A Variational Approach to Shape From Defocus

Abstract

We address the problem of estimating the three-dimensional shape and radiance of a surface in space from images obtained with different focal settings. We pose the problem as an infinite-dimensional optimization and seek for the global shape of the surface by numerically solving a partial differential equation (PDE). Our method has the advantage of being global (so that regularization can be imposed explicitly), efficient (we use level set methods to solve the PDE), and geometrically correct (we do not assume a shift-invariant imaging model, and therefore are not restricted to equifocal surfaces).

1 Introduction

Shape from defocus (SFD) consists of reconstructing the three-dimensional shape and radiance ("texture") of a scene from a number of images taken with different focal settings. Under some simplifying assumptions on the imaging device, this problem can be posed as the inversion of certain integral equations that describe the imaging process; having chosen an optimality criterion, the problem can be solved uniquely under suitable conditions on the radiance of the scene.

What makes SFD possible is the fact that the image of a scene at a certain position on the image plane (a "pixel") depends upon the radiance on a region of the scene, as well as on the shape of such a region. What makes SFD possible, however, also makes it difficult: the image at a given pixel is obtained by integrating the (unknown) radiance of the scene against an (unknown) kernel that depends upon its shape. Given values of the integral at each pixel, one needs to estimate both the radiance and the kernel, which is known to be a severely ill-posed inverse problem in its full generality.

Several approaches have been presented to address this problem, which is an instance of "blind deblurring", or "blind deconvolution" for the case of linear shift-invariant kernels, as we describe in Section 1.1. Typically, the depth of the scene along a projection ray through each pixel is computed after approximating (locally) the radiance of the scene using various classes of functions or filters[1].

In this paper, rather than estimating depth at each pixel, we formulate shape from defocus within a variational framework as the problem of estimating an infinite-dimensional surface in space. We derive the optimality conditions and design a numerical algorithm to efficiently reach a (local) minimum. We do not make the assumption that the imaging kernel is shift-invariant – one that is patently violated at occluding boundaries – and therefore can handle complex shapes easily.

1.1 Relation to previous work

In the literature of computational vision a number of algorithms have been proposed to estimate depth from focus and accommodation information. The main assumption common to most algorithms available in the literature (overtly or covertly) is that the scene is a plane parallel to the focal plane (the equifocal assumption) [1, 6, 7, 10, 12, 14, 15, 18, 19, 20, 21]. This assumption allows one to describe the imaging process as a linear convolution; the price to pay, however, is a fundamental tradeoff between noise and precision. In order to combat noise, one would want to integrate over regions that are as large as possible; in order for the equifocal assumption to be valid, one would want regions to be as small as possible. In particular, at occluding boundaries the equifocal assumption is violated altogether.

On the other hand, several algorithms have been proposed to solve the problem in the shift-variant case. For instance, [4] presents a handful of methods that range from using *block-variant blur methods*, where the assumption of local equifocal imaging is corrected by taking into account contributions from the neighboring regions, to applying complex spectrogram and Wigner distributions in a *space-frequency* framework, where regularization is imposed to assure depth smoothness, to employing a *Markov random field* model to obtain a *Maximum a posteriori* estimate of the blur parameter using simulated annealing.

There is also a vast body of related literature in the signal processing community, where the problem is known as "blind deconvolution" (or more generally "deblurring"). The equifocal assumption is equivalent to assuming a shift-invariant convolution kernel, which is also common to most of the literature. The interested reader can see the special issue [2] for references.

[1] Such approximation can be represented either in space or in frequency, but the representation does not change the nature of the method.

1.2 Contributions of this paper

The equifocal assumption is a well-known limitation. First, the depth estimation results are directly affected by modeling the shape with equifocal planes, since this is only a crude approximation of natural scenes. Second, as this assumption usually goes together with choosing a support window, some other errors are introduced implicitly. Just to mention a few we have: *image overlapping*, *windowing effects*, *edge bleeding*, etc. These are issues that impoverish the depth estimate, and make depth from defocus less appealing as a technique to retrieve shape. In particular, to take into account disruptions due to image overlapping, one has to choose images where the blurring is always small compared to the size of the window patches (for instance no more than 2 pixels when the image patch is of 5×5 pixels). However, having small blurring radii renders the estimation process more sensitive to noise, and thus high-precision cameras have to be employed. In our approach we forgo the equifocal assumption, so that we can integrate visual information over the entire image. This results in superior resistance to noise (as also noticed in [4]). Furthermore, in order to render the shape estimation well-posed, we formulate the problem within a variational framework, so that we can regularize the reconstruction process via imposing smoothness, and we do not make explicit approximations of the shape; rather, we estimate shape via optimization on the infinite-dimensional space of (piecewise) smooth surfaces. We compute the necessary optimality conditions and numerically implement a partial differential equation (PDE) to converge to a (local) minimum. Last, but not least, we achieve superior computational efficiency by estimating global shape (as opposed to depth at each pixel) since the radiance on overlapping regions does not need to be recomputed. Our current implementation takes in the order of minutes when properly initialized (on a personal computer equipped with a 1GHz Pentium processor), as we describe in the experimental section.

2 Optimal shape from defocus

Suppose P is a generic point on the surface s with coordinates $\mathbf{X} \in \mathbb{R}^3$. By exploiting the additive nature of energy transport phenomena, we model the image formation process as an integral of the form

$$I(\mathbf{x}) = \int h^s(\mathbf{X}) dR \tag{1}$$

where $\mathbf{x} = \pi(\mathbf{X})$ is the projection of P on the image plane (π depends on the geometry of the optics). The measure R indicates the radiance of the scene and the kernel h^s depends upon the shape of the scene as well as on the geometry of the imaging device. The image I is defined on a compact set $D \subset \mathbb{R}^2$. Our goal is to reconstruct both the radiance R and the shape of the scene from a collection of images.

2.1 Image formation and notation

Suppose we have L measurements, i.e. L images with different focal setting: $u_1, \dots u_L$. If we collect and organize them into an array $I \doteq [I_{u_1}, \dots, I_{u_L}]^T$, and so for the kernels $h^s_{u_l}$, we can get rid of the subscript u and write again $I(\mathbf{x}) = \int h^s(\mathbf{X}) dR$. The right-hand side can be interpreted as the "virtual image" of a given surface s radiating energy with a given (spatial) distribution R, $R(\tilde{\mathbf{X}})$: $\int h^s(\mathbf{X}, \tilde{\mathbf{X}}) dR(\tilde{\mathbf{X}})$. For scenes made with opaque objects, the integral is restricted to their surface, and therefore it is to be interpreted in the Riemannian sense [3]. With a suitable choice of coordinates we can express the shape $s = s(x, y)$ and the radiance $r = r(x, y)$ as graphs and write the integral as $\int h^s(x, y, \tilde{x}, \tilde{y}) r(\tilde{x}, \tilde{y}) d\tilde{x} d\tilde{y}$ for $\pi(\mathbf{X}) \in D$ with $\mathbf{X} = [x\ y\ s(x, y)]^T$. We call r the radiant density[2]. However, in the next sections we will show that the choice of parameterization is not crucial in our algorithm.

2.2 Cost functional

The problem of inverting integrals of the form (1) is well-known to be ill-posed. Furthermore, often (1) is only an approximation of the model that generates the data. We will therefore look for solutions that minimize a suitable *optimization criterion*. In his seminal paper [5], Csiszár presents a derivation of "sensible" optimization criteria for the problem above, and concludes that the only two that satisfy a set of consistency axioms are the L_2-norm – when the quantities at play are unconstrained – and the information-divergence – when both the radiance and the kernel are constrained to be non-negative, such as in our case, where the radiance represents an energy density and the kernel represents surface area. Therefore, without further discussion, we adopt information-divergence (or I-divergence) as a cost functional for the discrepancy between measured images I and the images generated by the image model J:

$$\Psi(I|J) = \int \Phi(I(\mathbf{x})|J(\mathbf{x})) dA \tag{2}$$

where

$$\Phi(I(\mathbf{x})|J(\mathbf{x})) = I(\mathbf{x}) \log \frac{I(\mathbf{x})}{J(\mathbf{x})} - I(\mathbf{x}) + J(\mathbf{x}) \tag{3}$$

$\mathbf{x} = \pi(\mathbf{X})$, $\mathbf{X}$ belongs to the shape s and dA is the area form of the s defined for each $\mathbf{X}$. To emphasize the dependence

[2] Strictly speaking, r is the Radon-Nikodym derivative of R and, as such, it is not an ordinary function but, rather, a distribution of measures. In what follows we will ignore such technicalities and assume that we can compute integrals and derivatives in the sense of distributions.

of the images J on the shape s and the radiance r, we write, with an abuse of notation, $J = J(s, r)$. Hence, the problem of retrieving both shape s and radiance r from a collection of images $I \doteq [I_{u_1}, \ldots, I_{u_L}]^T$ can be formulated as the problem of minimizing the I-divergence between I and $J = J(s, r)$:

$$\hat{s}, \hat{r} = \arg\min_{s,r} \Psi\big(I|J(s,r)\big). \tag{4}$$

2.3 Radiance and shape estimation

Tackling the problem of minimizing the above cost functional involves solving a nonlinear and infinite-dimensional optimization problem. Also, the simultaneous minimization of both shape s and radiance r is not an easy issue. Hence, we choose to split the optimization into two sub-problems through an *alternating minimization* technique. Suppose we are given an initial guess for the radiance r_0 and the surface s_0 (see Section 3.3 for more details), then the algorithm can be written as:

$$\begin{cases} \hat{r}_{k+1} = \arg\min_{r} \Psi(I|J(\hat{s}_k, r)) \\ \hat{s}_{k+1} = \arg\min_{s} \Psi(I|J(s, \hat{r}_{k+1})). \end{cases} \tag{5}$$

The enabling step to use such an alternating minimization relies on having two iterations that independently lower the value of the cost functional, so that their combination leads towards the (local) minimum. For the first step we employ an iterative formula on the radiance r, which is constrained to be strictly positive, obtained from the Kuhn-Tucker conditions [9] on the cost functional, while for the second step we use a gradient descent flow implemented using level set methods.

Radiance iteration

Since there is no closed-form solution for the Kuhn-Tucker conditions, we seek for an iterative procedure such that the radiance will converge to a fixed point. Following Snyder et al. [17] we define the iteration as:

$$\hat{r}_{k+1}(x,y) = \hat{r}_k(x,y) \frac{1}{\int h^{\hat{s}}(x,y,\tilde{x},\tilde{y}) d\tilde{x} d\tilde{y}} \frac{\int h^{\hat{s}}(x,y,\tilde{x},\tilde{y}) I(\pi(\tilde{\mathbf{X}})) d\tilde{x} d\tilde{y}}{\int h^{\hat{s}}(\tilde{x},\tilde{y},\bar{x},\bar{y}) r_k(\bar{x},\bar{y}) d\bar{x} d\bar{y}} \tag{6}$$

where $\tilde{\mathbf{X}} = [\tilde{x}\ \tilde{y}\ s(\tilde{x},\tilde{y})]^T$. It can be shown that this iteration provably minimizes the chosen cost functional (with respect to r) even when the kernel $h^{\hat{s}}$ is not provided with the correct shape s [17].

Gradient descent flow

Away from discontinuities, the surface can be locally approximated by its tangent plane. Thus, the model image J can be computed in the tangent space:

$$J(\pi(\mathbf{X})) = \int h(\mathbf{X}, K(x,y)) r(K(x,y)) dx dy \tag{7}$$

where K is the transformation bringing points from the tangent space to the scene space, which explicitly depends on the surface normal N.

Faugeras and Keriven [8] prove that the Euler-Lagrange equation for (2) takes the following form:

$$\kappa\Phi - \Phi_{\mathbf{X}} \cdot N - \kappa(\Phi_N \cdot N) + \mathrm{Tr}\left((\Phi_{\mathbf{X}N})_{T_S} + dN \circ (\Phi_{NN})_{T_S}\right) \tag{8}$$

where $Tr(\cdot)$ denotes the trace and T_s the tangent space of s at P; κ is the mean curvature, and $\Phi_{\mathbf{X}}$ and Φ_N stand for derivatives of Φ with respect to $\mathbf{X}$ and N respectively. Φ_{NN} and $\Phi_{\mathbf{X}N}$ are the second order derivatives of Φ. $(\Phi_{NN})_{T_S}$ and $(\Phi_{\mathbf{X}N})_{T_S}$ are their restrictions to the tangent plane T_s. dN is the differential of the Gauss map of the surface, which involves the second fundamental form. For more details, refer to [8].

By directly computing the gradient flow of the cost function, it can be shown that the following geometric flow minimize (2):

$$\begin{aligned} P_t = \ & (\kappa\Phi - \Phi_{\mathbf{X}} \cdot N - \kappa(\Phi_N \cdot N) + \\ & + \mathrm{Tr}\left((\Phi_{\mathbf{X}N})_{T_S} + dN \circ (\Phi_{NN})_{T_S}\right)) N. \end{aligned} \tag{9}$$

3 Implementation

3.1 Level set iteration

We implement the flow (9) using level set methods. The level set methods were originally developed by Osher and Sethian [13]. Since then, the method has gained more and more popularity in various fields. Many fast numerical schemes have been proposed based on it. For a complete account refer to [16].

The level set implementation of any geometric flow begins by embedding the initial interface $P(\mathbf{X}, 0)$, as a level set of a scalar function $\psi_0(\mathbf{X})$ which is then taken to be the initial condition for a function over time $\psi(\mathbf{X}, t)$:

$$\psi_0 : \mathbb{R}^3 \to \mathbb{R}, \quad \psi : \mathbb{R}^3 \times \mathbb{R}^+ \to \mathbb{R}, \quad \psi(\mathbf{X}, 0) = \psi_0(\mathbf{X})$$

The choice of a particular level set is arbitrary but is typically taken to be zero. The key point is that the interface is continuously embedded within the same fixed level set of ψ during all the time. Thus, choosing the zero level set we have

$$\psi_0(P(\mathbf{X}, 0)) = 0, \quad \text{and} \quad \psi(P(\mathbf{X}, t), t) = 0.$$

Differentiating with respect to t therefore yields

$$\psi_t + \nabla\psi \cdot P_t = 0 \quad \text{or} \quad \psi_t = -P_t \cdot \nabla\psi \tag{10}$$

an evolution equation for ψ (where $\nabla\psi = \psi_{\mathbf{X}}$) which evolves the interface $P(\mathbf{X}, t)$ described implicitly by $\psi(\mathbf{X}, t) = 0$ for all t.

3.2 Intersection with the surface

In the radiance iteration it is necessary to determine which point on the shape s corresponds to which point on the image plane, in order to establish the blurring radius of the kernel h^s. To this end one needs to compute the intersection of a line coming from the image with the shape s. Obtaining explicitly all the possible intersections with a discretized representation of the shape, turns out to be computationally expensive. Rather, it is possible to do this very efficiently by exploiting the advantage of an implicit formulation of the shape, i.e. the level set function ψ or the signed distance function. Let the line be defined by a point X_0 and a direction v. Let X be the intersection we are looking for. X satisfies the following ordinary differential equation:

$$\begin{cases} \frac{dX}{dt} &= c(X) \cdot v \\ X(0) &= X_0 \end{cases} \tag{11}$$

where $c(\cdot)$ is a scalar function defined as follows

$$c(X) = \begin{cases} \text{sign}(\psi(X)) & \text{if} \quad |\psi(X)| > 1 \\ \psi(X) & \text{if} \quad |\psi(X)| \leq 1 \end{cases} \tag{12}$$

Intuitively, we move X according to $c(\cdot)$ so that X is lead toward the shape. When X crosses the surface, $c(X)$ will change sign accordingly, and therefore X will be forced to come back. Hence, X will oscillate around the intersection of the line with the shape, reducing the overshoot at each step. Finally, we decide for X to be the intersection, when the oscillation remains within a fixed band around the shape. This happens typically within a few (3-5) iterations.

3.3 Initialization

As we have noticed in previous sections, to start the alternating minimization one needs to have an initial guess for both radiance and shape. Since we have no prior knowledge on either unknown, we proceed as follows: choose one of the input images I_{u_1}, taken with focal setting u_1; define the initial shape as a plane parallel to the focal plane passing through the focal depth u_1; compute the radiance by back-projecting the image I_{u_1} onto the defined shape. As we see in our experiments, such a choice is not crucial to the estimation process. However, we also noticed that a good initialization speeds up the minimization procedure considerably. Therefore, during the first steps of our algorithm we perform the shape estimation using a simple search of the minimum of the cost functional computed over a small grid of possible depths, using the equifocal assumption. This initial shape is then used for the radiance iteration step. After a few iterations, we substitute the search step with the level set iteration and proceed with the minimization as described in the previous sections.

4 Experiments

We have tested our algorithm on the real images shown in Figure 1. The images are obtained by changing the position of the image plane along the optical axis, and keeping the lens position fixed with respect to the scene. Moving the image plane necessarily involves scaling the images, which we avoid by employing telecentric optics (see [11]). The two images are taken with focal depths of approximately $0.9m$ and $1.1m$ respectively. The focal length is $30mm$ and the lens aperture is $F/8$. The scene has been chosen so as to test the performance of the proposed algorithm when the usual equifocal assumption does not hold. It can be noticed that the scene presents significant depth variations and several occluding boundaries. In particular, at the occluding boundaries of the statues and in the folds of the skirts, the planar approximation fails. Furthermore, the blurring radii are up to $4 - 5$ pixels, so that the window size would have to be at least of 10 pixels, which is beyond the usual patch size and does not allow for fine depth retrieval. Figure 2 shows three steps of the radiance iteration during the alternating minimization procedure. In Figure 3 we show the corresponding shape evolution from the level set iteration. (An MPEG movie of the evolution has been uploaded for reviewers' benefit). Then, in Figure 4 we show the final estimation of the shape coded in gray level (256 values) and three views of the final shape which has been texture-mapped with the final radiance.

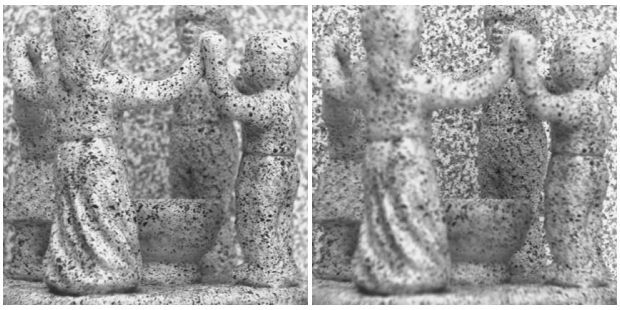

Figure 1: *Original images: the left image is near-focused ($0.9m$). The right image is far-focused ($1.1m$). As it can be noticed, in both images the blurring is quite consistent, the shape is non-trivial and presents several discontinuities due to occluding boundaries.*

5 Conclusion

In estimating shape from defocus, the equifocal assumption is a well-known limitation. It introduces several disruptions in the reconstruction process such as image overlapping, windowing effects, edge bleeding, etc. We present a

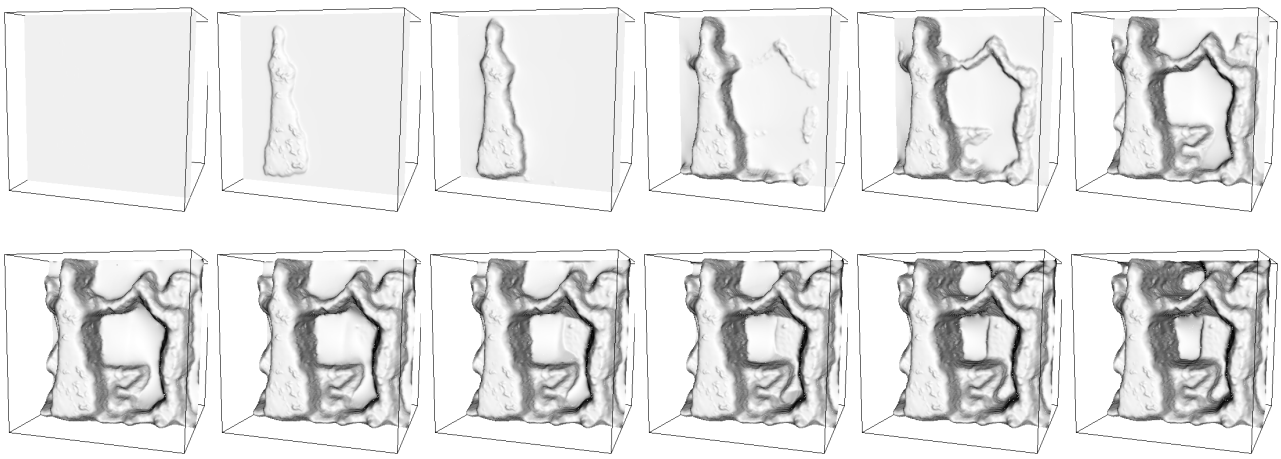

Figure 3: *Twelve snapshots of the shape evolution: the surface is gradually converging to the final shape, starting from a plane placed at depth* $0.9m$. An MPEG movie of the evolution has been uploaded for reviewers' benefit.

novel approach to shape from defocus based on an alternating minimization algorithm which does not make use of the equifocal assumption so as to overcome the above limitations. The radiance of the scene is estimated through an iterative equation which provably converges to a (local) minimum, while the shape is estimated using a gradient descent flow, which is then implemented numerically (efficiently) using level set methods. We show that the combination of these two steps leads to a (local) minimum of the discrepancy between the measured image and the modeled image. Also, by implementing the shape estimation with level set methods we implicitly impose smoothness on the estimated shape while preserving depth discontinuities at the occluding boundaries. These would be lost in a shift-invariant imaging model.

References

[1] N. Asada, H. Fujiwara, and T. Matsuyama. Edge and depth from focus. *Intl. J. of Comp. Vision*, 26(2):153–163, 1998.

[2] Various Authors. Special issue on blind system identification and estimation. *Proceedings of the IEEE*, 86(10), 1998.

[3] W. Boothby. *Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1986.

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

[5] I. Csiszár. Why least-squares and maximum entropy; an axiomatic approach to inverse problems. *Annals of statistics*, 19:2033–2066, 1991.

[6] T. Darell and K. Wohn. Depth from focus using a pyramid architecture. *Pattern Recognition Letters*, 11(2):787–796, 1990.

[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] O. Faugeras and R. Keriven. Variational principles, surface evolution pdes, level set methods and the stereo problem. *INRIA Technical report*, 3021:1–37, 1996.

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

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

[11] M.Watanabe and S.K.Nayar. Telecentric optics for constant magnification imaging. In *Technical Report CUCS-026-95*, pages Dept. of Computer Science, Columbia University, New York, USA, 1995.

[12] S. Nayar and Y. Nakagawa. Shape from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):824–831, 1994.

[13] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi equations. *J. of Comp. Physics*, 79:12–49, 1988.

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

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

[16] J. Sethian. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Material Science*. Cambridge University Press, 1996.

[17] D. Snyder and J. O'Sullivan. Deconvolution under non-negativity constraints. *IEEE Trans. Inform. Theory*, 1994.

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

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

[20] 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.

[21] D. Ziou. Passive depth from defocus using a spatial domain approach. In *Proc. of the Intl. Conf. of Computer Vision*, pages 799–804, 1998.

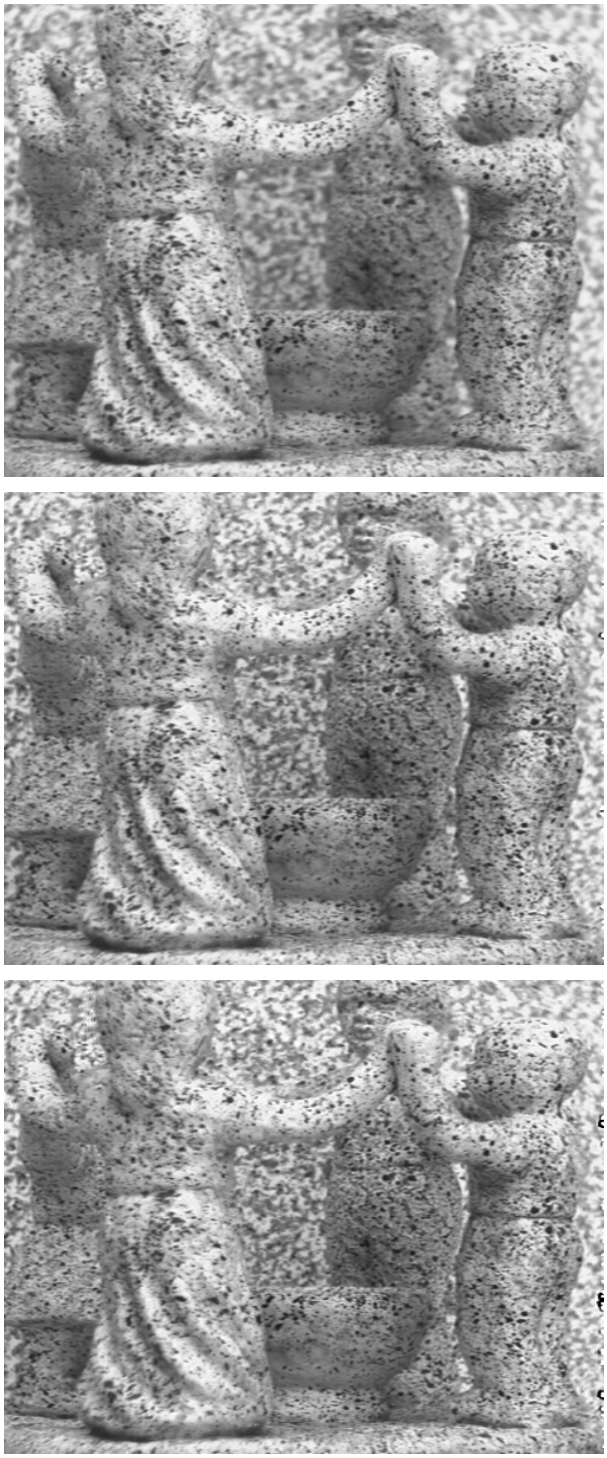

Figure 2: *Three snapshots from the radiance iteration. Top is the initial radiance obtained from the near focused image; middle is the radiance obtained after one iteration; and bottom is the radiance after three steps. It can be noticed that the radiance is gradually sharpening after each iteration, as is particularly visible in the background.*

Figure 4: *Top: depth rendered in gray levels (256 values); middle, bottom-left and bottom-right : three views of the final estimated shape, texture-mapped using the final estimated radiance.*