Seeing Beyond Occlusions
(and other marvels of a finite lens aperture)

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

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

Keywords: shape from defocus, early vision, image-based modeling, illumination and reflectance modeling, shape representation

Abstract

We present a novel algorithm to reconstruct the geometry and photometry of a scene with occlusions from a collection of defocused images. The presence of a finite lens aperture allows us to recover portions of the scene that would be occluded in a pin-hole projection, thus "uncovering" the occlusion. We estimate the shape of each object (a surface, including the occluding boundaries), and its radiance (a positive function defined on the surface, including portions that are occluded by other objects).

1. Introduction

In computer vision, we often try to suppress the fact that real cameras have a lens, and try to model images as if they were taken through a pin-hole[1]. However, lenses can do marvels. Images generated with a finite aperture contain spatial information that is lost in the ideal perspective projection. For instance, a lens on a finite aperture allows us to "see behind" occluding boundaries (Fig. 1) and can generate images that contain unequivocal information on the three-dimensional structure of the scene, given certain assumptions on the radiance and the shape in the scene that are discussed in detail in [8]. Such an effect depends on the aperture, the focal length, the size of the occluding object and its relative distance to the background, as will be seen in the Experiments section.

In this paper we introduce a model of the imaging process for an occlusion seen from an idealized thin lens. The measured images depend upon both the shape of the scene (which is unknown and non-smooth because of occlusions) and its radiance (which is also unknown and non-smooth). We present a novel algorithm that allows us to recover both the shape of the scene, the location of the occluding boundary, and its radiance, including portions of it which are occluded when imaged through a pin-hole. Since the unknowns (shape and radiance) are functions that live in infinite-dimensional spaces, our algorithm entails the solution of partial differential equations. Work in this area is, to the best of our knowledge, novel. We test our algorithm on both real and synthetic image sequences, the latter with ground truth.

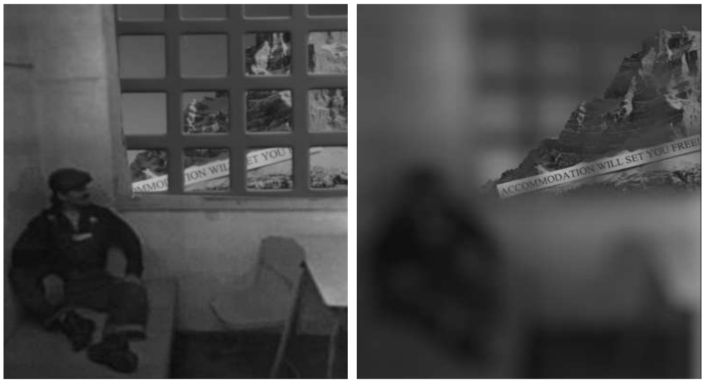

Figure 1: *The "pinhole prison": occluded portions of the background scene are not visible using a pinhole camera (left). Using a finite aperture and a lens, however, allows one to see past the occlusion (right). A similar effect can be seen in the scene in Figure 8.*

1.1. Relation to previous work and main contributions

The human eye comes equipped with a very sophisticated lens that can be actuated to change shape and aperture; such a sophistication comes at a cost, as everyone wearing eyeglasses knows (about 60% of the population in industrialized countries). However, it also comes with a consider-

[1] For some recent work on imaging models beyond the pin-hole approximation, see [1]

able benefit, since the variation of measured images under changing lens geometry conveys spatial information about the environment, as it has been shown in engineering practice [6, 7, 9, 14, 16, 17, 18, 20, 21] and theory [8]. However, of the visual field of the eye only a small portion is actually suited to processing variations associated with lens geometry, corresponding to about six degrees of visual field (the "fovea"). Therefore, it is evident that the human lens system is particularly suited to detecting large variations in depth that project in neighboring portions of the visual field [10]; this occurs at occluding boundaries (or "silhouettes"), as we show in Figure 1.

Despite the fact that occluding boundaries are particularly important, most of the research in this area of Computer Vision has so far concentrated to processing visual data away from occluding boundaries. Typically, a surface is approximated locally by *equifocal* planes[2]. This approximation results in a shift-invariant imaging model, which is violated at occluding boundaries. Therefore, most "shape from focus" (SFF) and "shape from defocus" (SFD) algorithms convey no spatial information exactly where it matters the most: at the occluding boundary.

To our knowledge, very little work has been done along this direction. [12] proposes to model occlusions by the *multicomponent blurring model*, where the image is a linear combination of the contributions from each surface. Unfortunately, this model holds only when the surfaces are perfectly transparent, i.e. when the energy they emit (reflect) adds linearly on the image plane. [19] only considers the problem of detection of a boundary. [3] analyzes the specific case of self-occlusions of a scene made of a single surface and a single radiance, and proposes an algorithm to improve the estimation of the depth of the scene. Both the surface and the radiance of a scene are modeled as Markov random fields and both of these fields are recovered using a maximum a-priori estimator. The algorithm is divided into two parts. First, the depth map of the scene is obtained without explicitly taking into account for occlusions. Second, occlusions are detected by locating discontinuities on the estimated depth map. At those discontinuities, the occlusion effect is compensated by modifying the imaging kernel so that the occluded area does not contribute to the image.

We consider a more general model for the occlusions, where the scene is composed of two distinct smooth surfaces, and where one is occluding the other. On the two surfaces, we define two radiance distributions. In addition, we consider that the occluding surface lies on a support that can be a single compact set or a collection of compact sets (in the next section we will give more details about the model here described). As analyzed in [2], these kind of occlusions cannot be modeled by a single convolutional equation (neither shift-invariant nor shift-variant) or by a linear combination of shift-invariant convolutional equations (as in the multicomponent blurring model of [12]). [2] proposes the *reversed projection blurring* model to describe the effect of occlusions when real aperture cameras are used. This model has been shown to be a good approximation of the physical image formation process, but no attempt has been done to estimate the parameters of the scene. In our solution, we adopt a model derived from the reversed projection blurring model and use it to infer all the geometric and radiometric parameters of the scene in a variational framework. We estimate the parameters by minimizing an energy functional composed of a discrepancy term between the measured images and the estimated images, and a number of regularization terms for the estimated parameters. The minimization is performed by evolving a set of partial differential equations. These minimization methods have been often used in the field of image processing for image restoration, or, more specifically, image deblurring [23, 22]. Our numerical implementation uses level set methods [13]. Since our work entails energy functional minimization evolving surfaces, our work relates also to [5, 4].

2. Modeling occlusions

We consider the scene as being composed of two surfaces: Object 1 and Object 2 (see Figure 2), represented by the functions $s_1 : \Omega_1 \subset \mathbb{R}^2 \mapsto \mathbb{R}_+$ and $s_2 : \Omega_2 \subset \Omega_1 \mapsto \mathbb{R}_+$ respectively. The radiance $r_1 : \Omega_1 \mapsto \mathbb{R}_+$ is defined on s_1, and the radiance $r_2 : \Omega_2 \mapsto \mathbb{R}_+$ is defined on s_2. We assume, without loss of generality, that Object 2 is closer to the lens than Object 1, i.e. that $s_1(\mathbf{x}) > s_2(\mathbf{x}) \;\; \forall \mathbf{x} \in \Omega_2$ (surfaces do not intersect). The intensity at a pixel $\mathbf{y} \in \Gamma \subset Z^2$ on the image plane is obtained by integrating the contribution from the two radiances over specific areas. The size and shape of these areas are determined by the mutual position of Object 1 and Object 2, as illustrated in Figure 2. Before introducing the imaging model, we need to define some functions that will be used in our notation. We denote with H the Heaviside function

$$H(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases} \qquad (1)$$

and with $\bar{H}$ the complement of H, i.e.

$$\bar{H}(z) = \begin{cases} 0, & \text{if } z \geq 0 \\ 1, & \text{if } z < 0. \end{cases} \qquad (2)$$

Since we will collect images of the scene by changing the geometry of the lens, we will use the symbol p as an index that indicates the camera parameters setting. π_p is the projection of s_2 onto s_1 through the point in focus $(-\frac{\mathbf{y}u_p}{v}, u_p)$ (in Figure 2 it is depicted as the point in space where all rays converge). The variable $w_2 : \Omega_1 \mapsto \mathbb{R}$, that we will call the

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

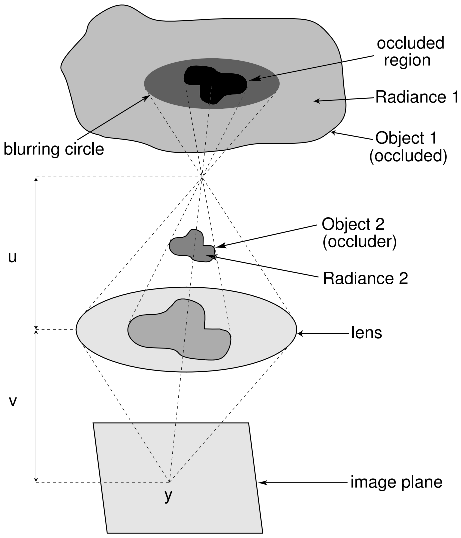

Figure 2: *Object 2 occludes Object 1. The irradiance collected at a point* $\mathbf{y}$ *on the image plane is made of contributions from both Object 1 and Object 2. The radiance of Object 1 is integrated over the blurring circle except for the region occluded by Object 2. This region depends on the projection of Object 2 through the intercept at depth* u *(the point in space where all rays converge). The parameter* v *denotes the distance between the image plane and the lens plane, and it is related to the focal position* u *via the* thin lens law*:* $\frac{1}{v}+\frac{1}{u}=\frac{1}{f}$*, where* f *is the focal of the lens.*

support function, defines the support Ω_2 of r_2 and s_2 via

$$\Omega_2 = \{\mathbf{x} \in \Omega_1 : w_2(\mathbf{x}) \geq 0\} \tag{3}$$

in a similar way as it is done in Chan and Vese [5]. An alternative to Eq. (3) is

$$\Omega_2 = \{\mathbf{x} \in \Omega_1 : H\big(w_2(\mathbf{x})\big) = 1\} \tag{4}$$

which we will use later in the integral expressions of the image formation model. Since we do not have any *a-priori* information about Ω_2, we need to reconstruct it from the data. The support Ω_2 might be a single connected set, or a collection of connected sets. Hence, we also need to be able to handle topological changes during the estimation of Ω_2. As we will see later, the above definition of Ω_2 via the support function w_2 allows for those changes (notice that w_2 can be seen as a level set function). We are now ready to introduce the image model for blurring in presence of occlusions. From the reversed projection blurring model [2], we have that the irradiance measured on the image plane $I_p : \Gamma \subset Z^2 \mapsto \mathbb{R}_+$ can be expressed as:

$$\begin{aligned} I_p(\mathbf{y}) \;&=\; \int_{\Omega_2} h_p(\mathbf{y},\mathbf{x},s_2(\mathbf{x}))r_2(\mathbf{x})d\mathbf{x} + \\ &\quad + \int_{\Omega_1} h_p(\mathbf{y},\mathbf{x},s_1(\mathbf{x}))\bar{H}\big(w_2(\pi_p(\mathbf{x}))\big)r_1(\mathbf{x})d\mathbf{x} \end{aligned} \tag{5}$$

or, equivalently, by

$$\begin{aligned} I_p(\mathbf{y}) \;&=\; \int_{\Omega_1} h_p(\mathbf{y},\mathbf{x},s_2(\mathbf{x}))H\big(w_2(\mathbf{x})\big)r_2(\mathbf{x})d\mathbf{x} + \\ &\quad + \int_{\Omega_1} h_p(\mathbf{y},\mathbf{x},s_1(\mathbf{x}))\bar{H}\big(w_2(\pi_p(\mathbf{x}))\big)r_1(\mathbf{x})d\mathbf{x} \end{aligned} \tag{6}$$

and both r_2 and s_2 are extended over the domain Ω_1 at will, since $H(w_2(\mathbf{x}))$ is an indicator function that restricts the integral to the domain Ω_2. The kernel $h_p : \Gamma \times \mathbb{R}^2 \times \mathbb{R}_+ \mapsto \mathbb{R}_+$ depends on the focal length f and the focal setting u_p, as well as on the surface of the scene (third argument of h_p). If we define $J_p : Z^2 \mapsto \mathbb{R}_+$ to be the measured image corresponding to the image model I_p, for $p = [p_1 ... p_N]$, where N is the total number of images, we can formulate the problem of recovering the unknown radiances, the surfaces and the support function, as the minimization of the following energy functional $E(r_1, r_2, s_1, s_2, w_2)$:

$$\begin{aligned} E = \sum_{i=1}^{N}\sum_{y} \phi\left(I_{p_i}, J_{p_i}\right) + \int_{\Omega_1} \alpha_1\left(\|\nabla r_1(\mathbf{x})\|\right)d\mathbf{x} + \\ + \int_{\Omega_2} \alpha_2\left(\|\nabla r_2(\mathbf{x})\|\right)d\mathbf{x} + \int_{\Omega_1} \beta_1\left(\|\nabla s_1(\mathbf{x})\|\right)d\mathbf{x} + \\ + \int_{\Omega_2} \beta_2\left(\|\nabla s_2(\mathbf{x})\|\right)d\mathbf{x} + \int_{\Omega_1} \gamma_2\left(\|\nabla H(w_2(\mathbf{x}))\|\right)d\mathbf{x} \end{aligned} \tag{7}$$

where $\phi : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}_+$, and $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_2$ are functions $\mathbb{R} \mapsto \mathbb{R}_+$. The first term accounts for the matching between the model image and the observed image. The remaining terms impose regularization on the five unknowns. In particular, the last term minimizes the length of the boundary of the support function w_2. Energy functionals like the one in Eq. 7 have been used in blind image restoration by You and Kaveh [23, 22].

The five unknowns r_1, r_2, s_1, s_2 and w_2 are inferred using a technique similar to gradient descent. Due to the iterative nature of gradient descent, our estimated functions will also be dependent on time. For example, the estimate of r_2 will be a function $\hat{r}_2 : \Omega_2 \times \mathbb{R}_+ \mapsto \mathbb{R}_+$, such that, if the solution is unique and convergence is guaranteed, $\lim_{t\to\infty} \hat{r}_2(\mathbf{x}, t) = r_2(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega_2$. We call *evolution* of r_2 the change of $\hat{r}_2$ in time. Now, let $\xi : \mathbb{R}^2 \times \mathbb{R}_+ \mapsto \mathbb{R}$ represent one of the estimated unknowns as it evolves in time, then, to minimize the energy functional (7), we use the following iteration scheme

$$\frac{d\xi}{dt} = -\nabla_\xi E \tag{8}$$

where $\nabla_\xi E$ is the Euler-Lagrange equation of the energy functional[3] with respect to ξ. The initialization procedure will be discussed in the implementation sections, and for now we assume that ξ is initialized with an "admissible" function.

3. Detecting occlusions

The amount of computations required by the algorithm described in the previous section is considerable, due to the infinite-dimensional formulation of the minimization scheme. One can reduce the time required to estimate the geometry and radiometry of the scene by first detecting occluding regions and then applying the minimization scheme to the detected regions. Rather than locating sharp discontinuities of the depth map, we detect occluding regions by examining the residual of the cost function when a single surface and a single radiance are assumed. Occluding boundaries result in a high residual, while smooth variations in depth do not elicit large residuals. Therefore, given a set of blurred images (see Figure 3), we first compute the residual with a single-surface/single-radiance model (see Figure 4); then, we select regions whose corresponding residuals are above a fixed threshold (see Figure 4), and restrict the occlusion model to operate within those regions. This step is conceptually not necessary, and it is only inserted to speed up the algorithm described below.

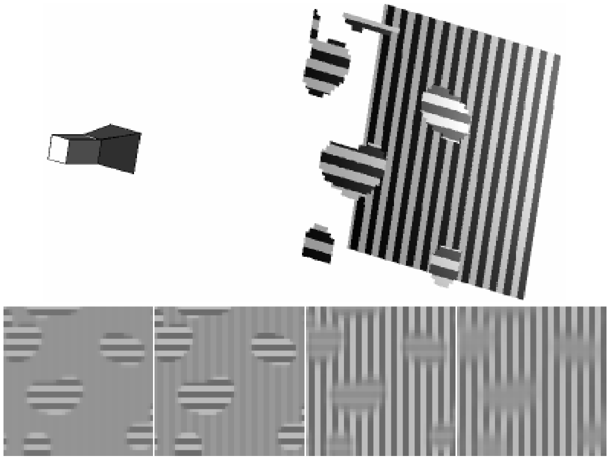

Figure 3: *Synthetic scene with occlusions. Top: setup. Bottom: data. From left to right the focal position moves away from the camera, thus, first focusing on the foreground object and then moving the focus towards the background object.*

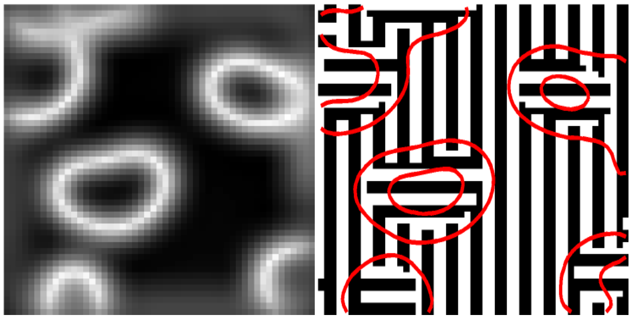

Figure 4: *Left: residual of the cost function when assuming a single-surface/single-radiance model (white denotes high residuals). High residuals correspond to occluding boundaries. Right: a pinhole image of the scene in Figure 3 with detected occlusions. Smooth variations in depth do not elicit large residuals.*

4. Implementation of the algorithm

In this section we will explain more in detail the minimization of the energy functional (7).
To simplify the notation that we are going to introduce, we will make use of the following change of coordinates

$$\hat{\mathbf{y}} = \mathbf{y}\frac{s_2(\hat{\mathbf{y}})}{v}. \quad (9)$$

Notice that this change of coordinates is in implicit form. However, since there is a one-to-one correspondence between $\mathbf{y}$ and $\hat{\mathbf{y}}$, we will not need to solve it explicitly, as variables originally defined in $\mathbf{y}$, will be defined with respect to the new coordinates $\hat{\mathbf{y}}$. This will be the case, for example, for the images I_{p_i}. Then, the operation of defining I_{p_i} with respect to $\hat{\mathbf{y}}$ corresponds to projecting the image I_{p_i} onto the surface s_2.

We are now interested in computing the projection function π_p:

$$\pi_p(\mathbf{x}) \doteq \bar{\mathbf{x}} = \mathbf{y}\frac{u_p}{v}\frac{s_1(\mathbf{x}) - s_2(\bar{\mathbf{x}})}{s_1(\mathbf{x}) - u_p} + \mathbf{x}\frac{s_2(\bar{\mathbf{x}}) - u_p}{s_1(\mathbf{x}) - u_p}. \quad (10)$$

Since s_2 appears as a function of $\bar{\mathbf{x}}$, this formula is also defined implicitly, and it can not be used directly in the energy functional. In order to obtain an explicit formula, it is necessary to use approximations of s_2. The simplest is to assume that s_2 is constant for each $\bar{\mathbf{x}}$ in a neighborhood of $\hat{\mathbf{y}}$, and therefore:

$$\pi_p(\mathbf{x}) \simeq \mathbf{y}\frac{u_p}{v}\frac{s_1(\mathbf{x}) - s_2(\hat{\mathbf{y}})}{s_1(\mathbf{x}) - u_p} + \mathbf{x}\frac{s_2(\hat{\mathbf{y}}) - u_p}{s_1(\mathbf{x}) - u_p}. \quad (11)$$

In addition to this approximation, we will also need a second one, due to the computation of the Euler-Lagrange

[3] Because of the parallel to gradient descent, the symbol $\nabla_\xi E$ has been chosen to represent the Euler-Lagrange equation of the energy functional with respect to ξ.

equations. Later, we will see that some of these equations require a change of coordinates that is implicit. To make the change of coordinate explicit, we impose that s_1 is constant for $\mathbf{x}$ in a neighborhood of $\hat{\mathbf{y}}$, and therefore:

$$\pi_p(\mathbf{x}) \simeq \hat{\mathbf{y}} \frac{u_p}{s_2(\hat{\mathbf{y}})} \frac{s_1(\hat{\mathbf{y}}) - s_2(\hat{\mathbf{y}})}{s_1(\hat{\mathbf{y}}) - u_p} + \mathbf{x} \frac{s_2(\hat{\mathbf{y}}) - u_p}{s_1(\hat{\mathbf{y}}) - u_p}. \tag{12}$$

We find experimentally that the current formulation is sufficiently accurate to infer the unknowns of the scene.

Now, we re-define the imaging model (6) under the change of coordinates (9) and the assumptions above, to yield

$$\begin{gathered}I_p(\hat{\mathbf{y}}) = \textstyle\int_{\Omega_1} h_p(\hat{\mathbf{y}}, \mathbf{x}, s_2(\hat{\mathbf{y}})) H\big(w_2(\mathbf{x})\big) r_2(\mathbf{x}) d\mathbf{x} + \\ + \textstyle\int_{\Omega_1} h_p\left(\hat{\mathbf{y}} \frac{s_1(\hat{\mathbf{y}})}{s_2(\hat{\mathbf{y}})}, \mathbf{x}, s_1(\hat{\mathbf{y}})\right) \bar{H}\big(w_2(\pi_p(\mathbf{x}))\big) r_1\left(\mathbf{x}\frac{s_2(\hat{\mathbf{y}})}{s_1(\hat{\mathbf{y}})}\right) d\mathbf{x}.\end{gathered} \tag{13}$$

Notice that in this formulation we defined the radiance r_1 on the surface s_2, rather than on the surface s_1, to have the same resolution of radiance r_2. Hence, when integrating in Ω_1 we have to re-project onto the surface s_2 the coordinates $\mathbf{x}$ defined on surface s_1.
Since our formulation of the problem requires the computation of gradients, which are not well-defined for the Heaviside function[4] in the context of regular functions, we define the following regularized version of the Heaviside function

$$H_\epsilon(z) \doteq \frac{1}{2}\left(1 + \frac{2}{\pi} \arctan\left(\frac{z}{\epsilon}\right)\right) \tag{14}$$

where ϵ is a *tuning* parameter and determines the degree of regularization of H_ϵ. In a similar fashion we define the derivative of $H_\epsilon(z)$ as the regularized Dirac delta function

$$\delta_\epsilon(z) \doteq \frac{\partial H_\epsilon(z)}{\partial z} = \frac{1}{\pi\epsilon} \frac{1}{1 + \left(\frac{z}{\epsilon}\right)^2}. \tag{15}$$

In the literature for depth from defocus, the kernel function h_p is typically chosen to be either a Gaussian or a pillbox function (see [6]). In our current implementation we use a Gaussian kernel, but the algorithm is not restricted to this choice. Hence, we have

$$h_p(\mathbf{y}, \mathbf{x}, s) = \frac{1}{2\pi\sigma_s^2} e^{-\frac{|\mathbf{y}-\mathbf{x}|^2}{2\sigma_s^2}} \tag{16}$$

where $\sigma_s = \lambda D \left|\frac{s}{u_p} - 1\right|$, λ is a scale factor, and D is the diameter of the lens.
Define ϕ to be the ℓ_2 norm, i.e.

$$\phi(I_{p_i}, J_{p_i}) = \sum_{\hat{\mathbf{y}}} (I_{p_i}(\hat{\mathbf{y}}) - J_{p_i}(\hat{\mathbf{y}}))^2. \tag{17}$$

Define also $\gamma_2(z) = \gamma|z|$ with γ a positive constant, and similarly for the functions α_1, α_2, β_1 and β_2. Now that all the relevant functions have been defined, we can proceed with computing the Euler-Lagrange equations of the energy functional with respect to the five unknowns.
The first equation we examine is $\nabla_{r_1} E$, and it is composed of two terms:

$$\begin{gathered}\nabla_{r_1} E(\mathbf{x}) = 2 \textstyle\sum_{i=1}^N \sum_{\hat{\mathbf{y}}} \left(I_{p_i}(\hat{\mathbf{y}}) - J_{p_i}(\hat{\mathbf{y}})\right) \frac{s_1^2(\hat{\mathbf{y}})}{s_2^2(\hat{\mathbf{y}})} \\ h_p\left(\hat{\mathbf{y}} \frac{s_1(\hat{\mathbf{y}})}{s_2(\hat{\mathbf{y}})}, \mathbf{x}\frac{s_1(\hat{\mathbf{y}})}{s_2(\hat{\mathbf{y}})}, s_1(\hat{\mathbf{y}})\right) \bar{H}_\epsilon(w_2(\pi_p(\mathbf{x}\frac{s_1(\hat{\mathbf{y}})}{s_2(\hat{\mathbf{y}})}))) - \\ -\nabla \cdot \left(\alpha_1'\left(\|\nabla r_1(\mathbf{x})\|\right) \frac{\nabla r_1(\mathbf{x})}{\|\nabla r_1(\mathbf{x})\|}\right)\end{gathered} \tag{18}$$

where α_1' is the first derivative of α_1 with respect to its only argument.
The first term comes from "matching" the image model and the measured image, and the other term comes from the regularization constraint on r_1. In particular, the second term can be interpreted as the anisotropic diffusion (see [15] for more details) of the reconstructed r_1, provided that the *diffusion coefficient* c is defined as $c(\mathbf{x}) \doteq \frac{\alpha_1'(\|\nabla r_1(\mathbf{x})\|)}{\|\nabla r_1(\mathbf{x})\|}$. Similarly, we obtain the Euler-Lagrange equation $\nabla_{r_2} E$:

$$\begin{gathered}\nabla_{r_2} E(\mathbf{x}) = 2 \textstyle\sum_{i=1}^N \sum_{\hat{\mathbf{y}}} \left(I_{p_i}(\hat{\mathbf{y}}) - J_{p_i}(\hat{\mathbf{y}})\right) h_p(\hat{\mathbf{y}}, \mathbf{x}, s_2(\hat{\mathbf{y}})) \\ H_\epsilon(w_2(\mathbf{x})) - \nabla \cdot \left(\alpha_2'\left(\|\nabla r_2(\mathbf{x})\|\right) \frac{\nabla r_2(\mathbf{x})}{\|\nabla r_2(\mathbf{x})\|}\right).\end{gathered} \tag{19}$$

The Euler-Lagrange equation $\nabla_{s_1} E$ involves a large number of terms. Rather than finding the explicit formula of $\nabla_{s_1} E$, we choose to approximate it pointwise via

$$\nabla_{s_1} E(\hat{\mathbf{y}}) \simeq \frac{E(s_1(\hat{\mathbf{y}}) + ds_1(\hat{\mathbf{y}})) - E(s_1(\hat{\mathbf{y}}))}{ds_1(\hat{\mathbf{y}})}. \tag{20}$$

In practice, due to the discretization of the functions in the implementation of the algorithm, it is sufficient to use a small $ds_1(\hat{\mathbf{y}})$ (between 0.001 m and 0.01 m for surfaces around 1 m) to compute a reasonable approximation of $\nabla_{s_1} E$. Similarly, we approximate $\nabla_{s_2} E$ pointwise by using

$$\nabla_{s_2} E(\hat{\mathbf{y}}) \simeq \frac{E(s_2(\hat{\mathbf{y}}) + ds_2(\hat{\mathbf{y}})) - E(s_2(\hat{\mathbf{y}}))}{ds_2(\hat{\mathbf{y}})}. \tag{21}$$

The computation of the last Euler-Lagrange equation $\nabla_{w_2} E$ returns:

$$\begin{gathered}\nabla_{w_2} E(\mathbf{x}) = 2 \textstyle\sum_{i=1}^N \sum_{\hat{\mathbf{y}}} \left(I_{p_i}(\hat{\mathbf{y}}) - J_{p_i}(\hat{\mathbf{y}})\right) \\ h_p(\hat{\mathbf{y}}, \mathbf{x}, s_2(\hat{\mathbf{y}})) \delta_\epsilon(w_2(\mathbf{x})) \\ \left(r_2(\mathbf{x}) - r_1\left(\hat{\mathbf{y}} \frac{u_p}{s_1(\hat{\mathbf{y}})} \frac{s_2(\hat{\mathbf{y}}) - s_1(\hat{\mathbf{y}})}{s_2(\hat{\mathbf{y}}) - u_p} + \mathbf{x} \frac{s_2(\hat{\mathbf{y}})}{s_1(\hat{\mathbf{y}})} \frac{s_2(\hat{\mathbf{y}}) - s_1(\hat{\mathbf{y}})}{s_2(\hat{\mathbf{y}}) - u_p}\right)\right) - \\ -\gamma \delta_\epsilon(w_2(\mathbf{x})) \nabla \cdot \left(\frac{\nabla w_2(\mathbf{x})}{\|\nabla w_2(\mathbf{x})\|}\right).\end{gathered} \tag{22}$$

5. Experiments

In this section we show some experiments on real and synthetically generated data. We consider scenes where one (or

[4] The derivative of the Heaviside is, however, well-defined in a distributional sense (see [11]).

a collection of) object(s) in the foreground occludes another in the background (see Figure 1, Figure 3 or Figure 8 for example). For each experiment we use 4 images: 2 with focal distance close to the foreground object(s) and the other 2 with focal distance close to the background object. The distances at play are such that the amount of blurring on either the foreground object or the background object is large.

5.1. Experiments on a synthetic scene

We generate 4 images from a scene made of an equifocal plane in the background, and a collection of surfaces in the foreground. The surfaces in the foreground all lie on another equifocal plane, that is closer to the camera (see Figure 3). We generate the images by changing the image plane distance from the lens, and then compensate for the scaling between the different images (see Figure 5). We estimate the radiance on the background surface (vertical stripes), and the radiance on the foreground surfaces (horizontal stripes). In Figure 5 we show a few snapshots of the estimation process of both the radiances, together with the ground truth (on the right). In Figure 6 we visualize the evolution of the zero level set of the support function w as a contour superimposed to the pinhole image of the scene. In Figure 7 we show some snapshots from the evolution of the surfaces estimation. In particular, the foreground surfaces are shown only on the estimated support, and both surfaces are texture mapped with the estimated radiances.

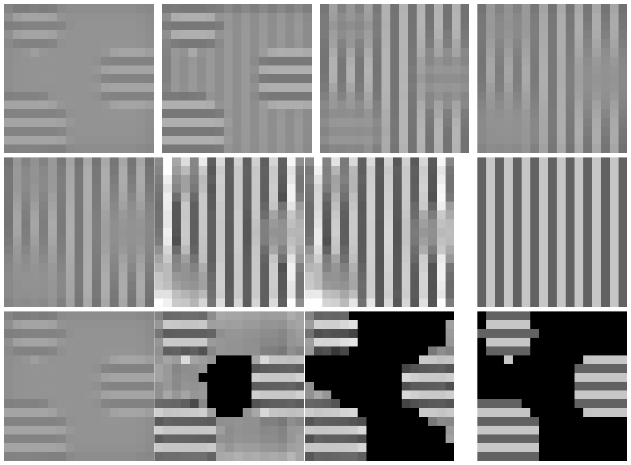

Figure 5: *Experiments on synthetic data. (Top row) 4 patches of 15x15 pixels each from synthetic data are shown. (Middle row) On the left we show 3 snapshots of the evolution of r_1, while on the right we show the ground truth for comparison. (Bottom row) On the left we show 3 snapshots of the evolution of r_2 together with the estimated support, while on the right we show the ground truth (on the true support).*

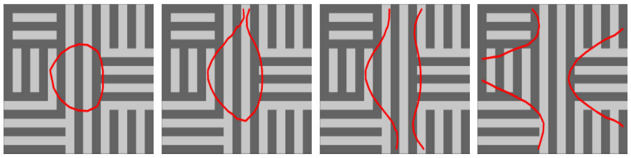

Figure 6: *Synthetic scene. Evolution of support function. The zero level set of the support function is showed (red contour) superimposed to the pinhole image of the scene.*

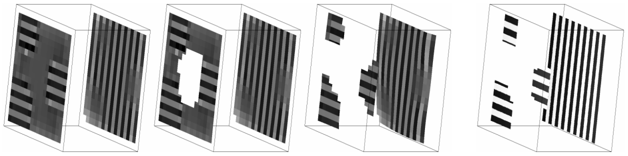

Figure 7: *Synthetic scene. From the left, 3 snapshots of the evolution of surfaces s_1 and s_2. Surface s_2 is shown only on the support estimated via w_2. On the right we show the true scene geometry.*

5.2. Real images

In Figure 8 it is shown the setup of the scene used in our experiments with real images. In the background we put the "IEEE" sign over a highly textured surface at a distance of $1.10m$ from the camera. In the foreground we placed a set of stripes at a distance of $.40m$, in such away as to completely cover the "IEEE" sign. However, when the background is focused, the finite size of the lens allows us to see "behind" the foreground object as it is evident in figures 8 and 9. To capture the images we used a Nikon AF NIKKON $35mm$ lens and focus at $.35m$, $.45m$, $1.05m$ and $1.15m$. Notice that the setup image in Figure 8 has been obtained by putting the camera and the objects closer to each other so that the stripes and the sign "IEEE" are both clearly visible.

As in the previous section, we estimate the radiance on the background surface (the "IEEE" sign), and the radiance on the foreground surfaces. In Figure 9 we show a few snapshots of the estimation process of both the radiances. On the right we put the "ground truth" for comparison. The image for the radiance in the background has been taken by simply removing the occluding object and by focusing on the background surface. Instead, the ground truth image for the radiance in the foreground object has been obtained by manually segmenting a focused image of the foreground object. In Figure 10 we visualize the evolution of the zero level set of the support function w as a contour superimposed to the pinhole image of the scene. In Figure 11 we show some snapshots from the evolution of the surfaces estimation. The foreground surfaces is shown only on the estimated support, and both surfaces are texture mapped with the estimated radiances.

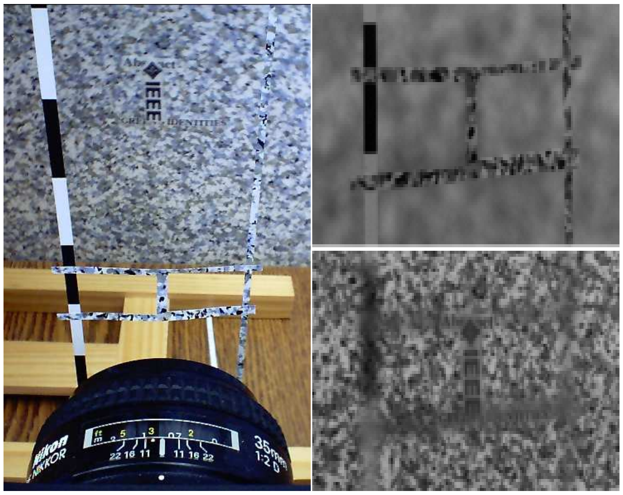

Figure 8: *Left: setup of the scene together with the camera. Right: two of the images captured for two different focal settings. As it can be seen, on the top right image the foreground object completely occludes the background object. By moving the plane in focus to the background, however, the sign “IEEE” on the background image becomes visible due to the finite-aperture of the lens (bottom-right image) similarly to Fig. 1.*

6. Summary and Conclusions

We have presented an algorithm to reconstruct the shape and radiance of a scene that features occlusions. This includes estimating the occluding boundaries of each object, its shape, and the radiance of each object, including portions of it that are occluded. To the best of our knowledge, we are the first to do so.

Our algorithm is a gradient descent on the first-order optimality condition of a cost functional that includes a data fidelity term and regularization terms for each of the unknown. The latter are necessary since the unknowns (shape and radiance) live in infinite-dimensional spaces. We have tested our algorithm on real and synthetic image sequences, including the uncovering of occlusions.

Acknowledgments

This research has been supported by funds AFOSR F49620-03-1-0095, NSF IIS-0208197/CCR-0121778, ONR N00014-02-1-0720 and Intel 8029.

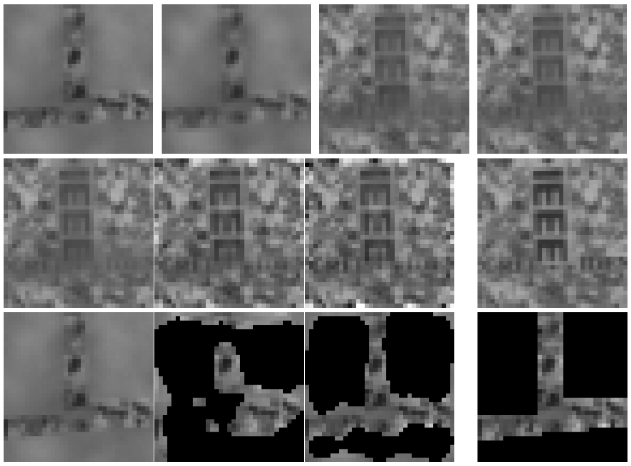

Figure 9: *Top row: 4 patches of 35x35 pixels from the real data of Figure 8. Middle row: on the left we show 3 snapshots of the evolution of r_1 (the radiance of the background object), while on the right we show the true radiance captured after removing the occluding object from the scene. Bottom row: on the left we show 3 snapshots of the evolution of r_2 together with its support, while on the right we show the radiance captured as ground truth.*

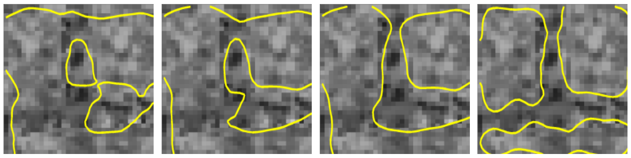

Figure 10: *Real scene. Evolution of the support function. The zero level set of the support function is showed (yellow contour) superimposed to the pinhole image of the scene.*

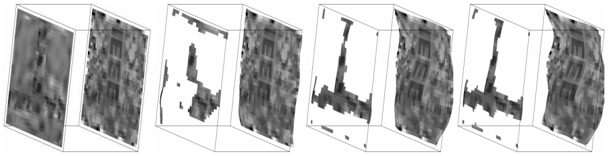

Figure 11: *Real data.* 4 *snapshots of the evolution of surfaces* s_1 *and* s_2. *Surface* s_2 *is shown only on the support estimated via* w_2.

References

[1] M. Aggarwal and N. Ahuja. A new imaging model. *Proceedings Eighth IEEE International Conference on Computer Vision*, 1:82–9, 2001.

[2] N. Asada, H. Fujiwara, and T. Matsuyama. Seeing

behind the scene: analysis of photometric properties of occluding edges by the reversed projection blurring model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):155–67, 1998.

[3] S. S. Bhasin and S. Chaudhuri. Depth from defocus in presence of partial self occlusion. *Proc. International Conference on Computer Vision*, 1(2):488–93, 2001.

[4] V. Caselles, R. Kimmel, and G. Sapiro. On geodesic active contour. *International Journal of Computer Vision*, 22(1):61–79, 1997.

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

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

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

[8] P. Favaro, A. Mennucci, and S. Soatto. Observing shape from defocused images. *International Journal of Computer Vision*, 52(1):25–43, 2003.

[9] S. Hiura and T. Matsuyama. Depth measurement by the multi-focus camera. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 935–9, 1998.

[10] J.A. Marshall, C.A. Burbeck, D. Ariely, J.P. Rolland, and K.E. Martin. Occlusion edge blur: a cue to relative visual depth. *Journal of the Optical Society of America A-Optics & Image Science*, 13(4):681–8, 1996.

[11] R. C. McOwen. *Partial Differential Equations: Methods and Applications.* Prentice Hall, 1995.

[12] T. C. Nguyen and T. S. Huang. Image blurring effects due to depth discontinuities: Blurring that creates emergent image details. *Proc. European Conference on Computer Vision*, pages 347–362, 1992.

[13] 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.

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

[15] 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.

[16] A.N. Rajagopalan and S. Chaudhuri. A variational approach to recovering depth from defocused images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, (no.10):1158–64, October 1997.

[17] Y.Y. Schechner and N. Kiryati. Depth from defocus vs. stereo: How different really are they? *International Journal of Computer Vision*, 39(2):141–62, 2000.

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

[19] P-S. Toh and A.K. Forrest. Occlusion detection in early vision. *Proceedings. Third International Conference on Computer Vision*, pages 126–32, 1990.

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

[21] 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.

[22] Y. You and M. Kaveh. Anisotropic blind image restoration. *Proceedings of 3rd IEEE International Conference on Image Processing*, 2:461–4, 1996.

[23] Y. You and M. Kaveh. Blind image restoration by anisotropic diffusion. *IEEE Trans. on Image Processing*, 8(3):396–407, 1999.