

Mumford-Shah on the Move: Region-Based Segmentation on Deforming Manifolds with Application to 3-D Reconstruction of Shape and Appearance from Multi-View Images

Hailin Jin · Anthony J. Yezzi · Stefano Soatto

Published online: 22 November 2007
© Springer Science+Business Media, LLC 2007

Abstract We address the problem of estimating the shape and appearance of a scene made of smooth Lambertian surfaces with piecewise smooth albedo. We allow the scene to have self-occlusions and multiple connected components. This class of surfaces is often used as an approximation of scenes populated by man-made objects. We assume we are given a number of images taken from different vantage points. Mathematically this problem can be posed as an extension of Mumford and Shah’s approach to static image segmentation to the segmentation of a function defined on a deforming surface. We propose an iterative procedure to minimize a global cost functional that combines geometric priors on both the shape of the scene and the boundary between smooth albedo regions. We carry out the numerical implementation in the level set framework.

Keywords Variational methods · Mumford-Shah functional · Image segmentation · Multi-view stereo · Level set methods · Curve evolution on surfaces · Dense shape reconstruction · Structure from motion · Visibility

H. Jin (✉)
Office of Technology, Adobe Systems Incorporated, San Jose, CA
95110, USA
e-mail: hljin@adobe.com

A.J. Yezzi
School of Electrical and Computer Engineering, Georgia Institute
of Technology, Atlanta, GA 30332, USA
e-mail: ayezzi@ece.gatech.edu

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

1 Introduction

Images of a scene taken from a moving vantage point (or from many cameras at different positions) are well-known to contain information about the three-dimensional (3-D) shape of the scene and its reflectance properties, which we refer to as “appearance”. However, it is also well-known that images alone are not sufficient to unravel such information: In the absence of any assumption about the shape of the scene and its reflectance, one could find infinitely many scenes that generate the images [20]. Traditionally, overt or covert assumptions have been used to establish “correspondence” between different images of the same scene, that is to determine which point (or region) in one image corresponds to which point (or region) in another image, in the sense of portraying the same location in space. Such corresponding points or regions are then used to triangulate the position in space of the patch of surface that generated them. These regions are necessarily sparse, since correspondence can only be established where the radiance profile is “distinctive enough” (a patch of white wall in one image can correspond to any other white patch in another image), and are often chosen small enough so that the assumptions made, the most common being planarity and Lambertian reflection [21], are reasonably well satisfied. If one desires a “dense reconstruction”, i.e. a collection of surfaces with their appearance rather than a bunch of points, some effort is required to “densify” the correspondence, triangulate a meshed surface, smooth it, and texture-map the images onto it.

Since for regions of different images to correspond there has to be a scene that can generate them, finding correspondence is just as difficult as reconstructing the scene [20]. It seems therefore ill-advised to make stringent assumptions to try to solve the correspondence problem as an intermediate step for 3-D reconstruction, when such a reconstruction is



Fig. 1 Man-made objects often exhibit piecewise smooth appearance. Approximating their radiances with global smooth functions would lead to gross error and “blurring” of the reconstruction. On the other

hand, the objects are not textured enough to establish dense correspondence among different views. However, we can clearly see radiance boundaries that divided the objects into smooth regions

needed (implicitly or explicitly) for correspondence in the first place. Furthermore, correspondence hides the assumptions in ways that make it difficult to control and verify them. Why not, instead, make explicit assumptions on the scene, and use them to reconstruct it directly, without intermediate steps? We have previously advocated this approach, that calls for matching *not* image-to-image, but all images to an underlying model, and relies on explicit assumptions. Such assumptions, in the specific case of this paper, are that the scene is *static*, i.e. there are no independently moving objects (although one could conceive extensions to independently moving objects), that it is *Lambertian* (we have addressed the case of non-Lambertian reflection in [8]), and that its *albedo is piece-wise smooth*, or piece-wise constant. These assumptions are by and large satisfied for man-made objects, such as those in Fig. 1, which makes them important enough to warrant dedicated treatment. This work continues the progression of our previous work where we have addressed the case of constant albedo [23], and smooth albedo [10], and our goal is to eventually have a complete hierarchy of models that can be used on a large variety of natural images.

While some of our assumptions can be lifted, the only crucial one we make is that the scene is static, that is all images portray the same scene, or object, and there is no intrinsic variability in it. In the recognition of object categories, or where objects can exhibit significant variability in different instances, we advocate sticking to a feature-based approach, where the sparseness and locality of the representation absorb in large part the intrinsic variability of the scene [21].

Given these assumptions, we will translate our problem into the language of the calculus of variations, and shows features that make it interesting even just on mathematical grounds. In particular, the problem amounts to extending the methods championed by Mumford and Shah [15] to segment static images to instead segment albedo profiles on manifolds, and moving ones at that. In fact, since at the beginning we know neither the shape nor the appearance of the scene, we will start with an initial guess of a surface, with

an initial guess of a radiance painted on it, and then simultaneously evolve the surface, and the piecewise smooth albedo function on it. Among the peculiarities of our solution is the fact that, despite being a gradient-based algorithm, it does not entail taking derivatives of the image. That is another advantage of having an explicit model: Instead of differentiating the noisy data, we can push the derivatives onto the model, which is differentiable by assumption. This is more principled, efficient, and robust than the practice commonly followed for dense 3-D reconstruction, which involves differentiating the images.

This work is an extension of our previous work [9]. Specifically, the present paper expands the formulation to objects with piecewise smooth radiances and provides detailed derivations to all the results that we obtained in [9]. We have also reported additional experiments on synthetic scenes that we feel are necessary to provide an evaluation against accurate ground truth.

2 Formalization of the Problem in a Variational Setting

In this section we will cast the problem of 3-D reconstruction of shape and appearance in the language of the calculus of variation. Before doing so we establish some notation.

2.1 Notation

We model the scene as a collection of smooth surfaces, one of them labeled as “background”. We denote such surfaces collectively with $S \subset \mathbb{R}^3$, noting that S can have multiple connected components. We denote with $\mathbf{X} = [X, Y, Z]^T$ the coordinates of a generic point on S with respect to a fixed reference frame. We assume to be able to measure n images of the scene $I_i : \Omega_i \subset \mathbb{R}^2 \rightarrow \mathbb{R}$, $i = 1, 2, \dots, n$, where Ω_i is the domain of each image with area element $d\Omega_i$.¹

¹More precisely, measured images are usually bounded discrete functions defined on (regular) grids. For ease of notation, we treat them as functions defined on compact domains in \mathbb{R}^2 and take values from the whole real line.

Table 1 Quick reference to the notation

Variable	Description
S	the surface of interest
B	the background surface
C	the segmenting curve
D_1, D_2	the region on the surface that supports a smooth radiance
$\rho : S \rightarrow \mathbb{R}$	the radiance function of the surface S
$h : B \rightarrow \mathbb{R}$	the radiance function of the background B
$\rho_i : D_i \rightarrow \mathbb{R}, i = 1, 2$	the radiance function of the region D_i
$\mathbf{X} = [X, Y, Z]^T$	the coordinates of a generic point on S
$\mathbf{X}_i = [X_i, Y_i, Z_i]^T$	\mathbf{X} with respect to the reference frame of the i -th camera
N	the inward unit normal to S
N_i	N with respect to the reference frame of the i -th camera
Ω_i	the domain of the image I_i
$\mathbf{x}_i = [x_i, y_i]^T$	the coordinates of a point in Ω_i
$\pi_i : S \rightarrow \Omega_i$	the projection transformation of the i -th camera
$\Theta_i : \Omega_i \rightarrow B$	the coordinate transformation from Ω_i to B
$Q_i = \pi_i(S)$	the projection of the surface S in the i -th image
$Q_i^c = \Omega_i \setminus \pi_i(S)$	the projection of the background B in the i -th image
$\chi_i : S \rightarrow \{0, 1\}$	the surface visibility function with respect to the i -th camera
$\sigma_i = d\Omega_i/dA$	the change of coordinates from dA to $d\Omega_i$

Each image is fully calibrated,² i.e., we know its intrinsic and extrinsic calibration parameters. Therefore, after pre-processing, each camera can be modeled as an ideal perspective projection $\pi_i : \mathbb{R}^3 \rightarrow \Omega_i; \mathbf{X} \mapsto \mathbf{x}_i \doteq \pi_i(\mathbf{X}) = \pi(\mathbf{X}_i) = [X_i/Z_i, Y_i/Z_i]^T$, where $\mathbf{X}_i = [X_i, Y_i, Z_i]^T$ are the coordinates for \mathbf{X} in the i -th camera reference frame. \mathbf{X} and \mathbf{X}_i are related by a rigid body transformation, which can be represented in coordinates by a rotation matrix $R_i \in SO(3)$ ³ and a translation vector $T_i \in \mathbb{R}^3$, such that $\mathbf{X}_i = R_i\mathbf{X} + T_i$. We assume that the background, denoted with B , covers the field view of each camera. Without loss of generality, we assume B to be a sphere with infinite radius, which can therefore be represented using angular coordinates $\Theta \in \mathbb{R}^2$. We can also define a one-to-one map from the image domain to the background $\Theta_i : \Omega_i \rightarrow B; \mathbf{x}_i \mapsto \Theta_i(\mathbf{x}_i)$. Since the scene is Lambertian, both foreground and background radiances can be modeled as scalar functions:

$$\rho : S \rightarrow \mathbb{R} \quad \text{and} \quad h : B \rightarrow \mathbb{R}. \tag{1}$$

We define the region $Q_i \doteq \pi_i(S) \subset \Omega_i$ and denote its complement with Q_i^c . Although the perspective projection π_i is not one-to-one (and therefore not invertible), we can still define the operator back-projecting a point \mathbf{x}_i from Q_i onto a given surface S as follows: We trace the ray starting from

the i -th camera center and passing through \mathbf{x}_i and define the first intersection point as the back-projection of \mathbf{x}_i onto S . Therefore, with an abuse of notation we denote this back-projection by $\pi_i^{-1} : Q_i \rightarrow S; \mathbf{x}_i \mapsto \mathbf{X}$.

Our assumption is that the foreground radiance ρ is *piecewise smooth*. For simplicity, we assume the background radiance h to be a *smooth* function. Extensions to piecewise smooth surfaces are conceptually straightforward, but significantly complicate the notation and implementation. Furthermore, we assume that the discontinuities of ρ can be modeled as a smooth closed curve C on the surface S , and C partitions S into two regions D_1 and D_2 such that $D_1 \cup D_2 = S$. We allow each region D_i to have multiple connected components. Extensions to more regions are straightforward, for instance following the work of Vese and Chan [22]. We can thus re-define ρ as follows:

$$\rho(\mathbf{X}) = \rho_i(\mathbf{X}) \quad \text{for } \mathbf{X} \in D_i, i = 1, 2 \tag{2}$$

where $\rho_i : D_i \rightarrow \mathbb{R}, i = 1, 2$. By construction, ρ_1 and ρ_2 are smooth within D_1 and D_2 respectively. We denote with $\pi_i(D_1)$ and with $\pi_i(D_2)$ the projections of D_1 and D_2 in the i -th image respectively. Please refer to Table 1 for quick references to the notations used in this paper.

2.2 The Cost Functional

Our task is to estimate S, C, ρ_1, ρ_2 , and h from the data $I_i, i = 1, 2, \dots, n$. In order to do so, we set up a cost that

²Given a sequence of images, there are systems that can automatically compute the intrinsic and extrinsic parameters of each image [7].

³ $SO(3) = \{R \mid R \in \mathbb{R}^{3 \times 3} \text{ s.t. } R^T R = I \text{ and } \det(R) = 1\}$.

measures the discrepancy between the prediction of the unknowns and the actual measurements. We can then adjust the unknowns to match the measured images. Since the unknowns live in infinite-dimensional spaces, we need to impose regularization to make the inference problem well-posed. In particular, we can leverage on our assumption that radiances are smooth in their supporting domains. However, this is not sufficient, for the estimated surface and curve could converge to very irregular shapes to match image noise and fine details. Therefore, we further introduce geometric priors to favor smooth surfaces and curves. These are the four main ingredients in our approach: A data fidelity term E_{data} , a radiance smoothness term E_{smooth} , and two geometric priors E_{surf} and E_{curv} . We consider the overall cost functional to be a weighted sum of these four terms:

$$E(S, C, \rho_1, \rho_2, h) = E_{data} + \alpha E_{surf} + \beta E_{curv} + \gamma E_{smooth}, \tag{3}$$

where $\alpha, \beta, \gamma \in \mathbb{R}^+$ control the relative weights among the terms.

The data fitness can be measured in the sense of \mathcal{L}^2 as:

$$E_{data} = \sum_{i=1}^n \left(\int_{\pi_i(D_1)} (I_i(\mathbf{x}_i) - \rho_1(\pi_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i + \int_{\pi_i(D_2)} (I_i(\mathbf{x}_i) - \rho_2(\pi_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i + \int_{Q_i^c} (I_i(\mathbf{x}_i) - h(\Theta_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i \right), \tag{4}$$

although other function norms would do as well. The geometric prior for S is given by the total surface area

$$E_{surf} = \int_S dA, \tag{5}$$

and that for C is given by the total curve length:

$$E_{curv} = \int_C ds, \tag{6}$$

where dA is the Euclidean area form of S and s is the arc-length parameterization for C . The smoothness of ρ_1, ρ_2 and h can be measured via a cost on the quadratic variation in their supporting domains:

$$E_{smooth} = \int_{D_1} \|\nabla_S \rho_1\|^2 dA + \int_{D_2} \|\nabla_S \rho_2\|^2 dA + \int_B \|\nabla_\Theta h\|^2 d\Theta, \tag{7}$$

where ∇_S and ∇_Θ denote the intrinsic gradient with respect to the surface and the background respectively. In general, one may choose different weights for the terms in (7). We

use one weight just for simplicity. The total cost takes the expression:

$$E_{total}(S, C, \rho_1, \rho_2, h) = E_{data} + \alpha E_{surf} + \beta E_{curv} + \gamma E_{smooth} = \sum_{i=1}^n \left(\int_{\pi_i(D_1)} (I_i(\mathbf{x}_i) - \rho_1(\pi_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i + \int_{\pi_i(D_2)} (I_i(\mathbf{x}_i) - \rho_2(\pi_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i + \int_{Q_i^c} (I_i(\mathbf{x}_i) - h(\Theta_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i \right) + \alpha \int_S dA + \beta \int_C ds + \gamma \left(\int_{D_1} \|\nabla_S \rho_1\|^2 dA + \int_{D_2} \|\nabla_S \rho_2\|^2 dA + \int_B \|\nabla_\Theta h\|^2 d\Theta \right). \tag{8}$$

This functional is in the spirit of the Mumford-Shah functional for image segmentation [15]. We conjecture that these terms are sufficient to define a unique solution to the minimization problem.

3 Optimization of the Cost Functional

In order to find the surface S , the radiances ρ_1, ρ_2, h and the curve C that minimize the cost (8), we set up an iterative procedure where we start from a generic initial condition (typically a big cube, sphere or cylinder for the surface for instance) and update the unknowns along their gradient directions until convergence to a (necessarily local) minimum is achieved.

3.1 Updating the Surface

The gradient descent flow for the surface geometric prior is given by the mean curvature flow:

$$S_t = 2\kappa N, \tag{9}$$

where κ is the mean curvature and N is the unit normal to S . Note that we have kept 2 in the expression in order to have the weights in the final flow match the weights in the cost (3). To facilitate computing the variation of the rest terms with respect to the surface, we introduce the radiance characteristic function ϕ to describe the location of C for a given surface S . We define $\phi : S \rightarrow \mathbb{R}$ such that

$$D_1 = \{\mathbf{X} \mid \phi(\mathbf{X}) > 0\}, \quad D_2 = \{\mathbf{X} \mid \phi(\mathbf{X}) < 0\}, \tag{10}$$

$$C = \{\mathbf{X} \mid \phi(\mathbf{X}) = 0\}.$$

ϕ can be viewed as the level set function of C . However, one has to keep in mind that ϕ is defined on S which is a smooth manifold. We can then express the curve length as [16]

$$\int_C ds = \int_S \|\nabla_S \mathcal{H}(\phi)\| dA = \int_S \delta(\phi) \|\nabla_S \phi\| dA \tag{11}$$

where \mathcal{H} is the Heaviside step function: $\mathcal{H}(x) = 0$ for $x < 0$ and $\mathcal{H}(x) = 1$ otherwise and δ denotes the one-dimensional Dirac distribution which is the derivative of \mathcal{H} : $\delta = \dot{\mathcal{H}}$. We prove in Appendix 1 that the gradient descent flow for the curve smoothness term takes the following expression

$$S_t = \frac{\delta(\phi)}{\|\nabla_S \phi\|} \mathbf{II}(\nabla_S \phi \times N)N, \tag{12}$$

where $\mathbf{II}(\mathbf{t})$ denotes the second fundamental form of a vector $\mathbf{t} \in T_P(S)$, i.e. the normal curvature along \mathbf{t} for $\|\mathbf{t}\| = 1$. $T_P(S)$ is the tangent space for S at P . Note that $\nabla_S \phi \times N \perp N$ and therefore $\nabla_S \phi \times N \in T_P(S)$. Since flow (12) is regulated by $\delta(\phi)$, it has effects only where ϕ is nonzero, i.e., on the curve C . This should not be surprising because $\int_C ds$

only controls the smoothness of C and points of the surface that do not belong to the curve are not affected by this term and consequently their motion should not be constrained by this term. The gradient descent flow that minimizes the radiance smoothness term is given by

$$S_t = 2((\mathbf{II}(\nabla_S \rho_1 \times N) - \kappa \|\nabla_S \rho_1\|^2) \mathcal{H}(\phi) + (\mathbf{II}(\nabla_S \rho_2 \times N) - \kappa \|\nabla_S \rho_2\|^2)(1 - \mathcal{H}(\phi)))N. \tag{13}$$

The derivation of (13) is similar the one in [10], to which we refer to the interested reader for more details.

To compute the variation of the data fitness term with respect to S , we need to introduce another two terms. Let $\chi_i : S \rightarrow \{0, 1\}$ be the surface visibility function with respect to the i -th camera, defined as follows: $\chi_i(\mathbf{X}) = 1$ for points on S that are visible from the i -th camera and $\chi_i(\mathbf{X}) = 0$ otherwise and σ_i be the change of coordinates from dA to $d\Omega_i$: $\sigma_i = \frac{d\Omega_i}{dA} = \langle \mathbf{X}_i, N_i \rangle / Z_i^3$, where N_i the unit normal N expressed in the i -th camera reference frame. We now can express the data term as follows:

$$\begin{aligned} & \sum_{i=1}^n \left(\int_{\pi_i(D_1)} (I_i(\mathbf{x}_i) - \rho_1(\pi_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i + \int_{\pi_i(D_2)} (I_i(\mathbf{x}_i) - \rho_2(\pi_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i + \int_{Q_i^c} (I_i(\mathbf{x}_i) - h(\Theta_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i \right) \\ &= \sum_{i=1}^n \left(\int_{\pi_i(D_1)} (I_i(\mathbf{x}_i) - \rho_1(\pi_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i + \int_{\pi_i(D_2)} (I_i(\mathbf{x}_i) - \rho_2(\pi_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i \right. \\ & \quad \left. - \int_{Q_i} (I_i(\mathbf{x}_i) - h(\Theta_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i + \int_{\Omega_i} (I_i(\mathbf{x}_i) - h(\Theta_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i \right) \\ &= \sum_{i=1}^n \left(\int_{D_1} \chi_i (I_i(\pi_i(\mathbf{X})) - \rho_1(\mathbf{X}))^2 \sigma_i dA + \int_{D_2} \chi_i (I_i(\pi_i(\mathbf{X})) - \rho_2(\mathbf{X}))^2 \sigma_i dA \right. \\ & \quad \left. - \int_S \chi_i (I_i(\pi_i(\mathbf{X})) - h(\Theta_i^{-1}(\pi_i(\mathbf{X}))))^2 \sigma_i dA + \int_{\Omega_i} (I_i(\mathbf{x}_i) - h(\Theta_i^{-1}(\mathbf{x}_i)))^2 d\Omega_i \right) \\ &= \sum_{i=1}^n \left(\int_S \chi_i \mathcal{H}(\phi) (I_i - \rho_1)^2 \sigma_i dA + \int_S \chi_i (1 - \mathcal{H}(\phi)) (I_i - \rho_2)^2 \sigma_i dA - \int_S \chi_i (I_i - h)^2 \sigma_i dA + \int_{\Omega_i} (I_i - h)^2 d\Omega_i \right) \\ &= \sum_{i=1}^n \int_S \chi_i (\mathcal{H}(\phi) (I_i - \rho_1)^2 + (1 - \mathcal{H}(\phi)) (I_i - \rho_2)^2 - (I_i - h)^2) \sigma_i dA + \sum_{i=1}^n \int_{\Omega_i} (I_i - h)^2 d\Omega_i. \tag{14} \end{aligned}$$

For ease of notation, we have dropped the arguments for I_i, ρ_1, ρ_2, h and ϕ in the last two steps of the above derivation. We can re-write (14) in a more concise form as

$$\sum_{i=1}^n \int_S \chi_i \Gamma_i \sigma_i dA + \sum_{i=1}^n \int_{\Omega_i} (I_i - h)^2 d\Omega_i, \tag{15}$$

where

$$\Gamma_i \doteq \mathcal{H}(\phi) (I_i - \rho_1)^2 + (1 - \mathcal{H}(\phi)) (I_i - \rho_2)^2 - (I_i - h)^2. \tag{16}$$

Since $\sum_{i=1}^n \int_{\Omega_i} (I_i - h)^2 d\Omega_i$ does not depend upon the unknown surface, we only need to compute the variation of the

first term $\sum_{i=1}^n \int_S \chi_i \Gamma_i \sigma_i dA$ in (15) with respect to S . It is proven in [24] that the gradient descent flow for minimizing cost functionals of the general form $\sum_{i=1}^n \int_S \chi_i \Gamma_i \sigma_i dA$ takes the form:

$$S_t = \sum_{i=1}^n \frac{1}{Z_i^3} (\Gamma_i \langle \chi_{i\mathbf{X}}, R_i^T \mathbf{X}_i \rangle - \chi_i \langle \Gamma_{i\mathbf{X}}, R_i^T \mathbf{X}_i \rangle) N, \quad (17)$$

where $\chi_{i\mathbf{X}}$ and $\Gamma_{i\mathbf{X}}$ denote the derivatives of χ_i and Γ_i with respect to \mathbf{X} respectively. We further note that $\langle \Gamma_{i\mathbf{X}}, R_i^T \mathbf{X}_i \rangle = 0$ [20] and obtain

$$\begin{aligned} \langle \Gamma_{i\mathbf{X}}, R_i^T \mathbf{X}_i \rangle &= \delta(\phi) ((I_i - \rho_1)^2 - (I_i - \rho_2)^2) \langle \nabla_S \phi, R_i^T \mathbf{X}_i \rangle \\ &\quad + 2(\rho_1 - I_i) \langle \nabla_S \rho_1, R_i^T \mathbf{X}_i \rangle \mathcal{H}(\phi) \\ &\quad + 2(\rho_2 - I_i) \langle \nabla_S \rho_2, R_i^T \mathbf{X}_i \rangle (1 - \mathcal{H}(\phi)). \end{aligned} \quad (18)$$

Substituting (18) into (17), we obtain that the gradient descent flow for the data term is given by

$$\begin{aligned} S_t &= \left(\sum_{i=1}^n \frac{1}{Z_i^3} (\mathcal{H}(\phi)(I_i - \rho_1)^2 + (1 - \mathcal{H}(\phi))(I_i - \rho_2)^2 \right. \\ &\quad - (I_i - h)^2) \langle \chi_{i\mathbf{X}}, R_i^T \mathbf{X}_i \rangle \\ &\quad - \frac{\chi_i}{Z_i^3} (\delta(\phi) ((I_i - \rho_1)^2 - (I_i - \rho_2)^2) \langle \nabla_S \phi, R_i^T \mathbf{X}_i \rangle \\ &\quad + 2(\rho_1 - I_i) \langle \nabla_S \rho_1, R_i^T \mathbf{X}_i \rangle \mathcal{H}(\phi) \\ &\quad \left. + 2(\rho_2 - I_i) \langle \nabla_S \rho_2, R_i^T \mathbf{X}_i \rangle (1 - \mathcal{H}(\phi)) \right) N. \end{aligned} \quad (19)$$

We remark that the flow (19) consists of four terms, which act on different regions of the surface. The first term, which is shared by the approach of [24], acts on the occluding boundaries because of the derivative of the surface visibility function (see more discussions in [24]). The second term acts on the curve, since it contains $\delta(\phi)$. Due to the presence of the radiance characteristic functions $\mathcal{H}(\phi)$ and $1 - \mathcal{H}(\phi)$, the third and fourth terms act in D_1 and D_2 respectively. Putting (9), (12), (13) and (19) together, we obtain the whole gradient flow for the cost (8) to be:

$$\begin{aligned} S_t &= \left(\sum_{i=1}^n \frac{1}{Z_i^3} (\mathcal{H}(\phi)(I_i - \rho_1)^2 + (1 - \mathcal{H}(\phi))(I_i - \rho_2)^2 \right. \\ &\quad - (I_i - h)^2) \langle \chi_{i\mathbf{X}}, R_i^T \mathbf{X}_i \rangle \\ &\quad - \frac{\chi_i}{Z_i^3} (\delta(\phi) ((I_i - \rho_1)^2 - (I_i - \rho_2)^2) \langle \nabla_S \phi, R_i^T \mathbf{X}_i \rangle \\ &\quad + 2(\rho_1 - I_i) \langle \nabla_S \rho_1, R_i^T \mathbf{X}_i \rangle \mathcal{H}(\phi) \\ &\quad + 2(\rho_2 - I_i) \langle \nabla_S \rho_2, R_i^T \mathbf{X}_i \rangle (1 - \mathcal{H}(\phi)) \\ &\quad \left. + 2\alpha\kappa + \beta \frac{\delta(\phi)}{\|\nabla_S \phi\|} \mathbf{II}(\nabla_S \phi \times N) \right) N, \end{aligned}$$

$$\begin{aligned} &+ 2\gamma (\mathbf{II}(\nabla_S \rho_1 \times N) - \kappa \|\nabla_S \rho_1\|^2) \mathcal{H}(\phi) \\ &+ 2\gamma (\mathbf{II}(\nabla_S \rho_2 \times N) - \kappa \|\nabla_S \rho_2\|^2) (1 - \mathcal{H}(\phi)) \Big) N. \end{aligned} \quad (20)$$

We remark that flow (20) depends only upon the image values, *not the image gradients*, together with the derivatives of the modeled radiances ρ_1, ρ_2 and h , which are smooth and differentiable by assumption. This property greatly improves the robustness of the resulting algorithm to image noise when compared to other variational approaches [5] to stereo based on image-to-image matching (i.e. less prone to becoming “trapped” in local minima).

3.2 Updating the Curve

We prove in Appendix 2 that for a fixed S , the optimal curve can be found by updating it according to the following gradient descent flow:

$$\begin{aligned} C_t &= \left(\sum_{i=1}^n \chi_i ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) \sigma_i + \beta \kappa_g \right. \\ &\quad \left. + \gamma (\|\nabla_S \rho_2\|^2 - \|\nabla_S \rho_1\|^2) \right) \mathbf{n}, \end{aligned} \quad (21)$$

where κ_g is the geodesic curvature and \mathbf{n} is the normal of the curve in $T_P(S)$ (commonly referred to as the *intrinsic normal* to the curve C). The direction of \mathbf{n} is defined as pointing from D_1 to D_2 . Note that since $\mathbf{n} \in T_P(S)$, C stays in S as it evolves according to (21).

3.3 Updating the Radiances

With the surface and the curve fixed, the cost functional can be minimized by evolving ρ_1, ρ_2 and h according to the following set of partial differential equations, which relate to the Euler-Lagrange equations for the cost (8).

$$\begin{cases} \frac{\partial}{\partial t} \rho_1 = \Delta_S \rho_1 - \frac{1}{\gamma} \sum_{i=1}^n \chi_i (\rho_1 - I_i) \sigma_i & \text{in } D_1, \\ \langle \nabla_S \rho_1, \mathbf{n} \rangle = 0 & \text{on } C, \end{cases} \quad (22)$$

$$\begin{cases} \frac{\partial}{\partial t} \rho_2 = \Delta_S \rho_2 - \frac{1}{\gamma} \sum_{i=1}^n \chi_i (\rho_2 - I_i) \sigma_i & \text{in } D_2, \\ \langle \nabla_S \rho_2, \mathbf{n} \rangle = 0 & \text{on } C, \end{cases} \quad (23)$$

$$\frac{\partial}{\partial t} h = \Delta_\Theta h - \frac{1}{\gamma} \sum_{i=1}^n \chi_i (h - I_i) \hat{\sigma}_i \quad \text{in } B \quad (24)$$

where Δ_S and Δ_Θ denote the intrinsic Laplacian (Laplace-Beltrami operator) over S and B respectively. $\hat{\sigma}_i$ denotes the change of coordinates from $d\Omega_i$ to $d\Theta$, i.e., $\hat{\sigma}_i = \frac{d\Omega_i}{d\Theta}$.

4 The Piecewise Constant Case

In this section, we discuss an important simplification of the model (8) when the radiance is piecewise constant. This simplification corresponds to giving an infinite weight to the radiance smoothness term, i.e., let $\gamma \rightarrow \infty$. Such a simplification has also been considered in region-based image segmentation [2, 15]. It has been shown to yield good segmentation results with much lower computational cost.

When the radiances are piecewise constant, our energy functional reduces to

$$\begin{aligned}
 E_{total}(S, C, \rho_1, \rho_2, h) &= E_{data} + \alpha E_{surf} + \beta E_{curv} \\
 &= \sum_{i=1}^n \left(\int_{\pi_i(D_1)} (I_i(\mathbf{x}_i) - \rho_1)^2 d\Omega_i \right. \\
 &\quad \left. + \int_{\pi_i(D_2)} (I_i(\mathbf{x}_i) - \rho_2)^2 d\Omega_i \right. \\
 &\quad \left. + \int_{Q_i^c} (I_i(\mathbf{x}_i) - h)^2 d\Omega_i \right) + \alpha \int_S dA + \beta \int_C ds \quad (25)
 \end{aligned}$$

where ρ_1, ρ_2 and h are not functions, but constants. The gradient descent flow for the surface is given by

$$\begin{aligned}
 S_t = & \left(\sum_{i=1}^n \frac{1}{Z_i^3} (\mathcal{H}(\phi)(I_i - \rho_1)^2 + (1 - \mathcal{H}(\phi))(I_i - \rho_2)^2 \right. \\
 & - (I_i - h)^2) \langle \chi_i \mathbf{X}, R_i^T \mathbf{X}_i \rangle \\
 & + \frac{\chi_i}{Z_i^3} \delta(c) ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) \langle \nabla_S \phi, R_i^T \mathbf{X}_i \rangle \\
 & \left. + 2\alpha\kappa + \beta \frac{\delta(\phi)}{\|\nabla_S \phi\|} \mathbf{II}(\nabla_S \phi \times N) \right) N, \quad (26)
 \end{aligned}$$

while the update to the curve simplifies to

$$C_t = \left(\sum_{i=1}^n \chi_i ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) \sigma_i + \beta \kappa_g \right) \mathbf{n}. \quad (27)$$

Finally, the optimization with respect to the radiances does not involve partial differential equations any more and can be solved in closed forms as:

$$\begin{cases}
 \rho_1 = \frac{\sum_{i=1}^n \int_{\pi_i(D_1)} I_i(\mathbf{x}_i) d\Omega_i}{\sum_{i=1}^n \int_{\pi_i(D_1)} d\Omega_i}, \\
 \rho_2 = \frac{\sum_{i=1}^n \int_{\pi_i(D_2)} I_i(\mathbf{x}_i) d\Omega_i}{\sum_{i=1}^n \int_{\pi_i(D_2)} d\Omega_i}, \\
 h = \frac{\sum_{i=1}^n \int_{Q_i^c} I_i(\mathbf{x}_i) d\Omega_i}{\sum_{i=1}^n \int_{Q_i^c} d\Omega_i},
 \end{cases} \quad (28)$$

i.e., the optimal values are the sample averages of the intensity values in the corresponding regions.

5 Level Set Implementation

In this section, we report details on implementing the proposed algorithm. Both the surface and curve evolutions are carried out in Osher and Sethian’s level set framework [16–18]. Level set methods are known for their ability to handle topological changes, which is important in our problem for we do not know the number of connected components of the scene surfaces and the number of photometrically consistent regions at the outset. Since there have already been textbooks [16, 18] on the method itself and a considerable amount of work on shape reconstruction using the method ([5, 10] and references therein), we will not elaborate more in this paper on the numerical implementation of the surface evolution. However, we would like to remark that one should exercise great caution in choosing the tuning parameters α and γ , because they are related to the stability of the flow (20). We refer the interested reader to [10] for more discussion on how to choose the tuning parameters. Many other techniques considered in [10] are also deployed in our implementation, such as the computation of the surface visibility function χ_i and the interpolation of the intensities. The implementation of (22), (23), and (24) have also been discussed extensively in [10].

Initialization can be performed in a number of ways. The simplest is to choose a generic shape, such as a large sphere or cube. More elaborate initialization can be performed if camera pose is known and if the background is sufficiently distinct from the object of interest. In this case, joint segmentation of the images and back-projection onto the scene produces an approximation of the “visual hull” of the object [12, 13]. Naturally, the closer the initial surface to the true one, the faster convergence will be achieved. We have tested our algorithm with both options; in the experimental section, in order to give a conservative estimate of the computational complexity of our algorithms, we use a generic initialization.

We will devote the rest of this section to the issues related to the implementation of the flow (27). Note that flow (21) and (27) are not simple planar curve evolutions. Our numerical implementation has to respect the fact that the curve, by definition, has to stay on the surface at any time. The way in which we approach the problem is to exploit the characteristic function of the radiance. Our approach is similar to the one considered by [3]. To simplify the notation, we write $C_t = \mathcal{F}_C \mathbf{n}$ as a general curve evolution, where $\mathcal{F}_C \in \mathbb{R}$ is the speed function. Recall that C is the zero level set of ϕ : $\phi(C, t) = 0$. Taking the derivative with respect to t , we get

$$\phi_t + \langle \nabla_S \phi, C_t \rangle = 0. \quad (29)$$

Substituting the expression for C_t , we obtain

$$\phi_t + \mathcal{F}_C \langle \nabla_S \phi, \mathbf{n} \rangle = 0. \quad (30)$$

Recall that \mathbf{n} is the intrinsic normal, which is the normal to the curve. Since $\langle \nabla_S \phi, C_s \rangle = 0$ where s is the arc-length parameterization of C , we can express \mathbf{n} using ϕ as (recall the direction of \mathbf{n} is from D_1 to D_2):

$$\mathbf{n} = -\frac{\nabla_S \phi}{\|\nabla_S \phi\|}. \tag{31}$$

Thus, we get

$$\phi_t = \mathcal{F}_C \|\nabla_S \phi\|. \tag{32}$$

Equation (32) is the level set motion for the corresponding curve motion on a surface. To eventually get an equation only involving ϕ , we need to express all the terms in \mathcal{F}_C using ϕ . In particular, the expression for the geodesic curvature is given by:

$$\begin{aligned} \kappa_g &= \nabla_S \cdot \left(\frac{\nabla_S \phi}{\|\nabla_S \phi\|} \right) = \frac{\nabla_S \cdot \nabla_S \phi}{\|\nabla_S \phi\|} - \left\langle \nabla_S \phi, \nabla_S \left(\frac{1}{\|\nabla_S \phi\|} \right) \right\rangle \\ &= \frac{\Delta_S \phi}{\|\nabla_S \phi\|} - \frac{\nabla_S^T \phi \nabla_S^2 \phi \nabla_S \phi}{\|\nabla_S \phi\|^3}, \end{aligned} \tag{33}$$

where $\nabla_S^2 \phi$ and $\Delta_S \phi$ denote the intrinsic Hessian and the intrinsic Laplacian of ϕ respectively. After representing the curve C with ϕ , we can implement the curve evolution by evolving ϕ on the surface, which is the problem of evolving functions on surfaces. The approach we take is closely related to the one considered by [1, 10]. We relax ϕ from being a function defined on S to being a function defined in \mathbb{R}^3 . We denote with φ the extended function:

$$\varphi : \mathbb{R}^3 \rightarrow \mathbb{R} \quad \text{s.t.} \quad \varphi(\mathbf{X}) = \phi(\mathbf{X}) \quad \forall \mathbf{X} \in S. \tag{34}$$

We can then express the intrinsic gradient as follows:

$$\nabla_S \phi = \nabla \varphi - \langle \nabla \varphi, N \rangle N, \tag{35}$$

and the intrinsic Hessian as follows:

$$\begin{aligned} \nabla_S^2 \phi &= (I - NN^T) \nabla^2 \varphi (I - NN^T) \\ &\quad - (N^T \nabla \varphi) \frac{(I - NN^T) \nabla^2 \psi (I - NN^T)}{\|\nabla \psi\|}, \end{aligned} \tag{36}$$

where ∇^2 stands for the standard Hessian in space and ψ is the level set function for S . $\Delta_S \phi$ can be computed as

$$\Delta_S \phi = \text{trace}(\nabla_S^2 \phi) = \Delta \varphi - 2\kappa N^T \nabla \varphi - N^T \nabla^2 \varphi N. \tag{37}$$

Finally we are ready to implement our curve evolution equations. In particular, flow (27) is given by updating the following partial differential equation

$$\begin{aligned} \phi_t &= \|\nabla_S \phi\| \sum_{i=1}^n \chi_i ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) \sigma_i \\ &\quad + \beta \left(\Delta_S \phi - \frac{\nabla_S^T \phi \nabla_S^2 \phi \nabla_S \phi}{\|\nabla_S \phi\|^2} \right), \end{aligned} \tag{38}$$

with ϕ replaced by φ and $\nabla_S \phi$, $\nabla_S^2 \phi$ and $\Delta_S \phi$ replaced by the corresponding terms of φ according to (35), (36) and (37).

We discretize the derivatives in (38) using finite difference schemes. The term involving $\|\nabla_S \phi\|$ yields a Hamilton-Jacobi equation for φ . We use Hamilton-Jacobi ENO methods [19]. The other term is parabolic and therefore we use standard central differencing. Time stepping is done by the simple forward Euler method. The CFL condition (Courant-Friedrichs-Lewy condition) is $\Delta t = c_1 \Delta x^2$, where Δt is the time increment and Δx is the size of the grid cell. c_1 depends on the actual speed function. We remark that the implementation of equation (38) can be made more efficient by using narrow-band techniques [18].

6 Experiments

In this section, we report experimental results for the proposed algorithm on several representative datasets. Our implementation for the numerical integration of the PDEs to estimate surface shape, radiance, and segmentation of the radiance is written in C++ and all the experiments are carried out on a 2.8 GHz single-CPU PC running Linux. We adopt the performance criterion proposed in [8] to evaluate shape errors, which is the ratio between the volume of the symmetric difference between the estimated shape and the true shape and the volume of the true shape. This measure is sensitive to pose errors, which makes it a conservative performance criterion; one could compute tighter error measures by looking at normal displacement along the estimated surface, but we feel that a conservative criterion already validates the quality of the reconstruction, and therefore opt for this simpler volume of symmetric difference.

In Fig. 2 we show 4 out of 26 views of a synthetic scene, which consists of two spheres. All the images in the dataset are rendered using OpenGL and are of size 257×257 . Both spheres are painted in black with the word ‘‘ECCV’’ and the rest is white. The background is gray. Clearly modeling this scene with one single constant radiance would lead to gross errors. One cannot even reconstruct either the white or the black part using the constant or smooth radiance model in [10, 24] due to occlusions. For comparison purpose, we report the results of our implementation of [24] in Fig. 3 (the right 2 images). The left 4 images in Fig. 3 show the final reconstructed shape using the proposed algorithm. The red curve marks the discontinuities of the radiance. The explicit modeling of radiance discontinuities may enable further applications. For instance, one can flatten the surface and the curve and perform character recognition of the letters. The numerical grid used in both algorithms is of the same size $128 \times 128 \times 128$. It requires total 2400 iterations for surfaces, radiances, curves to converge starting from a generic

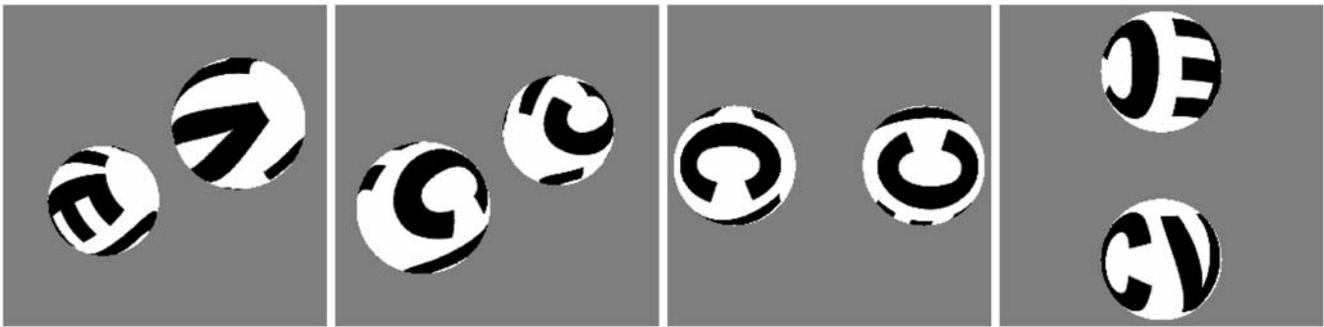


Fig. 2 4 images out of 26 views of a synthetic scene. The scene consists of two spheres, each of which is painted in black with the word “ECCV”. The rest of the spheres is white and the background is gray. Each image is of size 257×257

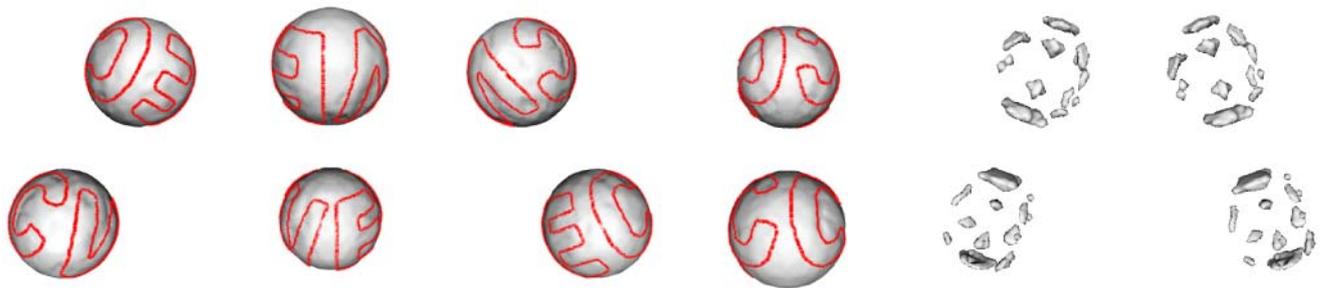


Fig. 3 The *first 4 images* are shaded views of the final shape estimated using the proposed algorithm. Radiance discontinuities have been rendered as red curves. The locations of the radiance discontinuities can be exploited for further purpose, for instance character recognition. The

last 2 images are the results by assuming the foreground has one constant radiance [24]. Note that the algorithm of [24] cannot capture all the white parts or all the black parts of the spheres, because that is not consistent with the input images due to occlusion

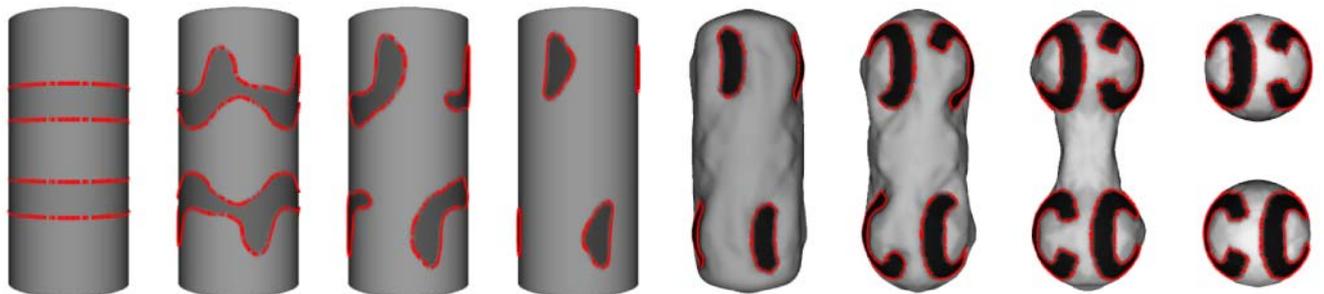


Fig. 4 Rendered surface during evolution. The foreground in all the images are rendered with the current estimate of the radiance values (ρ_1 and ρ_2) plus shading effects for ease of visualization

cylinder and takes around one hour. The shape error is 1.2%. Note, as we have already pointed out, that speed of execution depends crucially on initialization, and a simple strategy to compute the visual hull will significantly expedite the overall computation time. Here, for the sake of being conservative, we report total computation time starting from a generic shape that is not tailored to the object of interest. In Fig. 4 we show the surface evolving from a large cylinder to a final solid model. The foreground in all the images is rendered with its estimated radiance values (ρ_1 and ρ_2) and the segmenting curve is rendered in red. In Fig. 5 we show the

images reconstructed using the estimated surface, radiances and segmenting curve compared with one actual image in the original dataset.

In Fig. 6 we show 4 out of 16 views of another synthetic scene. The scenes consists of 4 objects: two bright spheres, one dark cube and one dark cylinder. All the images are rendered using OpenGL and each is of size 513×513 . In Fig. 7 we show the rendered surface evolving from an initial ellipsoid that neither contains nor is contained in the shape of the scene to the final objects. The objects in all the images are rendered with the current estimate of the radiances values



Fig. 5 The *first image* is just one view from the original dataset. The *remaining 6 images* are rendered using estimates from different stages of the estimation process. In particular, the *second image* is rendered using the initial data and the last image is rendered using the final estimates

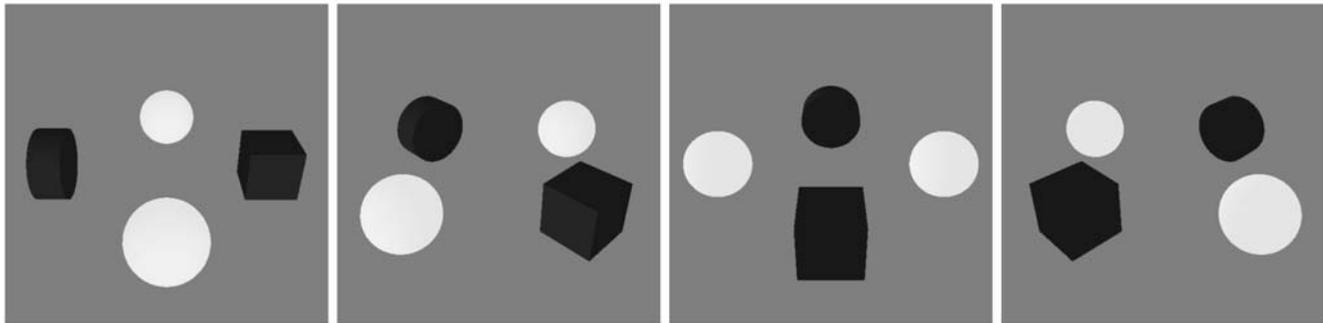


Fig. 6 4 out of 16 input images of a synthetic scene. The scene consists of 4 objects: two bright spheres, one dark cube and one dark cylinder. All the images are rendered using OpenGL and each is of dimension 513×513

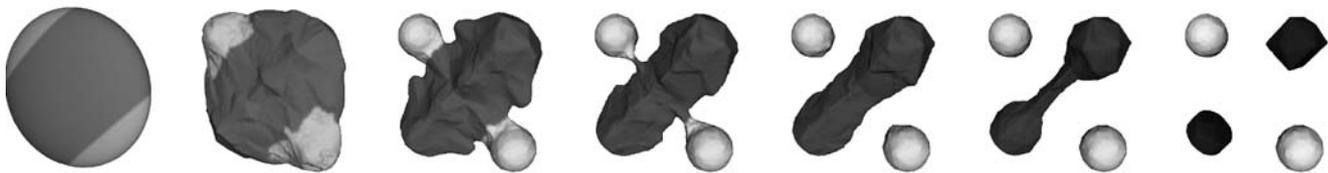


Fig. 7 Surface rendering through evolution. The objects in all the images are rendered with the current estimate of the radiance values (ρ_1 and ρ_2) plus some shading effects for ease of visualization

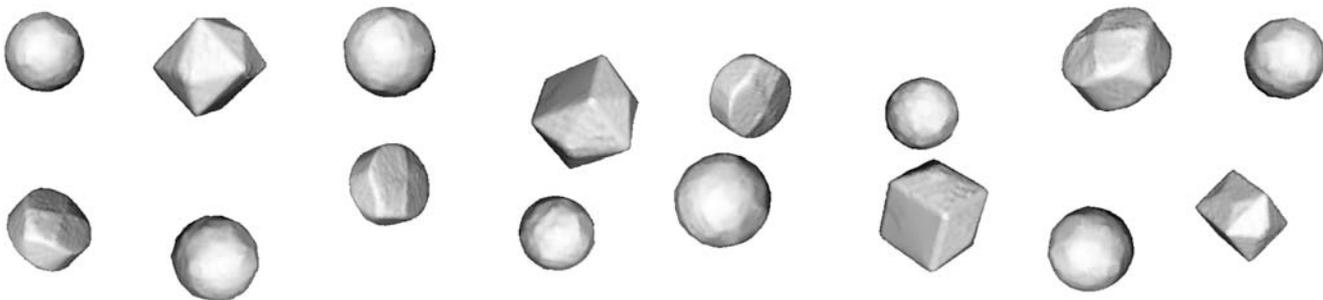


Fig. 8 4 shaded rendering of the final estimated surfaces

(ρ_1 and ρ_2) plus shading effects for ease of visualization. In Fig. 8 we show the shaded views of the final estimated shape from different vantage points. The numerical grid used here is of size $128 \times 128 \times 128$. It requires a total of 1500 iterations for surfaces, radiance and curves to converge starting from a generic ellipsoid and takes around two hours. (This one runs slower under few iterations because the input images are larger). The shape error is 3.7% which is also higher

compared to the first dataset. The reason for a higher shape error is that we do not have enough side views to constrain all the objects. As one can see from Fig. 8, both two flat sides of the cylinder and all sides of the cube are inflated. This is caused by the geometric prior which dominates when image data are not sufficiently strong in the cost function. Note that the reconstruction of spheres is not severely affected by insufficient number of images as they are the smoothest geo-

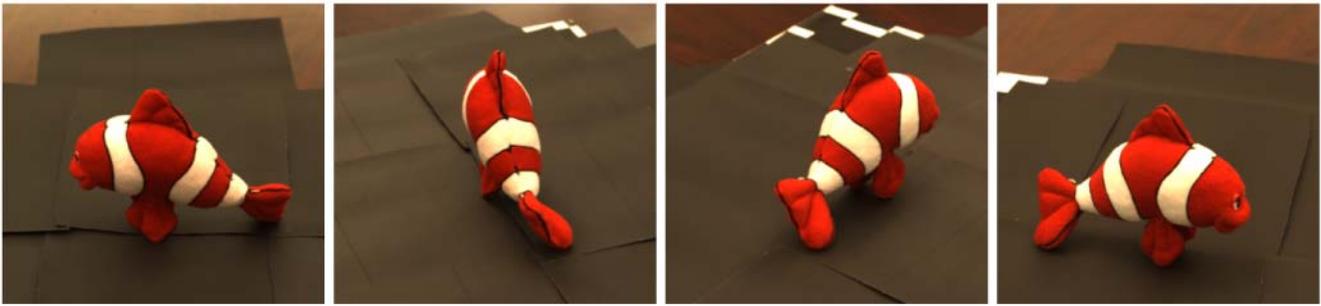


Fig. 9 4 images out of 31 images from “Nemo” dataset. Each image is of size 335×315 and calibrated manually using a calibration rig

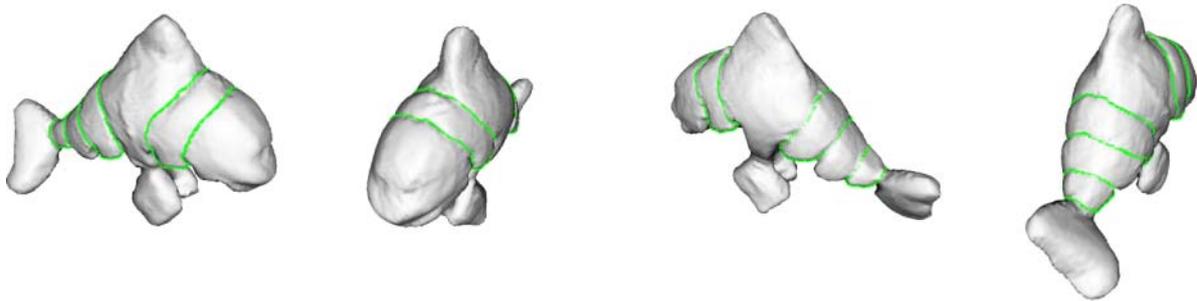


Fig. 10 Several shaded views of the final reconstructed surface. The radiance discontinuities have been highlighted in green

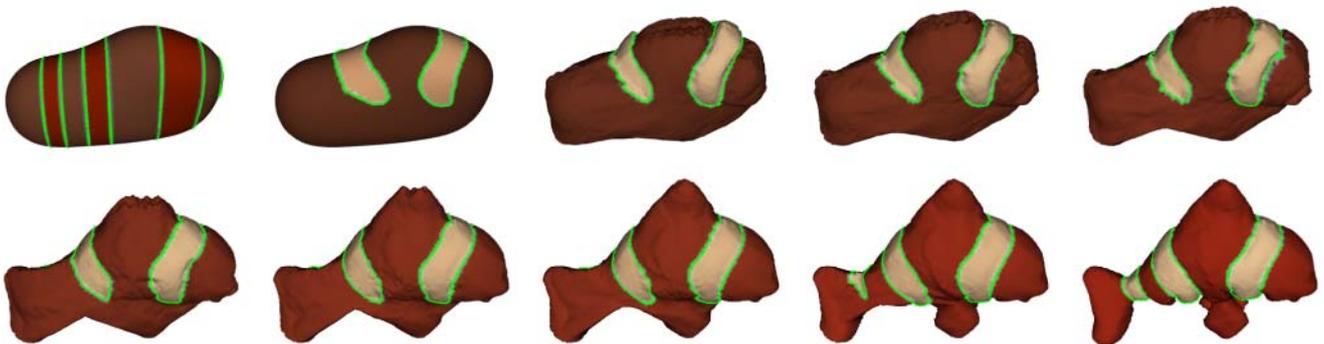


Fig. 11 Rendered surface during evolution. Notice that the initial surface is neither contained nor contains the actual object. The foreground in all the images are rendered with the current estimate of the radiance values (ρ_1 and ρ_2) plus some shading effects for ease of visualization

metric entities with a given volume. Again a simple computation of the visual hull to initialize the algorithm will significantly shorten the total computation time.

In Fig. 9 we show 4 out of 31 views of a real scene, which contains a plush model of the fish Nemo. The intrinsic and extrinsic parameters of the cameras have been calibrated off-line [14]. Each image is of size 335×315 . Nemo is red with white stripes. For the proposed algorithm to work with color images, we have extended the model (25) as follows: We consider images to take vector values (RGB color in our case) and modify the square error between scalars in (25) to the simple square of Euclidean vector norm. In Fig. 10 we show several shaded views of the final reconstructed shape

using the proposed algorithm. The radiance discontinuities are rendered as green curves. The numerical grid used here is of size $128 \times 60 \times 100$. On this dataset, it takes 4300 iterations for the algorithm to converge for a total computation time of about three and a half hours. In Fig. 11 we show the surface evolving from an initial shape that neither contains nor is contained in the shape of the scene, to a final solid model. The foreground in all the images is rendered with its estimated radiance values (ρ_1 and ρ_2) and the segmenting curve is rendered in green. In Fig. 12 we show the images reconstructed using the estimated surface, radiances and segmenting curve compared with one actual image from the original dataset.



Fig. 12 The *first image* is just one view from the original dataset. The *remaining 6 images* are rendered using estimates from different stages of the estimation process. In particular, the second image is rendered using the initial data and the last image is rendered using the final estimates

7 Conclusions

We have presented what to the best of our knowledge is the first algorithm to reconstruct the shape and radiance of a Lambertian scene with piecewise smooth or piecewise constant radiance from a collection of calibrated views. We set the problem in a variational framework and minimize a cost functional with respect to the unknown shape, unknown radiance values in each region, and unknown radiance discontinuities, by a gradient-descent partial differential equation that simultaneously evolves a surface in space (shape), a function defined on regions of that surface (radiance) and a curve defined on the surface (radiance discontinuities), implemented numerically using level set methods.

Among the features of our approach is the fact that, despite being a gradient-based algorithm, it does not involve the derivative of the (noisy) data, but only derivatives of the (analytical) model. This affords our approach significant robustness to noise when compared to the state of the art. On the flip side is the fact that, like other variational methods, our approach is not computationally light, especially compared to feature-based approaches. On the other hand, our algorithm provides a full dense reconstruction of shape and radiance, so its runtime should be compared to not just feature-based reconstruction, but to the entire pipeline that yield a texture-mapped model, which includes epipolar rectification, dense correspondence, triangulation, mesh simplification, and texture mapping. Furthermore, our approach is far more flexible, since it does not rely on the assumption that enough distinctive features are visible, and can handle regions with distinctive radiance profiles (e.g. edges, corners) as well as smooth profiles. Dense textures can be handled easily within this framework if one considers, rather than radiance, the output of filter banks as intermediate representations. This, however, is beyond our scope in this paper and is part of our future research agenda, along

with multi-scale implementations that would render our algorithm competitive also on computational grounds.

Acknowledgements This research was supported in part by AFOSR E-16-V91-G2/FA9550-06-1-0138, ONR67F-1080868 and Adobe Systems Incorporated. Part of this work was carried out while Hailin Jin was with the University of California at Los Angeles.

Appendix 1: Gradient Descent Flow for $\int_C ds$ with Respect to the Surface

In order to derive the results in this section and the next, we must parameterize the surface with fixed local coordinates (u, v) , which are independent of S itself. We will further assume that S may be covered by a single coordinate patch. This seems like a very restrictive assumption but, in the end, the resulting flows will have completely local expressions which do not rely upon this global assumption. We denote with E , F , and G the coefficients of the first fundamental form with respect to the local coordinates (u, v) , i.e., $E = \langle S_u, S_u \rangle$, $F = \langle S_u, S_v \rangle$, $G = \langle S_v, S_v \rangle$, and with e , f , and g the coefficients of the second fundamental form, i.e., $e = \langle S_{uu}, N \rangle$, $f = \langle S_{uv}, N \rangle$, $g = \langle S_{vv}, N \rangle$. We first rewrite (11) in local coordinates as

$$\begin{aligned} \int_C ds &= \int_S \delta(\phi) \|\nabla_S \phi\| dA \\ &= \iint \delta(\phi) \sqrt{\phi_u^2 G - 2\phi_u \phi_v F + \phi_v^2 E} dudv, \end{aligned} \quad (39)$$

where $\|\nabla_S \phi\|$, expressed in (u, v) coordinates, takes the expression $\sqrt{\frac{\phi_u^2 G - 2\phi_u \phi_v F + \phi_v^2 E}{EG - F^2}}$ (see [10] for more details on deriving this expression). Note that for ease of notation, we have omitted the integration domain for u and v . Taking the derivative of equation (39) with respect to (iteration) time, we obtain:

$$\begin{aligned} &\frac{\partial}{\partial t} \int_S \delta(\phi) \|\nabla_S \phi\| dA \\ &= \iint \delta(\phi) \frac{\partial}{\partial t} \sqrt{\phi_u^2 G - 2\phi_u \phi_v F + \phi_v^2 E} dudv \\ &= \iint \delta(\phi) \frac{\phi_u^2 \langle S_v, S_{vt} \rangle - \phi_u \phi_v (\langle S_u, S_{vt} \rangle + \langle S_v, S_{ut} \rangle) + \phi_v^2 \langle S_u, S_{ut} \rangle}{\sqrt{\phi_u^2 G - 2\phi_u \phi_v F + \phi_v^2 E}} dudv \end{aligned}$$

$$\begin{aligned}
 &= - \iint \delta(\phi) \frac{\phi_u^2 \langle S_{vv}, S_t \rangle - 2\phi_u \phi_v \langle S_{uv}, S_t \rangle + \phi_v^2 \langle S_{uu}, S_t \rangle}{\sqrt{\phi_u^2 G - 2\phi_u \phi_v F + \phi_v^2 E}} dudv \quad (\text{integration by parts}) \\
 &= - \iint \left\langle S_t, \delta(\phi) \frac{\phi_u^2 g - 2\phi_u \phi_v f + \phi_v^2 e}{\sqrt{EG - F^2} \sqrt{\phi_u^2 G - 2\phi_u \phi_v F + \phi_v^2 E}} N \right\rangle \sqrt{EG - F^2} dudv \quad (\text{ignoring tangential terms}) \\
 &= - \int_S \left\langle S_t, \frac{\delta(\phi)}{\|\nabla_S \phi\|} \frac{\phi_u^2 g - 2\phi_u \phi_v f + \phi_v^2 e}{EG - F^2} N \right\rangle dA.
 \end{aligned}$$

Note that, in the above derivation, we have dropped all the tangential terms whenever they appear in inner products with S_t , because the presence of such terms will not affect the shape of the surface, but only the parameterization [4]. If we choose

$$S_t = \frac{\delta(\phi)}{\|\nabla_S \phi\|} \frac{\phi_u^2 g - 2\phi_u \phi_v f + \phi_v^2 e}{EG - F^2} N, \tag{40}$$

then $\frac{\partial}{\partial t} \int_S \delta(\phi) \|\nabla_S \phi\| dA \leq 0$. Therefore, the cost is reduced under the action of flow (40). The term $\frac{\phi_u^2 g - 2\phi_u \phi_v f + \phi_v^2 e}{EG - F^2}$ is equivalent to $\mathbf{II}(\nabla_S \phi \times N)$ (see [10] for detailed derivations) and thus (40) simplifies to:

$$S_t = \frac{\delta(\phi)}{\|\nabla_S \phi\|} \mathbf{II}(\nabla_S \phi \times N) N. \tag{41}$$

Appendix 2: Derivation of the Curve Evolution

In this section, we will derive the curve evolution that minimizes that the cost (8), i.e., (21). We will derive the results in two steps. First we will show that the curve flow that minimizes its length is the geodesic curvature motion. Then we will derive the results related to the data and the radiance smoothness terms. Note that we do not need to consider the surface area term in the curve evolution, because it does not involve the curve.

Similar to our treatment of the surface, we parameterize the curve with a fixed parameterization p as:

$$\int_C ds = \int \|C_p\| dp \tag{42}$$

where s is the arc-length parameterization for C . Again for ease of notation, we have omitted the integration domain for p . The time derivative of (42) yields:

$$\begin{aligned}
 \frac{d}{dt} \int_C ds &= \int \frac{\langle C_{pt}, C_p \rangle}{\|C_p\|} dp = \int \langle C_{pt}, C_s \rangle dp \\
 &= - \int \langle C_t, C_{sp} \rangle dp = - \int \langle C_t, C_{ss} \rangle ds, \tag{43}
 \end{aligned}$$

where we have used integration by parts. C_{ss} can be decomposed as follows:

$$C_{ss} = \langle C_{ss}, \mathbf{n} \rangle \mathbf{n} + \langle C_{ss}, N \rangle N = \kappa_g \mathbf{n} + \kappa_N N, \tag{44}$$

where \mathbf{n} is the intrinsic normal to the curve, N is the unit normal to the surface, and κ_g and κ_N are called the *geodesic* and *normal* curvatures respectively. Note that $C_{ss} \perp C_s$. In fact, C_s , \mathbf{n} and N form a local orthonormal frame for \mathbb{R}^3 . Since C is constrained to stay on the surface, we should ignore the motion component along N for it will take C out of the surface. Therefore, the final flow minimizes the curve length is given by

$$C_t = \kappa_g \mathbf{n}. \tag{45}$$

Flow (45) is commonly referred to as the *geodesic curvature motion* [11].

Now we show the curve flow for the data and radiance smoothness terms. Since both terms have the same expressions, we will derive the results only for the data term. The cost functional of interest takes the expression:

$$\begin{aligned}
 &\sum_{i=1}^n \int_{\pi_i(D_1)} (I_i - \rho_1)^2 d\Omega_i + \sum_{i=1}^n \int_{\pi_i(D_2)} (I_i - \rho_2)^2 d\Omega_i \\
 &\quad + \sum_{i=1}^n \int_{Q_i^c} (I_i - h)^2 d\Omega_i \\
 &= \sum_{i=1}^n \int_{Q_i} (I_i - \rho_1)^2 d\Omega_i + \sum_{i=1}^n \int_{Q_i^c} (I_i - h)^2 d\Omega_i \\
 &\quad + \sum_{i=1}^n \int_{\pi_i(D_2)} ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) d\Omega_i. \tag{46}
 \end{aligned}$$

Since $\sum_{i=1}^n \int_{Q_i} (I_i - \rho_1)^2 d\Omega_i$ and $\sum_{i=1}^n \int_{Q_i^c} (I_i - h)^2 d\Omega_i$ do not depend on C , we just need to compute the variation of $\sum_{i=1}^n \int_{\pi_i(D_2)} ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) d\Omega_i$ with respect to C . We express this term on the surface instead of the image domain by using the surface visibility function χ_i and the change of coordinates σ_i :

$$\begin{aligned}
 &\sum_{i=1}^n \int_{\pi_i(D_2)} ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) d\Omega_i \\
 &= \sum_{i=1}^n \int_{D_2} \chi_i ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) \sigma_i dA
 \end{aligned}$$

$$\begin{aligned}
 &= \sum_{i=1}^n \int_{S^{-1}(D_2)} \chi_i((I_i - \rho_2)^2 - (I_i - \rho_1)^2) \\
 &\quad \times \sigma_i \sqrt{EG - F^2} dudv. \tag{47}
 \end{aligned}$$

In the last step, we have expressed the surface integral with the fixed local coordinates $(u, v) \in S^{-1}(D_2)$. We assume again that D_2 can be covered with a single coordinate patch in the (u, v) space, which may not be possible in some cases. Since in the end we will arrive at an expression that does not depend on any particular parameterization, this is not a limitation. We apply Green’s theorem [6] in the domain $S^{-1}(D_2)$ and obtain

$$\begin{aligned}
 \int_{S^{-1}(D_2)} (V_u - U_v) dudv &= \oint_{\partial(S^{-1}(D_2))} U du + V dv \\
 &= \int (Uu_s + Vv_s) ds, \tag{48}
 \end{aligned}$$

where U and V are two scalar functions $U, V : S^{-1}(D_2) \rightarrow \mathbb{R}$ and are assumed to satisfy

$$\begin{aligned}
 V_u - U_v &= \sum_{i=1}^n \chi_i((I_i - \rho_2)^2 - (I_i - \rho_1)^2) \\
 &\quad \times \sigma_i \sqrt{EG - F^2}. \tag{49}
 \end{aligned}$$

$\partial(S^{-1}(D_2))$ is the boundary of $S^{-1}(D_2)$ in the (u, v) domain and is assumed to be a piecewise smooth simple closed curve, oriented counterclockwise. Moreover, we have $C = S(\partial(S^{-1}(D_2)))$. This technique is also considered by Zhu and Yuille in their work on region competition [25].

Now we proceed with computing the time derivative of $\int (Uu_s + Vv_s) ds$.

$$\begin{aligned}
 &\frac{d}{dt} \int (Uu_s + Vv_s) ds \\
 &= \frac{d}{dt} \int (Uu_p + Vv_p) dp \\
 &= \int (U_t u_p + U u_{pt} + V_t v_p + V v_{pt}) dp \\
 &= \int ((U_u u_t + U_v v_t) u_p + U u_{pt} + (V_u u_t + V_v v_t) v_p \\
 &\quad + V v_{pt}) dp. \tag{50}
 \end{aligned}$$

We apply integration by parts $\int (U u_{pt} + V v_{pt}) dp$ and obtain

$$\begin{aligned}
 &\int (U u_{pt} + V v_{pt}) dp \\
 &= \int -(U_p u_t + V_p v_t) dp \\
 &= \int -((U_u u_p + U_v v_p) u_t + (U_u u_p U_v v_p) v_t) dp. \tag{51}
 \end{aligned}$$

Combining (50) and (51) together, we obtain that

$$\begin{aligned}
 &\frac{d}{dt} \int (Uu_s + Vv_s) ds \\
 &= \int (U_v v_t u_p - U_v v_p u_t + V_u u_t v_p - V_u u_p v_t) dp \\
 &= \int ((V_u - U_v) v_p u_t - (V_u - U_v) u_p v_t) dp \\
 &= \int (V_u - U_v) (u_t v_p - v_t u_p) dp \\
 &= \int (V_u - U_v) (u_t v_s - v_t u_s) ds. \tag{52}
 \end{aligned}$$

We can express $u_t v_s - v_t u_s$ as an inner product in the tangent plane of S as

$$u_t v_s - v_t u_s = \langle [S_u, S_v] \begin{bmatrix} u_t \\ v_t \end{bmatrix}, [S_u, S_v] M^{-1} \begin{bmatrix} v_s \\ -u_s \end{bmatrix} \rangle, \tag{53}$$

where $M \doteq \begin{bmatrix} E & F \\ F & G \end{bmatrix}$. The motivation to consider such an inner product in the tangent space of S is that we want to constraint the motion of C to be in S . Noting $C_t = [S_u, S_v] \begin{bmatrix} u_t \\ v_t \end{bmatrix}$, we obtain

$$\begin{aligned}
 &\frac{d}{dt} \int (Uu_s + Vv_s) ds \\
 &= \int \langle C_t, (V_u - U_v) [S_u, S_v] M^{-1} \begin{bmatrix} v_s \\ -u_s \end{bmatrix} \rangle ds. \tag{54}
 \end{aligned}$$

We note that $(V_u - U_v) [S_u, S_v] M^{-1} \begin{bmatrix} v_s \\ -u_s \end{bmatrix}$ lives in the tangent space of S and furthermore

$$\langle [S_u, S_v] M^{-1} \begin{bmatrix} v_s \\ -u_s \end{bmatrix}, [S_u, S_v] \begin{bmatrix} u_s \\ v_s \end{bmatrix} \rangle = 0, \tag{55}$$

i.e., $(V_u - U_v) [S_u, S_v] M^{-1} \begin{bmatrix} v_s \\ -u_s \end{bmatrix} \perp C_s$. Therefore, $(V_u - U_v) [S_u, S_v] M^{-1} \begin{bmatrix} v_s \\ -u_s \end{bmatrix}$ takes the expression $a\mathbf{n}$, where $a \in \mathbb{R}$. Since the direction of \mathbf{n} is free, we choose it pointing from D_1 to D_2 . Recall that $\partial(S^{-1}(D_2))$ is oriented counterclockwise. We have that $\mathbf{n} = [S_u, S_v] \begin{bmatrix} -v_s \\ u_s \end{bmatrix}$ and

$$\begin{aligned}
 a &= \langle (V_u - U_v) [S_u, S_v] M^{-1} \begin{bmatrix} v_s \\ -u_s \end{bmatrix}, [S_u, S_v] \begin{bmatrix} -v_s \\ u_s \end{bmatrix} \rangle \\
 &= -\frac{(V_u - U_v)}{EG - F^2} \|(Gv_s + Fu_s)S_u - (Fv_s + Eu_s)S_v\| \\
 &= -\frac{(V_u - U_v)}{\sqrt{EG - F^2}} \sqrt{Eu_s^2 + Gv_s^2 + 2Fu_s v_s} \\
 &= -\sum_{i=1}^n \chi_i((I_i - \rho_2)^2 - (I_i - \rho_1)^2) \sigma_i
 \end{aligned}$$

where we have plugged in the expression for $(V_u - U_v)$ from (49) and used the fact that s is the arc-length parameterization of C and thus we have

$$\begin{aligned}
 \|C_s\| &= \|S_u u_s + S_v v_s\| \\
 &= \sqrt{Eu_s^2 + Gv_s^2 + 2Fu_s v_s} = 1. \tag{56}
 \end{aligned}$$

Finally we arrive at

$$\begin{aligned} \frac{d}{dt} \sum_{i=1}^n \int_{\pi_i(D_2)} ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) d\Omega_i \\ = - \int \left\langle C_t, \sum_{i=1}^n \chi_i ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) \sigma_i \mathbf{n} \right\rangle ds. \end{aligned} \quad (57)$$

Therefore, the gradient descent flow is given by:

$$C_t = \sum_{i=1}^n \chi_i ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) \sigma_i \mathbf{n}. \quad (58)$$

Similarly, we can get the gradient flow for the radiance smoothness term:

$$C_t = (\|\nabla_S \rho_2\|^2 - \|\nabla_S \rho_1\|^2) \mathbf{n}. \quad (59)$$

Combining (45), (58) and (59) together, we get the final gradient descent curve flow for the cost (8):

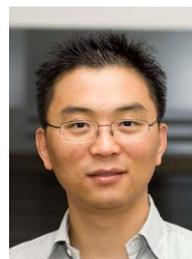
$$\begin{aligned} C_t = \left(\sum_{i=1}^n \chi_i ((I_i - \rho_2)^2 - (I_i - \rho_1)^2) \sigma_i + \beta \kappa_g \right. \\ \left. + \gamma (\|\nabla_S \rho_2\|^2 - \|\nabla_S \rho_1\|^2) \right) \mathbf{n}. \end{aligned} \quad (60)$$

This concludes our derivation.

References

1. Bertalmio, M., Cheng, L., Osher, S.J., Sapiro, G.: Variational problems and partial differential equations on implicit surfaces. *J. Comput. Phys.* **174**(2), 759–780 (2001)
2. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
3. Cheng, L.-T., Burchard, P., Merriman, B., Osher, S.J.: Motion of curves constrained on surfaces using a level-set approach. *J. Comput. Phys.* **175**(2), 604–644 (2002)
4. Epstein, C.L., Gage, M.: The curve shortening flow. In: *Wave Motion: Theory, Modelling, and Computation*, Berkeley CA, 1986. *Math. Sci. Res. Inst. Publ.*, vol. 7, pp. 15–59. Springer, New York (1987)
5. Faugeras, O., Keriven, R.: Variational principles, surface evolution, pdes, level set methods, and the stereo problem. *IEEE Trans. Image Process.* **7**(3), 336–344 (1998)
6. Greenberg, M.: *Advanced Engineering Mathematics*, 2 edn. Prentice Hall, New York (1998)
7. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge (2000)
8. Jin, H., Soatto, S., Yezzi, A.J.: Multi-view stereo reconstruction of dense shape and complex appearance. *Int. J. Comput. Vis.* **63**(3), 175–189 (2005)
9. Jin, H., Yezzi, A.J., Soatto, S.: Region-based segmentation on evolving surfaces with application to 3D reconstruction of shape and piecewise constant radiance. In: *Proc. Eur. Conf. on Computer Vision*, May 2004

10. Jin, H., Yezzi, A.J., Tsai, Y.-H., Cheng, L.-T., Soatto, S.: Estimation of 3d surface shape and smooth radiance from 2D images: a level set approach. *J. Sci. Comput.* **19**(1–3), 267–292 (2003)
11. Kimmel, R.: Intrinsic scale space for images on surfaces: the geodesic curvature flow. *Graph. Models Image Process.* **59**(5), 365–372 (1997)
12. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. *Int. J. Comput. Vis.* **38**(3), 199–218 (2000)
13. Laurentini, A.: The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Machine Intell.* **16**(2), 150–162 (1994)
14. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.: *An Invitation to 3-D Vision*. Springer, Berlin (2003)
15. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* **42**(5), 577–685 (1989)
16. Osher, S.J., Fedkiw, R.P.: *Level Set Methods and Dynamic Implicit Surfaces*, 1st edn. Springer, Berlin (2002)
17. Osher, S.J., Sethian, J.A.: Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J. Comput. Phys.* **79**(1), 12–49 (1988)
18. Sethian, J.A.: *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, 2nd edn. Cambridge University Press, Cambridge (1999)
19. Shu, C.-W., Osher, S.J.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **77**(2), 439–471 (1988)
20. Soatto, S., Yezzi, A.J., Jin, H.: Tales of shape and radiance in multi-view stereo. In: *Proc. of Int. Conf. on Computer Vision*, October 2003, vol. 1, pp. 171–178 (2003)
21. Vedaldi, A., Soatto, S.: Features for recognition: Viewpoint invariance for non-planar scenes. In: *Proc. of Int. Conf. on Computer Vision*, vol. 2, pp. 1474–1481 (2005)
22. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the Mumford and Shah model. *Int. J. Comput. Vis.* **50**(3), 271–293 (2002)
23. Yezzi, A.J., Soatto, S.: Deformation: deforming motion, shape average and the joint registration and approximation of structures in images. *Int. J. Comput. Vis.* **53**(2), 153–167 (2003)
24. Yezzi, A.J., Soatto, S.: Stereoscopic segmentation. *Int. J. Comput. Vis.* **53**(1), 31–43 (2003)
25. Zhu, S.C., Yuille, A.L.: Region competition: unifying snakes, region growing, and Bayes MDL for multiband image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(9), 884–900 (1996)



Hailin Jin received his Bachelor’s degree in Automation from Tsinghua University, Beijing, China in 1998. He then received his Master of Science and Doctor of Science degrees in Electrical Engineering from Washington University in Saint Louis in 2000 and 2003 respectively. Between fall 2003 and fall 2004, he was a post-doctoral researcher at the Computer Science Department, University of California at Los Angeles. He is currently a researcher at Adobe Systems Incorporated.

His research interests include curve and surface evolution, multi-view stereo reconstruction, structure and motion estimation, and nonlinear filtering. He is a member of the IEEE and the IEEE Computer Society.



Anthony J. Yezzi obtained his Ph.D. in 1997 through the Department of Electrical Engineering at the University of Minnesota. After completing a postdoctoral research position in the Laboratory for Information and Decision Systems (LIDS) at Massachusetts Institute of Technology, he joined the faculty of the School of Electrical and Computer Engineering at Georgia Institute of Technology in 1999 where he currently holds the position of Associate Profes-

sor. He has also consulted for a number of medical imaging companies including GE, Picker, and VTI, and has been an IEEE member since 1999. His research lies primarily within the fields of image processing and computer vision. He has worked on a variety of problems including image denoising, edge-detection, segmentation and grouping, shape analysis, multi-frame stereo reconstruction, tracking, and registration. Some central themes of his research include curve and surface evolution, differential geometry, and partial differential equations.



Stefano Soatto received his Ph.D. in Control and Dynamical Systems from the California Institute of Technology in 1996; he joined UCLA in 2000 after being Assistant and then Associate Professor of Electrical and Biomedical Engineering at Washington University, and Research Associate in Applied Sciences at Harvard University. Between 1995 and 1998 he was also Ricercatore in the Department of Mathematics and Computer Science at the University

of Udine - Italy. He received his D.Ing. degree (highest honors) from the University of Padova- Italy in 1992. He is the recipient of the David Marr Prize (with Y. Ma, J. Kosecka and S. Sastry of U.C. Berkeley) for work on Euclidean reconstruction and reprojection up to subgroups. He also received the Siemens Prize with the Outstanding Paper Award from the IEEE Computer Society for his work on optimal structure from motion (with R. Brockett of Harvard). He received the National Science Foundation Career Award and the Okawa Foundation Grant. He is a Member of the Editorial Board of the International Journal of Computer Vision (IJCV) and Foundations and Trends in Computer Graphics and Vision.