A variational framework for shape from contours

—

Research Report CERMICS-2002-221b

Renaud Keriven

Ecole Nationale des Ponts et Chaussees, CERMICS, France
keriven@cermics.enpc.fr

Abstract. We describe a new way to reconstruct the surface of an object from multiple views. We use the level sets method to get a surface such that its apparent contours are compatible with the images. We enforce, as much as possible, the apparent contours to be projected in each image in areas where the magnitude of the gradient of the grey level is high. Our approach is new for two reasons: we do not extract any contour in the images and we deal with the problem of occlusion. As a result, we do not obtain the classical "visual hull" [24] but a minimal surface which is closer to the actual shape of the object. Furthermore, our work can complement the approach of Faugeras and Keriven [14], thus getting a robust way to recover shape both from contours and from stereo-correlation.

Keywords: stereoscopic vision, shape from contours, variational methods.

1 Introduction

1.1 Previous Work

During the past decades many methods have been proposed to reconstruct the 3D shape of objects from images. Among these, some are based on feature matching [2] where a sparse surface map is produced. Some methods use texture correlation [3, 17, 21] to get a dense reconstruction. Other methods [6, 11, 16, 22, 31, 32, 26] use the apparent contours to compute the *visual hull* [24]. Space-carving was also proposed in [23, 29] where *photo-consistency* is used to carve voxels.

More recently, some methods based on a variational framework were proposed, using texture [14], texture and contours [13] or segmentation [34],

1.2 Motivation

In [13], Cross and Zisserman recently suggested to complement the variational approach of Faugeras and Keriven [14] with the ideas of shape from contours. Such a work is motivated by the complementarity of both following criterions: where the objects are textured, edges are somehow difficult to extract but similarity between images is meaningful; when there is no texture, similarity is useless but edges are correctly extracted and the apparent contours help to reconstruct the 3D shape of objects.

Our goal here, is to revisit this idea in order to make it compatible with what we think is the approach of [14]: using level sets and thinking in terms of "3D to 2D" rather than "2D to 3D".

2 Philosophy

Let us consider a bit more closely the work of [14], because we will see further that our work proceeds from the same philosophy. Their main idea is to take advantage of the variational framework to consider the standard problem of dense stereovision in the other way. For instance, in the case of two images, one generally tries to find, for each pixel (or feature) of the first image, the corresponding pixel in the second image as the most similar one (usually in term of normalized cross-correlation) satisfying the geometry of the cameras i.e. the epipolar constraint. A regularity constraint is also often added to the 3D surface reconstructed this way. This could be called a "2D to 3D" approach. In [14], the authors try to get a 3D surface, the projections of which in the images satisfy the similarity constraint. Therefore, it is the surface itself which is submitted to some optimization process such that it fits the chosen constraint in the best possible way. The advantages of this "3D to 2D" approach are:

- dealing naturally with more than two cameras.
- dealing with occlusions. At each step of the optimization process, an estimation of the surface is available. Thus, considering a point of the surface, one is aware of which cameras cannot see it because of some other object or

part of the same object. As a result, one can avoid the mistake of measuring between its projections in images where it his hidden by another point.

The classical shape from contours approaches take the somehow extracted 2D contours as a starting point and try to combine them to get the 3D shape. Even in [13] where a surface evolves toward the 3D shape, this surface tries to minimize its distance to the 3D cones generated by the each of the 2D contours.

In this paper, we try to mimic the approach of [14] consisting of verifying in the images themselves that the guessed surface fits the data. Doing so, we hope to get the same kind of advantages.

Our idea is the following: given different images of the same object, find a surface such that its apparent contours with respect to each of the images project in areas which can correspond to edges.

3 The geodesic active contours framework

The most appropriate framework for our purpose seems to be the one of geodesic active contours [9, 7, 8]. Given an image I and a parameterized closed curve $\mathcal{C}(q)$, the classical snake approach [19] consists in detecting a closed contour in I by minimizing the energy:

$$E(\mathcal{C}) = \alpha \int_0^1 |\mathcal{C}'|^2 dq + \beta \int_0^1 |\mathcal{C}''|^2 dq - \lambda \int_0^1 |\nabla I(\mathcal{C})| dq$$

In [7], the authors consider instead the minimization of only two terms, an internal (regularizing) energy and an external (linked to data) one :

$$E(\mathcal{C}) = \alpha \int_0^1 |\mathcal{C}'|^2 dq + \lambda \int_0^1 g(|\nabla I(\mathcal{C})|)^2 dq$$

where $g : [0, \infty[\rightarrow \mathbf{R}^+$ is a strictly decreasing function such that $g(r) \rightarrow 0$ as $r \rightarrow \infty$. Usually, the term $g(|\nabla I(\mathcal{C})|)$ is replaced by:

$$g = \frac{1}{1 + |\nabla \tilde{I}|^p}$$

where $\tilde{I}$ is a smoothed version of I and $p = 1$ or 2. Using the Maupertuis' Principle, this amounts to minimizing the following intrinsic energy:

$$E(\mathcal{C}) = \int_0^{L(\mathcal{C})} g(|\nabla I(\mathcal{C})|) ds \tag{1}$$

where $L(C)$ is the length of $\mathcal{C}$ and ds is the Euclidean arc-length. In order to minimize (1), they compute its Euler-Lagrange equation, resulting in a curve evolution for $\mathcal{C}$ (steepest descent):

$$\frac{\partial \mathcal{C}}{\partial t} = g(|\nabla I(\mathcal{C})|)\kappa \mathbf{n} - (\nabla g.\mathbf{n})\mathbf{n} \tag{2}$$

where κ is the curvature of $\mathcal{C}$ and $\mathbf{n}$ its unit normal. In order to detect non convex contours, they finally add a constant term c:

$$\frac{\partial \mathcal{C}}{\partial t} = g(c + \kappa)\mathbf{n} - (\nabla g.\mathbf{n})\mathbf{n} \tag{3}$$

In [8], the authors extend the so-called geodesic active contours to the 3D case, when I is a 3D image. $\mathcal{C}$ becomes a surface $\mathcal{S}$, the energy becomes the weighted area:

$$E(\mathcal{S}) = \int\int_{\mathcal{S}} g(|\nabla I(\mathcal{C})|)da$$

where da is the surface element of $\mathcal{S}$. The curve evolution (3) becomes a surface evolution:

$$\frac{\partial \mathcal{S}}{\partial t} = g(c + \mathbf{H})\mathbf{n} - (\nabla g.\mathbf{n})\mathbf{n} \tag{4}$$

where $\mathbf{H}$ is twice the mean curvature of $\mathcal{S}$.

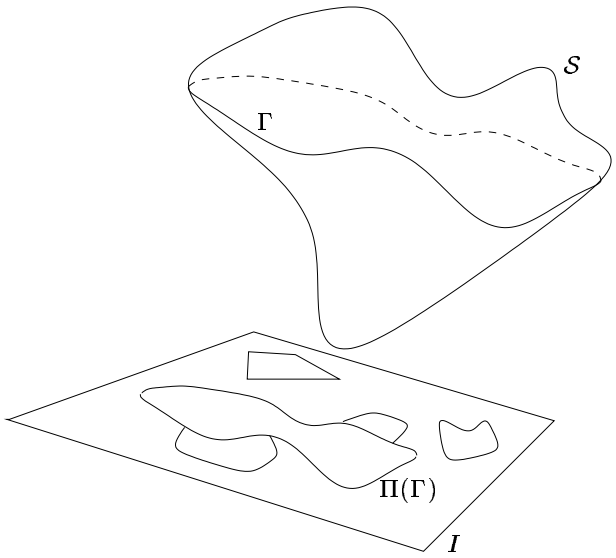

Fig. 1. The apparent contour Γ of the surface $\mathcal{S}$ should be projected in areas which can correspond to edges, i.e. where the magnitude of the gradient of the grey level of the image I is high: $g(|\nabla I|)$ should be small on $\Pi(\Gamma)$

4 Shape from contours

4.1 The main model: apparent contours

Let us come back to our problem. Given the surface $\mathcal{S}$ of an object, a camera $\mathbf{C}$ and the corresponding image I, let $\Gamma(\mathcal{S}, \mathbf{C})$ be the apparent contour of $\mathcal{S}$ viewed by $\mathbf{C}$, s the arc-length of Γ, $M(s)$ the associated 3D point and $\Pi(M)$ its projection into image I, our first idea is to minimize

$$\int_{\Gamma(\mathcal{S},\mathbf{C})} g(|\nabla I(\Pi(M(s)))|)ds$$

with respect to $\mathcal{S}$ (see figure (1)). To simplify notations, let $G(M) = g(|\nabla I(\Pi(M))|)$. Our energy becomes

$$\int_{\Gamma(\mathcal{S},\mathbf{C})} G(M(s))ds \tag{5}$$

Because this energy depends only upon the curve Γ, minimizing it with respect to a surface is an ill-posed problem. In order to deal with a smoothed version of Γ, let us introduce for any point M of $\mathcal{S}$ the indicator $h(M)$ of Γ ($h(M) = 1$ if $M \in \Gamma$, $h = 0$ otherwise), and $\tilde{h}$ a smoothed version of h ($\tilde{h} = 1$ on Γ, $\tilde{h} = 0$ outside a neighborhood of Γ and $\tilde{h}$ is smooth enough). We then replace energy (5) with

$$E(\mathcal{S}) = \int\int_{\mathcal{S}} \tilde{h}(M)G(M)da \tag{6}$$

Computing the Euler-Lagrange equation of energy (6) is not as straightforward as in the case of the geodesic surface. Actually, the point is that, because of $\tilde{h}$, E now depends not only upon M but also upon $\mathbf{n}$, the normal to $\mathcal{S}$ at M. Note that it was also the case in the work by Faugeras and Keriven. Computing the derivatives of $\tilde{h}$ with respect to $\mathbf{n}$ is possible. Yet, taking the derivatives of $\tilde{h}$ into account, even its derivative with respect to M, could lead to an undesired behavior of $\mathcal{S}$. If we do so, the points of Γ will tend not only to move toward regions of low G but also to deform locally the surface in order no to be on an apparent contour anymore (where $\tilde{h} = 0$!).

Considering this, instead of using the exact Euler-Lagrange equation of energy (6), we prefer to compute it at fixed $\tilde{h}$, thus yielding the following evolution for $\mathcal{S}$, where we add a constant term c as in equation (4):

$$\frac{\partial \mathcal{S}}{\partial t} = \tilde{h}[G(c + \mathbf{H}) - (\nabla G.\mathbf{n})]\mathbf{n} \tag{7}$$

Though they give the complete Euler-Lagrange of their energy, Faugeras and Keriven say they did not use all the terms of it in their actual implementation, arguing they would otherwise observe the same undesirable behavior: the surface would locally modify its normal to optimize the stereo-correlation, even when it is far from the solution.

4.2 Occlusion

Following [14], let us also take visibility into account. It is important that our method does not expect edges in the images were the is none because of occlusion.

Given the surface $\mathcal{S}$ and a camera $\mathbf{C}$, let $v(M)$ be a visibility indicator such that $v = 0$ when M is hidden from the camera by another part of the surface and $v = 1$ otherwise. Let $\tilde{v}$ be a smoothed version of v. We will now consider a corrected evolution for $\mathcal{S}$:

$$\frac{\partial \mathcal{S}}{\partial t} = \tilde{v}\tilde{h}[G(c + \mathbf{H}) - (\nabla G.\mathbf{n})]\mathbf{n} \tag{8}$$

Again, we do not claim the right hand side term of (8) to be the correct Euler-Lagrange equation of $\int\int_{\mathcal{S}} \tilde{v}(M)\tilde{h}(M)G(M)da$, but we justify our choice not only because the derivatives of $\tilde{v}\tilde{h}$ are intricate ($\tilde{v}$ depends on the whole surface!) but also because it is not a good idea to let the points of $\mathcal{S}$ directly act on the $\tilde{v}\tilde{h}$ term of the energy in order to minimize it.

4.3 Multiview

Given n views of the same object, taken by n cameras $\mathbf{C}_i$, it seems straightforward to consider the energy (with obvious notations):

$$\int\int_{\mathcal{S}} [\sum_i \tilde{v}_i(M)\tilde{h}_i(M)G_i(M)]da \tag{9}$$

and the evolution equation:

$$\frac{\partial \mathcal{S}}{\partial t} = [\sum_i \tilde{v}_i\tilde{h}_i[G_i(c + \mathbf{H}) - (\nabla G_i.\mathbf{n})]]\mathbf{n} \tag{10}$$

4.4 Coupling to stereo-correlation

Motivated by the complementarity of both approaches (see section 1.2), we could couple ours to the one of [14]. Because of the common points between the two methods (level sets, extension of velocities, visibility computation), this extension is quite natural. We do this simply adding the contribution of both methods with some scaling factor α. Let $\Phi(M, \mathbf{n})$ be the stereo-correlation of M corresponding to the best choice of two cameras where M is visible (see [14]). Following the actual implementation by Faugeras and Keriven, we keep only some part of the Euler-Lagrange of the energy $E_{ster}(\mathcal{S}) = -\int\int_{\mathcal{S}} \Phi(M)da$, thus yielding:

$$\frac{\partial \mathcal{S}}{\partial t} = [\sum_i \tilde{v}_i\tilde{h}_i[G_i(c + \mathbf{H}) - (\nabla G_i.\mathbf{n})] + \alpha(1 - \Phi)\mathbf{H} + \alpha\nabla\Phi.\mathbf{n}]\mathbf{n} \tag{11}$$

(see also [14] on why the term before $\mathbf{H}$ should be $(1 - \Phi)$ instead of $-\Phi$)

4.5 Correctness

In the case of the active contours [9, 7, 8], the authors proved the existence of a unique viscosity solution to their evolution equation. In our case, even though the equation (8) is far less complex than the one in [14], it seems that existence and uniqueness of a viscosity solution of (8) is not immediate. While we are aware that proving such results is essential, this is not the only direction to investigate. The problem of shape recovery from images using variational methods is still at its beginning. As far as we know, the ideal energy is still to be found. This energy should certainly take into account contours and stereo-correlation, but also the ideas of stereo segmentation [34] and the state of the art in shape from shading [18] and shape from texture [12, 5, 4, 20]. Solving the contour extraction problem with variational methods reached a first stage when the snakes model was proposed. Then, came geodesic active contours. Here, we are still trying to reach this first stage.

5 Implementation

Let us now go into some important implementation details.

5.1 The level sets method

Like the authors of [14] and of [9, 7, 8], we will use the now standard level sets method [27, 30]. This method is stable and deals with topological changes. The surface $\mathcal{S}$ is considered as the zero level set of a surface of $\mathbf{R}^4$ which is the graph of a function $u : \mathbf{R}^3 \rightarrow \mathbf{R}$, ie $\mathcal{S} = \{M \in \mathbf{R}^3 | u(M) = 0\}$. The evolution of $\mathcal{S}$ is obtained indirectly from the evolution of the embedding function u. If we want to implement the evolution equation $\mathcal{S}_t = \beta \mathbf{n}$, we simply let u depend upon time t and evolve according to $u_t = \beta |\nabla u|$. It can be shown [27] that the zero level set of $u(t, .)$ evolves according to the desired equation.

In our case, since $\mathbf{H} = div(\frac{\nabla u}{|\nabla u|})$ and $\mathbf{n} = \frac{\nabla u}{|\nabla u|}$, the evolution equation (8) gives for u:

$$\frac{\partial u}{\partial t} = \tilde{v}\tilde{h}[G(c + div(\frac{\nabla u}{|\nabla u|}))|\nabla u| + \nabla G.\nabla u] \tag{12}$$

Like in [10] and [14], our evolution velocity in equation (8) has only a meaning on $\mathcal{S}$ itself. Although G is defined in the whole domain of u, and although $\tilde{h}$ and $\tilde{v}$ could be given a meaning for each level set of u, we do not want all the level sets of u to evolve according to equation (8). Indeed, only its zero level set must project its apparent contours into the images in places where g is small. The other level sets should not fit the edges of the images! Therefore, like in [28], we use a PDE based procedure to extend the velocity from the surface to all the domain of u so that it remains constant along the normals to the surface. Note that, because we use the narrow-band approach [1] to make our scheme faster, this domain is just a small band around the zero level set. Thus, the extension step does not cost much time.

We proceed in the following way: given $\tilde{v}\tilde{h}G$ on the surface, we extend it to the domain of u as a function $\overline{\tilde{v}\tilde{h}G}$ such that $\nabla u.\nabla\overline{\tilde{v}\tilde{h}G} = 0$. In the same way, we extend $\tilde{v}\tilde{h}\nabla G$ as $\overline{\tilde{v}\tilde{h}\nabla G}$. Finally, the evolution equation for the embedding function u is:

$$\frac{\partial u}{\partial t} = [\overline{\tilde{v}\tilde{h}G}(c + div(\frac{\nabla u}{|\nabla u|}))|\nabla u| + \overline{\tilde{v}\tilde{h}\nabla G}.\nabla u] \tag{13}$$

rather than equation (12).

5.2 Computation of $\tilde{v}$

Like the authors of [14], we need to know, at each time step, which points of the surface are visible by each of the cameras. For such a purpose, we extract a triangulated approximation of the zero level set of u using the marching cubes algorithm [25], and fill a z-buffer for each camera in order to get v. It is quite straightforward to modify the z-buffer procedure to get a smoothed version $\tilde{v}$ of v. It is nowadays possible, though we did not try it, to use the hardware of the graphic card to do the job. Instead, we are currently investigating a way of getting $\tilde{v}$ directly from u without extracting the zero level set.

5.3 Computation of $\tilde{h}$

Given the optical center Ω of the camera and the normal $\mathbf{n}$ to the surface in M, it is easy to see that, if M is visible from the camera, then it is on the apparent contour Γ if and only if $\overrightarrow{\Omega M}.\mathbf{n} = 0$ ($\overrightarrow{\Omega M}.\mathbf{n}$ could also vanish outside Γ in points of non convex parts of $\mathcal{S}$, but such points are non visible). We thus have $h(M)v(M) = 1$ if $v(M) = 1$ and $\overrightarrow{\Omega M}.\mathbf{n} = 0$, and $h(M)v(M) = 0$ otherwise. For the smoothed version, this gives:

$$\tilde{h}\tilde{v} = f(|\frac{\overrightarrow{\Omega M}}{|\overrightarrow{\Omega M}|}.\mathbf{n}|)\tilde{v} \tag{14}$$

were f is a strictly decreasing function from 1 to 0 between 0 and ∞.

5.4 Computation of ∇G

Let us detail a straightforward computation just to show that ∇G is easily computed. Let $M = (x, y, z)^T$ and $m = (X, Y)^T = \Pi(M)$ be its projection into the image. We will follow the notation of [15]. Let $\tilde{M} = (x, y, z, 1)^T$ be the projective version of M, and $\tilde{P}$ be the 3×4 matrix associated with the camera. Let $\tilde{m} = (u, v, w)^T$ be the projective version of m. We have $\tilde{m} = \tilde{P}\tilde{M}$ and then $m = (u/w, v/w)^T$.

Let us denote:

$$\tilde{P} = \begin{pmatrix} \mathbf{r_1} & t_x \\ \mathbf{r_2} & t_y \\ \mathbf{r_3} & t_z \end{pmatrix}$$

where $\mathbf{r}_i$ are the three lines of the rotation associated with the camera and $(t_x, t_y, t_z)^T$ its translation. One can easily show that ∇G is linked to $\nabla g = (\frac{\partial g}{\partial X}, \frac{\partial g}{\partial Y})^T$ by:

$$\nabla G = \frac{\partial g}{\partial X} \frac{w\mathbf{r_1} - u\mathbf{r_3}}{w^2} + \frac{\partial g}{\partial Y} \frac{w\mathbf{r_2} - v\mathbf{r_3}}{w^2} \tag{15}$$

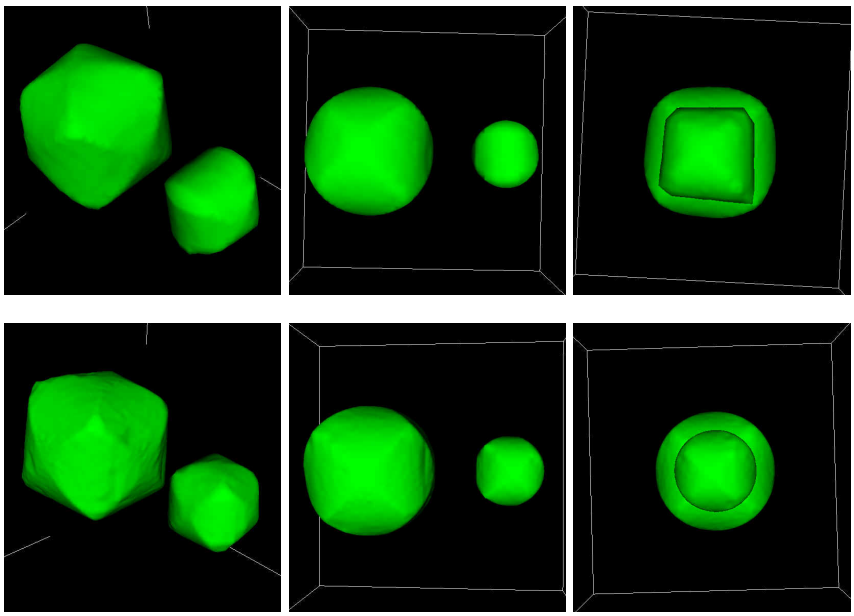

Fig. 2. *Our method versus the convex hull:* Top row: the right view, front view, left view and top view of two spheres. Middle row: the two spheres reconstructed with methods based on the visual hull. See how the small sphere is partially reconstructed (left image) and how it fits the front view (center image) but does not fit the right view (right image). Bottom row: the two spheres reconstructed with our method. See how the small sphere is reconstructed (left image) and how it fits the front view (center image) and the right view (right image)

6 Results and comparisons

We will now compare the results of our method to that of the other shape from contour methods. As expected, the advantages of our "3D to 2D" approach comes from avoiding the a-priori extraction of contours in the images.

6.1 Occlusions and inclusions

Because of occlusion, the apparent contours are not necessarily closed curves. Moreover, they can be included one into the other. This makes the extraction of a-priori contours the weak point of the other shape from contour methods.

For instance, let us consider the case of two objects (two spheres here), a big and a small one. Let us take four images of these objects: a front view, a right view, a left view and a top view (see figure (2) for a test with noisy synthetic images). In the right view, the apparent contour of the small object is inside the big object's apparent contour. In this view, an automatic contour extraction (eg using the geodesic active contours) only obtains the outer curve. Even if we extracted the inner contour too, how could we know a-priori whether it corresponds to a hole, to a texture on the big object, or to another object. As a result, classical "2D to 3D" approaches can only reconstruct an object based on the "visual hull" [24] obtained from the outer contours, thus yielding the objects shown in the middle row of figure (2) where the small object is not reconstructed with all the available information. With our approach, the projection of such an incomplete result into the right view does not fit its edges. Our method does not stop here and keeps evolving toward the result shown on the bottom row of figure (2): the small object is reconstructed in the best possible way. Thanks to occlusion detection, the small object does not generate any contour in the left view otherwise it would have tried to fit the edges of the big object or would have shrinked and would have disappeared.

6.2 Multiview coherence

An advantage of our approach is the ability to work even when the extraction of the contours in the 2D images are difficult, for instance when the images are noisy, or taken with bad lighting conditions or with a very contrasted or textured background. In these cases, it is difficult to get in the 2D images all the contours of the objects we want to reconstruct and only these contours. The 3D coherence inherent to our method can help to get rid of spurious contours or to keep those with low contrast. The idea here is that an image can come to help another one when things get bad. Figure (3) shows the reconstruction of a vase from eight noisy synthetic images taken all around it. The background is a layer of different rectangles so that it generates spurious contours in the images. Despite this, the shape is completely recovered. A close look a the second step of the evolution (the top right image) reveals edges (marked with arrows) caused by those spurious contours. Because they does not fit any contour in the other images, these edges quickly disappear.

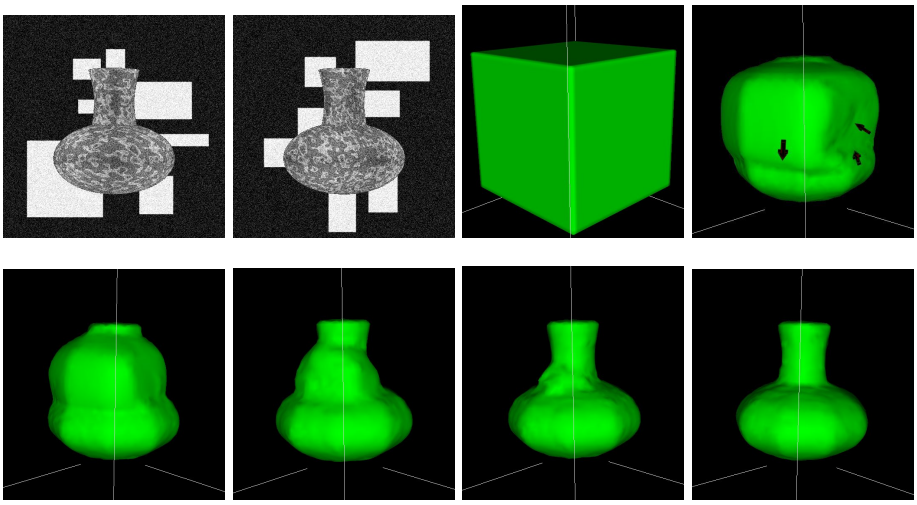

Fig. 3. *Multiview coherence:* Top left: two of the eight images of a vase observed spurious backgrounds. Other images: the evolution process. Despite the backgrounds, and their effects (see arrows in the top right image), the shape is completely recovered.

Fig. 4. *Contours and stereo-correlation (1):* Seven of the twenty images used to recover a human head despite bad lightning condition and few texture information (by courtesy of the RealViZ company). Bottom right: one of the four rough manual detouring used to generate the initial surface.

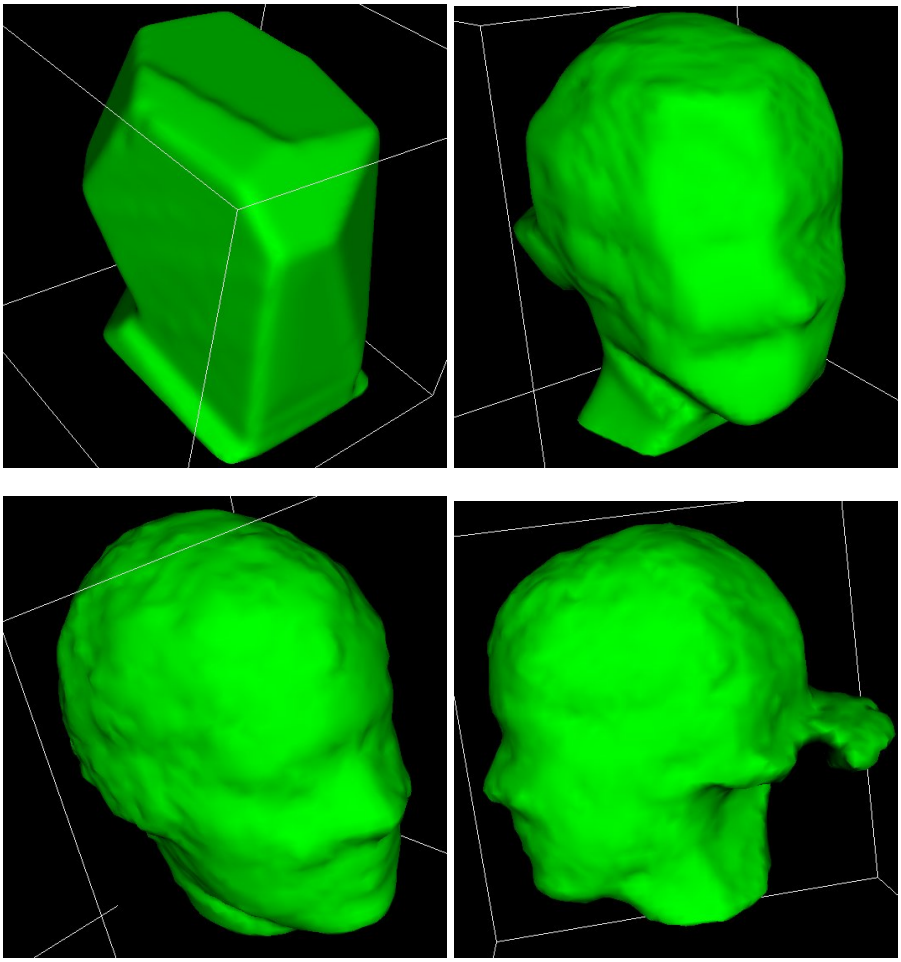

Fig. 5. *Contours and stereo-correlation (2):* Top left: the initial surface generated from rough manual detouring of four of the twenty images. Top right: an intermediate stage of the evolution. Bottom line: two views of the recovered shape.

6.3 Contours and stereo-correlation

Figures (4) and (5) show the ability of our method to deal with bad conditions when coupled to the stereo-correlation method [14]. Twenty images are used to recover a human head. Alone, our method does not work: some contours are not even visible in the images because of shades. Alone, the stereo-correlation method does not find enough texture (despite some black points painted on the face in order to accurately calibrate the cameras). When coupled, the two methods are able to use the contour and texture information where available and to recover the correct shape. Nevertheless, with a far initial condition, the evolution process was trapped into a local minimum. To prevent this, we had to use as a closer initial condition the convex hull generated from a very rough manual detouring of the head in four of the twenty images. Note that, even from this initial condition, none of the two methods works alone. Note also that 2D detouring with active contours is very hard in these images.

While the previous examples took less than five minutes on a standard 1GHz PentiumIII, this example took not less than one hour because of stereo but also because of the number of images.

7 Conclusion

We have presented a novel approach to shape from contours based on a "3D to 2D" paradigm. It is based on a variational framework and can be implemented with the level set method. Its advantages are: (i) to recover more completely the objects than with the visual hull, (ii) to deal with occlusions, (iii) to work even when 2D contours extraction is difficult and (iv) the ability to be coupled to the stereo-correlation criterion [14]. Our future work includes: investigating the possibility to prove existence and uniqueness of a solution to our variational problem or to a modified one; understanding better how to use 3D coherence to improve our method in bad conditions; coupling it to texture-based and shading based methods; improving speed; and finally extending it to deal with sequences of images in order to recover shape and motion simultaneously as in [33].

References

1. D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *J. Comp. Phys.*, 118(2):269–277, May 1995.
2. N. Ayache and O. Faugeras. Building a Consistent 3D Representation of a Mobile Robot Environment by Combining Multiple Stereo Views. In *IJCAI*, pages 808–810, 1987.
3. H. Baker and T. Binford. Depth from edge and intensity based stereo. In *IJICAI81*, pages 631–636, 1981.
4. Andrew Blake and Constantinos Marinos. Shape from texture: Estimation, isotropy and moments. *Artificial Intelligence*, 45(3):323–380, 1990.
5. Dorothea Blostein and Narendra 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.

6. E. Boyer and M. Berger. 3D Surface Reconstruction Using Occluding Contours. *IJCV*, 22(3):219–233, 1997.
7. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *ICCV95*, pages 694–699, 1995.
8. V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. 3D active contours. In *Images, Wavelets and PDEs*, volume 219 of *Lecture Notes in Control and Information Sciences*, pages 43–49. Springer, june 1996.
9. Vicent Caselles, Francine Catté, Tomeu Coll, and Françoise Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31, 1993.
10. S. Chen, B. Merriman, S. Osher, and P. Smereka. A simple level set method for solving Stefan problems. *J. Comput. Phys.*, 135(8), 1995.
11. R. Cipolla and A. Blake. Surface shape from deformation of apparent contours. *International Journal of Computer Vision*, 9:83–112, 1992.
12. Maureen Clerc and Stephane 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. Olivier Faugeras and Renaud 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.
15. Olivier D. Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.
16. Peter Giblin and Richard Weiss. Reconstruction of surfaces from profiles. Technical Report UM-CS-1987-026, 31, 1987.
17. E. L. Grimson. *From Images to Surfaces*. MIT Press, 1981.
18. Berthold K. P. Horn and Michael J. Brooks. *Shape from Shading*. The MIT Press, Cambridge, MA, 1989.
19. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: active contour models. *IJCV*, 1:321–331, 1988.
20. John R. Kender and Takeo 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.
21. R. Koch. 3D Surface Reconstruction from Stereoscopic Image Sequences. In *International Conference of Computer Vision*, 1995.
22. Kiriakos N. Kutulakos. Affine surface reconstruction by purposive viewpoint control. Technical Report TR581, 1996.
23. Kiriakos N. Kutulakos and Steven M. Seitz. A theory of shape by space carving. Technical Report TR692, 1998.
24. A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(2):150–162, February 1994.
25. 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.
26. W. Niem and J. Winbergmuhle. Automatic reconstruction of 3D objects using a mobile monoscopic camera. In *International Conference on Recent Advances in 3D Imaging and Modelling*, 1997.

27. S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
28. 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.
29. S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *Int. J. of Computer Vision*, 35(2):151–173, 1999.
30. J. A. Sethian. *Level Set Methods*. Cambridge University Press, 1998.
31. R. Szeliski and R. Weiss. Robust shape recovery from occluding contours using a linear smoother. *International Journal of Computer Vision*, 28(1):27–44, June 1998.
32. R. Vaillant and O. Faugeras. Using Extremal Boundaries for 3-D Object Modeling. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2):157–173, 1992.
33. S. Vedula, S. Baker, S. Seitz, and T. Kanade. Shape and motion carving in 6D. In *Proc. Computer Vision and Pattern Recognition Conf. (2)*, pages 592–598, 2000.
34. A. Yezzi and S. Soatto. Stereoscopic segmentation. In *Intl. Conf. on Computer Vision*, 2001.