Complete Dense Stereovision Using Level Set Methods

Olivier Faugeras[1] and Renaud Keriven[2]

[1] INRIA, France and MIT AI-Lab, USA (Olivier.Faugeras@inria.fr)
[2] CERMICS-ENPC, France (Renaud.Keriven@cermics.enpc.fr)

Abstract. We present a novel geometric approach for solving the stereo problem for an arbitrary number of images (greater than or equal to 2). It is based upon the definition of a variational principle that must be satisfied by the surfaces of the objects in the scene and their images. The Euler-Lagrange equations which are deduced from the variational principle provide a set of PDE's which are used to deform an initial set of surfaces which then move towards the objects to be detected. The level set implementation of these PDE's potentially provides an efficient and robust way of achieving the surface evolution and to deal automatically with changes in the surface topology during the deformation, i.e. to deal with multiple objects. Results of an implementation of our theory also dealing with occlusion and vibility are presented 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 [19] and then reformulated by Caselles, Kimmel and Sapiro [1] and Kichenassamy et al. [21] 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 partial implementation. The problem of curve evolution driven by a PDE has been recently studied both from the theoretical standpoint [13,14,26] and from the viewpoint of implementation [23,28,29] with the development of level set methods that can efficiently and robustly solve those PDE's. A nice recent exposition of the level set methods and of many of their applications can be found in [27]. The problem of surface evolution has been less touched upon even though some preliminary results have been obtained [29,2]. 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 minimisation of a functional (we will explore several such functionals) 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

H. Burkhardt, B. Neumann (Eds.): Computer Vision – ECCV '98,Vol I, LNCS 1406, pp. 379–394, 1998.

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 [18,15,16,17,7]. To explain the problem of stereo from the computational standpoint, we will refer the reader to Fig. 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 [7]), the problem is essentially (but not only) one of establishing correspondences between the views: one talks about the matching 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 Fig. 1.a again). In order to go any further,

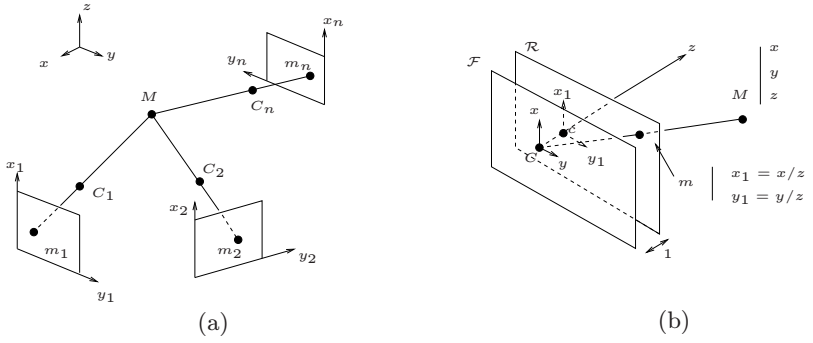

Fig. 1. (a) The multicamera stereo vision problem is, given a pixel m_1 in image 1, to find the corresponding pixel m_2 in image 2, ..., the corresponding pixel m_n in image n, i.e. the ones which are the images of the same 3D point M. Once such a correspondence has been established, the point M can 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 distance of 1 from it.

we need to be a little more specific about the process of image formation. We will assume here that the cameras perform a perpective projection of the 3D world on the retinal plane as shown in Fig. 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 Fig. 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. They are related in the sense that an increase along one axis usually implies a corresponding increase along the other. We start the paper with a short comparison of our work to previous work. In the next two sections we will 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). In Sect. 3 we consider an extremely simplified matching criterion which will allow us to convey to the reader the flavor of the ideas that we are trying to push here. We then move in Sect. 4 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. In the Section 5 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 as a set of general smooth surfaces in three space. The next step would of course be to relax the smoothness assumption but we will postpone this to a future paper.

Let us decide on some definitions and 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 as 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 this surface. Vectors and matrixes will generally be indicated in boldfaces, 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 will be indicated either using the ∂ symbol, e.g. $\frac{\partial f}{\partial \mathbf{x}}$, or as a lower index, e.g. $f_{\mathbf{x}}$.

2 Comparison with previous work

Our approach is an extension of previous work by Robert et al. and Robert and Deriche, [25,24], where the idea of using a variational approach for solving the stereo problem was proposed first 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 [5] 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.

Our work is also connected to that of Fua and Leclerc [12] and Fua [11] who have developed techniques and programs to integrate multiple stereo views. They use meshes and/or systems of particles to represent the surfaces of the objects and deform the mesh or move the particles to minimize a criterion that is not unlike the one we are using. The problems with this approach are well-known and described for example in [23,27]: vertices of the mesh or particles tend to cluster in areas of high curvature and the evolution may become unstable; moreover, the representation of complicated shapes with several connected components or nonzero genus as in the two tori of figure 4. The level set methods which we use to implement the evolution of the objects' surface was invented precisely because it solves elegantly those two problems [23,27]. Another main departure from Fua and Leclerc's approach is the use of a partial differential equation to drive this evolution. This puts our method on firmer mathematical grounds. We can potentially derive proofs of uniqueness of solutions in various functional spaces as well, prove convergence to the real scene as in [2] as well as benefit from the power of the level set method in our implementation.

There is also a connection to the work of Takeo Kanade and colleagues [22]. They build 3-D models of scenes from multiple cameras by merging the depth maps from different cameras into a common volumetric space. Just like in the case of the previous authors, we believe that their method suffers from the use of a mesh-like representation of the surface of the objects of the scene which makes merging difficult and unstable, the representation of objects with nonzero genus problematical and does not allow set the stage for proofs of correctness of the algorithm.

3 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 modelled 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 which minimizes the following integral:

$$C_1(f) = \int\int (I_1(m_1(x,y)) - I_2(m_2(x,y))^2 dxdy \tag{1}$$

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:

$$(I_1 - I_2)(\nabla I_1 \cdot \frac{\partial \mathbf{m}_1}{\partial f} - \nabla I_2 \cdot \frac{\partial \mathbf{m}_2}{\partial f}) = 0 \tag{2}$$

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 (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 (2), with some initial condition $f(x,y,0) = f_0(x,y)$. We thus see appear for the first time 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) 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), 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 [25,24]. We will postpone the solution of this problem until Sect. 5 and in fact solve it differently from the usual way which consists in adding a regularization term to $C_1(f)$.

Another strategy is to apply the level set idea [23,27]. Consider the family of surfaces S defined by $\mathbf{S}(x,y,t) = [x,y,f(x,y,t)]^T$. The parameters x and y are used to parameterize the surface, t is the time. The unit normal to this surface is the vector $\mathbf{N} = \pm\frac{1}{\sqrt{1+|\nabla f|^2}}[\nabla f^T, 1]^T$, the velocity vector is $\mathbf{S}_t = [0,0,f_t]^T$ and hence the evolution of the surface can be written

$$\mathbf{S}_t = \frac{\varphi(f)}{\sqrt{1+\mid \nabla f \mid^2}}\mathbf{N} \tag{3}$$

This expression of the evolution of the surface directly leads to a straightforward application of the level set methods. Consider a function $u(x, y, z, t)$ whose zero level set is the surface S, i.e. at each time instant t, the set of points (x, y, z) such that $u(x, y, z, t) = 0$ is identical to the surface S. Note that the function u can be considered a temporal sequence of volumetric images. The next question is, given the fact that the time evolution of S is given by (3), what should the evolution of u be? This question has been answered in [23] and the answer is:

$$u_t = \frac{\varphi(f)}{\sqrt{1+ \mid \nabla f \mid^2}} \mid \nabla u \mid$$

where ∇u is the gradient of u with respect to the first three variables. There are a couple of subtle points here. The first is that the level set methods have been designed for closed manifolds (curves or surfaces, say) but here the surface S is not closed in general, being a graph. This problem can be solved, as described for example in [3,27]. The second point is that the coefficient of the term $\mid \nabla u \mid$ in the previous equation is defined only on the surface S and not in the whole (x, y, z) volume. But this term is needed at all points to solve for u.

We will not delve further into the last issue because it will be solved as we proceed toward better models.

4 A better functional for matching

It is clear that the error measure (1) is a bit simple for practical applications. We can extend 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. In the first case we move along the matching criterion axis, in the second we move both along the shape and matching criterion complexity axes.

We explore those two avenues in the next sections.

4.1 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 on M:

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

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

$$\overline{I_k}(m_k) = \frac{1}{4pq}\int_{-p}^{+p}\int_{-q}^{+q} I_k(m_k + m')dm' \quad k = 1, 2 \tag{5}$$

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:

$$C_2(f) = -\int\int \frac{\langle I_1, I_2\rangle}{\mid I_1 \mid \cdot \mid I_2 \mid} dxdy = \int\int {}_2\Phi(f, x, y)dxdy \tag{6}$$

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 [8]. We can then proceed to solve the Euler-Lagrange equation as described in the previous section. But we will not pursue this task and explore rather a better functional.

4.2 Taking into account the tangent plane to the object

We now take into account the fact that the rectangular window centered at m_2 is not rectangular but is the image in the second retina of the backprojection 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 Fig. 2.a). In esssence, we approximate the object S in a neighbourhood 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:

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

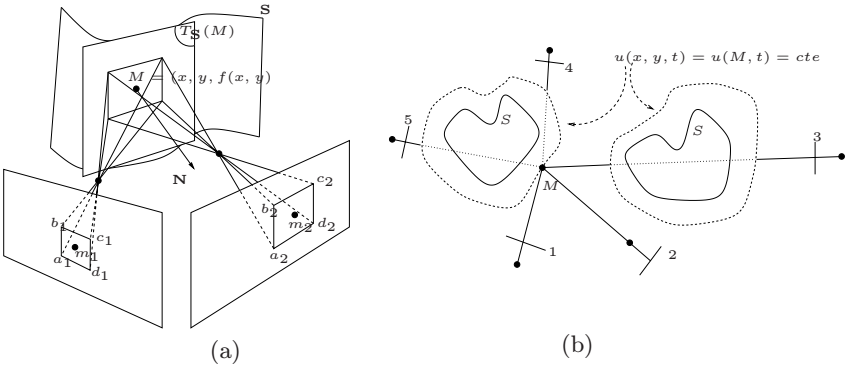

Fig. 2. (a) The square window (a_1, b_1, c_1, d_1) in the first image is back projected onto the tangent plane to the object S at point M and reprojected in the retinal plane of the second camera where it is generally not square. The observation is that the distortion between (a_1, b_1, c_1, d_1) and (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. (b) Occlusion is taken into account: only the cameras viewing the point according to the current surface are used,thus avoiding any irrelevant correlation.

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 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:

$$\mathbf{K} = d\mathbf{R}^T - \mathbf{R}^T\mathbf{t}\mathbf{N}^T \tag{7}$$

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 [5] 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 (4) must be modified as follows:

$$\langle I_1, I_2\rangle(f, \nabla f, x, y) = \int^* (I_1(m_1+m)-\overline{I_1}(m_1))(I_2(K(m_1+m))-\overline{I_2}(m_2))dm, \tag{8}$$

Note that, the definition of $\langle I_1, I_2\rangle$ is no longer symmetric, because of K. In order to make it symmetric, we should define it as:

$$\begin{aligned}\langle I_1, I_2\rangle(f, \nabla f, x, y) = &\int^* (I_1(m_1+m)-\overline{I_1}(m_1))(I_2(K(m_1+m))-\overline{I_2}(m_2))dm \\ &+ \int^* (I_1(K^{-1}(m_2+m'))-\overline{I_1}(m_1))(I_2(m_2+m')-\overline{I_2}(m_2))dm' \end{aligned} \tag{9}$$

The definition (5) of $\overline{I_1}$ (resp. of $\overline{I_2}$) is not modified in the first (resp. second) integral of the right hand side, that of $\overline{I_2}$ (resp. of $\overline{I_1}$), on the other hand, must be modified as follows:

$$\overline{I_2}(m_2) = \int^* I_2(K(m_1+p))dp, \quad \overline{I_1}(m_1) = \int^* I_1(K^{-1}(m_2+p'))dp' \tag{10}$$

Since this new definition does not modify the fundamental ideas exposed in this paper but makes the computations significantly more complex, we will assume the definition (8) in what follows, acknowledging the fact that in practice (9) should be used.

We now want to minimize the following error measure:

$$C_3(f, \nabla f) = -\int\int {}_3\Phi(f, \nabla f, x, y)dxdy, \quad {}_3\Phi = \frac{\langle I_1, I_2\rangle}{\mid I_1 \mid \cdot \mid I_2 \mid}(f, \nabla f, x, y) \tag{11}$$

Since the functional ${}_3\Phi$ now depends on 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 will 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 computation (see [8] for details) since we present in Sect. 5 a more elaborate model that encompasses this one and for which we will perform the corresponding computation.

5 An even more refined model

In this section we 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 the zero

level set of a function $\hat{u} : \mathbf{R}^3 \to \mathbf{R}$ which we assume to be smooth, i.e. C^2. The coordinates (x, y, z) of the points in the scene which are on the surface of the objects present are thus defined by the equation $\hat{u}(x, y, z) = 0$. This approach has at least two advantages. First, by relaxing the graph assumption, it potentially allows us to use an arbitrary number of cameras to analyze the scene and second, it leads very naturally to an implementation of a surface evolution scheme through the level set method as follows.

Let us consider a family of smooth surfaces $S : (v, w, t) \to \mathbf{S}(v, w, t)$ where (v, w) parameterize the surface and t is the time. It is in general not possible to find a single mapping S from $\mathbf{R}^2$ to $\mathbf{R}^3$ that describes the entire surface of the objects (think of the sphere for example where we need at least two) but we do not have to worry about this since our results will in fact be independent of the parametrization we choose. The objects in the scene correspond to a surface $\hat{\mathbf{S}}(v, w)$ and our goal is, starting from an initial surface $\mathbf{S}_0(v, w)$, to derive a partial differential equation

$$\mathbf{S}_t = \beta \mathbf{N}, \tag{12}$$

where $\mathbf{N}$ is the inner unit normal to the surface, which, when solved with initial conditions $\mathbf{S}(v, w, 0) = \mathbf{S}_0(v, w)$, will yield a solution that closely approximates $\hat{\mathbf{S}}(v, w)$. 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 the evolution equation (12) can be solved using the level set method which has the advantage of coping automatically with several objects in the scene. In detail, the surfaces $\mathbf{S}$ are at each time instant the zero level sets of a function $u : \mathbf{R}^4 \to \mathbf{R}$:

$$u(\mathbf{S}, t) = 0$$

Taking derivatives with respect to v, w, t, noticing that $\mathbf{N}$ can be chosen such that $\mathbf{N} = -\frac{\nabla u}{|\nabla u|}$, where ∇ is the gradient operator for the first three coordinates of u, one finds easily that the evolution equation for u is:

$$u_t = \beta \mid \nabla u \mid \tag{13}$$

Using the same ideas as in the section 4.2, we can define the following error measure:

$$C_4(\mathbf{S}, \mathbf{N}) = \int\int {}_4\Phi(\mathbf{S}, \mathbf{N}, v, w) d\sigma, \; {}_4\Phi = -\sum_{i,j=1, i\neq j}^{n} \frac{1}{\mid I_i \mid \cdot \mid I_j \mid} \langle I_i, I_j \rangle \tag{14}$$

In this equation, the indexes i and j range from 1 to n, the number of views. In practice it is often not necessary to consider all possible pairs but it does not change our analysis of the problem. In (14), the integration is carried over with respect to the area element $d\sigma$ on the surface S. With the previous notations, we have

$$d\sigma = \mid \mathbf{S}_v \times \mathbf{S}_w \mid dvdw = h(v, w) dvdw$$

$d\sigma$ plays the role of $dx\,dy$ in our previous analysis, $\mathbf{S}$ that of f, and $\mathbf{N} = \frac{\mathbf{S}_v \times \mathbf{S}_w}{|\mathbf{S}_v \times \mathbf{S}_w|}$, the unit normal vector to the surface S, that of ∇f.

Note that this is a significant departure from what we had before because we are multiplying our previous normalized cross-correlation score with the term $|\ \mathbf{S}_v \times \mathbf{S}_w\ |$. This has two dramatic consequences: (i) It automatically regularizes the variational problem like in the geodesic snakes approach [1], and (ii) it makes the problem intrinsic, i.e. independent of the parametrization of the objects in the scene.

Note also that each integral that appears in (14) is only computed for those points of the surface S which are visible in the two concerned images. Thus, *visibility and occlusion* are modelled in this approach (fig. 2.b). This is essential not to pretend the surface is at a wrong place when it actually is at the right place.

The rest of the derivation is extremely similar, although technically more complicated, to the derivations in the previous section, namely we write the Euler-Lagrange equations of the variational problem (14), consider their component β along the normal to the surface, set up a surface evolution equation (12) and implement it by a level-set method. This is all pretty straightforward except for the announced result that the resulting value of β is *intrinsic* and does not depend upon the parametrization of the surface S.

We will in fact prove a more general result. Let $\Phi : \mathbf{R}^3 \times \mathbf{R}^3 \longrightarrow \mathcal{S}_2$ be a smooth function of class at least C^2 defined on the product of the three-dimensional space $\mathbf{R}^3$ where the surface S "lives" and the two-dimensional unit-radius sphere $\mathcal{S}_2$ of $\mathbf{R}^3$ where the unit normal $\mathbf{N}$ to the surface S "lives". We note $\Phi(\mathbf{X}, \mathbf{N})$ the value of Φ at the point $\mathbf{X}$ of $\mathbf{R}^3$ and the point $\mathbf{N}$ of $\mathcal{S}_2$. Let us now consider the following error measure:

$$C(\mathbf{S}, \mathbf{S}_v, \mathbf{S}_w) = \int\int \Phi(\mathbf{S}(v,w),\, \mathbf{N}(v,w)) h(v,w) dv\, dw \tag{15}$$

where the integral is taken over the surface S.

We prove in [8] 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 (15) along the normal to the surface is the product of h with an intrinsic factor, i.e. which does not depend upon the parametrization (v, w). Furthermore, this component is equal to*

$$h(\Phi_{\mathbf{X}}\mathbf{N} - 2H(\Phi - \Phi_{\mathbf{N}}\mathbf{N}) + Trace((\Phi_{\mathbf{XN}})_{T_S} + d\mathbf{N} \circ (\Phi_{\mathbf{NN}})_{T_S})) \tag{16}$$

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, $\Phi_{\mathbf{XN}}$ and $\Phi_{\mathbf{NN}}$ are the second order derivatives of Φ, $(\Phi_{\mathbf{XN}})_{T_S}$ and $(\Phi_{\mathbf{NN}})_{T_S}$ their restrictions to the tangent plane T_S of the surface at the point S.

The symbol $\circ$ represents the composition of applications. Note that the error criterion (14) is of the form (15) if we define Φ to be

$$-\sum_{i,j=1,i\neq j}^{n}\frac{1}{\mid I_i \mid \cdot \mid I_j \mid}\langle I_i, I_j\rangle$$

According to the theorem 1, in order to compute the velocity β along the normal in the evolution equations (12) or (13) 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. Using the fact that the function Φ is a sum of functions $\Phi_{ij} = -\frac{1}{|I_i|\cdot|I_j|}\langle I_i, I_j\rangle$, the problem is broken down into the problem of computing the corresponding derivatives of the Φ_{ij}'s, which, for the first order derivatives is extremely similar to what we have done in the Sect. 4.2. The computations are carried out in [8]. In terms of the level set implementation, we ought to make a few remarks. The first is to explain how we compute β in (13) at each point (x, y, z) rather than on the surface S. It should be clear that we do not have any problem for computing $\mathbf{N} = -\frac{\nabla u}{|\nabla u|}$ and $2H = div(\frac{\nabla u}{|\nabla u|})$ and $d\mathbf{N}$ which is the differential of the Gauss map of the level set surface going through the point (x, y, z). The vectors $\Phi_{\mathbf{X}}$, $\Phi_{\mathbf{N}}$, the matrices $\Phi_{\mathbf{XN}}$, $\Phi_{\mathbf{NN}}$ are computed as explained in [8].

The second remark is that we can now write (13) as follows:

$$\begin{aligned} u_t = \mid \nabla u \mid div(\Phi\frac{\nabla u}{\mid \nabla u \mid}) - \Phi_{\mathbf{N}}(D\mathbf{N} + trace(D\mathbf{N})\mathbf{I}_3)\nabla u \\ - Trace((\Phi_{\mathbf{XN}})_{T_S} + d\mathbf{N} \circ (\Phi_{\mathbf{NN}})_{T_S}) \mid \nabla u \mid \end{aligned} \tag{17}$$

where $D\mathbf{N}$ is the 3×3 matrix of the derivatives of the normal with respect to the space coordinates, $\mathbf{I}_3$ the identity matrix, and at each point (x, y, z) the tangent plane T_S is that of the level set surface $u = constant$ going through that point. Note that $trace(D\mathbf{N}) = -div(\frac{\nabla u}{|\nabla u|})$. The first term $\mid \nabla u \mid div(\Phi\frac{\nabla u}{|\nabla u|})$ is identical to the one in the work of Caselles, Kimmel, Sapiro and Sbert [2] on the use of minimal surfaces or geodesic snakes to segment volumetric images. Our other terms come from the particular process that we are modelling, i.e. stereo. We believe and hope that we can prove in the near future that, under some reasonable assumptions, (17) is well-posed. As a first step in that direction, we report on some important implementation details: (i) Near the solution, Φ is close to -1 and the term $\mid \nabla u \mid div(\Phi\frac{\nabla u}{|\nabla u|})$ becomes anti-diffusive! As a consequence, we used $\Phi' = \Phi + 1$ instead of Φ,which takes values between 0 and +2, as a new error measure. This is equivalent to introducing in the criterion a term that tends to minimize the total area of the objects since it has the effect of adding to our original criterion the term $\int\int d\sigma$ which is precisely equal to that area. (ii) Regarding the image smoothness assumptions, a Gaussian modulated correlation could be used [10]. Actually, image intensities and their derivatives are extracted using recursively implemented Gaussian filters [4]. (iii) Concerning the problems of visibility and occlusion, the total error measure C_4 assumes the choice of certain camera pairs. Due to lack of place, we will not go into the

details and just say that our implementation handles this problem such that C_4 is at least continuous. For more details, see [20] and [9].

6 Results

We now present some results obtained from both synthetic and real images. The corresponding animated recovering processes as well as other results can be downloaded at: `http://cermics.enpc.fr/~keriven/stereo.html`. We first synthesized two crossing tori, shot from enough points of views so that each part was seen at least twice (actually 24 images – fig. 3). See how the surface splits after some iterations and how even the internal parts are recovered (fig. 4). We also used real images of a real objet (two human heads) that was rotated before the cameras (fig. 4). All viewed parts (ie. neither the top nor the bottom) are correctly recovered (fig. 5). In both cases the image textures are mapped on the surface by standard texture mapping techniques.

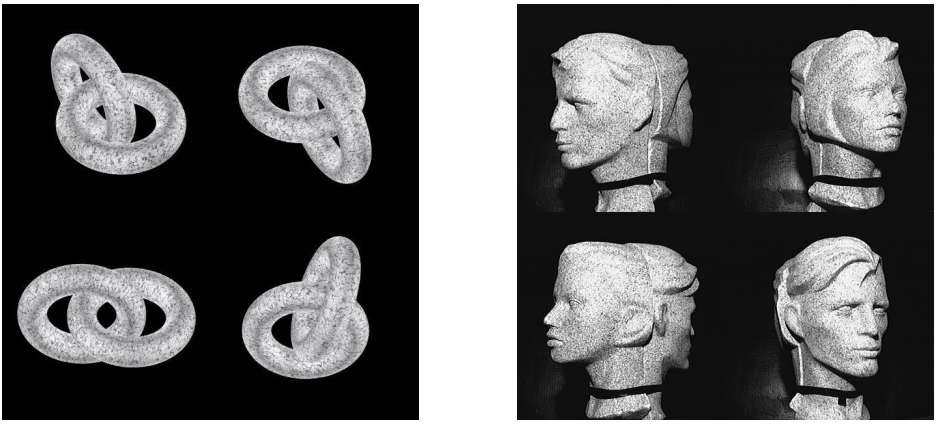

Fig. 3. Multicamera images of 3D objets. On the left hand side, two crossing synthetic tori (24 images). On the right hand side, real images: two human heads (18 images).

7 Conclusion

We have presented a novel geometric approach for solving the stereo problem from an arbitrary number of views. It is based upon writing a variational principle that must be satisfied by the surfaces of the objects to be detected. The design of the variational principle allows us to clearly incorporate the hypotheses we make about the objects in the scene and how we obtain correspondences

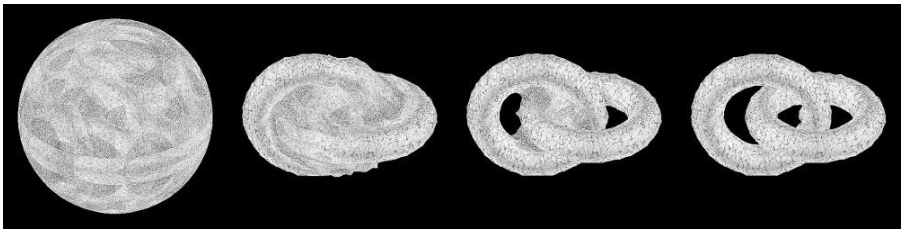

Fig. 4. Evolution of the surface for the two tori.

Fig. 5. Evolution of the surface for the two heads.

between image points. The Euler-Lagrange equations which are deduced from the variational principle provide a set of PDE's which are used to deform an initial set of surfaces which then move towards the objects to be detected. The level set implementation of these PDE's provides an efficient and robust way of achieving the surface evolution and to deal automatically with changes in the surface topology during the deformation. The whole objects (at least parts seen from two or more cameras) are recovered and visibility and occlusion are taken into account.

References

1. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proceedings of the International Conference on Computer Vision*, pages 694–699, 1995. 379, 389
2. V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. 3d active contours. In M-O. Berger, R. Deriche, I. Herlin, J. Jaffre, and J-M. Morel, editors, *Images, Wavelets and PDEs*, volume 219 of *Lecture Notes in Control and Information Sciences*, pages 43–49. Springer, June 1996. 379, 382, 390
3. David L. Chopp. Computing minimal surfaces via level set curvature flow. *Journal of Computational Physics*, 106:77–91, 1993. 384
4. R. Deriche. Recursively implementing the gaussian and its derivatives. Technical Report 1893, INRIA, Unité de Recherche Sophia-Antipolis, 1993. 390
5. Rachid Deriche, Stéphane Bouvin, and Olivier 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. 382, 386
6. M. P. DoCarmo. *Differential Geometry of Curves and Surfaces.* Prentice-Hall, 1976.
7. Olivier Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint.* MIT Press, 1993. 380, 380
8. Olivier Faugeras and Renaud Keriven. Variational Principles, Surface Evolution, PDE's, Level Set Methods and the Stereo Problem. Technical Report 3021, INRIA, November 1996. 385, 387, 389, 390, 390
9. Olivier Faugeras, Renaud Keriven, and Imahd Eddine Srairi. A first implementation of a complete dense surface reconstruction from 2D images. Technical report, ENPC-CERMICS, October 1997. 391
10. L.M.J. Florack. *The syntactical structure of scalar images.* PhD thesis, Utrecht University, 1993. 390
11. Pascal Fua. From multiple stereo views to multiple 3-d surfaces. *The International Journal of Computer Vision*, 24(1):19–35, August 1997. 382
12. Pascal Fua and Yves G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *The International Journal of Computer Vision*, 16(1):35–56, September 1995. 382
13. M. Gage and R.S. Hamilton. The heat equation shrinking convex plane curves. *J. of Differential Geometry*, 23:69–96, 1986. 379
14. M. Grayson. The heat equation shrinks embedded plane curves to round points. *J. of Differential Geometry*, 26:285–314, 1987. 379
15. W.E.L. Grimson. *From Images to Surfaces.* MIT Press : Cambridge, 1981. 380
16. H. L. F. von Helmholtz. *Treatise on Physiological Optics.* New York: Dover, 1925. Translated by J.P. Southall. 380
17. Berthold Klaus Paul Horn. *Robot Vision.* MIT Press, 1986. 380
18. Bela Julesz. *Foundations of Cyclopean perception.* The University of Chicago Press, Chicago and London, 1971. 380
19. M. Kass, A. Witkin, and D. Terzopoulos. SNAKES: Active contour models. *The International Journal of Computer Vision*, 1:321–332, January 1988. 379
20. Renaud 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. 391
21. S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. In *Proc. Fifth International Conference on Computer Vision*, Boston, MA, June 1995. IEEE Computer Society Press. 379
22. P.J. Narayanan, P.W. Rander, and Takeo Kanade. Constructing virtual worlds using dense stereo. In *Proceedings of the 6th International Conference on Computer Vision*, Bombay, India, January 1998. IEEE Computer Society Press. 382
23. 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. 379, 382, 382, 383, 384
24. L. Robert and R. Deriche. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. In Bernard Buxton, editor, *Proceedings of the 4th European Conference on Computer Vision*, Cambridge, UK, April 1996. 382, 383
25. L. Robert, R. Deriche, and O.D. Faugeras. Dense depth recovery from stereo images. In *Proceedings of the European Conference on Artificial Intelligence*, pages 821–823, Vienna, Austria, August 1992. 382, 383

26. Guillermo Sapiro and Allen Tannenbaum. Affine Invariant Scale Space. *The International Journal of Computer Vision*, 11(1):25–44, August 1993. 379
27. J. A. Sethian. *Level Set Methods*. Cambridge University Press, 1996. 379, 382, 382, 383, 384
28. J.A. Sethian. Numerical algorithms for propagating interfaces: Hamilton-jacobi equations and conservation laws. *Journal of Differential Geometry*, 31:131–161, 1990. 379
29. J.A. Sethian. Theory, algorithms, and applications of level set methods for propagating interfaces. Technical Report PAM-651, Center for Pure and Applied Mathematics, University of California, Berkeley, August 1995. To appear Acta Numerica. 379, 379