3D Shape from Anisotropic Diffusion

P. Favaro
Dept. of Electrical
Engineering
Washington University
St. Louis, MO 63130
fava@ee.wustl.edu

S. Osher
Dept. of Applied
Mathematics
UCLA
Los Angeles, CA 90095
sjo@math.ucla.edu

S. Soatto
Dept. of
Computer Science
UCLA
Los Angeles, CA 90095
soatto@ucla.edu

L. Vese
Dept. of Applied
Mathematics
UCLA
Los Angeles, CA 90095
lvese@math.ucla.edu

Abstract

We cast the problem of inferring the 3D shape of a scene from a collection of defocused images in the framework of anisotropic diffusion. We propose a novel algorithm that can estimate the shape of a scene by inferring the diffusion coefficient of a heat equation. The method is optimal, as we pose it as the minimization of a certain cost functional based on the input images, and fast. Furthermore, we also extend our algorithm to the case of multiple images, and derive a 3D scene segmentation algorithm that can work in the presence of pictorial camouflage.

1. Introduction

When imaging a scene through a lens, objects at different distances are blurred by different amounts, and therefore a collection of defocused images contains information about the shape of the scene. The problem of retrieving the 3D shape, or depth map, of a scene, given two or more blurred images is known as depth/shape from defocus in the field of computer vision. The typical assumption is to approximate locally the depth map of the scene with a plane parallel to the focal plane [4, 12, 15, 16, 18, 14, 6], although work has also been done in more general cases [3, 8].

In this paper we present a novel algorithm to optimally recover shape from two defocused images. 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 [9]. The idea consists of running a forward heat equation on regions where one image is more focused than the other, until both images become identical. The diffusion coefficient required to match the two images at any given point has a one-to-one correspondence with the depth of the scene at that point. Although we pose our problem as a cost functional minimization, our algorithm is surprisingly fast given the dimension of the space of the

unknowns at play. The minimization is performed by a few iterations of a gradient descent, which is initialized by a fast approximated solution.

This algorithm, which can be though of as performing "simulation-based" inference, avoids the proverbial ill-posedness associated to reconstructing shape from defocus. The latter corresponds to solving a blind deconvolution task, an inverse problem in the infinite-dimensional space of unknown shapes and radiances of the scene. Our algorithm, instead, returns an estimate of the shape of the entire scene without the need for time-consuming alternating minimizations that are typically employed in blind deconvolution problems.

The literature on anisotropic diffusion is quite substantial and, therefore, this work relates to a large number of other works. In particular, to the extensive literature in image processing, for instance [7, 11, 1, 13, 17] and references therein.

We also propose an extension of our algorithm to the case when more defocused images are available. We consider that images are collected through a controlled change in the focus setting, for example by moving the lens clockwise at a constant velocity and by capturing images at a fixed frame rate. Our algorithm yields a segmentation of the scene based on its shape, and not on pictorial cues, and can therefore be useful in the presence of camouflage, where pictorial segmentation is challenging.

2. Defocusing via diffusion

In this section, we first introduce the convolutional model for defocusing. In the simple case of uniform blurring, that occurs whenever we are imaging a plane parallel to the focal plane, the convolutional model is known to be shift-invariant. Then, we recall that the same model can also be formulated as the (isotropic) heat equation, within the context of partial differential equations (PDE). However,

1

when blurring is not uniform, the isotropic heat equation cannot be employed. As an alternative, we propose to use the anisotropic diffusion equation. Based on the properties of this parabolic differential equation, we devise a scheme to infer the shape of the scene by reconstructing the space-varying diffusion coefficients.

2.1. A convolutional model for defocusing

Consider a scene with a smooth *Lambertian*[1] surface S. We take images of the scene from the same point of view and assume that the scene and the illumination are static with respect to the camera . Under these conditions we can represent the surface S with a function $s : \mathbb{R}^2 \mapsto \mathbb{R}^+$, and the radiance map (or, equivalently, the reflectance map) on the surface S with a function $r : \mathbb{R}^2 \mapsto \mathbb{R}_+$. If we use a real aperture camera, the irradiance (the image intensity) $I : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}_+$ measured on the image plane with focal distance p of the optics (the distance between the lens plane and the plane focused by the lens), can be approximated by the following equation:

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

where $h : \Omega \times \mathbb{R}^2 \times \mathbb{R}_+ \mapsto \mathbb{R}_+$ is the *point spread function* (PSF) of the optics. The point spread function depends on the *blurring radius* b that is function of the focal distance p, and the distance between the surface S and the image plane. The blurring radius b determines the size of the region of the radiance r that contributes to the image intensity $I(\mathbf{y})$ at each point $\mathbf{y} \in \Omega$. Under some hypotheses (see [16]) on the scene and the focal settings, the map between b and the surface S is monotonic. Therefore, the problem of inferring the surface S from blurred images, is equivalent to the problem of inferring the blurring b. From the *energy conservation principle*, h is a function that satisfies the normalization property

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

for any shape and parameters of the optics. When we are imaging an *equifocal plane*[2], we have $s(\mathbf{x}) = s \ \forall \mathbf{x} \in \Omega$. The PSF h turns out to be a shift-invariant function, i.e. $h(\mathbf{x}, \mathbf{y}, b(s(\mathbf{x}), p)) = h(\mathbf{x} - \mathbf{y}, b(s, p))$ and $b(s, p)$ is a constant. It can be argued, by using notions of radiometry, that a good approximation for this PSF is the Gaussian function, with variance $\sigma^2 \doteq \gamma b$ for a constant $\gamma > 0$ (this constant can be determined with a calibration procedure). The Gaussian function has been widely used in the literature of depth from defocus [3].

---

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

[2] An equifocal plane is a plane parallel to the lens plane.

2.2. Equifocal imaging as isotropic diffusion

When the PSF is approximated by a shift-invariant Gaussian function, it is well-known [9] that the imaging model of Eq. (1) can be formulated in terms of the *isotropic heat equation*:

$$\begin{cases} \dot{u}(\mathbf{y}, t) = c \triangle u(\mathbf{y}, t) & c \in \mathbb{R}_+ \ \ t \geq 0 \\ u(\mathbf{y}, 0) = r(\mathbf{y}) & \forall \mathbf{y} \in \Omega \end{cases} \quad (3)$$

where the solution $u : \mathbb{R}^2 \times \mathbb{R}_+ \mapsto \mathbb{R}_+$ taken at a specific time $t = \tau$, plays the role of an image $I(\mathbf{y}) = u(\mathbf{y}, \tau) \ \forall \mathbf{y} \in \Omega$. Since we restrict ourselves to solutions that represent images, which are spatially bounded, we are guaranteed that a solution to (3) exists and is unique. The dot-notation is defined as $\dot{u} \doteq \frac{\partial u}{\partial t}$, and the symbol $\triangle$ denotes the Laplacian operator $\sum_{i=1}^{2} \frac{\partial^2}{\partial y_i^2}$ with $\mathbf{y} \doteq [y_1 \ y_2]$. The parameter c is called the *diffusion coefficient* and it is nonnegative. By computing the Green's function[3] relative to the isotropic heat equation (3), it is easy to verify that the blurring parameter b is related to the diffusion coefficient via

$$b = \frac{2tc}{\gamma}. \quad (5)$$

Notice that the diffusion coefficient c and the time variable t are determined up to a common scale factor. Later, we will solve this ambiguity by fixing the time t to a reference time, and scaling c accordingly, or, vice versa, by fixing c and scaling the time t.

Now, suppose that we collect two images I_1 and I_2 for two different blurring parameters b_1 and b_2. Using Eq. (5) with a fixed diffusion coefficient c, we can obtain the two time instants t_1 and t_2, so that the solution of Eq. (3) verifies $I_1(\mathbf{y}) = u(\mathbf{y}, t_1)$ and $I_2(\mathbf{y}) = u(\mathbf{y}, t_2) \ \forall \mathbf{y} \in \Omega$. Now, suppose $b_1 < b_2$ (i.e. image I_1 is more focused than image I_2), then also $t_1 < t_2$ from Eq. (5), and we can substitute Eq. (3) with

$$\begin{cases} \dot{u}(\mathbf{y}, t) = c \triangle u(\mathbf{y}, t) & c \in \mathbb{R}_+ \\ u(\mathbf{y}, t_1) = I_1(\mathbf{y}) & \forall \mathbf{y} \in \Omega \end{cases} \quad (6)$$

and have that the solution u verifies $u(\mathbf{y}, t_2) = I_2(\mathbf{y}) \ \forall \mathbf{y} \in \Omega$. Similarly, if $t_1 > t_2$, the same equations hold by switching the role of the images I_1 and I_2.

The isotropic diffusion of Eq. (6) models the *relative blurring* between image I_1 and image I_2. In fact, by using Eq. (1) in the shift-invariant Gaussian case, the relative

---

[3] The Green's function $G : \Omega \times \mathbb{R}^2 \times \mathbb{R}_+ \mapsto \mathbb{R}_+$ corresponding to Eq. (3) is defined as the *impulse response* of Eq. (3), i.e. $G(\mathbf{y}, \mathbf{x}, t) = u(\mathbf{y}, t)$ where $u(\mathbf{y}, t)$ is the solution of Eq. (3) with initial conditions $u(\mathbf{y}, 0) = \delta(\mathbf{x} - \mathbf{y})$. The function G satisfies the equation

$$u(\mathbf{y}, t) = \int_{\mathbb{R}^2} G(\mathbf{y}, \mathbf{x}, t) u(\mathbf{x}, 0) d\mathbf{x} = \int_{\mathbb{R}^2} G(\mathbf{y}, \mathbf{x}, t) r(\mathbf{x}) d\mathbf{x}. \quad (4)$$

2

blurring $\Delta b \doteq b_1 - b_2$ verifies

$$\Delta b = \frac{2(t_1 - t_2)c}{\gamma} = \frac{2\Delta tc}{\gamma}. \tag{7}$$

One can view the relative time Δt as the variable encoding the change in focal settings, and the diffusion coefficient c as the variable encoding the shape of the surface S.

The problem of inferring shape from defocus, can then be posed as the following minimization:

$$\hat{c} = \arg \min_{c} \int_{\mathbb{R}^2} \phi(u(\mathbf{y}, t_2), I_2(\mathbf{y})) d\mathbf{y} \tag{8}$$

where $u(\mathbf{y}, t)$ is the solution of (6), and ϕ is a discrepancy measure.

2.3. Non-equifocal imaging model

When the surface S is not an equifocal plane, the corresponding PSF is, in general, shift-variant, and the equivalence with the isotropic heat equation does not hold. Rather than seeking an approximation for the shift-variant PSF, we propose a model based on a generalization of the isotropic heat equation that satisfies the energy conservation principle (2). To take into account the space-variant nature of the non-equifocal case, we propose using the *anisotropic diffusion equation*, and define a diffusion tensor $c \doteq \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}$ where $c_{ij} : \mathbb{R}^2 \mapsto \mathbb{R}$ for $i, j = 1, 2$. We assume that $c_{ij} \in C^1(\mathbb{R}^2)$ (i.e. the space of functions with continuous partial derivatives in $\mathbb{R}^2$) for $i, j = 1, 2$, and[4] $c(\mathbf{y}) \geq 0$ $\forall \mathbf{y} \in \mathbb{R}^2$. Let the open set $\mathcal{O} \doteq \{\mathbf{y} : c(\mathbf{y}) > 0\} \subset \Omega$, and assume $\mathcal{O}$ is such that $\mathcal{O} \in C^1$ (i.e. $\mathcal{O}$ is bounded and the boundary of $\mathcal{O}$ can be locally mapped by functions in $C^1(\mathbb{R})$). The anisotropic diffusion equation is then defined as

$$\begin{cases} \dot{u}(\mathbf{y}, t) = \nabla \cdot (c(\mathbf{y})\nabla u(\mathbf{y}, t)) & t \geq 0 \\ u(\mathbf{y}, t_1) = I_1(\mathbf{y}) & \forall \mathbf{y} \in \Omega \end{cases} \tag{9}$$

where the symbol ∇ is the gradient operator $\begin{bmatrix} \frac{\partial}{\partial y_1} \\ \frac{\partial}{\partial y_2} \end{bmatrix}$ with $\mathbf{y} = [y_1 \ y_2]$, and the symbol $\nabla \cdot$ is the divergence operator $\sum_{i=1}^{2} \frac{\partial}{\partial y_i}$. Here we assume for simplicity that I_1 is more focused than I_2, i.e. $t_1 < t_2$. It is easy to verify that (9) satisfies the energy conservation principle (2) by using the divergence theorem and the assumptions on the diffusion tensor (for more details see [5]).

By assuming that the surface S is smooth, we can relate the diffusion tensor c to a *relative blurring tensor* Δb (a 2-dimensional covariance matrix of the Gaussian kernel) via:

$$\Delta b(s(\mathbf{y}), p) \simeq \frac{2\Delta tc(\mathbf{y})}{\gamma} \tag{10}$$

---

[4]Since c is a tensor, the notation $c(\mathbf{y}) \geq 0$ means that $c(\mathbf{y})$ is positive semi-definite.

and the solution $u(\mathbf{y}, t)$ of Eq. (9) verifies $u(\mathbf{y}, t_2) \simeq I_2(\mathbf{y})$.

The original problem (8) can then be posed again as

$$\hat{c} = \arg \min_{c} \int_{\mathbb{R}^2} \phi(u(\mathbf{y}, t_2), I_2(\mathbf{y})) d\mathbf{y} \tag{11}$$

where $u(\mathbf{x}, t)$ is now the solution of (9). This problem is known to be ill-posed [10], and regularization needs to be added in order to retrieve the diffusion tensor c:

$$\hat{c} = \arg \min_{c} \int_{\mathbb{R}^2} \phi(u(\mathbf{y}, t_2), I_2(\mathbf{y})) d\mathbf{y} + \alpha \int_{\mathbb{R}^2} \|\nabla c(\mathbf{y})\|^2 d\mathbf{y} \tag{12}$$

where $\alpha > 0$ is a *tuning* parameter that determines the degree of regularization of c, and

$$\|\nabla c(\mathbf{y})\|^2 = \sum_{i,j=1}^{2} \|\nabla c_{ij}(\mathbf{y})\|^2. \tag{13}$$

In general, the relative blurring tensor Δb may be neither positive definite nor negative definite, so that the diffusion tensor c does not satisfy $c \geq 0$, and the initial assumption that I_1 is more focused than I_2 is not true over the whole Ω. In this case, the minimization (12) would involve backward diffusion as well as forward diffusion. We avoid the instability issues of backward diffusions, by defining two diffusion tensors c_+ and c_-, and a partition $\{\mathcal{O}_-, \mathcal{O}_+\}$ of $\mathbb{R}^2$, such that they satisfy $c_+(\mathbf{y}) = c(\mathbf{y}) \geq 0 \ \forall \mathbf{y} \in \mathcal{O}_+$ and $c_-(\mathbf{y}) = -c(\mathbf{y}) \geq 0 \ \forall \mathbf{y} \in \mathcal{O}_-$. Then, we can divide the original model (9) into the following two models:

$$\begin{cases} \dot{u}(\mathbf{y}, t) = \nabla \cdot (c_+(\mathbf{y})\nabla u(\mathbf{y}, t)) & t \geq 0 \\ u(\mathbf{y}, 0) = I_1(\mathbf{y}) & \forall \mathbf{y} \in \mathcal{O}_+ \\ u(\mathbf{y}, \Delta t_+) \simeq I_2(\mathbf{y}) & \forall \mathbf{y} \in \mathcal{O}_+ \end{cases} \tag{14}$$

and

$$\begin{cases} \dot{u}(\mathbf{y}, t) = \nabla \cdot (c_-(\mathbf{y})\nabla u(\mathbf{y}, t)) & t \geq 0 \\ u(\mathbf{y}, 0) = I_2(\mathbf{y}) & \forall \mathbf{y} \in \mathcal{O}_- \\ u(\mathbf{y}, \Delta t_-) \simeq I_1(\mathbf{y}) & \forall \mathbf{y} \in \mathcal{O}_- \end{cases} \tag{15}$$

which satisfy the initial assumptions.

Similarly, we can also divide the original problem (12) into the two subproblems:

$$\hat{c}_+ = \arg \min_{\substack{c_+(\mathbf{y}) \geq 0 \\ \forall \mathbf{y} \in \mathcal{O}_+}} \int_{\mathcal{O}_+} \phi(u(\mathbf{y}, \Delta t_+), I_2(\mathbf{y})) d\mathbf{y} + \alpha_+ \int_{\mathbb{R}^2} \|\nabla c_+(\mathbf{y})\|^2 d\mathbf{y} \tag{16}$$

3

and

$$\hat{c}_- = \arg \min_{\substack{c_-(\mathbf{y}) \geq 0 \\ \forall \mathbf{y} \in \mathcal{O}_-}} \int_{\mathcal{O}_-} \phi(u(\mathbf{y}, \Delta t_-), I_1(\mathbf{y})) d\mathbf{y} + \alpha_- \int_{\mathbb{R}^2} \|\nabla c_-(\mathbf{y})\|^2 d\mathbf{y} \tag{17}$$

with some constants $\alpha_+ > 0$ and $\alpha_- > 0$.

Notice that in both problems (16) and (17) the diffusion tensors c_+ and c_- are constrained to be positive semidefinite, and therefore, the corresponding models (14) and (15) involve only forward diffusions. An additional benefit of this formulation, that we will see in Section 3, is that when solving these two problems, we can obtain a fast approximated solution by simulating the corresponding model just once.

3. Shape from defocus via anisotropic diffusion

As we have seen in the previous sections, the problem of estimating the depth map s of a scene from blurred images, can be posed as the problem of inferring the diffusion tensors c_- and c_+.

For now, consider Eq. (14) and the corresponding problem (16) on c_+. For ease of notation, we define

$$\int_{\mathbb{R}^2} \psi(c_+(\mathbf{y})) d\mathbf{y} = \int_{\mathbb{R}^2} \phi(u(\mathbf{y}, \Delta t_+), I_2(\mathbf{y})) d\mathbf{y} + \alpha_+ \int_{\mathbb{R}^2} \|\nabla c_+(\mathbf{y})\|^2 d\mathbf{y}. \tag{18}$$

The evolution of the diffusion coefficient c_+ can be performed via the gradient descent

$$\dot{c}_+(\mathbf{y}) = -\kappa (\nabla \psi(c_+))(\mathbf{y}) \tag{19}$$

with $\kappa > 0$ a scalar tuning parameter. It can be shown that in the case of $\phi(a, b) = (a - b)^2$, where $a, b \in \mathbb{R}$, the gradient $\nabla \psi(c_+)$ is (for more details see [5]):

$$(\nabla \psi(c_+))(\mathbf{y}) = -2 \int_0^{\Delta t_+} \nabla v(\mathbf{y}, \tau) \nabla u(\mathbf{y}, \tau)^T d\tau - \alpha_+ \triangle c_+(\mathbf{y}) \tag{20}$$

where $(\cdot)^T$ denotes the transpose of a vector, and $v : \mathbb{R}^2 \times \mathbb{R}_+ \mapsto \mathbb{R}$ is defined as

$$v(\mathbf{y}, \tau) \doteq \int_{\mathbb{R}^2} G(\mathbf{y}, \mathbf{x}, \Delta t_+ - \tau)(u(\mathbf{x}, \Delta t_+) - I_2(\mathbf{x})) d\mathbf{x}. \tag{21}$$

G is the Green's function associated with u in Eq. (14).

Once $u(\mathbf{y}, \tau)$ has been computed by simulating model (14) with the current estimate of c_+, we can compute $v(\mathbf{y}, \tau)$ by simulating the same model, but with initial conditions $u(\mathbf{y}, \Delta t_+) - I_2(\mathbf{y})$.

Since in this paper we are interested in deriving a fast scheme for 3D shape estimation, we consider only the case of a scalar diffusion coefficient c, which is equivalent to having $c_{ij} = c$ for $i, j = 1, 2$. In this case the gradient above becomes simply:

$$(\nabla \psi(c_+))(\mathbf{y}) = -2 \int_0^{\Delta t_+} \nabla v(\mathbf{y}, \tau)^T \nabla u(\mathbf{y}, \tau) d\tau - \alpha_+ \triangle c_+(\mathbf{y}) \tag{22}$$

which reduces of about four times the total amount of required computations. Furthermore, we also propose to initialize the above gradient descent by using a fast approximated solution for c_+. As a first step, we substitute the integration in τ with a summation over only the two time instants $\tau = 0$ and $\tau = \Delta t_+$. Then, we approximate the Green's function $G(\mathbf{y}, \mathbf{x}, \Delta t_+)$ with a shift invariant Gaussian kernel with variance $\frac{2\Delta t_+ c_M}{\alpha}$, where $c_M = \max_{\mathbf{y} \in \mathcal{O}_+} c_+(\mathbf{y})$. We call $(\nabla \varphi(c_+))(\mathbf{y})$ the approximated gradient.

To compensate for all the simplifications made so far, we propose substituting the gradient iteration (19) with the following iteration

$$\dot{c}_+(\mathbf{y}) = \begin{cases} \kappa & \text{if } (\nabla \varphi(c_+))(\mathbf{y}) < 0 \\ 0 & \text{if } (\nabla \varphi(c_+))(\mathbf{y}) \geq 0. \end{cases} \tag{23}$$

Then, to further reduce the amount of computations for this approximated solution, we change the initial condition of model (14) with the solution $u(\mathbf{y}, \tau)$ computed at the previous gradient descent step and use $\dot{c}_+$ as diffusion coefficient instead of c_+. Finally, we obtain c_+ by integrating (23) in time. In this way, we obtain a first initial diffusion coefficient c_+ by running a single forward simulation of model (14), and by simultaneously guaranteeing that $c_+ \geq 0$.

Although the iteration above has been derived by using very crude approximations, we will see in the experimental section that it returns an initial estimate of the diffusion coefficient already very close to the ground truth (which is available in the case of synthetic data).

We use this estimate of c_+ to initialize iteration (19). Then, from Eq. (19) the total amount of diffusion c_+ can be obtained by integrating $\dot{c}_+$ in time. The same scheme applies to the diffusion coefficient c_-. We also impose that the diffusion coefficients c_+ and c_- are strictly positive only on disjoint regions, as the sets $\mathcal{O}_+$ and $\mathcal{O}_-$ are disjoint by definition. We do that by setting to zero the smallest of the two coefficients on the regions where the gradient iterations make both coefficients positive. Finally, the surface S can be derived directly from c_+ and c_- via Eq. (10).

In short, one can see the proposed scheme as a technique to diffuse two given blurred images I_1 and I_2 in the regions where one is more focused than the other, until they both become identical. The amount of diffusion required to match

4

the two images carries information about the shape of the scene.

4. 3D shape segmentation

When it is possible to collect N images in a controlled way, for example by moving the lens clockwise at a constant velocity and by capturing images at a fixed frame rate, we can reconstruct the surface S of the scene by using a partitioning scheme derived from the previous section.

Consider two images I_1 and I_2 from the N images available, captured with focal planes at depth p_1 and p_2 respectively, and suppose that the depth map of the scene is contained between p_1 and p_2. This means that

$$p_1 < s(\mathbf{x}) < p_2 \quad \forall \mathbf{x} \in \Omega. \tag{24}$$

The blurring radius b is typically defined as

$$b(s(\mathbf{x}), p_i) = \beta D \left| 1 - \frac{s(\mathbf{x})}{p_i} \right| \quad i = 1, 2 \tag{25}$$

with β a scale factor, and D the lens diameter. Then, points where the amount of blurring of the two images is the same, i.e. where $b(s(\mathbf{x}), p_1) = b(s(\mathbf{x}), p_2)$, lie on the equifocal plane at depth

$$Z = \frac{2p_1 p_2}{p_1 + p_2}. \tag{26}$$

Points that belong to regions where I_2 is more focused than I_1 correspond to depths larger than (26), while points that belong to regions where I_1 is more focused than I_2 correspond to depths smaller than (26). We call the collection of the first group of points the *background*, and the collection of the second group of points the *foreground*.

Our procedure consists in computing the diffusion coefficients c_+ and c_-, and then setting the foreground as the region where the estimated diffusion coefficient c_+ is strictly positive and the background as the region where c_- is strictly positive. We have seen experimentally that segmentation is achieved with sufficient precision by iterating only once the approximated gradient descent (22).

By repeating the segmentation on other couples of images satisfying the constraint (24) we can build a "layered" representation of the surface of the scene. Suppose that the N images have been collected by increasing the focal plane depth p of a constant step Δp, so that the i-th image corresponds to the focal plane depth $p_i = p + i\Delta p$. During the segmentation procedure, we consider two images with focal depth p_i and p_{i+M} with $i = 1 \ldots N - M$. We choose M so that (24) is satisfied for all $i = 1 \ldots N - M$. The couple $(i, i + M)$ defines the equifocal plane at depth $Z_i = \frac{2p_i p_{i+M}}{p_i + p_{i+M}}$ as prescribed by (26). Then, the region of the image domain, whose depth is between Z_j and Z_{j+1}, will be segmented as background when the couple $(j, j + M)$ is considered, and as foreground when the couple $(j + 1, j + 1 + M)$ is considered.

5. Experiments

To test our algorithms we use both synthetically generated and real images. First, we show experiments by using the gradient descent described in Section 3 on sets of 2 images. Then, we show experiments on a set of 60 images by implementing the 3D shape segmentation algorithm described in the previous section.

5.1. Experiments: anisotropic diffusion

We numerically implement the gradients required by our algorithm by using forward and backward finite difference schemes. We will not give further details, as this implementation follows a very standard scheme, that can be found in [2], for example.

Synthetic images: In our synthetic setup (see Figure 1) we generate a "wave-shaped" surface and a radiance with intensities that are uniformly distributed between 50 and 150, where the gray-scale ranges between 0 and 255. We capture two images with two different focal settings. In this case a shift-invariant approach would fail to capture the depth variation of the scene, especially at the ridges of the waves (see Figure 3). We retrieve a first estimate of the surface by employing the approximated gradient in Eq. (22). Figure 2 shows 6 snapshots of the surface estimation in gray-levels (white corresponds to small depth values, while black corresponds to large depth values). Notice that the surface closely resembles the correct shape (see Figure 1), although it has been estimated via the approximated gradient. An alternative visualization of the surface evolution is given in Figure 3, where it is easier to perceive the reconstructed 3D geometry. Finally, in Figure 4 we show the final shape after applying the gradient descent (22), texture mapped with one of the input images. We found that the difference between the final shape and the one obtained from the approximated iteration is minimal.

Real images: Our real data setup (see Figure 5) is composed of a cylinder and a square sponge placed on top of a box. The objects lie between $0.6m$ and $0.9m$ in front of the camera, which is equipped with a Nikon AF NIKKON $35mm$ lens. We capture two images by focusing first at $0.6m$ and then at $0.9m$ (see Figure 5). Figure 6 shows 6 snapshots of the surface estimation in gray-levels. The same evolution is rendered in 3D in Figure 7. Notice that at the edges of the sponge the segmentation fails, due to the fact that the diffusion coefficient is not smooth and the image formation process cannot be captured by Eq. (1). Finally, in Figure 8 we show the final shape after applying the gradient descent (22), texture mapped with one of the input images.

5.2. Experiments: 3D shape segmentation

We test the 3D shape segmentation algorithm on real images from the scene in Figure 9. We capture 60 images by
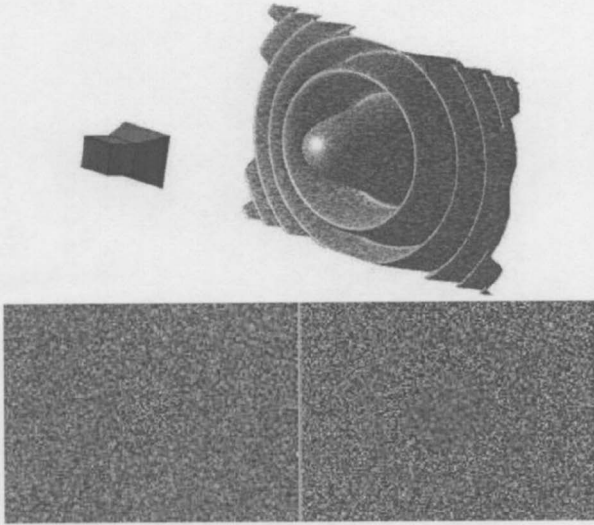
5

Figure 1: *Top: setup for the synthetic scene. Bottom: 2 input images synthetically generated.*
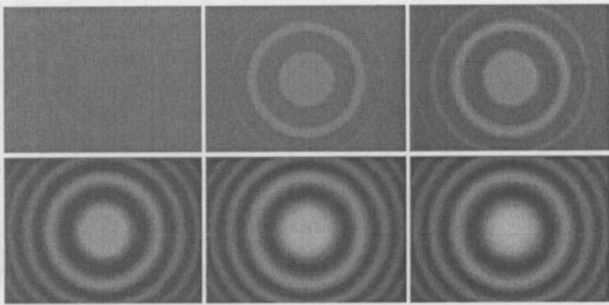

Figure 2: *6 snapshots from the evolution of the estimated surface by using the approximated gradient in Eq. (22). The depth map is shown in gray-levels (white corresponds to small depth values, while black corresponds to large depth values).*
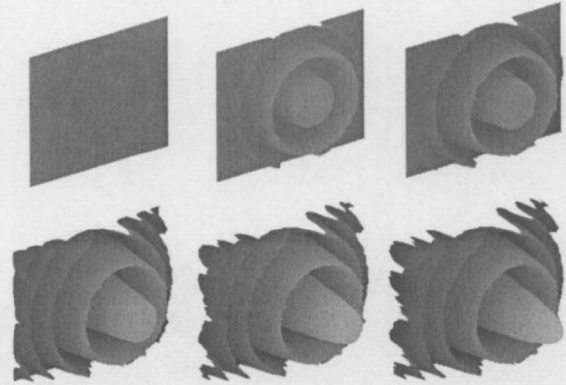

Figure 3: *6 snapshots from the evolution of the surface estimation by using the approximated gradient in Eq. (22) are shown as a gray-level mesh (white corresponds to small depth values, black corresponds to large depth values).*
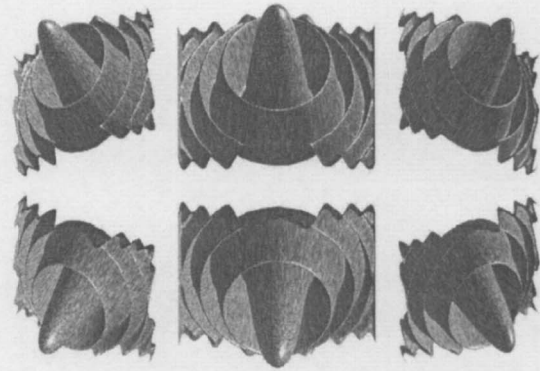
Figure 4: *6 novel views of the estimated shape with texture mapping, after refining the diffusion coefficients with Eq. (20).*

changing the focal setting of a constant step. In Figure 9 we show a sample of 2 of the 60 images. We choose $M = 33$ and segment the 3D shape in 27 levels (see Figure 10 and Figure 11). 6 snapshots of the reconstructed surface after smoothing are then shown in Figure 12.

6. Summary and Conclusions

We have presented novel algorithms to infer shape from defocus and 3D segmentation based on true shape and not on pictorial cues. The first algorithm retrieves 3D shape of an entire scene by inferring the diffusion coefficient of a heat equation. Inference is performed by a gradient descent minimization and numerical instabilities are avoided by consid-ering only forward diffusions. The algorithm is robust, in view of its "simulation" nature, and fast because it avoids alternating minimization, usually employed in blind deconvolution algorithms. The second algorithm segments the 3D shape of the scene based on a partitioning procedure derived from the properties of anisotropic diffusion.
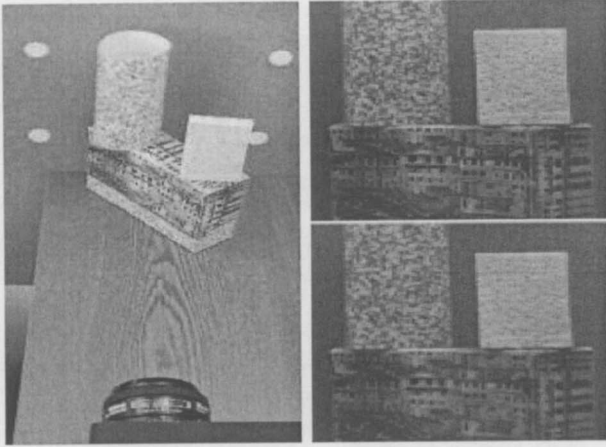
Acknowledgments

Figure 5: *Left: setup of the real scene. Right: 2 input images with different focal settings.*
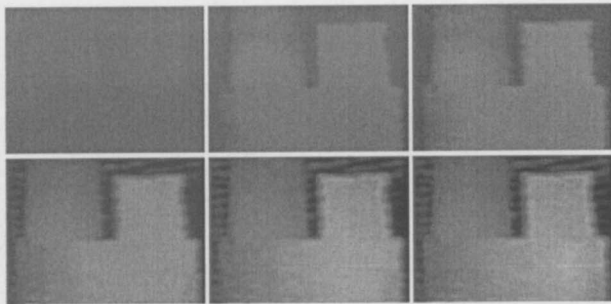

Figure 6: *6 snapshots from the evolution of the estimated surface obtained by employing the approximated gradient in Eq. (22). The depth map is shown in gray-levels (white corresponds to small depth values, while black corresponds to large depth values).*

References

[1] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*, 7(3):421–32, 1998.

[2] T. Chan and L. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–77, 2001.

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

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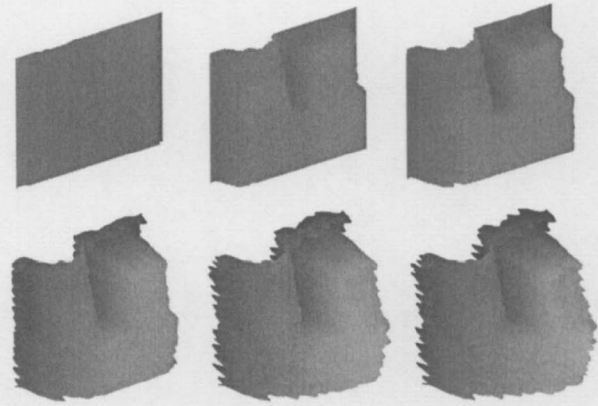

Figure 7: *6 snapshots from the evolution of the surface estimation by using the approximated gradient in Eq. (22) are shown as a gray-level mesh (white corresponds to small depth values, black corresponds to large depth values).*
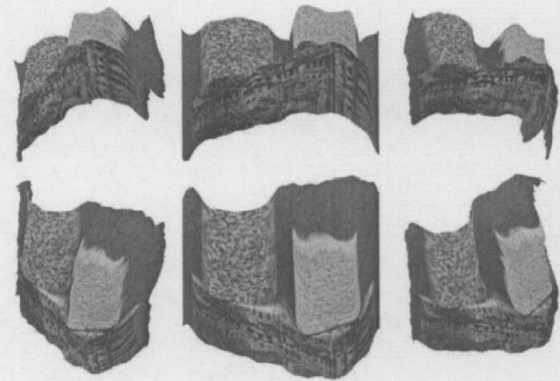

Figure 8: *6 novel views of the estimated shape with texture mapping, refined by applying the gradient descent (20).*

[5] P. Favaro, S. Osher, L. Vese, and S. Soatto. 3d shape from anisotropic diffusion. In *Technical Report UCLA*, January 2003.

[6] P. Favaro and S. Soatto. Shape and reflectance estimation from the information divergence of blurred images. In *European Conference on Computer Vision*, pages 755–768, June 2000.

[7] 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*, 19(4):342–52, 1997.

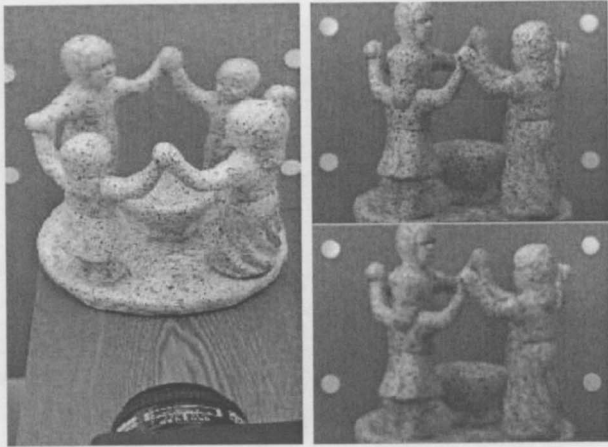[8] H. Jin and P. Favaro. A variational approach to shape from defocus. In *ECCV (2)*, pages 18–30, 2002.

IEEE
COMPUTER
SOCIETY

Figure 9: *Left: setup of the real scene. Right: 2 input images among the 60 that have been collected.*
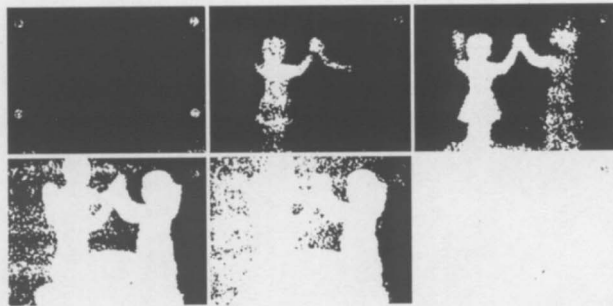

Figure 10: *6 snapshots of the segmentation foreground/background of the 3D shape by using the technique described in Section 4.*
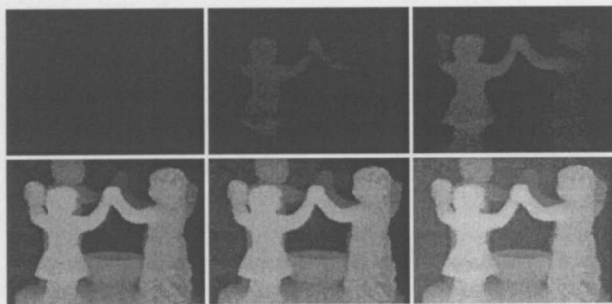

Figure 11: *6 snapshots from the evolution of the estimated surface. The depth map is shown in gray-levels (white corresponds to small depth values, while black corresponds to large depth values).*
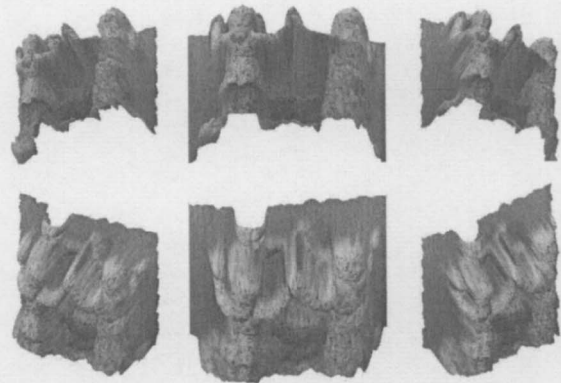

Figure 12: *6 novel views of the estimated shape with texture mapping, after smoothing.*

[10] R. Lagnado and S. Osher. A technique for calibrating derivative security pricing models: numerical solution of an inverse problem. *J. Comput. Finance*, 1(1):13–26, 1997.

[11] S. Osher and S. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulation. *J. of Comput. Phys.*, 79:12–49, 1998.

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

[13] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–39, 1990.

[14] S. Soatto and P. Favaro. A geometric approach to blind deconvolution with application to shape from defocus. In *Intl. Conf. on Computer Vision and Pattern Recognition*, pages 10–17, June 2000.

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

[16] M. Watanabe and S. Nayar. Rational filters for passive depth from defocus. *Intl. J. of Comp. Vision*, 27(3):203–225, 1998.

[17] J. Weickert. *Anisotropic Diffusion in Image Processing*. B.G.Teubner Stuttgart, 1998.

[18] Y. Xiong and S. Shafer. Depth from focusing and defocusing. In *Proc. of the Intl. Conf. of Comp. Vision and Pat. Recogn.*, pages 68–73, 1993.

[9] J. J. Koenderink. The structure of images. *Biol. Cybern.*, 50:363–370, 1984.