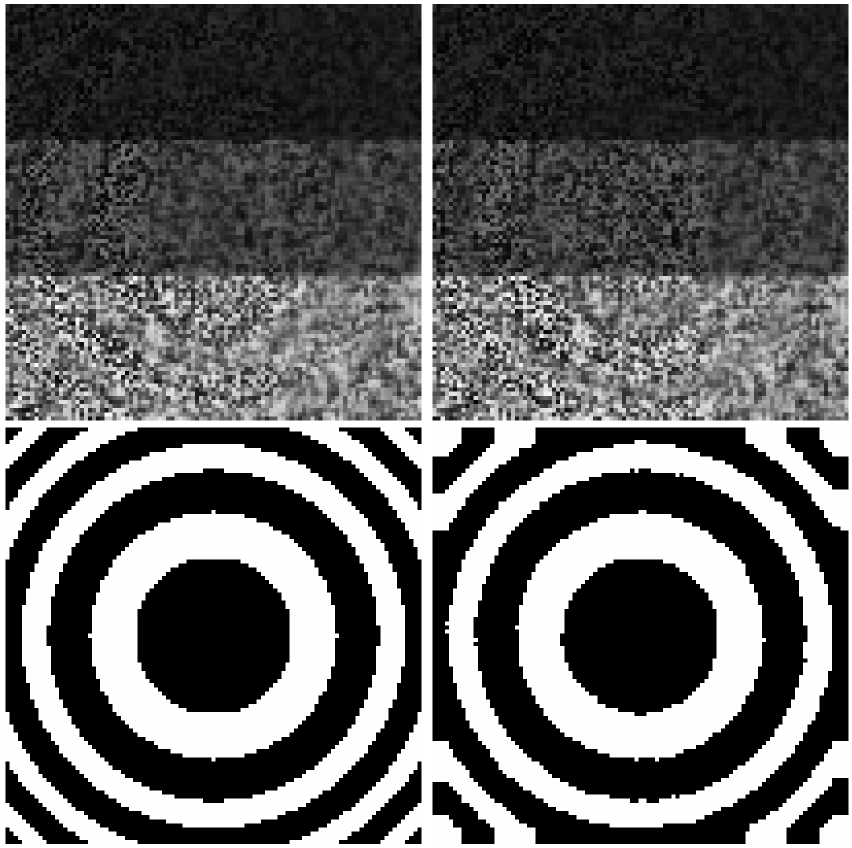

Fig. 3

WAVE DATA SET. TOP-LEFT: NEAR-FOCUSED ORIGINAL IMAGE. TOP-RIGHT: FAR-FOCUSED ORIGINAL IMAGE. BOTTOM-LEFT: TRUE REGION Ω_+. BOTTOM-RIGHT: ESTIMATED REGION Ω_+.

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:

$$\begin{aligned} ERR_0 &= \sqrt{E\left[(s-\hat{s})^2\right]} \\ ERR_1 &= \sqrt{E\left[\left(\frac{\hat{s}}{s}-1\right)^2\right]} \end{aligned} \tag{49}$$

where s is the true depth map and $\hat{s}$ is the estimated depth map. $E[\cdot]$ denotes the average over[8] Ω. The first one, ERR_0, measures the absolute error in meters (m); the second, ERR_1, measures

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

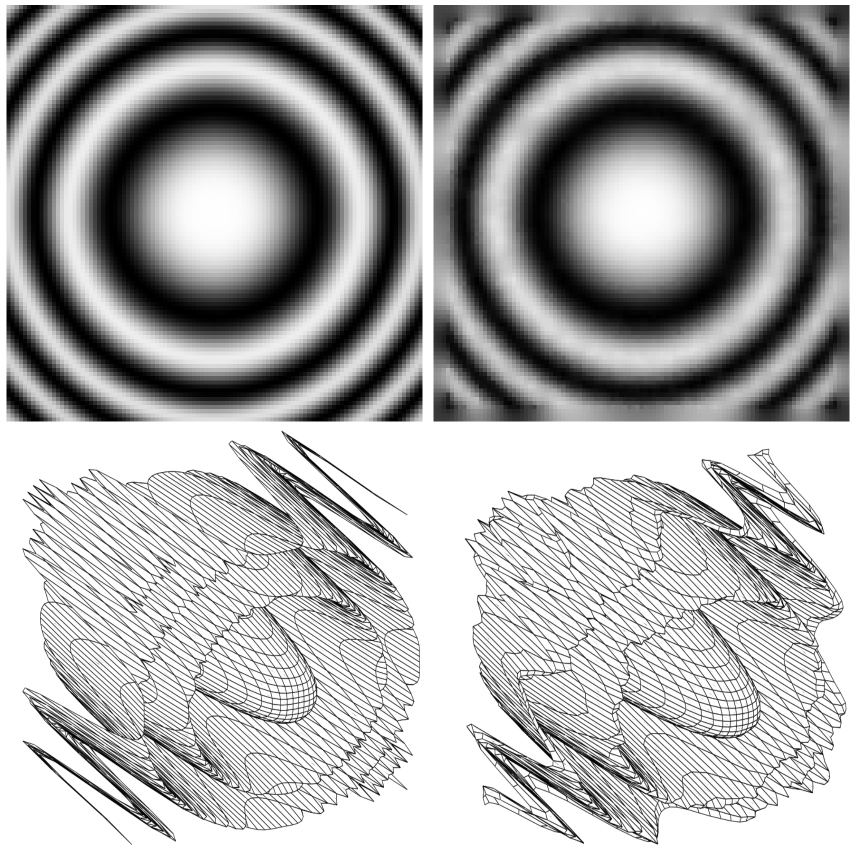

Fig. 4

WAVE DATA SET. TOP-LEFT: ORIGINAL DEPTH MAP IN GRAYSCALE INTENSITIES. TOP-RIGHT: ESTIMATED DEPTH MAP IN GRAYSCALE INTENSITIES. BOTTOM-LEFT: ORIGINAL DEPTH MAP DISPLAYED AS A MESH. BOTTOM-RIGHT: ESTIMATED DEPTH MAP DISPLAYED AS A MESH.

the relative error and it is therefore unitless. We show the absolute and relative errors for each of the synthetic experiments in Table II.

B. Experiments with real images

In the case of real data we have the same camera settings as in the case of the synthetic data apart from γ that is set to $3 \cdot 10^4 pixel^2/mm^2$. The scene is composed of a dark box at the bottom and a grey cylinder and a small white box on the top (see top-row in Figure 9), which have been purposely chosen to challenge any global optimization method for shape from

		wave	slope	boxes
depth map	ERR_0	$0.016m$	$0.0009m$	$0.015m$
	ERR_1	2.2%	0.13%	2.3%

TABLE II

ABSOLUTE AND RELATIVE DISCREPANCIES BETWEEN GROUND TRUTH AND ESTIMATED DEPTH MAPS. THE ABSOLUTE DISCREPANCIES, ERR_0, ARE IN METERS (m).

defocus. We capture the two defocused images shown in the top-row in Figure 9 and obtain the region Ω_+ shown in the bottom-left image and the depth map in the bottom-right image. The depth map is also displayed as a mesh and viewed from different vantage points in Figure 10 to better evaluate the quality of the reconstruction. As one can see, despite all the objects have texture with a different intensity and frequency content, the proposed algorithm yields a high quality map of the scene.

X. CONCLUSIONS

Using the notion of relative blur we have been able to formulate the problem of shape from defocus as a variable-stopping-time forward diffusion process, that is numerically stable and yields a global estimate of the depth map of a scene, without making any assumption on the regularity of the underlying radiance.

Our approach marries the benefits of local approaches (separation of shape from radiance, hence reduced computational complexity) with those of global ones (optimal estimate 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.

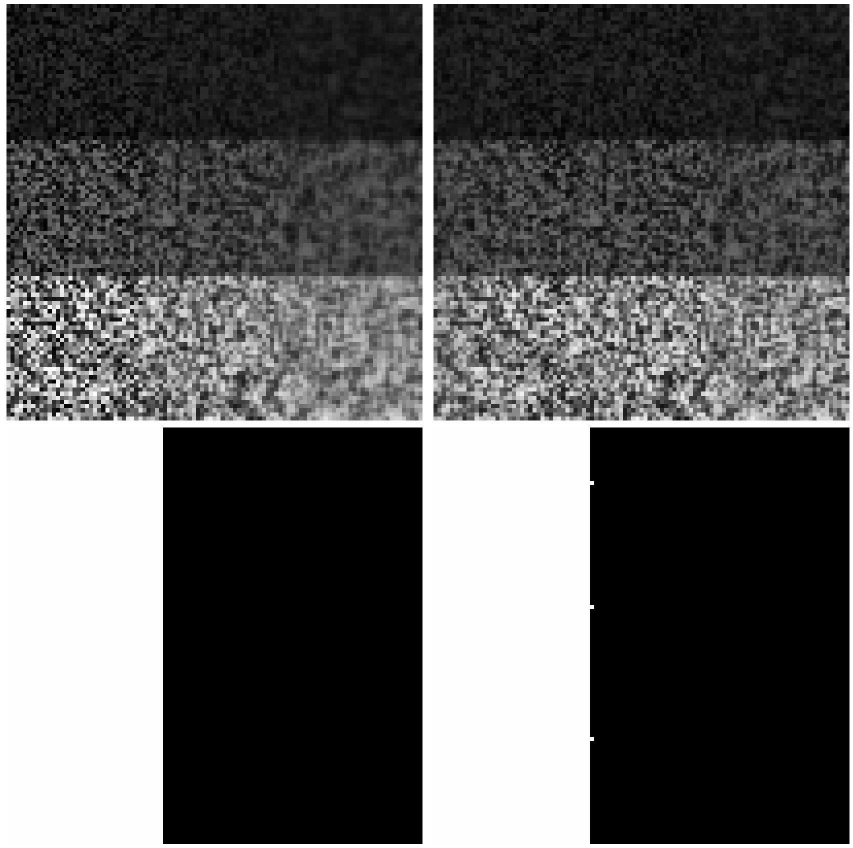

Fig. 5

SLOPE DATA SET. TOP-LEFT: NEAR-FOCUSED ORIGINAL IMAGE. TOP-RIGHT: FAR-FOCUSED ORIGINAL IMAGE. BOTTOM-LEFT: TRUE REGION Ω_+. BOTTOM-RIGHT: ESTIMATED REGION Ω_+.

APPENDIX I
COMPUTATION OF THE GRADIENTS

To compute the gradients (39) we need to derive the first-order variation of the cost functional (35). In the case of

$$E_3 = \int_\Omega \|\nabla s(y)\|^2 dy \tag{50}$$

the computation amounts to:

$$E_3'(s)h = \int_\Omega 2\nabla s(y) \cdot \nabla h(y) dy = \int_\Omega -2(\nabla \cdot \nabla s(y))h(y)dy = 0 \tag{51}$$

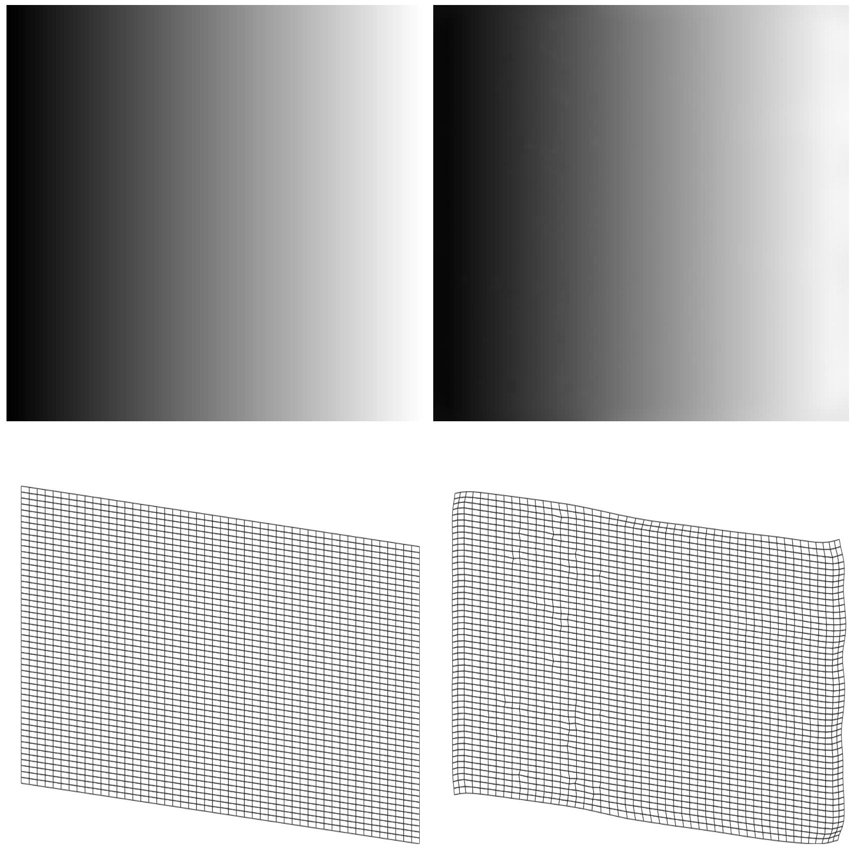

Fig. 6

SLOPE DATA SET. TOP-LEFT: ORIGINAL DEPTH MAP IN GRAYSCALE INTENSITIES. TOP-RIGHT: ESTIMATED DEPTH MAP IN GRAYSCALE INTENSITIES. BOTTOM-LEFT: ORIGINAL DEPTH MAP DISPLAYED AS A MESH. BOTTOM-RIGHT: ESTIMATED DEPTH MAP DISPLAYED AS A MESH.

for an arbitrary h such that $h(y) = 0$, $\forall y \in \partial\Omega$. Hence, the functional derivative with respect to E_3 is

$$E_3'(s)(y) = -2\triangle s(y). \tag{52}$$

The other two 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 complication we meet in computing the functional derivative relative to E_1 is due to the fact that the function u is not given in explicit form, but as the solution of a

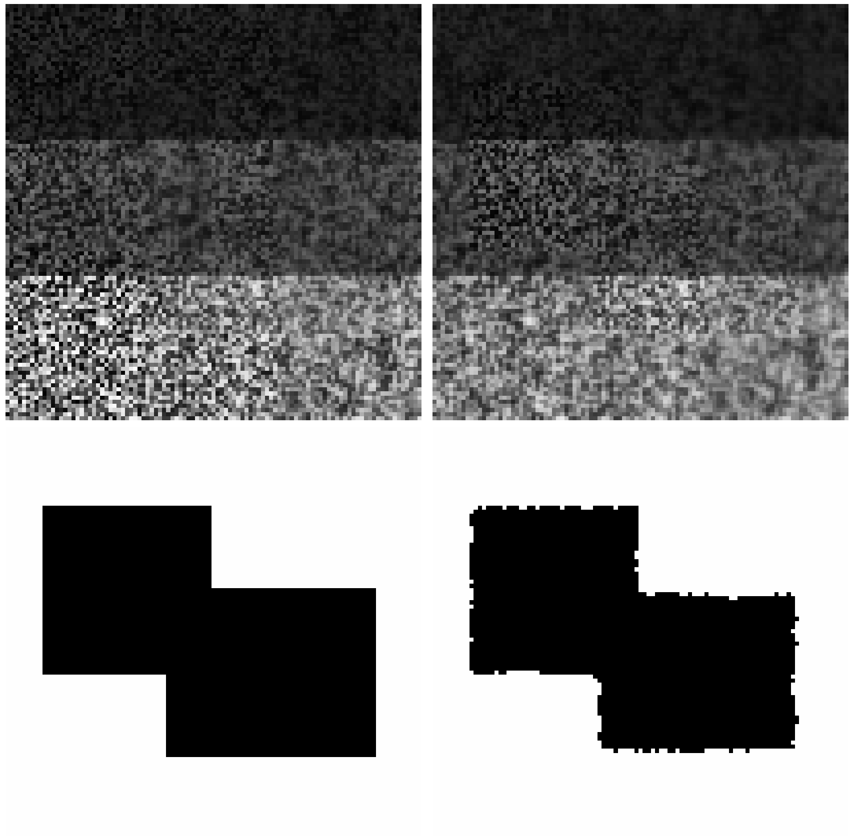

Fig. 7

PIECEWISE-CONSTANT DATA SET. TOP-LEFT: NEAR-FOCUSED ORIGINAL IMAGE. TOP-RIGHT: FAR-FOCUSED ORIGINAL IMAGE. BOTTOM-LEFT: TRUE REGION Ω_+. BOTTOM-RIGHT: ESTIMATED REGION Ω_+.

PDE. When $x \in \Omega_+$, u is the solution of

$$\begin{cases} \dot{u}(y,t) = \nabla \cdot (|c(y)|\nabla u(y,t)) & t \in [0,\infty) \\ u(y,0) = I_1(y) & \forall y \in \Omega_+ \\ \\ u(y,\Delta t) = I_2(y) & \forall y \in \Omega_+ \end{cases} \tag{53}$$

The first-order variation [49] of the term

$$\tilde{E}_1(c) \doteq \int_{\Omega_+} (u(y,\Delta t) - I_2(y))^2 dy \tag{54}$$

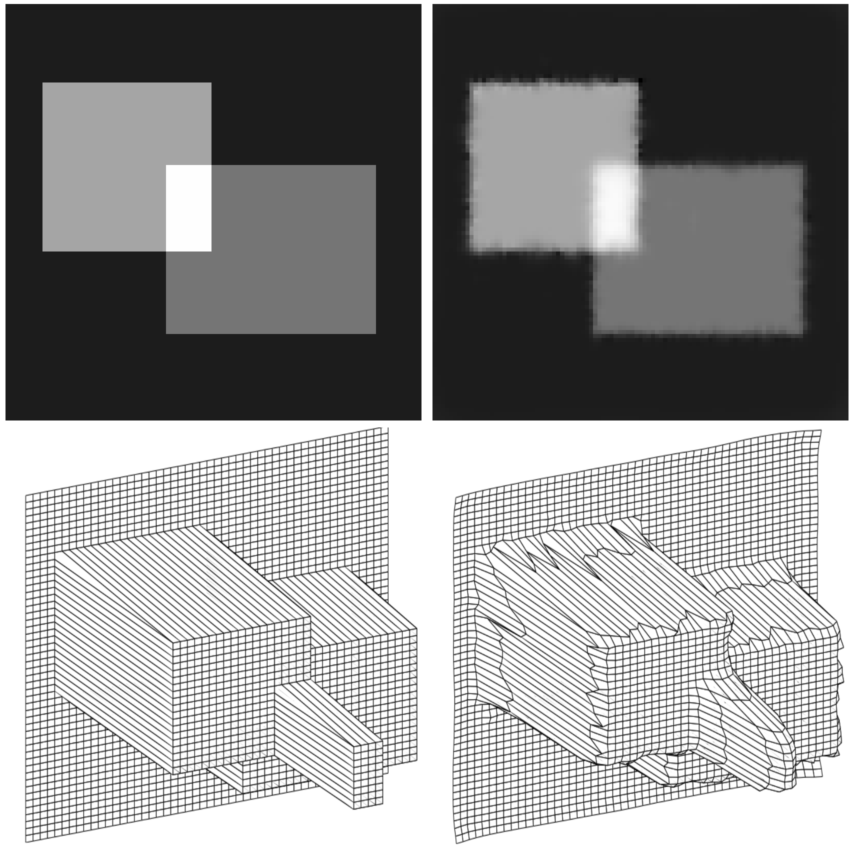

Fig. 8

PIECEWISE-CONSTANT DATA SET. TOP-LEFT: ORIGINAL DEPTH MAP IN GRAYSCALE INTENSITIES. TOP-RIGHT: ESTIMATED DEPTH MAP IN GRAYSCALE INTENSITIES. BOTTOM-LEFT: ORIGINAL DEPTH MAP DISPLAYED AS A MESH. BOTTOM-RIGHT: ESTIMATED DEPTH MAP DISPLAYED AS A MESH.

is

$$\begin{aligned} \tilde{E}_1'(c)h &= 2\int_{\Omega_+} (u(y,\Delta t) - I_2(y))\delta u(y,\Delta t)dy \\ &\quad + \int \delta(c(y))\left(u(y,\Delta t) - I_2(y)\right)^2 h(y)dy \end{aligned} \tag{55}$$

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

$$\delta u(y,t) \doteq \lim_{\epsilon \to 0} \frac{u(y,t)\big|_{c+\epsilon h} - u(y,t)\big|_c}{\epsilon} \tag{56}$$

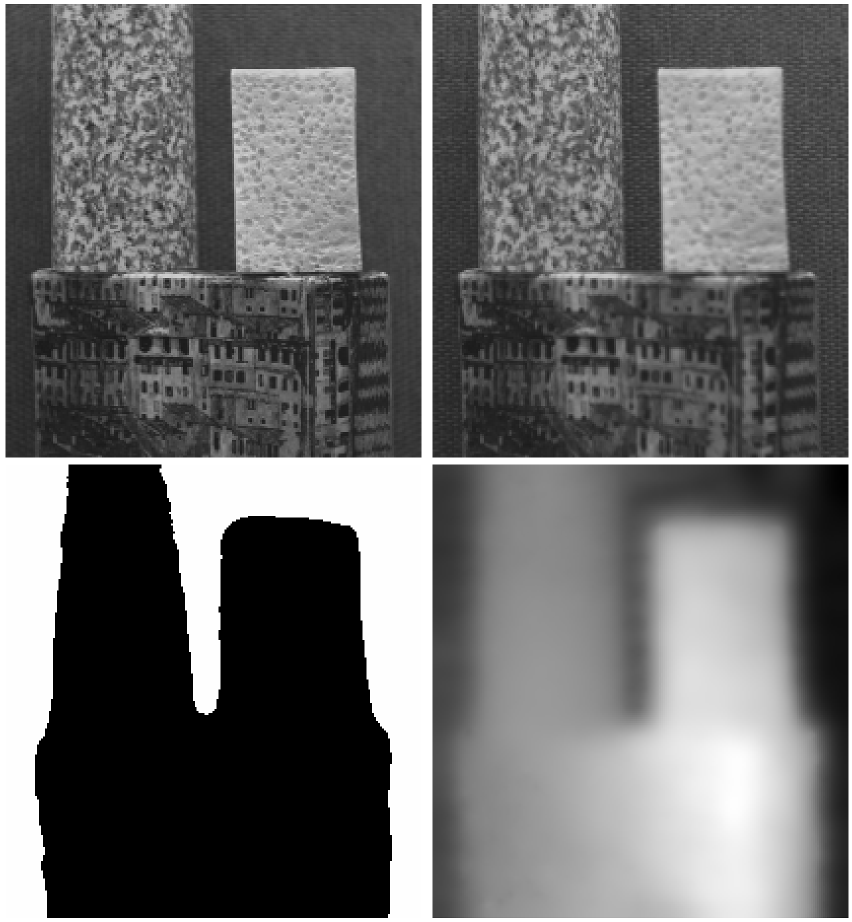

Fig. 9

REAL DATA SET. TOP-LEFT: NEAR-FOCUSED ORIGINAL IMAGE. TOP-RIGHT: FAR-FOCUSED ORIGINAL IMAGE. BOTTOM-LEFT: ESTIMATED REGION Ω_+. BOTTOM-RIGHT: ESTIMATED DEPTH MAP.

we see that δu satisfies

$$\begin{cases} \delta\dot{u}(y,t) = \nabla \cdot (c(y)\nabla\delta u(y,t) + h(y)\nabla u(y,t)) & t \in (0,\infty) \\ \delta u(y,0) = 0 & \forall y \in \Omega_+ \end{cases} \tag{57}$$

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

$$z(y,\Delta t) = u(y,\Delta t) - I_2(y). \tag{58}$$

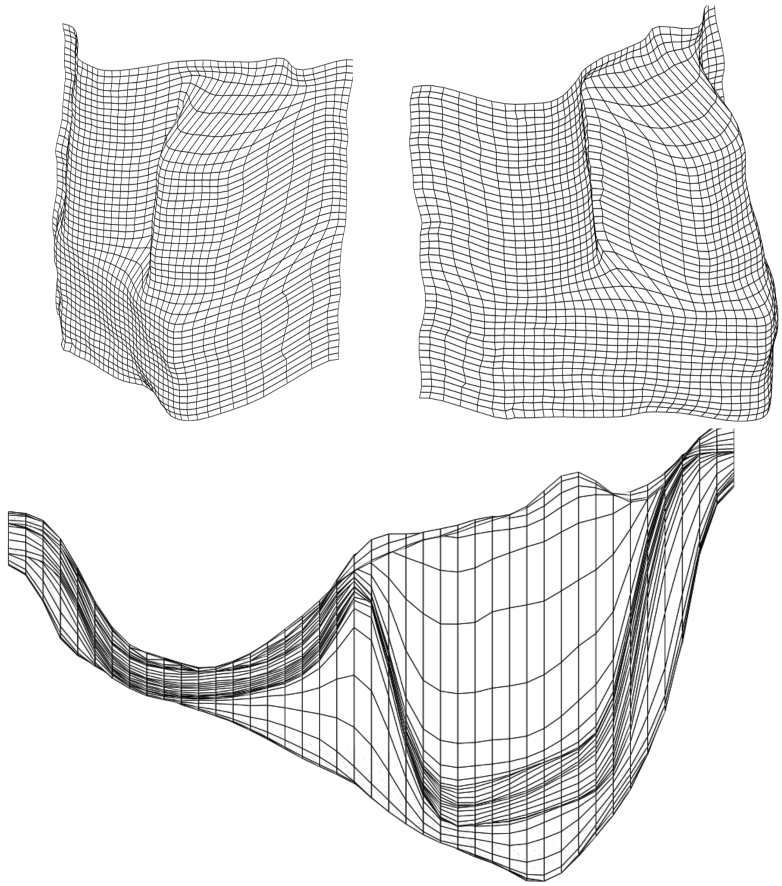

Fig. 10

REAL DATA SET. TOP-LEFT: ESTIMATED DEPTH MAP VIEWED FROM THE RIGHT. TOP-RIGHT: ESTIMATED DEPTH MAP VIEWED FROM THE LEFT. BOTTOM: ESTIMATED DEPTH MAP VIEWED FROM THE TOP.

Then, by substituting z in eq. (55), we obtain:

$$
\begin{aligned}
\tilde{E}_1'(c)h &= 2\int_{\Omega_+} z(y,\Delta t)\delta u(y,\Delta t)dy \\
&\quad + \int \delta(c(y))\left(u(y,\Delta t) - I_2(y)\right)^2 h(y)dy \\
&= 2\int_{\Omega_+} z(y,0)\delta u(y,0)dy \\
&\quad + 2\int_0^{\Delta t}\int_{\Omega_+} \dot{z}(y,t)\delta u(y,t) + z(y,t)\delta\dot{u}(y,t)dydt \\
&\quad + \int \delta(c(y))\left(u(y,\Delta t) - I_2(y)\right)^2 h(y)dy
\end{aligned}
\tag{59}
$$

which becomes

$$\begin{aligned}\tilde{E}_1'(c)h &= 2\int_0^{\Delta t}\int_{\Omega_+}\Big(\dot{z}(y,t)+\nabla\cdot(c(y)\nabla z(y,t))\Big)\delta u(y,t)\\ &\quad -\ h(y)\nabla z(y,t)\cdot\nabla u(y,t)dydt\\ &\quad +\ \int\delta(c(y))\left(u(y,\Delta t)-I_2(y)\right)^2 h(y)dy\end{aligned}\tag{60}$$

after integrating by parts twice and using the fact that both h and the diffusion coefficient c vanish at the boundary of Ω_+, and noticing that the initial conditions of δu are $\delta u(y,0)=0$. To simplify the above equation we choose the adjoint function z in the interval $(0,\Delta t)$ such that

$$\begin{cases}\dot{z}(y,t)=-\nabla\cdot(c(y)\nabla z(y,t)) & t\in(0,\Delta t]\\ z(y,\Delta t)=u(y,\Delta t)-I_2(y) & \forall y\in\Omega_+.\end{cases}\tag{61}$$

The substitution of z in eq. (60) yields

$$\begin{aligned}\tilde{E}_1'(c)h &= -2\int_0^{\Delta t}\int_{\Omega_+}h(y)\nabla z(y,t)\cdot\nabla u(y,t)dydt\\ &\quad +\ \int\delta(c(y))\left(u(y,\Delta t)-I_2(y)\right)^2 h(y)dy.\end{aligned}\tag{62}$$

Now, define $w(y,t)=z(y,\Delta t-t)$, then we obtain the *adjoint equation* (also notice that this is the definition of eq. (41) and eq. (42))

$$\begin{cases}\dot{w}(y,t)=\nabla\cdot(c(y)\nabla w(y,t)) & t\in(0,\Delta t]\\ w(y,0)=u(y,\Delta t)-I_2(y) & \forall y\in\Omega_+\end{cases}\tag{63}$$

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

$$\begin{aligned}\tilde{E}_1'(c)h &= -2\int_0^{\Delta t}\int_{\Omega_+}h(y)\nabla w(y\Delta t-t)\cdot\nabla u(y,t)dydt\\ &\quad +\ \int\delta(c(y))\left(u(y,\Delta t)-I_2(y)\right)^2 h(y)dy.\end{aligned}\tag{64}$$

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

$$\begin{aligned}\tilde{E}_1'(c) &= -2H(c(y))\int_0^{\Delta t}\nabla w(y,\Delta t-t)\cdot\nabla u(y,t)dt\\ &\quad +\ \delta(c(y))\left(u(y,\Delta t)-I_2(y)\right)^2.\end{aligned}\tag{65}$$

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 discuss some mathematical properties of the modified gradient flow (48). For the analysis of the flow we assume that H and δ are smoothed (at least continuously differentiable) approximations of the Heaviside function and Dirac' delta, so that all functionals are differentiable. For simplicity we assume to know the boundary values of s, so that $-\Delta$ is a positive definite operator on

$$u_D + H_0^1 = \{u + u_D \in L^2 \mid \nabla u \in L^2, u|_{\partial\Omega} = 0\}. \tag{66}$$

Here u_D is a function with $\nabla u_D \in L^2$ with the prescribed boundary values and we also make the natural assumption that $u_D \in L^\infty$, i.e., it is globally bounded on Ω. Finally, we approximate $c'(s)$ by the formula

$$c'(s) = \frac{\gamma^2 D^2 (v_2 - v_1)}{4 \max\{\hat{\epsilon}, s\}^2 \Delta t} \left[(v_2 + v_1) \left(\frac{1}{F} - \frac{1}{\max\{\hat{\epsilon}, s\}} \right) - 1 \right], \tag{67}$$

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

$$|c'(s)| \leq \frac{\gamma^2 D^2 \, |v_2 - v_1|}{4\hat{\epsilon}^2 \Delta t} \left[(v_2 + v_1) \left(\frac{1}{F} + \frac{1}{\hat{\epsilon}} \right) + 1 \right], \tag{68}$$

We start by verifying the existence of the preconditioned flow (48). Defining an operator

$$R(s) = -M(s)(E_1'(s) + E_2'(s)), \tag{69}$$

the flow can be rewritten as

$$\frac{\partial s}{\partial \tau} = 2\Delta s + R(s), \tag{70}$$

i.e., we obtain a heat equation with a nonlocal source term. One can verify that R maps L^∞ to L^∞ and is even Lipschitz on these spaces. Hence, by standard results for parabolic equations (cf. [50]) we obtain the existence of a bounded solution, more precisely:

Proposition 2: Let I_1 and I_2 be positive and bounded. Then there exists a unique solution s of eq. (48) such that $s(\tau) \in L^\infty$ for almost any $\tau \in \mathbb{R}^+$.

Since eq. (48) is possibly no gradient flow for a single 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 we provide a different argument for the existence of a bounded solution in the following. The equation for a stationary solution is given by $E'(s) = 0$, i.e.,

$$E_3'(s) = -2\alpha\Delta s = R(s) = -M(s)(E_1'(s) + E_2'(s)). \tag{71}$$

We shall then prove the existence of a stationary solution by a fixed argument for the equation in the form

$$s = F(s) = \frac{1}{2\alpha}(-\Delta)^{-1}R(s). \tag{72}$$

As noticed above, R is a continuous operator on L^∞. Moreover, the inverse Laplacian $(-\Delta)^{-1}$ 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 $L^\infty \cap (u_D + H_0^1)$. Moreover, we have

$$\begin{aligned} \left|\left(M(s)\tilde{E}_1'(c(s))\right)(y)\right| &= \left| H(c(y))\frac{(u(y,\Delta t) - I_2(y))u'(y,\Delta t)}{\frac{\epsilon}{2} + I_2(y)|u'(y,\Delta t)|} \right.\\ &\quad \left. + \frac{1}{2}\delta(c(y))\frac{(u(y,\Delta t) - I_2(y))^2}{\frac{\epsilon}{2} + I_2(y)|u'(y,\Delta t)|}\right| \\ &\leq C_H\left(\frac{|u(y,\Delta t)|}{\epsilon} + 1\right) + C_\delta \frac{(u(y,\Delta t) - I_2(y))^2}{\epsilon}, \end{aligned} \tag{73}$$

where the constants C_H and C_δ are the maximal values of the smoothed Heaviside and delta function, respectively. Now finally, let C_I be an upper bound for I_1, I_2, and u_D. Then, a maximum principle for parabolic equations (cf. [51]) implies $|u(y,\Delta t)| \leq C_I$ for all $y \in \Omega$. Hence, we obtain

$$\sup_{y\in\Omega}\left|\left(M(s)\tilde{E}_1'(c(s))\right)(y)\right| \leq C_H\left(\frac{C_I}{\epsilon} + 1\right) + C_\delta\frac{4C_I^2}{\epsilon}, \tag{74}$$

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

$$R(s) = M(s)(\tilde{E}_1'(c(s))c'(s) + \tilde{E}_2'(c(s))c'(s)) \tag{75}$$

maps into a bounded set in L^∞. Finally, since $(-\Delta)^{-1}$ is compact, we can conclude that the operator F even maps into a compact set in L^∞ (and in particular this compact set into itself). Thus, by Schauder's fixed point theorem (cf. [52]) we conclude the existence of a solution of eq. (72), which is equivalent to eq. (71). We summarize this result in the following Proposition:

Proposition 3: Let I_1 and I_2 be positive and bounded. Then there exists a solution $s \in L^\infty \cap (u_D + H_0^1)$ of eq. (71).

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] P. Favaro, S. Soatto, L. A. Vese, and S. J. Osher, "Shape from anisotropic diffusion," in *CVPR*, June 2003, pp. I–179–186.

[3] P. Favaro and S. Soatto, *3D Shape Reconstruction and Image Restoration: Exploiting Defocus and Motion-Blur.* Springer-Verlag, 2006.

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

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

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

[7] J. Ens and P. Lawrence, "An investigation of methods for determining depth from focus," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, pp. 97–108, 1993.

[8] P. Favaro, A. Mennucci, and S. Soatto, "Observing shape from defocused images," *IJCV*, vol. 52, no. 1, pp. 25–43, April 2003.

[9] P. Favaro and S. Soatto, "Learning shape from defocus," in *ECCV02*, 2002, p. II: 735 ff.

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

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

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

[13] A. Pentland, "A new sense for depth of field," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, pp. 523–531, 1987.

[14] A. Pentland, T. Darrell, M. Turk, and W. Huang, "A simple, real-time range camera," in *Computer Vision and Pattern Recognition*, 1989, pp. 256–261.

[15] 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, November 1994.

[16] 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. 10–7, 2000.

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

[18] A. N. Rajagopalan, S. Chaudhuri, and U. Mudenagudi, "Depth estimation and image restoration using defocused stereo pairs." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1521–1525, 2004.

[19] M. Bertero and P. Boccacci, "Introduction to inverse problems in imaging," *Institute of Physics Publishing, Bristol and Philadelphia*, 1998.

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

[21] Y. Schechner and N. Kiryati, "The optimal axial interval in estimating depth from defocus," in *Proc. of the Intl. Conf. of Comp. Vision*, 1993, pp. 843–848.

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

[23] H. N. Nair and C. V. Stewart, "Robust focus ranging," in *Computer Vision and Pattern Recognition*, 1992, pp. 309–314.

[24] N. Asada, H. Fujiwara, and T. Matsuyama, "Edge and depth from focus," *Intl. J. of Comp. Vision*, vol. 26, no. 2, 1998.

[25] J. Marshall, C. Burbeck, and D. Ariely, "Occlusion edge blur: a cue to relative visual depth," *Intl. J. Opt. Soc. Am. A*, vol. 13, pp. 681–688, 1996.

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

[27] G. Schneider, B. Heit, J. Honig, and J. Bremont, "Monocular depth perception by evaluation of the blur in defocused images," in *Proc. International Conference on Image Processing*, vol. 2, 1994, pp. 116–9.

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

[29] B. Girod and S. Scherock, "Depth from focus of structured light," in *SPIE*, 1989, pp. 209–15.

[30] S. K. 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, December 1996.

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

[32] Y. Xiong and S. Shafer, "Depth from focusing and defocusing," in *Proc. of the Intl. Conf. of Comp. Vision and Pat. Recogn.*, 1993, pp. 68–73.

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

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

[35] B. Fischl and E. L. 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," *J. of Comput. Phys.*, vol. 79, pp. 12–49, 1998.

[37] M. J. Black, G. Sapiro, D. H. 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. A. Forsyth and J. Ponce, *Computer Vision: a modern approach.* Prentice Hall, 2002.

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

[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," in *CVPR03*, 2003, pp. I: 651–656.

[44] H. H. Hopkins, "The frequency response of a defocused optical system," *Proc. R. Soc. London Ser. 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," *J. Comput. Finance*, vol. 1, no. 1, pp. 13–26, 1997.

[46] G. Aubert and K. P., "Mathematical problems in image processing: Partial differential equations and the calculus of variations," *Springer Verlag, Applied Mathematical Sciences*, vol. 147, November 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] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

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

[50] H. Amann, *Linear and Quasilinear Parabolic Problems, Volume I: Abstract Linear Theory*. BirkhŁuser, Basel, 1995.

[51] M. Protter and H. Weinberger, *Maximum Principles in Differential Equations*. Prentice-Hall, Englewood Cliffs, NJ, 1967.

[52] E. Zeidler, *Nonlinear Functional Analysis and its Applications. Part 1: Fixed-Point Theorems*. Springer New York, 1998.

PLACE PHOTO HERE

Paolo Favaro Paolo Favaro received the D.Ing. degree from the University of 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.

PLACE PHOTO HERE

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.

PLACE PHOTO HERE

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.

PLACE PHOTO HERE

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.