Efficient initialization for constrained active surfaces, applications in 3D Medical images

Roberto Ardon[1,2] **and Laurent D. Cohen**[2]

[1] MEDISYS-Philips France,
51, rue Carnot, 92156 Suresnes, France
[2] CEREMADE-Universit Paris Dauphine,
Place du Marchal de Lattre, 75016 Paris, France
roberto.ardon@philips.com

Abstract. A novel method allowing simplified and efficient active surface initialization for 3D images segmentation is presented. Our method allows to initialize an active surface through simple objects like points and curves and ensures that the further evolution of the active object will not be trapped by unwanted local minima. Our approach is based on minimal paths that integrate the information coming from the user given curves and from the image volume. The minimal paths build a network representing a first approximation of the initialization surface. An interpolation method is then used to build a mesh or an implicit representation based on the information retrieved from the network of paths. From this initialization, an active surface converges quickly to the expected solution. Our paper describes a fast construction obtained by exploiting the Fast Marching algorithm. The algorithm has been successfully applied to synthetic images and 3D medical images.

1 Introduction

Since their introduction by Kass *et al.* in [7], deformable models have been extensively used to find single and multiple objects in 2D and 3D images. The common use of these models consists in introducing an initial object in the image and transforming it until it reaches a wanted target. In most applications, the evolution of the object is done in order to minimize an energy attached to the image data until a steady state is reached. One of the main drawbacks of this approach is that it suffers from local minima 'traps'. This happens when the steady state reached by the active object does not correspond with the target but with another local minimum. This is why, when considering the problem of a single object segmentation in real 3D images, the initialization of the active object is a major issue: if the initial position of the active object is too far from the target, local minima can block the evolution, hence missing the objective. Since the publication of [7], much work has been done in order to free the model from this initialization dependency. A balloon force was early proposed in [5] to cope with the shrinking problem of the first model, but this force supposed a known direction in the evolution. Geodesic active objects [2, 3] were then a major step toward the improvement of the energy term, but still did not guarantee the uniqueness of a minimum in the general case. The introduction of region dependent energies [11] and the use of the level set technique [10,9] that allows topological changes of the active model, contributed to create a more robust framework. Nonetheless, when dealing with 3D images and when looking for a precise object (like the left ventricle in 3D ultrasound images) the initialization of the model still is a fundamental step that is often made by simple geometric objects (spheres, cylinders) too far from the objective, or by tedious hand drawing.

The main contribution of our work is to provide an initialization surface that will allow the active object to avoid unwanted local minima and increase its convergence speed. This surface will be obtained through the interpolation of a network of 3D global minimal paths that will be generated between simple objects like points or curves introduced by the user and known to belong to the object of interest.

Our paper outline is as follows: we begin in section 2 by recalling the principles of geodesic active contours and surfaces as well as the global minimal paths framework. In section 3 we explain how minimal paths can be used to build a network of paths that is not concerned by the problem of local minima traps. In section 4 we present an improvement of the network construction where we project the minimal paths into suitable subspaces. In section 5 we give the final step of our algorithm which is the generation of the initialization surface from the network of paths. At last, in section 6 we show some examples on synthetic images and real medical images.

2 Active Surfaces and Minimal Paths

Active surfaces as well as minimal paths derived from deformable models introduced with the snake model [7]. This model consists in introducing a curve $\mathcal{C}$ into the image and make it evolve in order to minimize the energy,

$$E(\mathcal{C}) = \int \{\alpha. \|\mathcal{C}'(s)\| + \beta. \|\mathcal{C}''(s)\| \, ds\} + \int \mathcal{P}(\mathcal{C}(s))ds. \tag{1}$$

The two first terms maintain the regularity of the curve and the last one is the data attachment. $\mathcal{P}$ represents an edge detector that has lower values on edges, for example if I is the image, a choice can be $\mathcal{P} = (1 + |\nabla I|^2)^{-1}$.

Caselles *et al* improved the energy formulation in [2,3] by introducing the geodesic active contour model and its surface extension. In their approach the evolution of the initial curve $\mathcal{C}_0$ or surface $\mathcal{S}_0$ was driven by the minimization of the geodesic energy

$$E(\mathcal{C}) = \int \mathcal{P}(\mathcal{C}(s)) \, \|\mathcal{C}'(s)\| \, ds, \tag{2}$$

$$E(\mathcal{S}) = \int\int \mathcal{P}(\mathcal{S}(u,v)) \, \|\mathcal{S}_u \times \mathcal{S}_v\| \, dudv \tag{3}$$

Even though these models are only edge-driven most of the current approaches that integrates other informations (region, texture, a priori) are actually extensions. This is why in this paper we will concentrate on this models.

The most popular approach for solving the minimization problem is to consider the Euler-Lagrange equations for (2) and (3)(first variation of the energy) and derive from it the corresponding gradient descent schemes that will drive the model in the direction of steepest descent :

$$\frac{\partial \mathcal{C}}{\partial t} = (\mathcal{P}\kappa - \nabla\mathcal{P}.\boldsymbol{n})\,\boldsymbol{n} \text{ with } \mathcal{C}(.,0) = \mathcal{C}_0, \tag{4}$$

$$\frac{\partial \mathcal{S}}{\partial t} = (\mathcal{P}H - \nabla\mathcal{P}.\boldsymbol{N})\,\boldsymbol{N} \text{ with } \mathcal{S}(.,.,0) = \mathcal{S}_0, \tag{5}$$

where H and κ are respectively the mean curvature of the surface and the curvature of the curve. $\boldsymbol{N}$ and $\boldsymbol{n}$ are their outward normals.
However, in next section we recall a method introduced in [6] that allows to find the global minimum for the contour energy (2) when imposing the two end points and that does not use the evolution equation 4. Unfortunately such a method is not available for surfaces, we thus have to deal with the steepest gradient descent. The choice of $\mathcal{S}_0$ will then have a major impact on how well and how fast the model will recover the expected surface. In our approach to build a convenient $\mathcal{S}_0$ we made extensive use of the possibility to find the global minimum of the geodesic energy for curves.

2.1 Global minimal paths between two end points

It was established in [8] that the global minimal path connecting p_0 and p_1 is the curve obtained by following the opposite gradient direction on a map $\mathcal{U}_{p_0}$ starting from p_1 until p_0 is reached, thus solving the problem:

$$\frac{d\mathcal{C}}{ds}(s) = -\nabla\mathcal{U}_{p_0}, \text{with } \mathcal{C}(0) = p_1 \text{ and } \mathcal{C}(L) = p_0. \tag{6}$$

It was also shown in [8] that the map $\mathcal{U}_{p_0}$ is the solution of the Eikonal equation

$$\|\nabla\mathcal{U}_{p_0}\| = \mathcal{P} \text{ and } \mathcal{U}_{p_0}(p_0) = 0. \tag{7}$$

Equation 6 can be numerically solved by simple ordinary differential equations techniques like Newton's or Runge-Kutta's. Hence, the fundamental point is the computation of the minimal action map $\mathcal{U}_{p_0}$. To solve equation 7 numerically, classic finite differences schemes tend to be unstable. In [14] Tsitsiklis introduced a new method that was independently reformulated by Sethian in [12] that relies on a one-sided derivative looking in the up-wind direction of the front, and giving the correct viscosity solution. This algorithm is known as the Fast Marching algorithm and is now widely used and understood, details can be found in [13]. Nevertheless it is important to highlight the major interest of this algorithm. After a simple initialization of $\mathcal{U}_{p_0}$ over the grid domain, setting $\mathcal{U}_{p_0}(p_0) = 0$ and $\mathcal{U}_{p_0}(p) = \infty$ for any other point p, only one grid pass is needed, and by using min-heap data structure, an $O(N\log(N))$ complexity can be ensured on a grid of N nodes for the computation of $\mathcal{U}_{p_0}$.

To summarize, we are able, by imposing the two end points, to build a 3D global minimum path for the geodesic energy. On the other hand, an active surface's goal is to locate a local minimum of energy (3) that agrees with some user's criteria. The problem is that during the evolution process the surface can be trapped by other local minima.

In our work we have used the global minimum property of the paths to generate an initial surface $\mathcal{S}_0$ from points and/or curves drawn by the user.

3 First approximation of the initial surface $\mathcal{S}_0$

In order to determine a first approximation of $\mathcal{S}_0$ we propose to build a network of global minimal paths between a curve (henceforth noted $\mathcal{C}_1$) and a point (henceforth noted p) or other curves (henceforth noted $\mathcal{C}_2$, $\mathcal{C}_3$ and so on). Clearly, when building a minimal path network between $\mathcal{C}_1$ and p, one will compute only one action map (noted $\mathcal{U}_p$ as in previous section) centered in p and the network will be built by gradient descents from the points of a discretized version of $\mathcal{C}_1$. Which yields in this way a very numerically efficient algorithm. Figure 1.a and 1.b give an example of this construction. From a closed cylinder synthetic image, we generated minimal paths between a point in the upper part of the cylinder and a circle drawn on the opposite side.

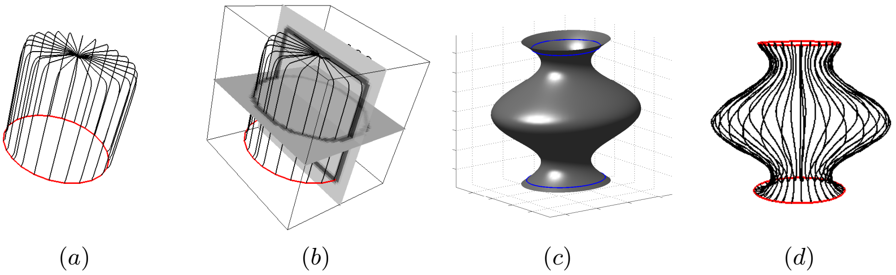

Fig. 1: (a) Set of Minimal Paths between a point and a curve lying on a cylinder. (b) The paths are minimal with respect to a potential that takes small values on the boundaries of the cylinder.(c) is the original vase surface, (d) is the set of Minimal Paths between two curves lying on a synthetic image of this vase. The paths are minimal with respect to a potential that takes small values on the vase's boundaries.

When it comes to building a set of minimal paths between two curves, an extension of this approach has to be considered. In the general situation we do not want to associate all the points of $\mathcal{C}_1$ with all the points of $\mathcal{C}_2$, first because this would be too computationally expensive (at least N actions maps to build and $N \times N$ gradient descents, if N is the number of points of the discretized versions of $\mathcal{C}_1$ and $\mathcal{C}_2$), and second because we are only interested in the most relevant associations. This is why we introduced in [1] a way to compute the optimal path between a curve and a single point. In [1] we showed that this problem can also be solved by applying the Fast Marching algorithm to solve the equation $\|\nabla \mathcal{U}_{\mathcal{C}_1}\| = \mathcal{P}$ initializing $\mathcal{U}_{\mathcal{C}_1}$ by $\mathcal{U}_{\mathcal{C}_1}(q) = 0$ if $q \in \mathcal{C}_1$ and $\mathcal{U}_{\mathcal{C}_1}(q) = \infty$ otherwise. And the minimal path is obtained by a gradient

descent initialized on p, just like in equation (6).
Consider now a discretized version of $\mathcal{C}_2$ containing n_2 points $\{q_i\}_{i=1\dots n_2}$. For each and every point q_i, we build the minimal path between this point and $\mathcal{C}_1$, thus generating a set of minimal paths from $\mathcal{C}_1$ to $\mathcal{C}_2$, $\left\{\mathcal{G}_i^{\mathcal{C}_1}\right\}_{i=1\dots n_2}$.
An illustration of this construction is given in figure 1.d, a potential adapted to finding the surface of a vase is used and the network is built between two curves drawn on it.
It is important to see that the construction of the network of paths is not symmetric. The set of paths $\left\{\mathcal{G}_i^{\mathcal{C}_1}\right\}_{i=1\dots n_2}$ is different from its homologue the set $\left\{\mathcal{G}_i^{\mathcal{C}_2}\right\}_{i=1\dots n_1}$.
One can use this feature to generate a more dense set of paths and thus contributing to the generation of an interpolated surface $\mathcal{S}_0$.
In practice, this scheme produces well-behaved paths which provide enough information for the construction of an interpolated surface $\mathcal{S}_0$ (Figures 1.a, 1.c and 3.a). Unfortunately in some particular situations it is not the case. In fact, even though the number of paths can be controlled by the number of discretization points on each curve, in some images, this will not improve the information one can extract from the network because minimal paths tend to merge (see Figure 2.b). This not only further complicates the problem of interpolation but if a surface is generated it is usually not the surface expected.
In order to cope with this problem we propose a simple method to constrain the paths between $\mathcal{C}_1$ and $\mathcal{C}_2$ and thus produce a better distributed network around the surface we are looking for.

4 Constraining the minimal path

Figure 2 illustrates a simple situation where the set of paths described in the previous section is not distributed in a satisfactory manner. The potential is minimal and constant on a surface which is the blending of a plane and half a sphere. Minimal paths will cut around the sphere rather than 'climbing' on it because the potential has no influence (being constant on the surface) and the length of the paths becomes the predominant factor.
We propose a simple but effective method to obtain a network of paths that will be better distributed. We shall constrain the network to different planes.
First introduced in [1] this approach now relies on a sounder theoretical ground which is presented in more details in the appendix. Here we show why this restriction provides suitable paths for our purpose.
Indeed, in the appendix to this article (section 8) we have derived a local necessary condition a curve $\mathcal{C}$ traced on a surface $\mathcal{S}$ must satisfy in order to be a local minimum of energy (2). $\mathcal{C}$ must satisfy

$$\nabla\mathcal{P}(\mathcal{C}).\boldsymbol{B} - \mathcal{P}(\mathcal{C})\kappa\big(\boldsymbol{n}.\boldsymbol{B}\big) = 0, \tag{8}$$

where $\boldsymbol{B} = \boldsymbol{T} \wedge \boldsymbol{N}$, $\boldsymbol{N}$ is the Gauss map of $\mathcal{S}$, $\boldsymbol{T}$ is the tangent vector of $\mathcal{C}$, and n is the normal vector to the curve. Now, if the constraining surface $\mathcal{S}$ is a plane, the normal $\boldsymbol{n}$ to the curve is contained in this same plane. Therefore $\boldsymbol{T} \wedge \boldsymbol{N} = \boldsymbol{n}$ since

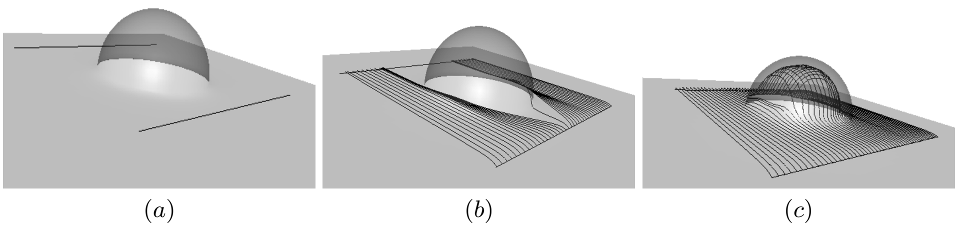

Fig. 2: (a) represents a half-sphere blended on a plane (transparent visualization) and $\mathcal{C}_1$ and $\mathcal{C}_2$ (black segments). b) Result without constraints:set of paths $\left\{\mathcal{G}_i^{\mathcal{C}_1}\right\}_{i=1\ldots n_2}$ missing the half-sphere. c) Result with constrains.

by definition, $\boldsymbol{n}$ is normal to $\boldsymbol{T}$ and $\boldsymbol{N}$ is normal to both. Equation (8) boils down to

$$\nabla\mathcal{P}(\mathcal{C}).\boldsymbol{n} - \mathcal{P}(\mathcal{C})\kappa = 0,$$

which is exactly the Euler-Lagrange equation of the geodesic energy (2) in two dimensions.

In practice we geometrically restrict the back-propagation procedure that builds the minimal paths (equation 6), considering the construction of $\left\{\mathcal{G}_i^{\mathcal{C}_1}\right\}_{i=1\ldots n_2}$, for a point p_i of the discretized version of $\in \mathcal{C}_2$ we choose a plane Π_{p_i} containing p_i and having $\vec{n_{p_i}}$ as its normal vector. We build path $\mathcal{G}_i^{\mathcal{C}_1}$ by solving the projected equation on Π_{p_i}:

$$\frac{d\mathcal{C}}{ds}(s) = -\nabla\mathcal{U}(\mathcal{C}) + \left(\nabla\mathcal{U}(\mathcal{C}).\,\vec{n_{p_i}}\right).\,\vec{n_{p_i}}\;. \tag{9}$$

Figure 2.c illustrates this construction, the network is obtained through the restriction of the paths to parallel planes which are orthogonal to $\mathcal{C}_1$ and $\mathcal{C}_2$ (n_{p_i} does not depend on p_i). In a more general case, for each point p_i, one can define plane Π_{p_i} by three points: G_1, center of mass of $\mathcal{C}_1$, G_2 center of mass of $\mathcal{C}_2$ and p_i, and thus

$$\vec{n_{p_i}} = \frac{\overrightarrow{G_1G_2} \wedge \overrightarrow{G_1p_i}}{\left\|\overrightarrow{G_1G_2} \wedge \overrightarrow{G_1p_i}\right\|}.$$

As the point p_i varies along $\mathcal{C}_2$, the plane Π_{p_i} will "rotate" around the principal axis G_1G_2.

5 Generating the initialization surface $\boldsymbol{\mathcal{S}_0}$

The final step for the generation of $\mathcal{S}_0$ is its effective construction through the interpolation of the information given by the network of paths. We have chosen two different approaches to generate $\mathcal{S}_0$.

The first one assumes that we have enough information with only one of the unconstrained sets $\left\{\mathcal{G}_i^{\mathcal{C}_j}\right\}_{i=1\ldots n_j, j=1,2}$, which means that the paths will cross only if they merge and that the network is simple enough to use an analytical interpolation. In [1] we proposed an analytical method inspired by splines, that integrates information coming from both the network of paths and the curves $\mathcal{C}_1$ and $\mathcal{C}_2$, and produces a smooth mesh. An example of this construction is given in Figure 3.b, we have interpolated a network of paths obtained from an ultrasound image of the left ventricle. This first method produces a mesh representation, and is thus adapted to the use of active meshes. Nonetheless, if needed one can produce a Level set representation using the Fast Marching method initialized with the mesh and with a constant potential of value 1.

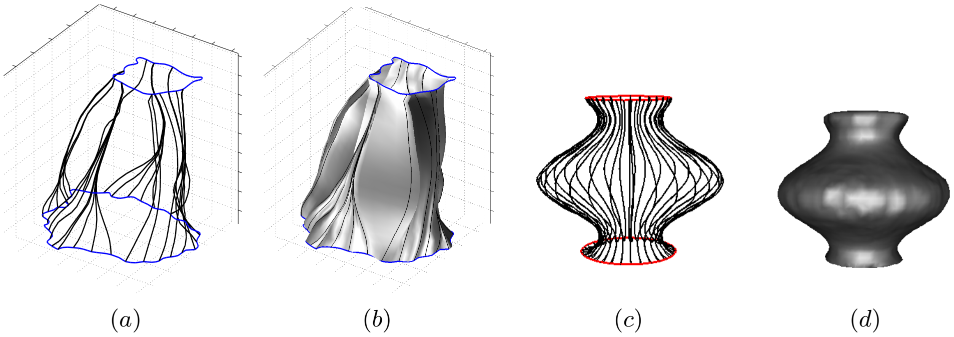

Fig. 3: (a) is the network of minimal paths obtained from an ultrasound image of the left ventricle. The user initialized the model by drawing the upper and lower closed curves. (b) is the analytically interpolated surface. (c) is the network obtained from our synthetic vase image, (d) it the variational scattered data points interpolation.

In spite of its great speed and good results, this method is only applicable under the conditions we stated. When considering the symmetrical network construction or when constraining the paths, the numerous crossings makes it very difficult to exploit the fact that we are interpolating curves and a scattered points approach is better suited.

The second interpolation method we have used is thus the one proposed by Zhao et al. in [15] where a variational approach is considered for the interpolation. Figure 3.d gives an example on the vase.

6 Application to synthetic and medical images

It is a common practice in 3D medical images to perform, as a first step, 2D segmentation on a slice. When images are of very low quality the practitioner does the

segmentation by hand. Our algorithm allows to rapidly build a good initialization and by so a 3D model. Figure 4 shows the result of the segmentation of a 3D ultra sound image of the left ventricle. As can be seen in Figure 4.b the restricted set of paths has already rebuilt perceptually the ventricle. Figure 4.c represents the interpolated surface with the variational method. We then applied a classical level set method evolution to refine the segmentation (Figure 4.d and 4.e).
In figure 5 we have given an other application of our method. We have considered the segmentation of a 3D magnetic resonance image representing an aneurysm.

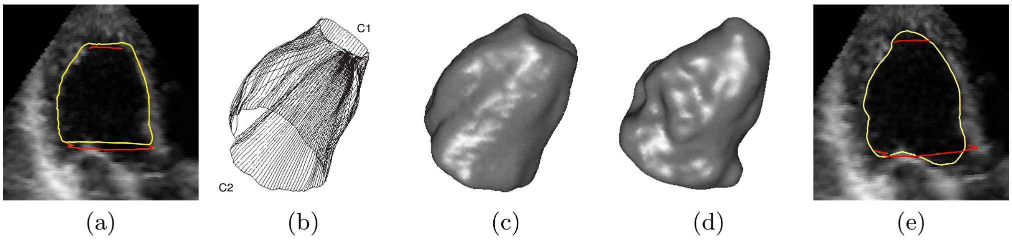

Fig. 4: (a) A slice of the 3D ultrasound image, we also have drawn the projection of the user given curves and the intersection of our interpolated surface with this plane. (b) Set of paths. (c) Interpolated surface. (d) final segmentation after a few iterations of the level set, (e) Planar view of the same slice, intersection with the model evolved as a level set.

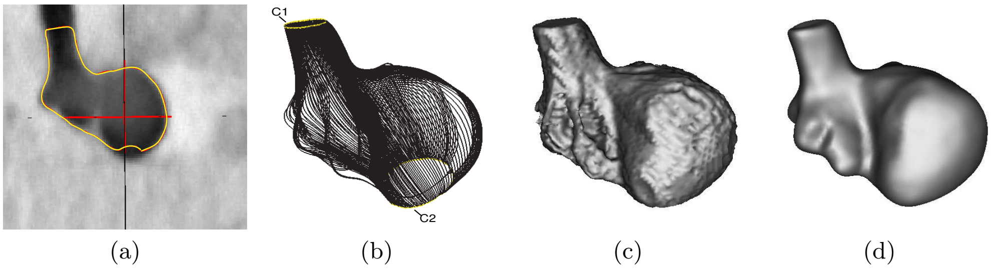

Fig. 5: (a) Intersection of our interpolated surface with a slice of a 3D MR image. (b) Set of paths. (c) Interpolated surface. (d) final segmentation after a few iterations of a level set.

In figure 6 we compare our method to a classical active surface. Figure 6.a is a difficult to segment image, it was generated from three 'S' shaped tubes placed one inside the other. The difficulty resides in the extraction of the middle one. Without a good initialization, a gradient descent will fail to recover the surface because of the presence of many local minima. Our method manages to generate a suitable initialization (figure 6.c) with the only information of two curves lying on it. An active surface, initialized with a cylinder containing the initialization curves (Figure 6.e and 6.f), will fail to recover the middle tube (Figure 6.g and 6.h).

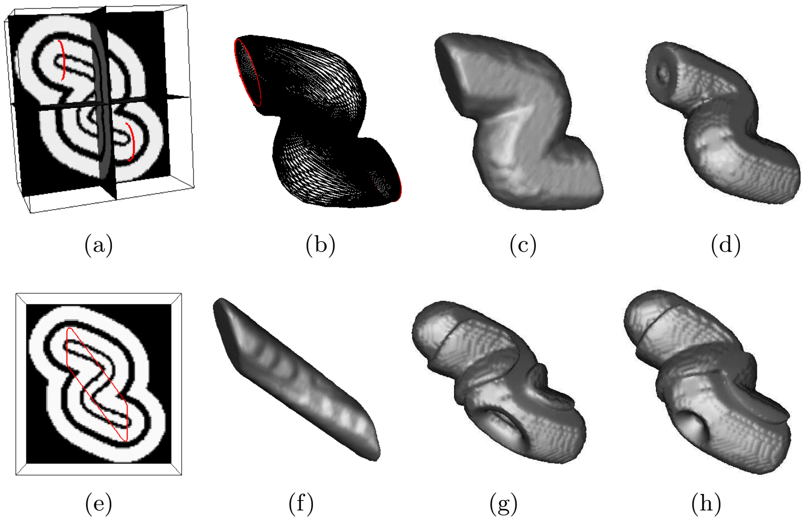

Fig. 6: (a) View of different intersecting planes of a 3D volume with the two constraining curves drawn on it. (b) Network of paths obtained with our method. (c) Interpolated surface. (d) Surface after few iterations of a level set. (e) and (f) Simple initialization of an active object. (g) surface after 150 iterations and (h) after 500 iterations.

7 Conclusion

In this paper we have presented a method that allows to greatly simplify the initialization process of active surfaces. The model can be initialized by simple objects like curves and points instead of volumes. Our approach is also capable of taking a maximum advantage of the information given by the user through the initialization curves, since the surface it generates is constrained to include those curves. Our method uses globally minimal paths to generate a surface which is the initialization of an active surface model. Hence, the final surface is not concerned by the problem of the local minima traps as all other active objects approach are. It is particularly well suited for medical image segmentation, in particular for ultrasound images segmentation. In cases where the image quality is very poor, our approach handles the introduction of additional information coming from the practitioner in a very natural manner. A few 2D segmentations can be enough to generate a coherent complete surface.

8 Appendix

In this appendix, inspired by the methods used in [4], we give a proof of the necessary condition a curve $\mathcal{C}$, traced on a surface $\mathcal{S}$, should satisfy in order to be a local minimum of the geodesic energy (3). We should note Ψ the signed distance function to $\mathcal{S}$, $\boldsymbol{T} = \frac{\mathcal{C}_t}{|\mathcal{C}_t|}$ the tangent vector along the curve, $\boldsymbol{N}(\alpha)$ the normal vector to $\mathcal{S}$ on α, $\boldsymbol{n}$ the normal vector to $\mathcal{C}$, and $\boldsymbol{B} = \boldsymbol{T} \wedge \boldsymbol{N}$(figure 7).

Let U_δ ($\delta \geq 0$) be a neighborhood of $\mathcal{S}$ defined by $U_\delta = \{\alpha | \Psi(\alpha) \leq \delta\}$. It can be shown that if δ is small enough, we have $\forall \alpha \in U_\delta \ \nabla\Psi(\alpha) = \boldsymbol{N}(\alpha_{\mathcal{S}})$, where $\alpha_{\mathcal{S}}$ is the closest point to α on $\mathcal{S}$. And, since Ψ is the distance function to $\mathcal{S}$, we have that, in that case $\alpha_{\mathcal{S}} = \alpha - \Psi(\alpha)\nabla\Psi(\alpha)$. We then define the the operator $\Pi(\alpha) = \alpha - \Psi(\alpha)\nabla\Psi(\alpha)$ on U_δ, with δ small enough. Let $\mathcal{C}_\lambda$ be a variation of $\mathcal{C}$, $\mathcal{C}_\lambda = \mathcal{C} + \lambda\eta$, where η is a function which has the suitable regularity properties and boundary conditions, $\lambda \in \mathbb{R}$. This variation of $\mathcal{C}$ is not correct in our case, since $\mathcal{C}_\lambda$ is not necessarily traced on $\mathcal{S}$. Hence we will consider $\Pi(\mathcal{C}_\lambda) = \mathcal{C}_\lambda + \Psi(\mathcal{C}_\lambda)\nabla\Psi(\mathcal{C}_\lambda)$ instead and will be interested in the value of the derivative of

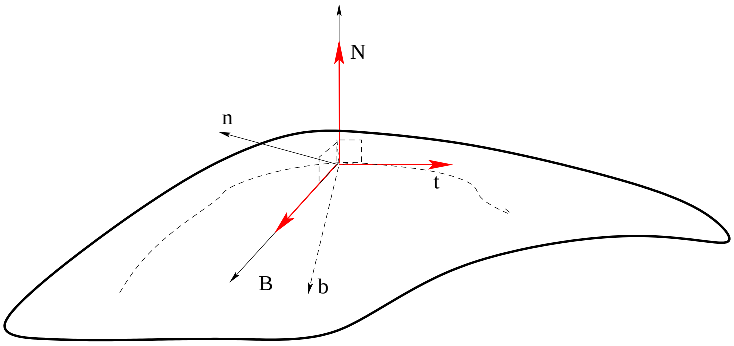

Fig. 7: Illustration showing the notation we use.

$$E(\lambda) = \int \mathcal{P}(\Pi(\mathcal{C}_\lambda)) \left|\Pi(\mathcal{C}_\lambda)_t\right| dt \text{ at } \lambda = 0, \tag{10}$$

$$\left.\frac{dE}{d\lambda}\right|_{\lambda=0} = \underbrace{\int \frac{d}{d\lambda}\{\mathcal{P}(\Pi(\mathcal{C}_\lambda))\} . |\Pi(\mathcal{C}_\lambda)_t| \, dt\Big|_{\lambda=0}}_{I_1} + \underbrace{\int \mathcal{P}(\Pi(\mathcal{C}_\lambda)). \frac{d}{d\lambda}\{|\Pi(\mathcal{C}_\lambda)_t|\} \, dt\Big|_{\lambda=0}}_{I_2}$$

If curve $\mathcal{C}$ is a minimum of E this derivative should be zero.

Note that $\mathcal{C}$ being traced on $\mathcal{S}$, $\Psi(\mathcal{C}) = 0$ and $\nabla\Psi(\mathcal{C}).\mathcal{C}_t = \boldsymbol{N}(\mathcal{C}).\boldsymbol{T}(\mathcal{C}) = 0$. Thus

$$\begin{aligned}\left.\frac{d}{d\lambda}\left\{\mathcal{P}(\Pi(\mathcal{C}_\lambda))\right\}\right|_{\lambda=0} &= \nabla\mathcal{P}(\mathcal{C}).\left(\eta - \nabla\Psi(\mathcal{C}).\eta\nabla\Psi(\mathcal{C}) - \underbrace{\Psi(\mathcal{C})}_{=0}(...)\right)\\ &= \big(\nabla\mathcal{P}(\mathcal{C}) - \big(\boldsymbol{N}(\mathcal{C}).\nabla\mathcal{P}(\mathcal{C})\big)\boldsymbol{N}(\mathcal{C})\big).\eta\end{aligned} \tag{11}$$

and

$$\Pi(\mathcal{C}_\lambda)_t|_{\lambda=0} = \mathcal{C}_t - \underbrace{(\nabla\Psi(\mathcal{C}).\mathcal{C}_t)}_{=0}\nabla\Psi(\mathcal{C}) - \underbrace{\Psi(\mathcal{C})}_{=0}(...) = \mathcal{C}_t \tag{12}$$

Using this relations we obtain:

$$I1 = \int \big(\nabla\mathcal{P}(\mathcal{C}) - \big(\nabla\Psi(\mathcal{C}).\nabla\mathcal{P}(\mathcal{C})\big)\nabla\Psi(\mathcal{C}).\eta\big)|\mathcal{C}_t|\,dt \tag{13}$$

Now concerning I_2, from (12) its second factor can be written as

$$\left.\frac{d}{d\lambda}\left\{|\Pi(\mathcal{C}_\lambda)_t|\right\}\right|_{\lambda=0} = \frac{\mathcal{C}_t}{|\mathcal{C}_t|}.\left.\frac{d}{d\lambda}\left\{\Pi(\mathcal{C}_\lambda)_t\right\}\right|_{\lambda=0} = \boldsymbol{T}.\left.\frac{d}{d\lambda}\left\{\Pi(\mathcal{C}_\lambda)_t\right\}\right|_{\lambda=0} \tag{14}$$

And

$$\begin{aligned}\left.\frac{d}{d\lambda}\left\{\Pi(\mathcal{C}_\lambda)_t\right\}\right|_{\lambda=0} &= \frac{d}{dt}\left\{\left.\frac{d}{d\lambda}\left\{\Pi(\mathcal{C}_\lambda)\right\}\right|_{\lambda=0}\right\} = \frac{d}{dt}\left\{\eta - \big(\nabla\Psi(\mathcal{C}).\eta\big)\nabla\Psi(\mathcal{C})\right\}\\ &= \eta_t - \frac{d}{dt}\left\{\big(\boldsymbol{N}(\mathcal{C}).\eta\big)\right\}\boldsymbol{N}(\mathcal{C}) - \big(\boldsymbol{N}(\mathcal{C}).\eta\big)H_\Psi(\mathcal{C})\mathcal{C}_t\end{aligned}$$

H_Ψ being the Hessian of Ψ. And going back to 14, noticing that $\boldsymbol{N}(\mathcal{C}).\boldsymbol{T} = 0$,

$$\boldsymbol{T}.\left.\frac{d}{d\lambda}\left\{\Pi(\mathcal{C}_\lambda)_t\right\}\right|_{\lambda=0} = \boldsymbol{T}.\eta_t - \big(\boldsymbol{N}(\mathcal{C}).\eta\big)\big(\boldsymbol{T}.H_\Psi(\mathcal{C})\mathcal{C}_t\big)$$

following that

$$\begin{aligned}I_2 &= \int \mathcal{P}(\mathcal{C})\big(\boldsymbol{T}.\eta_t\big)dt - \int \mathcal{P}(\mathcal{C})\big(\nabla\Psi(\mathcal{C}).\eta\big)\big(\boldsymbol{T}.H_\Psi(\mathcal{C})\boldsymbol{T}\big)|\mathcal{C}_t|\,dt\\ &= -\int \big(\,(\nabla\mathcal{P}.\boldsymbol{T})\boldsymbol{T} + \mathcal{P}\kappa\boldsymbol{n} + \mathcal{P}\big(\boldsymbol{T}.H_\Psi(\mathcal{C})\boldsymbol{T}\big)\boldsymbol{N}(\mathcal{C})\big)\,.\,\eta\,|\mathcal{C}_t|\,dt,\end{aligned}$$

thanks to an integration by parts, and where κ is the curvature of $\mathcal{C}$. Combining I_1 and I_2 we get

$$\left.\frac{dE}{d\lambda}\right|_{\lambda=0} = \int \big(\nabla\mathcal{P} - \big(\boldsymbol{N}.\nabla\mathcal{P}\big)\boldsymbol{N} - \big(\,\nabla\mathcal{P}.\boldsymbol{T}\big)\boldsymbol{T} - \mathcal{P}\big(\kappa\boldsymbol{n} + \big(\boldsymbol{T}.H_\Psi(\mathcal{C})\boldsymbol{T}\big)\boldsymbol{N}\big)\big)\,.\,\eta\,|\mathcal{C}_t|\,dt.$$

Finally, taking the following relations into account

$$\begin{aligned}&\nabla\mathcal{P} - \big(\boldsymbol{N}.\nabla\mathcal{P}\big)\boldsymbol{N} - \big(\,\nabla\mathcal{P}.\boldsymbol{T}\big)\boldsymbol{T} = \big(\nabla\mathcal{P}.\boldsymbol{B}\big)\boldsymbol{B},\\ &\kappa\boldsymbol{n} + \big(\boldsymbol{T}.H_\Psi(\mathcal{C})\boldsymbol{T}\big)\boldsymbol{N} \;=\; \kappa\boldsymbol{n} - \big(\boldsymbol{n}.\nabla\Psi\big)\kappa\boldsymbol{N} = \kappa\big(\boldsymbol{n} - \big(\boldsymbol{n}.\boldsymbol{N}\big)\boldsymbol{N}\big) = \big(\boldsymbol{n}.\boldsymbol{B}\big)\kappa\boldsymbol{B}\end{aligned}$$

we get

$$\left.\frac{dE}{d\lambda}\right|_{\lambda=0} = \int \big((\nabla\mathcal{P} - \mathcal{P}\kappa\boldsymbol{n})\ .\ \boldsymbol{B}\big)\big(\boldsymbol{B}.\ \eta\big)|\mathcal{C}_t|\,dt$$

Since this integral is equal to zero for every function η, we get that for every t of the parameterization domain of $\mathcal{C}$

$$\nabla\mathcal{P}(\mathcal{C}).\boldsymbol{B}(\mathcal{C}) - \mathcal{P}(\mathcal{C})\kappa\big(\boldsymbol{n}.\boldsymbol{B}\big)= 0$$

References

1. R. Ardon and L.D. Cohen. Fast Constrained surface extracion by minimal paths. *2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision*, pages 233–244, October 2003.
2. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
3. V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. Minimal-surfaces based object segmentation. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 19(4):394–398, April 1997.
4. Li-Tien Cheng. *The Level Set Method Applied to Geometrically Based Motion, Materials Science, and Image Processing.* PhD thesis, University of California, Los Angeles, 2000.
5. L.D. Cohen. On active contour models and balloons. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 53(2):211–218, 1991.
6. L.D. Cohen and R. Kimmel. Global minimum for active contour models: A minimal path approach. *International Journal of Computer Vision*, 24(1):57–78, August 1997.
7. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
8. R. Kimmel and J.A. Sethian. Optimal algorithm for shape from shading and path planning. *Journal of Mathematical Imaging and Vision*, 14(3):237–244, 2001.
9. R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modelling with front propagation: A level set approach. *IEEE Transactions On Pattern Analysis and Machine Intelligence*, 17(2):158–175, February 1995.
10. S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: algorithms based on the hamilton-jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
11. N. Paragios and R. Deriche. A pde-based level-set approach for detection and tracking of moving objects. Technical Report 3173, ROBOTVIS Project, INRIA Sophia-Antipolis, May 1997.
12. J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the Natural Academy of Sciences*, 93(4):1591–1595, February 1996.
13. J.A. Sethian. *Level set methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences.* Cambridge University Press, University of California, Berkeley, 2nd edition, 1999.
14. J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, Septembe 1995.
15. H. Zhao, S. Osher, and R. Fedkiw. Fast surface reconstruction using the level set method. *Worksohp on Variational and Level Set Methods In Computer Vision*, pages 194–201, July 2001.