Computational stereo: a variational method.

O. Faugeras, J. Gomes, R. Keriven

ABSTRACT Given a set of simultaneously acquired images of a scene, the stereo problem consists of reconstructing the three-dimensional shape of the objects in the scene. Traditionally, this problem has been tackled by separating the *correspondence* problem, i.e. that of establishing which points in the images are the projections in the retina of the same 3D point, and the *reconstruction* problem, i.e. that of reconstructing the 3D shapes from the correspondences.
The approach described in this chapter does not separate the two problems. Given a set of objects (surfaces) in 3D space, we propose a function that measures the adequacy of these objects with the measured images. The problem is then to find a set of surfaces in the scene which maximize our function. This can be achieved by a variational approach. A level set implementation is described and results are given on synthetic and real images.

1 Introduction and preliminaries

The idea that is put forward in this paper is that the methods of curve and surface evolutions which have been developed in computer vision under the name of snakes [28] and then reformulated by Caselles, Kimmel and Sapiro [6] and Kichenassamy et al. [32] in the context of PDE driven evolving curves can be used effectively for solving 3D vision problems such as stereo and motion analysis.

As a first step in this direction we present a mathematical analysis of the stereo problem in this context as well as a level set implementation.

The problem of curve evolution driven by a PDE has been recently studied both from the theoretical standpoint [19, 22, 41] and from the viewpoint of implementation [37] with the development of level set methods that can efficiently and robustly solve those PDE's. The problem of surface evolution has been less touched upon even though some preliminary results have been obtained [7].

The path we will follow to attack the stereo problem from that angle is, not surprisingly, a variational one. In a nutshell, we will describe the stereo problem (to be defined more precisely later) as the minimization of a functional with respect to some parameters (describing the geometry of the

scene); we will compute the Euler-Lagrange equations of this functional, thereby obtaining a set of necessary conditions, in effect a set of partial differential equations, which we will solve as a time evolution problem by a level set method.

Stereo is a problem that has received considerable attention for decades in the psychophysical, neurophysiological and, more recently, in the computer vision literatures. It is impossible to cite all the published work here, we will simply refer the reader to some basic books on the subject [27, 23, 24, 25, 15]. To explain the problem of stereo from the computational standpoint, we will refer the reader to figure 1.a. Two, may be more, images of the world are taken simultaneously. The problem is, given those images, to recover the geometry of the scene. Given the fact that the relative positions and orientations and the internal parameters of the cameras are known which we will assume in this article (the cameras are then said to be calibrated [15]), the problem is essentially (but not only) one of establishing correspondences between the views: one speaks about the *matching* or *corespondence* problem. The matching problem is usually solved by setting up a matching functional for which one then tries to find extrema. Once a pixel in view i has been identified as being the image of the same scene point as another pixel in view j, the 3D point can then be reconstructed by intersecting the corresponding optical rays (see figure 1.a again): this is the *reconstruction* problem.

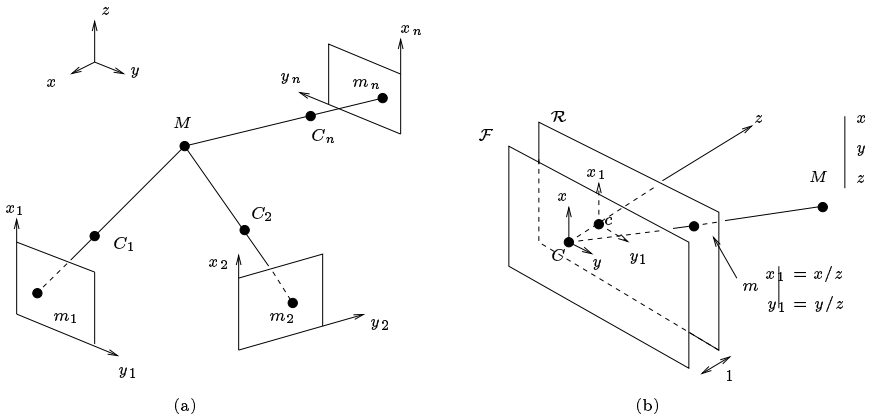

FIGURE 1. (a) The multi-camera stereo vision problem is, given a pixel m_1 in image 1, to find the corresponding pixel m_2 in image 2, ..., the corresponding pixel m_n in image n, i.e. the pixels which are the images of the same 3D point M. Once such a correspondence has been established, the point M can be reconstructed by intersecting the optical rays $\langle m_i, C_i \rangle$, $i = 1, \cdots, n$. (b) The focal plane (x, y) is parallel to the retinal plane (x_1, y_1) and at a unit distance from it.

The approach described in this chapter is different in the sense that it does not separate the *matching* and the *reconstruction* problems. Given a

set of objects (surfaces) in 3D space, we propose a function that measure the adequacy of these objects with the measured images. The problem is then to find a set of surfaces in the scene which maximize our function. This can be achieved by establishing the corresponding Euler-Lagrange equations with respect to the objects shape and applying a steepest infinitesimal gradient descent by solving the resulting parabolic equations. These parabolic equations describe the way the surfaces of the objects evolve toward a local maximum of our function.

We represent the objects shapes by the signed distance function to their boundary, this allows us to perform the implementation of the evolution equations through the level set method. Among the advantages of the level set method, the more important here is that the changes in the topology of the evolving surface (mainly its number of connected components) are handled automatically: typically, the initial shape is a large sphere enclosing the whole scene which shrinks down to the objects surfaces, splitting itself seamlessly into several surfaces whenever this is required. An unfortunate feature of the level set method is that the evolution does not in general preserve the distance function. We show that a modification of the level set equation can guarantee this invariance, resulting in a significant gain in computing time.

In order to go any further, we need to be a little more specific about the process of image formation. We will assume here that the cameras perform a perspective projection of the 3D world on the retinal plane as shown in figure 1.b. The optical center, noted C in the figure, is the center of projection and the image of the 3D point M is the pixel m at the intersection of the optical ray $\langle C, m\rangle$ and the retinal plane $\mathcal{R}$. As described in many recent papers in computer vision, this operation can be conveniently described in projective geometry by a matrix operation. The projective coordinates of the pixel m (a 3×1 vector) are obtained by applying a 3×4 matrix $\mathbf{P}_1$ to the projective coordinates of the 3D point M (a 4×1 vector). This matrix is called the perspective projection matrix. If we express the matrix $\mathbf{P}_1$ in the coordinate system (C, x, y, z) shown in the figure 1.b, it then takes a very simple form:

$$\mathbf{P}_1 = [\mathbf{I}_3\, \mathbf{0}]$$

where $\mathbf{I}_3$ is the 3×3 identity matrix. If we now move the camera by applying to it a rigid transformation described by the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$, the expression of the matrix $\mathbf{P}$ changes accordingly and becomes:

$$\mathbf{P}_2 = [\mathbf{R}^T \; -\mathbf{R}^T\mathbf{t}]$$

With these preliminaries in mind we are ready to proceed with our program which we will do by progressing along two related axes. The first axis is that of object complexity, the second axis is that of matching functional complexity.

In section 2, we first consider a simple object model which is well adapted to the binocular stereo case where it is natural to consider that the objects in the scene can be considered mathematically as forming the graph of an unknown smooth function (the depth function in the language of computer vision). We start with an extremely simplified matching criterion which allows us to convey to the reader the flavor of the ideas that we are trying to push here. We then move to a more sophisticated albeit classical matching criterion which is at the heart of the techniques known in computer vision as correlation based methods. Within the framework of this model we study two related shape models. Finally in section 3 we introduce a more general shape model in which we do not assume anymore that the objects are the graph of a function and model them instead as a set of general smooth surfaces in three dimensional space. The next steps would be to relax the smoothness assumption and to consider more complex matching criteria but we do not address them here.

Section 4 goes into some important implementation details, while section 5 presents results with synthetic and real images.

Let us give some definitions and fix notations. Images are denoted by I_k, k taking some integer values which indicate the camera with which the image has been acquired. They are considered as smooth (i.e. C^2, twice continuously differentiable) functions of pixels m_k whose coordinates are defined in some orthonormal image coordinate systems (x_k, y_k) which are assumed to be known. We note $I_k(m_k)$ or $I_k(x_k, y_k)$ the intensity value in image k at pixel m_k. We will use the first and second order derivatives of these functions, i.e. the gradient ∇I_k, a 2×1 vector equal to $[\frac{\partial I_k}{\partial x_k}, \frac{\partial I_k}{\partial y_k}]^T$, and the Hessian $\mathbf{H}_k$, a 2×2 symmetric matrix.

The pixels in the images are considered to be functions of the 3D geometry of the scene, i.e. of some 3D point M on the surface of an object in the scene, and of the unit normal vector $\mathbf{N}$ to the surface at this point.

Vectors and matrices are generally represented in boldface, e.g. $\mathbf{x}$. The dot or inner product of two vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted by $\mathbf{x} \cdot \mathbf{y}$. The cross-product of two 3×1 vectors $\mathbf{x}$ and $\mathbf{y}$ is noted $\mathbf{x} \times \mathbf{y}$.

Partial derivatives are represented either with the ∂ symbol, e.g. $\frac{\partial f}{\partial \mathbf{x}}$, or with a lower index, e.g. $f_{\mathbf{x}}$.

Our approach is an extension of previous work by Robert et al. and Robert and Deriche, [40, 39], where the idea of using a variational approach for solving the stereo problem was first proposed in the classical Tikhonov regularization framework and then by using regularization functions more proper to preserve discontinuities. Our work can be seen as a 3D extension of the approach proposed in [14] where we limit ourselves to the binocular case, to finding cross-sections of the objects with a fixed plane, and do not take into account the orientation of the tangent plane to the object. Preliminary versions of our work can be found in [16, 17, 18, 30, 21].

2 The simplified models

Let us now describe the different object models and the functions that measure the adequacy of the objects with the images. We will proceed from the simplest to the most sophisticated one. Note that, in order to turn our variational problem into a minimization problem, we consider error measures instead of adequacy measures.

2.1 A simple object and matching model

This section introduces in a simplified framework some of the basic ideas of this paper. We assume, and it is the first important assumption, that the objects which are being imaged by the stereo rig (a binocular stereo system) are modeled as the graph of an unknown smooth function $z = f(x, y)$ defined in the first retinal plane which we are trying to estimate. A point M of coordinates $[x, y, f(x, y)]^T$ is seen as two pixels m_1 and m_2 whose coordinates $(g_i(x, y), h_i(x, y)), i = 1, 2$, can be easily computed as functions of $x, y, f(x, y)$ and the coefficients of the perspective projection matrices $\mathbf{P}_1$ and $\mathbf{P}_2$. Let I_1 and I_2 be the intensities of the two images. Assuming, and it is the second important assumption, that the objects are perfectly Lambertian, we must have $I_1(m_1) = I_2(m_2)$ for all pixels in correspondence, i.e. which are the images of the same 3D point.

This reasoning immediately leads to the variational problem of finding a suitable function f defined, to be rigorous, over an open subset of the focal plane of the first camera and which minimizes the following integral:

$$C_1(f) = \int\int (I_1(m_1(x, y)) - I_2(m_2(x, y))^2 dxdy \tag{1.1}$$

computed over the previous open subset. Our first variational problem is thus to find a function f in some suitable functional space that minimizes the error measure $C_1(f)$. The corresponding Euler-Lagrange equation is readily obtained:

$$(I_1 - I_2)(\nabla I_1 \cdot \frac{\partial \mathbf{m}_1}{\partial f} - \nabla I_2 \cdot \frac{\partial \mathbf{m}_2}{\partial f}) = 0 \tag{1.2}$$

The values of $\frac{\partial \mathbf{m}_1}{\partial f}$ and $\frac{\partial \mathbf{m}_2}{\partial f}$ are functions of f which are easily computed. The terms involving I_1 and I_2 are computed from the images. In order to solve (1.2) one can adopt a number of strategies.

One standard strategy is to consider that the function f is also a function $f(x, y, t)$ of time and to solve the following PDE:

$$f_t = \varphi(f)$$

where $\varphi(f)$ is equal to the left hand side of (1.2), with some initial condition $f(x, y, 0) = f_0(x, y)$. We thus see for the first time appear the idea that the

shape of the objects in the scene, described by the function f, is obtained by allowing a surface of equation $z = f(x, y, t)$ to evolve over time, starting from some initial configuration $z = f(x, y, 0)$, according to some PDE, to hopefully converge toward the real shape of the objects in the scene when time goes to infinity. This convergence is driven by the data, i.e. the images, as expressed by the error criterion (1.1) or the Euler-Lagrange term $\varphi(f)$. It is known that if care is not taken, for example by adding a regularizing term to (1.1), the solution f is likely not to be smooth and therefore any noise in the images may cause the solution to differ widely from the real objects. This is more or less the approach taken in [40, 39]. We will postpone the solution of this problem until section 3 and in fact solve it differently from the usual way which consists in adding a regularization term to $C_1(f)$.

It is clear that the error measure (1.1) is a bit simple for practical applications. We can extend it in at least two ways. The first is to replace the difference of intensities by a measure of correlation, the hypothesis being that the scene is made of fronto parallel planes. The second is to relax this hypothesis and to take into account the orientation of the tangent plane to the surface of the object. We explore those two avenues in the next two sections.

2.2 Fronto parallel correlation functional

To each pair of values (x, y), corresponds a 3D point M, $\mathbf{M} = [x, y, f(x, y)]^T$ which defines two image points m_1 and m_2 as in the previous section. We can then classically define the unnormalized cross-correlation between I_1 and I_2 at the pixels m_1 and m_2. We note this cross-correlation $\langle I_1, I_2 \rangle(f, x, y)$ to acknowledge its analogy with an inner product and the fact that it depends upon M:

$$\langle I_1, I_2 \rangle(f, x, y) = \frac{1}{4pq} \int_{-p}^{+p} \int_{-q}^{+q} (I_1(m_1 + m) - \overline{I_1}(m_1)) \\ (I_2(m_2 + m) - \overline{I_2}(m_2)) dm \tag{1.3}$$

equation where the averages $\overline{I_1}$ and $\overline{I_2}$ are classically defined as:

$$\overline{I_k}(m_k) = \frac{1}{4pq} \int_{-p}^{+p} \int_{-q}^{+q} I_k(m_k + m') dm' \quad k = 1, 2 \tag{1.4}$$

Finally, we note $\mid I \mid^2$ the quantity $\langle I, I \rangle$. Note that $\langle I_1, I_2 \rangle = \langle I_2, I_1 \rangle$.

To simplify notations we write $\int^*$ instead of $\frac{1}{4pq} \int_{-p}^{+p} \int_{-q}^{+q}$ and define a matching functional which is the integral with respect to x and y of minus the normalized cross-correlation score $-\frac{\langle I_1, I_2 \rangle}{|I_1| \cdot |I_2|}$:

$$C_2(f) = -\int\int \frac{\langle I_1, I_2 \rangle}{\mid I_1 \mid \cdot \mid I_2 \mid} dxdy = \int\int {}_2\Phi(f, x, y) dxdy \tag{1.5}$$

the integral being computed, as in the previous section, over an open set of the focal plane of the first camera. The functional ${}_2\Phi$ is $-\frac{\langle I_1, I_2 \rangle}{|I_1| \cdot |I_2|}(f, x, y)$. This quantity varies between -1 and +1, -1 indicating the maximum correlation. We have to compute its derivative with respect to f in order to obtain the Euler-Lagrange equation of the problem. The computations are simple but a little fastidious. They can be found in [16]. We could then proceed to the corresponding steepest gradient descent as mentioned in the previous section. But we will not pursue this task and explore rather a better functional.

2.3 Taking into account the tangent plane to the object

We now take into account the fact that the rectangular window centered at m_2 should be the image in the second retina of the back-projection on the tangent plane to the object at the point $M = (x, y, f(x, y))$ of the rectangular window centered at m_1 (see figure 2). In essence, we approximate the object S in a neighborhood of M by its tangent plane but without assuming, as in the previous section, that this plane is fronto parallel, and in fact also that the retinal planes of the two cameras are identical. Let us first study the correspondence induced by this plane between the two images.

Image correspondences induced by a plane

Let us consider a plane of equation $\mathbf{N}^T\mathbf{M} - d = 0$ in the coordinate system of the first camera. d is the algebraic distance of the origin of coordinates to that plane and $\mathbf{N}$ is a unit vector normal to the plane. This plane induces a projective transformation between the two image planes. This correspondence plays an essential role in the sequel.

To see why we obtain a projective transformation, let M be a 3D point in that plane, $\mathbf{M}_1$ and $\mathbf{M}_2$ be the two 3D vectors representing this point in the coordinate systems attached to the first and second cameras, respectively. These two 3×1 vectors are actually coordinate vectors of the two pixels m_1 and m_2 seen as projective points (see Sect. 1). Furthermore, they are related by the following equation:

$$\mathbf{M}_2 = \mathbf{R}^T(\mathbf{M}_1 - \mathbf{t})$$

Since M belongs to the plane, $\mathbf{N}^T\mathbf{M}_1 = d$, and we have:

$$\mathbf{M}_2 = (\mathbf{R}^T - \frac{\mathbf{R}^T\mathbf{t}\mathbf{N}^T}{d})\mathbf{M}_1$$

which precisely expresses the fact that the two pixels m_1 and m_2 are related by a collineation, or projective transformation K. The 3×3 matrix representing this collineation is $(\mathbf{R}^T - \frac{\mathbf{R}^T\mathbf{t}\mathbf{N}^T}{d})$. This transformation is one

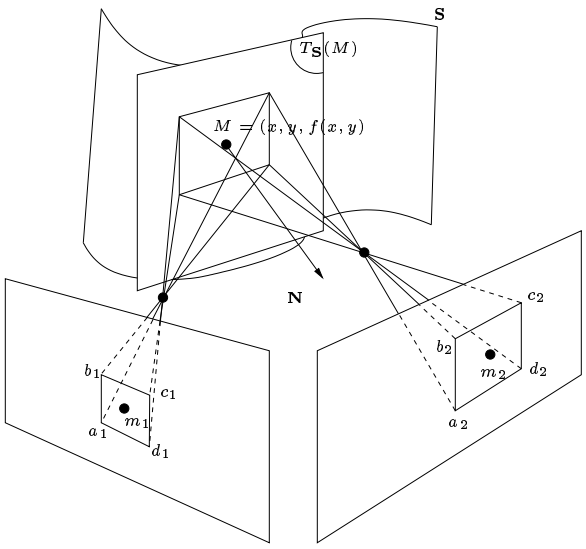

FIGURE 2. The square window (a_1, b_1, c_1, d_1) in the first image is back projected onto the tangent plane to the object S at point M and reprojected in the retinal plane of the second camera where it is generally not square. The observation is that the distortion between (a_1, b_1, c_1, d_1) and (a_2, b_2, c_2, d_2) can be described by a collineation which is function of M and the normal $\mathbf{N}$ to the surface of the object.

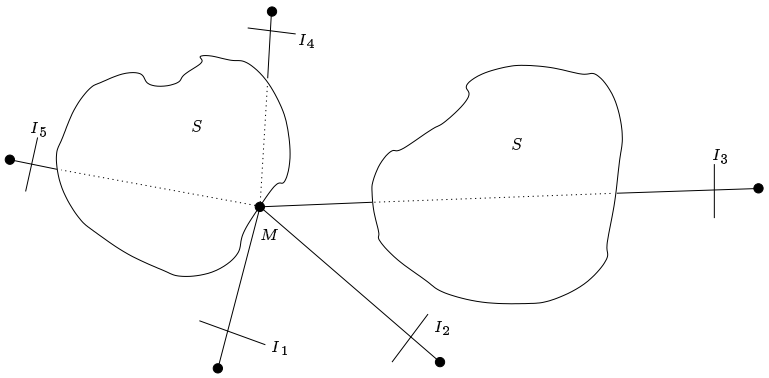

FIGURE 3. Occlusion and visibility are taken into account: only the cameras seeing the points on the current surface are used in the error criterion, thus avoiding the computation of irrelevant correlation. Here, only the cameras 1 and 2 are considered for point M

to one except when the plane goes through one of the two optical centers when it becomes degenerate. We will assume that it does not go through either one of those two points and since the matrix of K is only defined up to a scale factor we might as well take it equal to:

$$\mathbf{K} = d\mathbf{R}^T - \mathbf{R}^T\mathbf{t}\mathbf{N}^T \tag{1.6}$$

The new criterion and its Euler-Lagrange equations

We just saw that a plane induces a collineation between the two retinal planes. This is the basis of the method proposed in [14] although for a very different purpose. The window alluded to in the introduction to this section is therefore the image by the collineation induced by the tangent plane of the rectangular window in image 1. This collineation is a function of the point M and of the normal to the object at M. It is therefore a function of f and ∇f that we denote by K. It satisfies the condition $K(m_1) = m_2$. The inner product (1.3) must be modified as follows:

$$\begin{aligned}\langle I_1, I_2\rangle(f, \nabla f, x, y) = \int^* & (I_1(m_1 + m) - \overline{I_1}(m_1)) \\ & (I_2(K(m_1 + m)) - \overline{I_2}(m_2))dm,\end{aligned} \tag{1.7}$$

Note that, the definition of $\langle I_1,\ I_2\rangle$ is no longer symmetric, because of K. This problem could be address as mentioned in [18]. Yet, we will see in section 3 that our final error measure will solve it in a more natural way.

We now want to minimize the following error measure:

$$\begin{aligned}C_3(f, \nabla f) &= \int\int {}_3\Phi(f, \nabla f, x, y)dxdy \\ &\text{with } {}_3\Phi = -\frac{\langle I_1, I_2\rangle}{\mid I_1 \mid \cdot \mid I_2 \mid}(f, \nabla f, x, y)\end{aligned} \tag{1.8}$$

Since the functional ${}_3\Phi$ now depends upon both f and ∇f, its Euler-Lagrange equations have the form ${}_3\Phi_f - div({}_3\Phi_{\nabla f}) = 0$. We must therefore recompute ${}_3\Phi_f$ to take into account the new dependency of K upon f and compute ${}_3\Phi_{\nabla f}$.

We could simplify the computations by assuming that the collineation K can be well approximated by an affine transformation. Because of the condition $K(m_1) = m_2$, this transformation can be written:

$$K(m_1 + m) \approx m_2 + \mathbf{A}\mathbf{m}$$

where $\mathbf{A}$ is a 2×2 matrix depending upon f and ∇f.

In practice this approximation is often sufficient and we will assume that it is valid in what follows. We will not pursue this derivation (see [16] for details) since we present in section 3 a more elaborate model that encompasses this one and for which we will perform the corresponding computation.

3 The complete model

We are now ready to consider the case when the objects in the scene are not defined as the graph of a function of x and y as in the previous sections, but as a surface $\hat{S}$ of $\mathbf{R}^3$ which we assume to be smooth, i.e. C^2. Note that, by relaxing the graph assumption, it potentially allows us to use an arbitrary number of cameras to analyze the scene. Let us consider a family of smooth surfaces $S(t)$ where t is the time. Our goal is, starting from an initial surface S_0, to derive a partial differential equation

$$\mathbf{S}_t = \beta \mathbf{N}, \tag{1.9}$$

where $\mathbf{S}$ denotes a point on S and $\mathbf{N}$ is the inner unit normal to the surface at $\mathbf{S}$, which, when solved with initial condition $S(0) = S_0$, will yield a solution that closely approximates $\hat{S}$.

The function β is determined by the matching functional that we minimize in order to solve the stereo problem. We define such a functional in the next paragraph. An interesting point is that, if β is intrinsic (see below), the evolution equation (1.9) can be solved using the level set method which has the advantage of coping automatically with several objects in the scene.

Using the same ideas as in the section 2.3, we can define the following error measure:

$$C_4(\mathbf{S}, \mathbf{N}) = \int_S {}_4\Phi(\mathbf{S}, \mathbf{N}) d\sigma \tag{1.10}$$

where the integration is carried over with respect to the area element $d\sigma$ on the surface S and where ${}_4\Phi(\mathbf{S}, \mathbf{N})$ measures the error made when assuming that there is an object at point $\mathbf{S}$ with a normal $\mathbf{N}$. Here, $d\sigma$ plays the role of $dx\,dy$ in our previous analysis, $\mathbf{S}$ that of f, and $\mathbf{N}$ that of ∇f. Note that this is a significant departure from what we had before because we are multiplying our error criterion Φ with an area element on the surface. As we will see, this has two dramatic consequences, like in the geodesic snakes approach [6]:

1. it automatically regularizes the variational problem, and
2. it makes the problem intrinsic, i.e. independent from the parametrization of the surface, thus allowing a level set implementation.

3.1 Error criterion

While we still compute our error criterion Φ from the correlation as in the section 2.3, we will see that some changes are needed. First, we address the problems of regularization and symmetry that we have left aside so far.

Then, we deal with a more complex case, using more than two cameras and taking occlusions into account.

Given a point $\mathbf{S}$ on S and its normal $\mathbf{N}$, we could still let ${}_4\Phi$ be the opposite of $\rho_{12} = \frac{\langle I_1, I_2 \rangle}{|I_1| \cdot |I_2|}(\mathbf{S}, \mathbf{N}, m_1)$, the correlation between image I_1 and image I_2 taken at point m_1, image of $\mathbf{S}$ in I_1, defined as in section 2.3. Yet, it is now required that ${}_4\Phi$ stays positive: because we want to minimize its integral over S with respect to $d\sigma$, the surface would tend to maximize its area in the regions where ${}_4\Phi$ is negative, thus yielding an unstable evolution. ${}_4\Phi$ should also be small when correlation is high, otherwise the surface would tend to be as smallest as possible, regardless the correlation measure. Like in the geodesic snakes approach [6], we take ${}_4\Phi = g(\rho_{12})$, where $g : [-1, 1] \to \mathbf{R}^+$ is some positive decreasing function with $g(1) = 0$. We will see in section 3.3 that doing so solves the regularization problem.

The symmetry of the correlation could also be recovered, just taking ${}_4\Phi = g(\frac{1}{2}(\rho_{12} + \rho_{21}))$, the mean correlation $\frac{1}{2}(\rho_{12} + \rho_{21})$ being a symmetric value between -1 and $+1$. Actually, we could use the same idea to address the multi-camera problem. Given n images I_n $(1 \leq i \leq n)$, we could use $g(\frac{1}{n(n-1)} \sum_{i \neq j} \rho_{ij})$, but we go further and model *visibility and occlusion.* This is essential to avoid making mistakes by using incorrect information from cameras which do not see some parts of the objects (see figure 3). Given the surface S and a point $\mathbf{S}$ on it, it is possible to determine the cameras where $\mathbf{S}$ is visible, i.e. not hidden by another part of S, and to consider the correlation ratio for those cameras only. Let $\Gamma(\mathbf{S}, S)$ be the set of those cameras, we thus take:

$$ {}_4\Phi(\mathbf{S}, \mathbf{N}) = g\left(\frac{1}{|\Gamma|(|\Gamma| - 1)} \sum_{i,j \in \Gamma,\ i \neq j} \rho_{ij}\right) \tag{1.11} $$

This will be our symmetric, regularizing, multi-camera error criterion, taking *visibility* into account.

3.2 The Euler-Lagrange equations

In order to set up a surface evolution equation such as (1.9) and implement it by a level-set method, we have to write the Euler-Lagrange equations of the variational problem (1.10) and consider their component $-\beta$ along the normal to the surface. Although technically more complicated, this is similar to the derivations in the previous section. This is all fairly straightforward except for the announced result that the resulting value of β is *intrinsic* i.e. does not depend upon the parametrization of the surface S.

We have in fact proved a more general result. Let $\Phi : \mathbf{R}^3 \times \mathbf{R}^3 \longrightarrow \mathbf{R}$ be a smooth function of class at least C^2 defined on the surface S and depending upon the point $\mathbf{S}$ and the unit normal $\mathbf{N}$ at this point, which

we denote by $\Phi(\mathbf{S}, \mathbf{N})$. Let us now consider the following error measure:

$$C(\mathbf{S}, \mathbf{N}) = \int_S \Phi(\mathbf{S}, \mathbf{N}) d\sigma \tag{1.12}$$

We prove in [16] the following theorem:

Theorem 1 *Under the assumptions of smoothness that have been made for the function Φ and the surface S, the component of the Euler-Lagrange equations for criterion (1.12) along the normal to the surface is intrinsic, i.e. it does not depend upon the parametrization of S. Furthermore, this component is equal to*

$$\begin{aligned} -\beta = &(\Phi_{\mathbf{S}} + 2H\Phi_{\mathbf{N}})\mathbf{N} - 2H\Phi \\ &+ Trace((\Phi_{\mathbf{SN}})_{T_S} + d\mathbf{N} \circ (\Phi_{\mathbf{NN}})_{T_S}) \end{aligned} \tag{1.13}$$

where all quantities are evaluated at the point $\mathbf{S}$ of normal $\mathbf{N}$ of the surface, T_S is the tangent plane to the surface at the point $\mathbf{S}$. $d\mathbf{N}$ is the differential of the Gauss map of the surface, H is its mean curvature, $\Phi_{\mathbf{SN}}$ and $\Phi_{\mathbf{NN}}$ are the second order derivatives of Φ, $(\Phi_{\mathbf{SN}})_{T_S}$ and $(\Phi_{\mathbf{NN}})_{T_S}$ their restrictions to the tangent plane T_S of the surface at the point S.

Note that the error criterion (1.10) is of the form (1.12). According to the theorem 1, in order to compute the velocity β along the normal in the evolution equation (1.9), we only need to compute $\Phi_{\mathbf{S}}$, $\Phi_{\mathbf{N}}$, $\Phi_{\mathbf{SN}}$ and $\Phi_{\mathbf{NN}}$ as well as the second order intrinsic differential properties of the surface S. The problem is obviously broken down into the problem of computing the corresponding derivatives of the ρ_{ij}'s, which, for the first order derivatives is extremely similar to what we have done in the section 2.3. The computations are carried out in [16].

3.3 The normal velocity and the regularization property

The normal velocity β given by equation (1.13) looks a bit compicated. Actually, this is just an extension of the one obtained for 3D geodesic active contours by Caselles, Kimmel, Sapiro and Sbert [7] to the case where the error criterion Φ depends not only upon the point $\mathbf{S}$ but also upon the normal $\mathbf{N}$. Had we taken some error criterion $\Phi(\mathbf{S})$ with no dependency upon the normal (e.g. by restricting the model to the fronto parallel case), we would have obtained $\beta = 2H\Phi - \nabla\Phi.\mathbf{N}$ as in [7].

Our case is more complicated since our error criterion is a function $\Phi(\mathbf{S}, \mathbf{N})$ of the point and its normal but we see that our normal velocity is the sum of the following three terms:

$$\beta = \left\{ \begin{array}{lll} & 2H\Phi & \text{regularization term} \\ - & (\Phi_{\mathbf{S}} + 2H\Phi_{\mathbf{N}})\mathbf{N} & \text{first order data term} \\ - & Trace((\Phi_{\mathbf{SN}})_{T_S} + d\mathbf{N} \circ (\Phi_{\mathbf{NN}})_{T_S}) & \text{higher order data term} \end{array} \right. \tag{1.14}$$

The regularization term is the same as usual. The first order data term is an extension of the one in [6]. The higher order data term is certainly a bit tricky to understand. Nevertheless, our experiments seem to indicate that its influence is negligible.

We see that, as announced, the regularization problem has been solved in a natural way by using an intrinsic definition of the error criterion. We note the fact that Φ should be positive otherwise the curvature term induces instability, and that it should be small for high correlation values this preventing the surface to disappear, just like it does with the mean curvature flow [19].

3.4 Correctness

In the case of the active contours [6], the authors proved the existence of a unique viscosity solution to their evolution equation. In our case, it seems that existence and uniqueness of a viscosity solution is not immediate. While we are aware that proving such results is important, this is not the only research direction to investigate.

In effect, the problem of shape recovery from images using variational methods is still in its early days. As far as we know, the ideal energy is still to be found. This energy should certainly take into account stereo-correlation, but also contours [13, 10, 20, 5, 35], shape from shading [26], shape from texture [12, 4, 3, 29] and the ideas around stereo segmentation [45].

An interesting parallel can be drawn with the problem of contour extraction where a first significant progress was achieved with the proposal of the snake model [28] which was mostly heuristic. Formalisation and mathematical proofs of well-posedness came later, as a second stage, with the geodesic snakes model [6, 7]. In the case of stereo we are still struggling to reach the corresponding first stage.

4 Level Set Implementation

Let us now go into some very important implementation details. Our normal velocity β being intrinsic, we use the level set method. Among the difficulties we will have to overcome, the so-called *velocity extension problem* will be crucial. We give a solution to this problem that also eliminates another problem attached to the level set method, the need to reinitialize periodically the distance function.

4.1 The velocity extension problem

Let us denote by $u(\mathbf{X},t)$ the standard level set function, the zero level set of which is our moving surface S. The evolution equation (1.9) thus becomes an equation for u:

$$\frac{\partial u}{\partial t}(\mathbf{X},t) = \beta(\mathbf{X},t) \mid \nabla u(\mathbf{X},t) \mid \tag{1.15}$$

It is important to notice that β in (1.15) is defined in $\mathbf{R}^3$ whereas in (1.9) it is defined on the surface S. The extension of β from S to the whole domain $\mathbf{R}^3$ is a crucial point for the analysis and implementation of (1.15). There are mainly two ways of doing this.

(i) Most of the time this extension is natural. For example, if $\beta(\mathbf{S}) = H$, the mean curvature of S in (1.9), one can choose $\beta(\mathbf{X}) = H_u$, the mean curvature of the level set of u passing though $\mathbf{X}$ in (1.15). In the geodesic active contours approach [7], $\beta(\mathbf{S}) = 2H\Phi - \nabla\Phi.\mathbf{N}$, where $\Phi(\mathbf{X}) = g(|\nabla I(\mathbf{X})|)$ for a given image I and some real function g. Although one only wants to minimize the sum of Φ along S, just because Φ is defined in $\mathbf{R}^3$, $\beta(\mathbf{X}) = 2H_u\Phi - \nabla\Phi.\mathbf{N_u}$ is often taken for equation (1.15). Everything happens as if every level set of u was evolving according to the surface equation (1.9), i.e. was trying to reach a contour.

(ii) In some cases [9, 42], this extension is not possible. Then one may choose to assign to $\beta(\mathbf{X})$ in (1.15) the value of $\beta(\mathbf{S})$ in (1.9) where $\mathbf{S}$ is the closest point to $\mathbf{X}$ belonging to S. One may also choose to extend β so that it remains constant along the characteristics of u (the characteristics of u are the integral curves of ∇u). This is what is done in by Adalsteinsson and Sethian in [1], and by Peng et al. in [38], where a PDE based procedure extends β.

Note that, if u is the signed distance to S, then the characteristics of u are lines passing through the closest point of S, so that the two previous choices become one (see next section).

Our evolution velocity (1.14) only makes sense on S. It does not seem either reasonable or meaningful to consider some correlation measure or some visibility test for all the level sets of u: we are in case **(ii)**. Another point is that the computation of our β is expensive. In our case, computing β on S and extending it will be faster than computing it everywhere in the domain of u, even if this domain is only a narrow-band around S [2].

4.2 Preserving the distance function

In [21], we propose to use the following equation

$$\begin{aligned} &\frac{\partial u}{\partial t}(\mathbf{X},t) = \beta(\mathbf{X} - u\nabla u) \\ &u(\mathbf{X},0) = \text{signed distance function to } S(0) \end{aligned} \tag{1.16}$$

instead of the classical level set one (1.15). We show that u remains the signed distance function to its zero level, the evolution of which is still the desired one (1.9). Moreover, $\mathbf{X} - u\nabla u$ is the closest point to $\mathbf{X}$ on S, and our PDE is exactly (1.15) when β is extended via the closest point principle or, which is equivalent in this case, via the characteristics of u (Remember that $|\nabla u| = 1$ for the distance function). A detailed description of an implementation can be found in [21]. Also note that standard implementations of the level set method periodically reinitialize the function u to be a distance function. With our method, this step becomes unnecessary.

4.3 Error criterion

The estimation of β requires that of ${}_4\Phi$ and its derivatives. The function g in the definition (1.11) of ${}_4\Phi$ can be very simple. Our implementation uses $g(x) = 1 - x$.

Despite the use of a narrow-band and the fact that β is only computed on S and then extended, one may still find the computation of β too expensive. Depending upon the images, one may want to use only two cameras instead of all in the set Γ in (1.11). Good candidates are the cameras of Γ whose optical axes are the closest to the normal to the surface S at point $\mathbf{S}$. Experiments indicate that the higher order term for β in equation (1.14) can be ignored without any significant difference in the results.

4.4 Visibility

Estimating ${}_4\Phi$ requires the crucial step of computing the hidden parts of the surface for all the cameras. We first extract the zero level set of u as a triangulated mesh using the marching cube algorithm [36]. We then use a Z-buffer algorithm [8] to project this mesh onto each image with visibility information. This is not the most expensive part of our method. Purists will object that an evolution of u relying on the extraction of its zero level set does not agree with the philosophy of the level set method. It should probably be easy to adapt the work on visibility in an implicit framework by Tsai et al. [44].

4.5 Why and how well does it work?

We will see in section 5, that the results are promising. Yet the role of visibility in the reconstruction process is still an open question. Determining what are the local minima and to what extent they are avoided is also a hard problem. As a preliminary step toward answering these questions, we make a few remarks:

(i) Correlation based techniques are often fooled by false matches. In our case, we observed that the points in space inducing high correlation

scores were often isolated. As a result, thanks to the regularization term, the evolving surface does not get "stuck" at such points.

(**ii**) Local minima may be avoided using a multi-scale approach. The problem may be solved at a rough 3D scale first and its solution refined at finer scales. This technique could also be used to increase the convergence speed, although adaptive methods [43] seem an even better choice.

(**iii**) Depending on the images, the derivatives of the correlation function may not help the minimization process when the surface is too far from a minimum, i.e. may fail to predict the right direction of descent. In such cases, the regularization term, acting as a deflating term, helps to alleviate the problem. Thus one should choose an initial surface surrounding the objects to be recovered. In the active contours case, smoothing the images is often used to make the data term more efficient far from the contours. In our case, doing so would delete the texture information needed for the correlation. Other texture matching criteria are being investigated [11] to address this problem.

4.6 A simple but important case

Let us consider the case where the cameras are set up in such a way that every part of the objects is seen by at least two of them that are more or less close to each other. This happens when many cameras are used, or when the images are acquired with stereo rigs. If so we can, without much loss of information, use only those two cameras when estimating Φ, or if several choices are possible, the two cameras that are the most "in front" of the considered point $\mathbf{S}$. We can even rectify [15] the two corresponding images as a preliminary stage. Then, not only can the correlation be efficiently computed but it also does not depend anymore upon the normal to the surface, a square window in the first image approximately corresponding to a window of the same size in the second image[1]. As a result, such a situation yield fast correlation as well as a simplified normal velocity, similat to the active contours case: $\beta = 2H\Phi - \nabla\Phi.\mathbf{N}$

5 Results

We now present some results obtained with both synthetic and real images.

We first synthesized images of two tori, seen from enough points of views so that each part was seen at least twice (a total of 24 views – figure 4 left). See how the initial surface splits and how even the internal parts are

[1] The choice of the best two cameras depends upon the normal and even upon the whole surface through the visibility estimation. Yet, the Euler-Lagrange equations have been derived given such a choice.

FIGURE 4. Multi-camera images of 3D objects. Left: synthetic images of two tori (24 images). Right: real images of two heads (18 images).

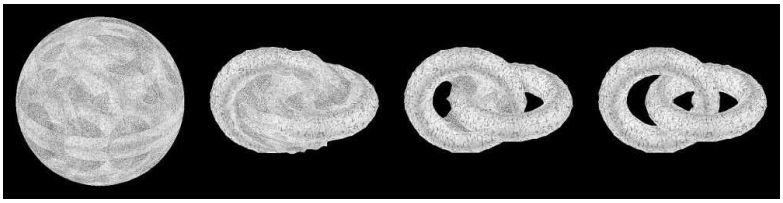

FIGURE 5. Four steps of the stereo algorithm for the two tori.

FIGURE 6. Four steps of the stereo algorithm for the two heads.

FIGURE 7. Some views of the reconstructed 3D object.

FIGURE 8. Eight of the twenty images used to reconstruct a human head despite bad lighting conditions and little texture information (courtesy of the RealViZ company).

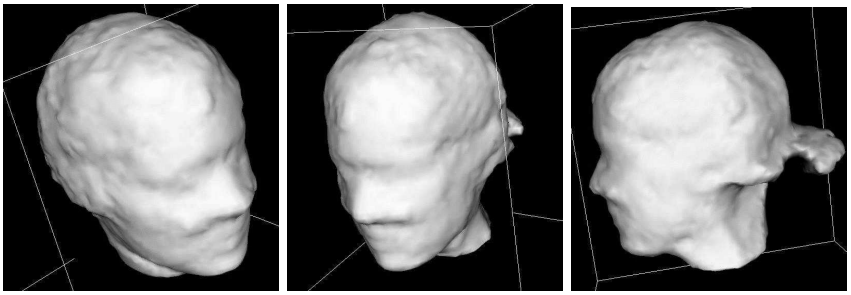

FIGURE 9. Three views views of the reconstructed shape. Note that the recovery of poorly textured areas like the hairs is made possible by an extension of the method presented in this chapter and described in [31].

reconstructed (figure 5).

We then used real images of two heads acquired by rotating them in front of the camera (figure 4 right). All visible parts (i.e.. neither the top nor the bottom) are correctly recovered (figures 6 and 7)

We finally applied the method to twenty images of a real human head (figure 8). As expected, the poorly textured regions such as the hairs were not reconstructed well. The results shown in figure 9, where these regions are nonetheless reconstructed, were obtained with a modified version of the functional to minimize, taking into account the contours information (see [31]).

6 Conclusion

We have presented in this chapter a novel approach for solving the stereo problem from an arbitrary number of views. It is based upon a variational principle that must be satisfied by the surfaces of the objects to be reconstructed. The design of the variational principle allows us to cleanly incorporate the hypotheses we make about the objects in the scene (smooth, opaque, lambertian), the methods used to rate the correspondences between image points (mostly cross-correlation). The Euler-Lagrange equations which are deduced from the variational principle provide a set of PDE's that are used to deform an initial set of surfaces which move toward the objects to be detected. The level set implementation of these PDE's provides an efficient and robust way of achieving the surface evolution, of dealing automatically with changes in the surface topology during the evolution and of taking into account visibility and occlusion. The whole objects (at least the parts seen by at least two cameras) are reconstructed without any constraints on the cameras positions unlike in methods such as space carving [34] or graph cuts [33].

7 References

[1] D. Adalsteinsson and J. Sethian. The fast construction of extension velocities in level set methods. Technical Report PAM-738, Center for Pure and Applied Mathematics. University of California at Berkeley, 1997.

[2] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *J. Comp. Phys.*, 118(2):269–277, May 1995.

[3] A. Blake and C. Marinos. Shape from texture: Estimation, isotropy and moments. *Artificial Intelligence*, 45(3):323–380, 1990.

[4] D. Blostein and N. Ahuja. Shape from texture: Integrating texture-element extraction and surface estimation. *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, PAMI-11(12):1233–1251, 1989.

[5] E. Boyer and M. Berger. 3D Surface Reconstruction Using Occluding Contours. *IJCV*, 22(3):219–233, 1997.

[6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *International Conference on Computer Vision*, pages 694–699, 1995.

[7] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. 3d active contours. In *International Conference on Analysis and Optimization of Systems*, pages 43–49, 1996.

[8] E. E. Catmull. *A Subdivision Algorithm for Computer Display of Curved Surfaces.* PhD thesis, Dept. of CS, U. of Utah, Dec. 1974.

[9] S. Chen, B. Merriman, S. Osher, and P. Smereka. A simple level set method for solving Stefan problems. *J. Comput. Phys.*, 135(8), 1995.

[10] R. Cipolla and A. Blake. Surface shape from deformation of apparent contours. *International Journal of Computer Vision*, 9:83–112, 1992.

[11] M. Clerc. Wavelet-based correlation for stereopsis. In *European Conference on Computer Vision*, 2002.

[12] M. Clerc and S. Mallat. Shape from texture through deformations. In *ICCV (1)*, pages 405–410, 1999.

[13] G. Cross and A. Zisserman. Surface reconstruction from multiple views using apparent contours and surface texture. In A. Leonardis, F. Solina, and R. Bajcsy, editors, *Confluence of Computer Vision and Computer Graphics*, pages 25–47. Kluwer, 2000.

[14] R. Deriche, S. Bouvin, and O. Faugeras. A level-set approach for stereo. In *Fisrt Annual Symposium on Enabling Technologies for Law Enforcement and Security - SPIE Conference 2942 : Investigative Image Processing.*, Boston, Massachusetts USA, November 1996.

[15] O. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint.* MIT Press, 1993.

[16] O. Faugeras and R. Keriven. Variational Principles, Surface Evolution, PDE's, Level Set Methods and the Stereo Problem. Technical Report 3021, INRIA, November 1996.

[17] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. In *European Conference on Computer Vision*, pages 379–393, 1998.

[18] O. Faugeras and R. Keriven. Variational principles, surface evolution, PDEs, level set meth ods, and the stereo problem. *IEEE Transactions on Image Processing. Special Issue on Geometry Driven Diffusion and PDEs in Image Processing*, 7(3):336–344, March 1998.

[19] M. Gage and R.S. Hamilton. The heat equation shrinking convex plane curves. *J. of Differential Geometry*, 23:69–96, 1986.

[20] P. Giblin and R. Weiss. Reconstruction of surfaces from profiles. Technical report, UM-CS-1987-026, 31, 1987.

[21] J. Gomes and O. Faugeras. Reconciling distance functions and level sets. *Journal of Visual Communication and Image Representation*, 11:209–223, 2000.

[22] M. Grayson. The heat equation shrinks embedded plane curves to round points. *J. of Differential Geometry*, 26:285–314, 1987.

[23] W.E.L. Grimson. *From Images to Surfaces.* MIT Press : Cambridge, 1981.

[24] H. L. F. von Helmholtz. *Treatise on Physiological Optics.* New York: Dover, 1925.

[25] B. K. P. Horn. *Robot Vision.* MIT Press, 1986.

[26] B. K. P. Horn and M. J. Brooks. *Shape from Shading.* The MIT Press, Cambridge, MA, 1989.

[27] B. Julesz. *Foundations of Cyclopean perception.* The University of Chicago Press, Chicago and London, 1971.

[28] M. Kass, A. Witkin, and D. Terzopoulos. SNAKES: Active contour models. *The International Journal of Computer Vision*, 1:321–332, January 1988.

[29] J. R. Kender and T. Kanade. Mapping image properties into shape constraints: Skewed symmetry and affine-tramsfornable patterns, and the shape-from-texture paradigm. In *National Conference on Artificial Intelligence*, pages 4–6, 1980.

[30] R. Keriven. *Equations aux Drives Partielles, Evolutions de Courbes et de Surfaces et Espaces d'Echelle: Applications la Vision par Ordinateur.* PhD thesis, Ecole Nationale des Ponts et Chausses, Dec. 1997.

[31] R. Keriven. A variational framework to shape from contours. Technical Report 2002-221, CERMICS, ENPC, 2002.

[32] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *International Conference on Computer Vision*, 1995.

[33] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proceedings of the 7th European Conference on Computer Vision*, volume 3, Copenhagen, Denmark, May 2002. Springer–Verlag.

[34] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *The International Journal of Computer Vision*, 38(3):199–218, July 2000.

[35] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(2):150–162, February 1994.

[36] W.E. Lorensen and H.E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. In *Proceedings of the SIGGRAPH '87 Conference*, volume 21, pages 163–170, Anaheim, CA, July 1987.

[37] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed : algorithms based on the Hamilton-Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[38] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155(2):410–438, 1999.

[39] L. Robert and R. Deriche. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. In *European Conference on Computer Vision*, 1996.

[40] L. Robert, R. Deriche, and O.D. Faugeras. Dense depth recovery from stereo images. In *European Conference on Artificial Intelligence*, pages 821–823, 1992.

[41] G. Sapiro and A. Tannenbaum. Affine Invariant Scale Space. *The International Journal of Computer Vision*, 11(1):25–44, August 1993.

[42] J. Strain and J. Sethian. Crystal growth and dendritic solidification. *Journal of Computational Physics*, 98:231–253, 1992.

[43] M. Sussman, A.S. Almgren, J.B. Bell, P. Colella, L. Howell, and M. Welcome. An adaptive level set approach for incompressible two-phase flow. *J. Comput. Phys.*, 148:81–124, 1999.

[44] R. Tsai, L. Cheng, P. Burchard, S. Osher, and G. Sapiro. Dynamic visibility in an implicit framework. Technical Report CAM TR 02-06, UCLA, 2002.

[45] A. Yezzi and S. Soatto. Stereoscopic segmentation. In *International Conference on Computer Vision*, 2001.