Fast Constrained Surface Extraction by Minimal Paths

Roberto Ardon[1,2]
[1] MEDISYS-Philips France
51, rue Carnot, Suresnes 92156
roberto.ardon@philips.com

Laurent D. Cohen[2]
[2] CEREMADE, Université Paris Dauphine
75775 Paris, France
cohen@ceremade.dauphine.fr

Abstract- A new approach performing surface extraction from 3D images under user defined geometrical constraints is presented. The constraints are introduced through boundary curves given by the user or by preprocessing. Our approach is based on minimal paths that integrate the information coming from the constraint curves and from the image through a potential representing the features to extract. The minimal paths build a network of curves that represent a first approximation of the desired 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. Our paper describes a fast construction obtained by exploiting the Fast Marching algorithm and a fast analytical interpolation method. Moreover, a model extension as well as a Level set method can be used to refine the segmentation when higher accuracy is required. The algorithm has been successfully applied to 3D ultrasound heart images and synthetic images.

1 Introduction

Since their introduction by Kass et al. [9], deformable models have been extensively used to find objects in 2D and 3D images through the minimization of an energy. Two of the main difficulties these models suffer from are local minima "traps" and integrating external information. Since the energy used for the minimization is not convex, the existence of local minima can prevent the active object from finding the wanted minimum. Moreover, when image quality is very low, energy minima do not necessarily represent the desired segmentation and the model should be able to take into account information given by the user.

In this article we address the problem of introducing the user information and using it to build a global minimum of the energy. We consider a situation where the extra information is given by 2D contours as segmentations in two parallel planes extracted from the original 3D image. This is a common situation in medical imaging; figure 1 shows different views of a 3D left ventricle ultrasound image with the two planar segmentations.

The main idea of this work is to first estimate the surface as a set of 3D minimal paths that join the two given curves. As will be shown, it is possible to build minimal paths [7, 8] as global minima of the energy. Thus our segmentation will not suffer from local minima problems, as would other active surface methods like [4, 3]. By constraining the extremities of the paths to lie on the user defined curves we introduce strong constraints on the resulting minimum. We thus obtain a dense set of paths that perceptually reconstruct the surface of the object of interest. In order to get the complete surface, we introduced an adapted interpolation method. If needed, a Level Set model [10, 3] can be used afterwards to refine the segmentation since we now have a very close initialization. While the example of figure 1 is typical, and could maybe be dealt with a slice by slice method [4], our method works as well on general surfaces as seen in figures 3 or 15. The article structure is as follows: we begin in section 2 by recalling the principles of finding minimal paths between points with respect to an image dependent energy. In section 3 we extend this method to the construction of minimal paths be-

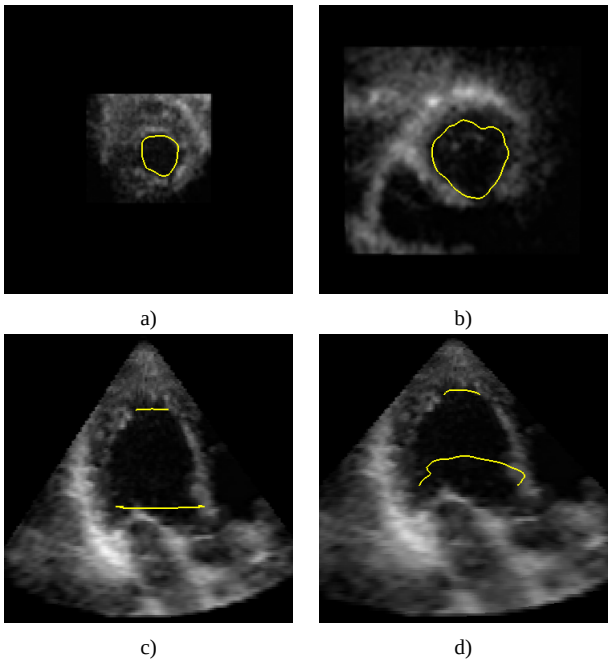


Figure 1: Ultrasound Example. a) and b) show constraint curves as 2D segmentations of two slices of a 3D ultrasound heart image. c) and d) show different views of the 3D image and the constraint curves.

tween a point and a curve and we exploit this extension in the construction of a network of feature-based minimal paths. In section 4 we present a very fast interpolation method adapted to our problem which allows to derive our segmenting surface from the paths. In section 5 we show some examples on synthetic images and ultrasound heart images.

2 Background on Minimal Paths

In this section we will present the main background to our work. Section 2.1 recalls the method allowing to find minimal energy paths between two points in a 2D image introduced in [7]. In section 2.2 we rapidly review the Fast Marching method [12] and its 3D extension proposed in [8].

2.1 Minimal paths as global minima

A known classical approach to boundary segmentation was proposed by authors of [9] who defined the active contour energy by

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

where $\mathcal{C}$ was a curve in a 2D image, Ω its definition domain and $\mathcal{P}$ the data attraction potential. The deformation of $\mathcal{C}$ was driven by the minimization of this energy.
Authors of [7] proposed to use minimal paths to find the global minimum of this energy. They showed that energy (1) could be simplified and that the initialization of $\mathcal{C}$ could be reduced to giving only its two end points (p_0 and p_1). Their simplified version,

$$E(\mathcal{C}) \;=\; \int_0^L \{\omega + \mathcal{P}(\mathcal{C}(s))\}\,ds = \int_0^L \left\{\tilde{\mathcal{P}}(\mathcal{C}(s))\right\} ds, \tag{2}$$

where L is the length of $\mathcal{C}$ and $\tilde{\mathcal{P}} = \omega + \mathcal{P}$ is assumed to be positive, achieves the smoothing of the curve by the only influence of the constant ω.

The solution to this minimization problem is obtained through the calculation of the minimal action map $\mathcal{U}$ (which is the minimal energy integrated along a path between the starting point p_0 and any given point p of the plane):

$$\mathcal{U}(p) = \inf_{\mathcal{C}\in\mathcal{A}_{p_0 p}} \left\{ \int_0^L \tilde{\mathcal{P}}(\mathcal{C}(s)ds \right\} \tag{3}$$

where $\mathcal{A}_{p_0 p}$ is the set of all paths between p_0 and p.
Once $\mathcal{U}$ is obtained, the search of the minimal action path connecting p_0 and point p_1 consists in following the opposite gradient direction on $\mathcal{U}$ starting from p_1 until p_0 is reached, solving:

$$\frac{d\mathcal{C}}{ds}(s) = -\nabla\mathcal{U}, \text{with } \mathcal{C}(0) = p_1 \;\text{ and } \mathcal{C}(L) = p_0 \tag{4}$$

This back propagation procedure is a simple steepest gradient descent and can be performed as proposed in [7]. The fundamental point of this algorithm is the calculation of the minimal action map $\mathcal{U}$, this is done in a very effective manner through the Fast Marching algorithm described in the next section.

2.2 Fast Marching Resolution

As mentioned in [7], the problem of finding the map $\mathcal{U}$ is directly related to the evolution of a curve $\mathcal{L}$ from an infinitesimal circle shape around p_0 and driven by the evolution equation

$$\frac{\partial\mathcal{L}}{\partial t}(s,t) \;=\; \frac{1}{\tilde{P}}.\,\vec{n}\,(s,t), \tag{5}$$

where $\vec{n}$ is the normal to the curve. More precisely, the level sets of $\mathcal{U}$ satisfy

$$\begin{aligned} \forall t,\quad \mathcal{U}^{-1}(t) &= \left\{p \in \mathbb{R}^2 \mid \mathcal{U}(\mathrm{p}) = \mathrm{t}\right\} \\ &= \left\{p \in \mathbb{R}^2 \mid \exists \mathrm{s},\ \mathcal{L}(\mathrm{s},\mathrm{t}) = \mathrm{p}\right\} \\ &= \mathcal{L}(.,t), \end{aligned} \tag{6}$$

thus the knowledge of $\mathcal{L}$ implies the knowledge of $\mathcal{U}$. It is straightforward to see that $\mathcal{U}$ also satisfies the Eikonal equation

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

To solve this equation numerically, classic finite differences scheme tend to be unstable. The Fast Marching method introduced in [12] relies on a one-sided derivative that looks in the up-wind direction of the front, and gives the correct viscosity solution. A similar algorithm was also proposed in [13]. Details on this algorithm can be found for example in [6, 8].
The principal interest of this method is its efficiency. Only one pass is needed over the grid domain, and by using min-heap data structure, an $O(N\log(N))$ complexity can be ensured on a grid of N nodes.
As shown in [8] the extension of the previous problem to a 3D image is straightforward. Using a 6-connexity neighborhood, the Fast Marching method can be extended, allowing to solve the problem with the same complexity as in the 2D case, thus discretizing equation (7) on a square grid by

$$\begin{aligned} & (max\,\{u - \mathcal{U}_{i-1,j,k}, u - \mathcal{U}_{i+1,j,k}, 0\})^2 \\ + \;& (max\,\{u - \mathcal{U}_{i,j-1,k}, u - \mathcal{U}_{i,j+1,k}, 0\})^2 \\ + \;& (max\,\{u - \mathcal{U}_{i,j,k-1}, u - \mathcal{U}_{i,j,k+1}, 0\})^2 \;= \mathcal{P}_{i,j}^2 \end{aligned} \tag{8}$$

For more details on the algorithm refer to [8].

In [5] the authors expanded the method presented in this section to finding minimal paths between two regions. In the next section we present an extension that allows to find minimal paths with respect to a potential between a curve and a point in a 3D space. Our new approach iterates this process in order to obtain a set of paths which then generate our segmenting surface.

3 Segmenting Paths

Our objective is to generate a surface that extracts an 3D object from image features. This surface will also be constrained to contain two given 3D curves. We propose to build a first approximation of the surface by a set of minimal paths. In section 3.1 we present an extension of the methods presented in section 2 that allows to find minimal paths between a point and a curve in a 3D domain. In section 3.2 we show this leads to the construction of a set of minimal paths that can perceptually rebuild a segmenting surface. In section 3.3 we present improvements on this approach.

3.1 Minimal paths between points and curves

Let us denote by γ a curve defined in a 3D image ($\gamma : \mathbb{R} \rightarrow \mathbb{R}^3$) and by p a point of $\mathbb{R}^3$. We call path between γ and p, a curve $\mathcal{C}$ such that $\mathcal{C}(0) = p$ and $\mathcal{C}(L) \in \gamma$ (L being the length of $\mathcal{C}$, parameterized by its arc-length).
In this case, the minimal action map $\mathcal{U}$ is defined as the function that associates to each point $p \in \mathbb{R}^3$ the energy value of the minimal path to γ

$$\mathcal{U}(p) = \inf_{\mathcal{C} \in \mathcal{H}} \left\{ \int_0^L \tilde{\mathcal{P}}(\mathcal{C}(s))ds \right\} \tag{9}$$

where $\mathcal{H}$ is the set of all paths between γ and p.
As for section 2.2 the problem of finding the map $\mathcal{U}$ is the same as solving the partial differential problem

$$\frac{\partial \Gamma(u,v,t)}{\partial t} = \frac{1}{\tilde{P}} . \vec{n}\ (u,v,t) \tag{10}$$

which is the evolution equation of a surface (Γ) evolving under a normal force ($\frac{1}{\tilde{P}} . \vec{n}$). Its initial state will be set to be, for example, a tube around γ. The minimal action map $\mathcal{U}$ satisfies, as in previous section, the Eikonal equation (7), with $\mathcal{U} = 0$ on γ.
Numerically, this equation can also be solved by applying the Fast Marching algorithm, only a minor change is done on its initialization. As detailed in [5], instead of initializing with a seed point, we initialize with all points of γ.
In order to find the minimal path, the back-propagation will start from a given point p in space and will stop when a point of γ is reached.
Next section shows how this extension can be used to produce a set of minimal paths between two curves with respect to image features.

3.2 Free Minimal Paths Network

Given some potential $\mathcal{P}$ that takes lower values near the edges or features of a 3D image and two curves $\mathcal{C}_1$ and $\mathcal{C}_2$ defined in the same 3D image domain, our goal is to find a surface $\mathcal{S}$ that satisfies the following conditions

- $\mathcal{S}$ contains the curves $\mathcal{C}_1$ and $\mathcal{C}_2$,
- $\mathcal{S}$ is close to the features of the image expressed through low values of $\mathcal{P}$.

The first step of our algorithm consists in the generation of a network of minimal paths with respect to $\mathcal{P}$ that join curves $\mathcal{C}_1$ and $\mathcal{C}_2$.
In order to generate this network we compute for each point p of $\mathcal{C}_2$ the minimal path g_p between this point and $\mathcal{C}_1$:

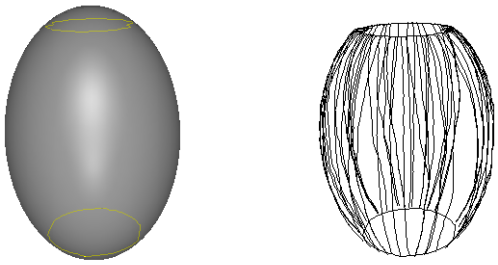
$$g_p = \underset{\mathcal{C} \in \mathcal{H}_p}{ArgMin} \left(\int_0^L \tilde{\mathcal{P}}(\mathcal{C}(s))ds \right), \tag{11}$$

where $\mathcal{H}_p$ is the set of all paths joining p to $\mathcal{C}_1$.

As presented in section 3.1 this is done by solving the Eikonal equation (7) related to $\mathcal{P}$ using Fast Marching and initializing $\mathcal{U} = 0$ on $\mathcal{C}_1$. For each point p of $\mathcal{C}_2$, a back propagation procedure (4) is performed in order to find the path g_p.

Note that for every p, the path g_p is a global minimum of the energy $E(C) = \int \{\mathcal{P}(C)\}$, thus it best approaches lower values of $\mathcal{P}$ and at the same time it joins $\mathcal{C}_1$ and $\mathcal{C}_2$.

An illustration of this construction is given in figure 2, a potential adapted to finding the external surface of an ellipsoid is used ($\mathcal{P}$ has a small value on the ellipsoid, and a large value in the background). It is clear from this example that the generated set of minimal paths $\{g_p\}_{p \in \mathcal{C}_2}$ perceptually rebuilds the object of interest between the two curves, giving an approximation of the searched surface $\mathcal{S}$.



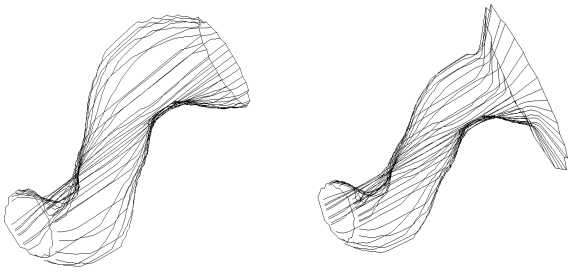
a) Original Ellispoid to be found and curves $\mathcal{C}_1$ and $\mathcal{C}_2$.

b) Set of minimal Paths between $\mathcal{C}_1$ and $\mathcal{C}_2$.

Figure 2: Path network obtained on a synthetic 3D image.

It should also be noted that even when the given 3D curves $\mathcal{C}_1$ and $\mathcal{C}_2$ are not located on the object features, the set of paths will choose a compromise between proximity to features and joining $\mathcal{C}_1$ and $\mathcal{C}_2$ with a minimal cost. This is illustrated in figure 3 where the image potential drives the paths towards a tubular shape. In the second example, curve $\mathcal{C}_2$ does not lie on the tube.

The previously described construction of the set of paths is numerically very effective, the fast marching algorithm is only performed once and each path is determined by a very



a) 3D Curves on features. b) $\mathcal{C}_2$ no longer on features.

Figure 3: Path network obtained on an synthetic 3D 'S' shaped tube. (a) The two initial curves are situated on the external surface of the tube, the image shows the obtained network of minimal paths. (b) although the upper curve is not situated on the external surface of the tube, paths join this surface in a minimal manner.

low complexity gradient descent method. Nevertheless its main drawback resides in the fact that minimal paths between the points of $\mathcal{C}_2$ and the curve $\mathcal{C}_1$ will tend to merge like rivers do when descending from mountains to a valley and thus missing some areas of the surface of interest (see figure 7a). This behavior can hardly be avoided because it is mainly driven by the geometrical nature of the features this potential represents. In the next section we present some solutions to handle situations where the minimal paths alone fail to perceptually build the object of interest.

3.3 Set of Restricted Minimal Paths

Figure 4 illustrates a usual situation where the set of paths described in the previous section fails to correctly recover the object of interest. The potential is minimal on a surface which is the blending of a plane and a half-sphere. Minimal paths prefer to take a short cut around the sphere rather than 'climbing' on it. The reason for this behavior is that since the path can join any point on curve $\mathcal{C}_2$, it is shorter and thus of lower energy to turn around the sphere.

In more complicated image situations like ultrasound heart images, the majority of the paths merge and few points of the destination curve ($\mathcal{C}_2$) are reached. This is problematic in order to extract a surface since too much information is lost (see figure 7a).

To cope with this problem, a simple but effective method is to geometrically restrict the back-propagation procedure that builds the minimal paths (equation 4). In our case a manner to obtain a more uniformly distributed set of paths is to restrict their construction to planes. When backpropagating from a point $p \in \mathcal{C}_2$ we define a plane Π_p containing p and given through its normal vector $\vec{n_p}$. Instead of building the solution path from (4) we build $\mathcal{C}_p$ by projecting the

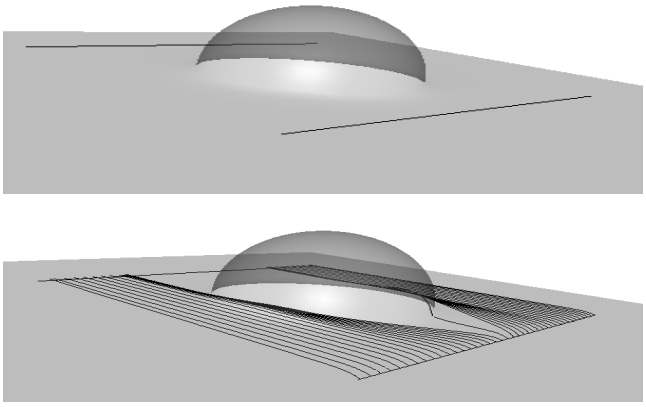


Figure 4: Data (above): half-sphere blended on a plane (transparent visualization) and $\mathcal{C}_1$ and $\mathcal{C}_2$ (black segments). Result (below): set of paths joining $\mathcal{C}_1$ and $\mathcal{C}_2$. The network of paths misses the half-sphere.

evolution equation (4) on Π_p.

$$\frac{d\mathcal{C}_p}{ds}(s) = -\nabla\mathcal{U} + \left(\nabla\mathcal{U}.\,\vec{n_p}\right).\,\vec{n_p} \qquad (12)$$

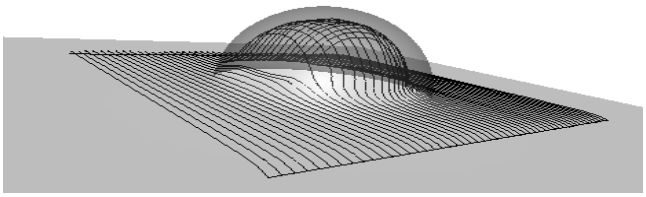


Figure 5: Network obtained by the restricted back propagation. Compared with Figure 4, the network is denser, perceptually the half-sphere is nearly complete.

We show in figure 5 the result obtained with restriction of the paths to parallel planes which are orthogonal to $\mathcal{C}_1$ and $\mathcal{C}_2$ (n_p is a constant vector).
In practice, the two given curves are usually closed contours segmenting 2D slices of the 3D volume. In such cases we can intuitively define the plane Π_p by three points: G_1 (center of mass of $\mathcal{C}_1$), G_2 (center of mass of $\mathcal{C}_2$) and p, and thus

$$\vec{n_p} = \frac{\vec{G_1G_2} \wedge \vec{G_1p}}{\left\|\vec{G_1G_2} \wedge \vec{G_1p}\right\|} \qquad (13)$$

As the point p varies along $\mathcal{C}_1$, the plane Π_p will "rotate" around the principal axis G_1G_2 (see figure 6) . This method will prove to be quite effective when dealing with objects that present rotational symmetry or nearly. At each of its positions, plane Π_p will naturally be close to a meridian plane. Thus, minimal paths will generate 2D segmentations on those planes. When applied to ultrasound heart images of the left ventricle, our method is very efficient (see figure

7b and section 5.3) since the left ventricle is nearly radially symmetric around an axis.

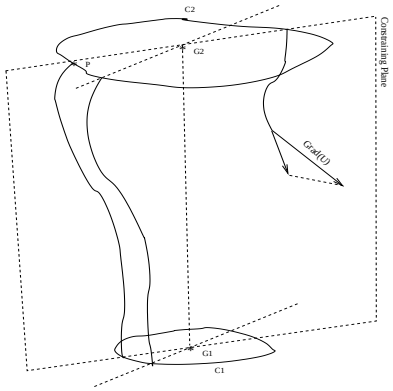
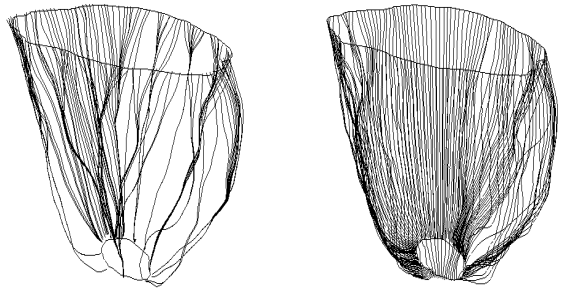


Figure 6: Illustration of the projected back propagation



a) Set of paths b) Restricted set of paths

Figure 7: Results on a left ventricle ultrasound image.

4 The Interpolated Surface

In this section we describe the final step of our algorithm, which consists in obtaining a surface representation derived from the above calculated paths.
Our goal in now to find a surface interpolating the information given by the network of paths.
First we consider a discrete representation of curve $\mathcal{C}_2$ with nodes $\left\{P_2^i\right\}$ uniformly spaced with unit step with respect to arc-length. The paths $\left\{g^i\right\}_{i\in\{1,2,..,n\}}$ correspond to back-propagation from the nodes $\left\{P_2^i\right\}$ to $\mathcal{C}_1$.

By construction two paths belonging to this network may either have an empty intersection or merge. Based on this fact, we have introduced a novel and simple interpolation method inspired by splines that integrates information coming from both the network of paths and the constraining curves $\mathcal{C}_1$ and $\mathcal{C}_2$. If paths do not merge, the method generates a surface which is at least continuously differentiable.

4.1 Analytical path interpolation

Our surface interpolation method is based on a local linear interpolation of each surface sector. A sector is defined by two successive minimal paths and the two portions of curves $\mathcal{C}_1$ and $\mathcal{C}_2$ (see Figure 8).

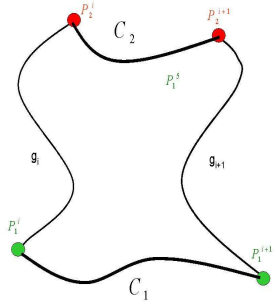


Figure 8: illustration of a sector defined by two paths

Let s_1 and s_2 be the arc-length parameterizations of $\mathcal{C}_1$ and $\mathcal{C}_2$ and $\mathcal{C}_1^i$, $\mathcal{C}_2^i$ their restrictions to the i^{th} sector. The paths network is noted $\left\{g^i\right\}_{i\in\{1,2,..,n\}}$ and $\left\{P_1^i\right\}$, $\left\{P_2^i\right\}$ are the intersection points of $\mathcal{C}_1$, $\mathcal{C}_2$ with paths $\left\{g^i\right\}$ ($\mathcal{C}_j \cap g^i = \mathcal{C}_j(s_j(P_j^i)), j = 1, 2$).

Our aim is to generate a parameterized surface D that is continuously differentiable and is parameterized with u and v.
The essential constraint on D is to contain curves $\mathcal{C}_1, \mathcal{C}_2$ and all paths. In order to obtain continuity on the frontiers of sectors, if D^i is the restriction to the i^{th} sector of D, it must verify:

$$(\mathcal{P}_1) \quad \left\{ \begin{array}{rcl} D^i\big(.,v(P^i)\big) & \equiv & g^i \\ D^i\big(0,.\big) & \equiv & C_1^i \\ D^i\big(1,.\big) & \equiv & C_2^i \end{array} \right.$$

If we now impose on D to satisfy the following condition for every $u \in [0,1]$

$$(\mathcal{P}_2) \quad \frac{\partial D^i}{\partial v}\big(u,v(P^{i+1})\big) = \frac{\partial D^{i+1}}{\partial v}\big(u,v(P^{i+1})\big) \qquad (14)$$

D will be at least a continuously differential surface parameterized by u and v.
A fundamental step of the construction of the surface is the introduction of a function σ which is strictly increasing, of class at least C^1 and creates the following correspondence between the arc-lengths of curves $\mathcal{C}_1$ and $\mathcal{C}_2$: $s_2\left(P_2^i\right) = \sigma\left(s_1\left(P_1^i\right)\right)$. More precisely, σ can be chosen as a cubic spline from $[0,1]$ into itself, that verifies the

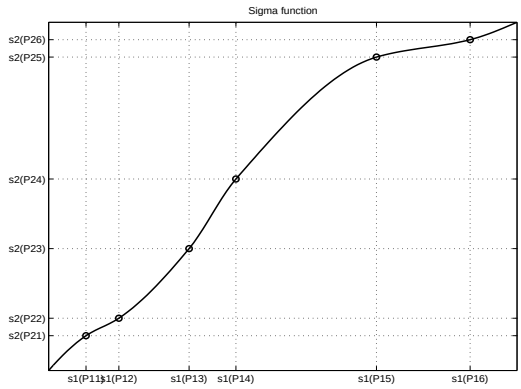


Figure 9: illustration of function σ that builds the wanted correspondence between arc-lengths.

previous equalities (see figure 9).
This allows to use a common parameterization, noted v, on both $\mathcal{C}_1$ and $\mathcal{C}_2$ and thus the same abscissa for the intersecting points $\{P^i\}$. Notice that we need only to change the parameter on $\mathcal{C}_2$ (since v can be chosen identical to s_1).
In the same manner, we will parameterize all paths $\{g^i\}$ with the same parameter u which takes its values on the interval $[0,1]$. In figure 10 we give an illustration of the coorrespondace established by D between $[0,1]\times[0,1]$ and our surface.

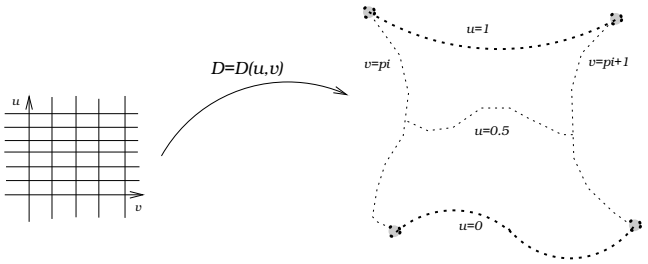


Figure 10: Illustration of the correspondance established by D

It can be shown that the following expression of D satisfies ($\mathcal{P}_1$) and ($\mathcal{P}_2$),

$$\begin{aligned} D^i(u,v) &= \alpha^i(u,v).\left[C^i(u,v) - \tilde{g}^i(u)\right] \\ &+ (1-\alpha^i(u,v)).\left[C^i(u,v) - \tilde{g}^{i+1}(u)\right] \end{aligned} \tag{15}$$

where we have the following definitions

- $C^i(u,v) = (1-f(u)).C_1^i(v) + f(u).C_2^i(\sigma(v))$
- $\tilde{g}^i(u) = g^i(u) - C^i\big(u, v(P^i)\big)$
- $\alpha^i = (1-\tau_i(v))\left\{1+\tau_i(v)\left[\mu_i(u)-1\right].h\circ\tau_i(v)\right\}$
- $\tau_i(v) = \frac{v-v(P^i)}{v(P^{i+1})-v(P^i)}$
- $\mu_i(u) = \frac{\tilde{g}^{i+2}(u)-\tilde{g}^{i+1}(u)}{\tilde{g}^{i+1}(u)-\tilde{g}^{i}(u)}.\tau_{i+1}(v(P^i))$

The two functions f and h can be chosen among all the differentiable functions on $[0,1]$ and must verify $h(1)=f(0)=1$ and $f(1)=h(0)=0$.
A proof of these statements can be found in [1].
The main interest of this interpolation method is its calculation speed. Only elementary calculations are needed to generate the surface (there is no matrix inversion) and both information from the paths and from the initial curves are integrated in the process. Because of its capacity to integrate the information of the given curves, even when many paths are lacking the interpolation is still satisfactory (see Figure 11).

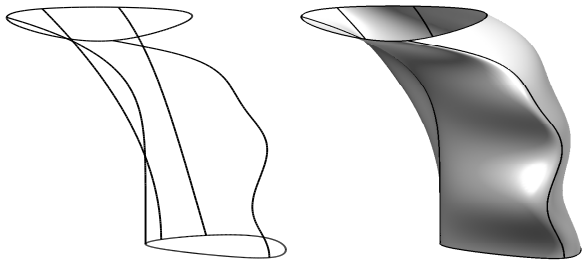


Figure 11: Interpolated surface from $\mathcal{C}_1$ and $\mathcal{C}_2$ (the lower and upper curves) and 4 paths. The path network is here limited for visibility.

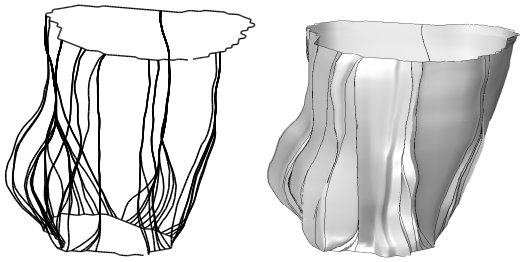


Figure 12: Paths network obtained when applying our method to a ultrasound left ventricle image and the analytical interpolating surface.

4.2 Using a Level Set method

Depending on the nature of the image, the obtained surface can be considered as the actual segmentation or, in more difficult cases, as a good initialization for level set active contours ([3, 10]). From the regular mesh obtained through the analytical method a distance map, ϕ, that will satisfy $\phi^{-1}(0) = D$ can be computed as a distance map using the *Fast Marching* method initialized with D and with a constant potential of value 1.
Once ϕ is obtained, a classical Level set evolution method gives a more accurate segmentation. Note that few iterations of the numerical evolution of ϕ will be needed since the surface we produced is already close to image features and its

energy is nearly the minimum we are searching for. Compared to using a level set approach from the beginning, our approach is much faster, needs no initialisation, avoids local minima and allows to impose hard constraints.

5 Application

In this section we will apply our algorithm to some synthetic and ultrasound images. Firstly we will give some indications on a possible procedure for generating a potential function capable of detecting edges.

5.1 Choosing the Potential

In order to find a relevant object in a given image, we are going to use a potential (or cost function) that will allow the propagating front $\mathcal{U}$ (see section 2) to rapidly advance in regions where edges are likely to be present. The form of the potential we used is the following

$$\mathcal{P} = \alpha.h(|\nabla I_\sigma|) + (1-\alpha) * h_{gap}(\Delta I_\sigma), \qquad (16)$$

where h and h_{gap} are two functions bounded to $[0,1]$ and I_σ is the convolution of the given image with a gaussian kernel of variance σ.
The right choice for functions h and h_{gap} is restricted by the fact that the cost function should be high in areas where it is unlikely to encounter an edge. A simple choice for h can be the classical form: $h(x) = \frac{1}{1+x^2/\lambda^2}$ ([2, 10]), where λ is a user defined contrast factor that can be computed as an average gradient value, $\lambda = \frac{\int \|\nabla I\| d\mathbf{x}}{\int d\mathbf{x}}$.
The h_{gap} function is chosen to be a zero crossing detector that depends on a user defined constant gap. Because of the noisy nature of the Laplacian of an image, h_{gap} is set to be a binary map that detects only relevant zero crossing points of the Laplacian. In order to automatically detect the contribution factor α, a learning method could be used as in [11], collecting information from the image on a neighborhood of the given contours. In practice it is very difficult to set automatically this value because most of the time the given contours are drawn in zones where the image has features that are very hard to detect. The knowledge of the practitioner is thus needed.

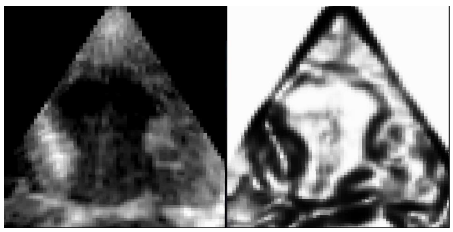


Figure 13: View of a slice of a 3D ultra sound image of the left ventricle. On the right, cost function.

5.2 Results on synthetic images

In this section we present some segmenting surfaces obtained using our method on some synthetic images.
Figure 14 represents the reconstruction of our 'S' shaped tube. This example shows the advantages introduced by our method: after the fast constructions of the paths and the very rapid interpolation we have nearly reconstructed the object (Figure 14 c)). We also show the capability of building a surface which is a compromise between the hard constraints and the image features.

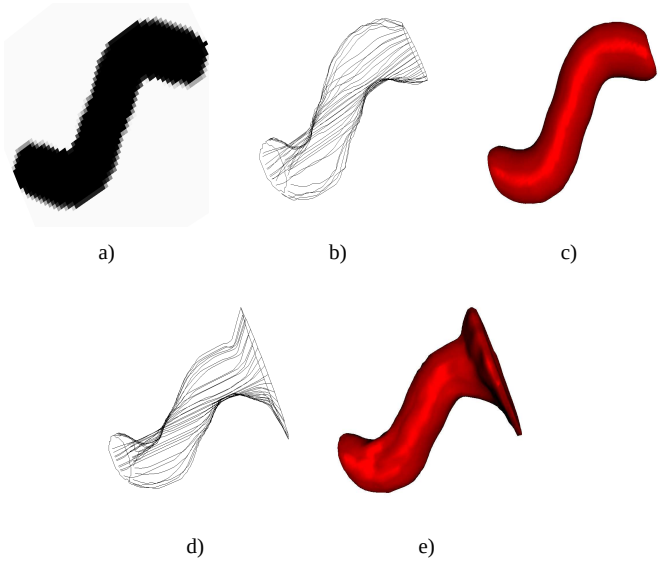


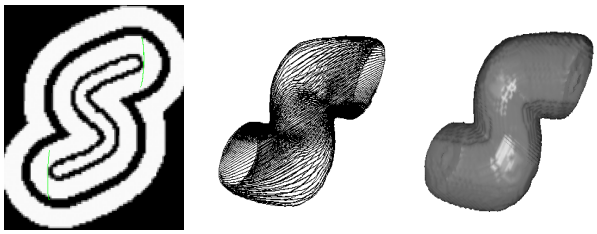
Figure 14: Reconstruction of a tubular synthetic image: a) Intersection of the 3D 'S' shaped tube image with a plane. b) Network of paths obtained when the two constraining curves are situated at the ends of the tubular surface. c) Interpolated surface. d) The network and surface e) obtained when segmenting the same image but no longer imposing to $\mathcal{C}_2$ to be on the tubular surface.

Figure 15 shows the segmentation of a much more complicated image (Figure 15a) that contains internal and external surfaces close to the desired surface. If one wishes to obtain the middle 'S' shaped tube most of variational methods will fail, unless a very close initialization is given, because of the presence of many local minima. Our method manages to extract the object with the only extra information of two curves lying on it. Minimal paths find the global minimum of the energy.

In next section we present a real image result. In real images local minima are very difficult to avoid, and this is one aspect of the importance of our contribution.

5.3 Reconstruction of the left ventricle from ultrasound images

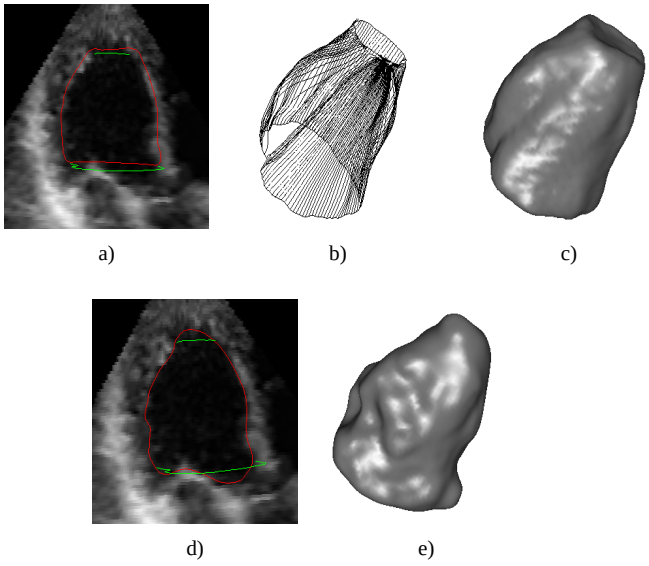
It is a common practice in ultrasound heart images to perform as a first step 2D segmentation on a slice. When images are of very low quality the practitioner does the segmentation



a) b) c)

Figure 15: a) Intersection of the 3D image with a plane. b) Obtained set of paths. c) Interpolated surface.

by hand. Our algorithm allows to rapidly build a 3D model that approximates quite well the left ventricle between two segmented slices. Figure 16 shows the result of the segmentation of the 3D image with the constraining paths presented in Figure 1. As can be seen in Figure 16b the restricted set of paths has already rebuilt perceptually the ventricle. Figure 16c represents the interpolated surface with our method. We then applied a classical level set method evolution to refine the segmentation.



a) b) c)

d) e)

Figure 16: a) A slice of the 3D ultrasound image, we also have drawn the intersection of our interpolated surface with this plane. b) Set of paths. c) Interpolated surface. d) Planar view of the same slice, intersection with the model evolved as a level set. e) final segmentation after a few iterations of the level set.

6 Conclusion and future work

In this article we presented a new method to build a surface constrained to lie on two curves using minimal paths as guides for the generation of an interpolated surface. We also presented a novel interpolation method which is characterized by its simplicity and its speed.
This construction has two major interests. First, in the region defined between the two given curves, the obtained surface avoids local minima and is very close to the actual minimizing surface. Second, the obtained surface is constrained to contain the given curves.
A future perspective of this work could be the search for particular points during the construction of the minimal paths in order to allow surface reconstruction outside the region between the given curves.

References

[1] R. Ardon and L.D. Cohen. Fast constrained surface extraction by minimal paths. Technical report, Les cahiers du Cérémade, Septembe 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. *PAMI*, 19(4):394–398, April 1997.

[4] L.D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2D and 3D images. *PAMI*, 15(11):1131–1147, November 1993.

[5] L.D Cohen and T. Deschamps. Grouping conected components using minimal path techniques. application to reconstruction of vessels in 2D and 3D images. *CVPR01*, 2001.

[6] L.D. Cohen and R. Kimmel. Fast marching the global minimum of active contours. In *ICIP96*, volume 1, pages 473–476, Lausanne, Switzerland,, sept 1996.

[7] 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.

[8] T. Deschamps and L.D. Cohen. Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Medical Image Analysis*, 5(4), December 2001.

[9] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[10] 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.

[11] Myriam Greff O. Gerard, T. Deschamps and Laurent D. Cohen. Real-time interactive path extraction with on-the-fly adaptation of the external forces. *ECCV02*, may 2002.

[12] J.A. Sethian. *Level set methods and Fast Marching methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge University Press, University of California, Berkeley, 2nd edition, 1999.

[13] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, Septembe 1995.