

# From a Single Point to a Surface Patch by Growing Minimal Paths

Fethallah Benmansour and Laurent D. Cohen

CEREMADE, UMR CNRS 7534, Université Paris Dauphine,  
Place du Maréchal De Lattre De Tassigny, 75775 PARIS CEDEX 16, France  
{benmansour,cohen}@ceremade.dauphine.fr

**Abstract.** We introduce a novel implicit approach for surface patch segmentation in 3D images starting from a single point. Since the boundary surface of an object is locally homeomorphic to a disc, we know that the boundary of a small neighboring domain intersects the surface of interest on a single closed curve. Similarly to active surfaces, we use a cost potential which penalizes image regions of low interest. First, Using a front propagation approach from the source point chosen by the user, one can see that the closed curve corresponds to a valley line of the arrival time from the source point. Next, we use an implicit 3D segmentation method. It assumes that the object boundary contains two known constraining curves. In our case, the first curve is reduced to a point and the other one is automatically detected by our approach. A partial differential equation is introduced and its solution is used for segmentation. The zero level set of this solution contains the valley line and the source point as well as the set of minimal paths joining them. We present a fast implementation which has been successfully applied to 3D biomedical and synthetic images.

## 1 Introduction

In this paper we are interested in interactive segmentation of a surface in a 3D image by clicking a single point on the boundary of an object and obtaining a patch of the desired surface around the given point. For this we use energy minimizing techniques and partial differential equations. Energy minimization techniques have been applied to a broad variety of problems in image processing and computer vision. Since the original work on snakes [1], they have notably been used for boundary detection. An active contour model, or snake, is a curve that deforms its shape in order to minimize an energy combining an internal part which smooths the curve and an external part which guides the curve toward particular image features. One of the main drawbacks of this approach is that it suffers from local minima 'traps'. Consequently, results strongly depend on the model initialization. Since the publication of [1], much work has been done in order to free active models from the problem of local minima.

Cohen and Kimmel [2] introduced an approach to globally minimize the geodesic active contour energy, provided that two endpoints of the curve are

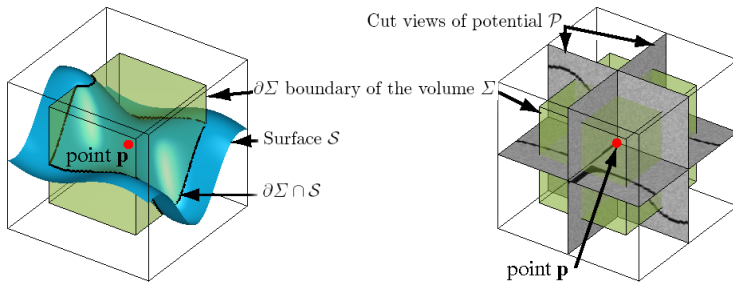
initially supplied by the user. This energy is of the form  $\int_{\gamma} \tilde{\mathcal{P}}$  where the incremental cost  $\tilde{\mathcal{P}}$  is chosen to take lower values on the contour of the image, and  $\gamma$  is a path joining the two points. The solution of this minimization problem is obtained through the computation of *the minimal action map* associated to a source point. The minimal action map can be regarded as the arrival times of a front propagating from the source point with velocity  $(1/\tilde{\mathcal{P}})$ , and it satisfies the Eikonal equation. Therefore, we can compute efficiently the minimal action map with the *Fast Marching Method* as will be detailed in section 2. However their approach [2] cannot be directly extended to find the global minimum for an active surface in a 3D image. Nevertheless, this approach has been extended to surfaces in a 3D image by extracting a *minimal* surface laying on two given curves [3]. The advantage of this method is that it does not suffer from local minima problems, as would other active surface methods like [4, 5].

In this work, we focus on a novel approach for 3D object segmentation. Our aim is to generate a local surface patch from a single point. The method presented herein can be seen as an extension of the Eulerian approach presented by Ardon et al in [3] for surface extraction from a couple of 'constraining' closed curves. But in our case, one of the curves is reduced to a single point and the other one is unknown. Let  $\tilde{\mathcal{P}} : \Omega \rightarrow \mathbb{R}^+$  be a potential, where  $\Omega \subset \mathbb{R}^3$ , such that  $\tilde{\mathcal{P}}$  takes lower values on the surface of the object to be extracted, noted  $\mathcal{S}$  and unknown. Having a single point  $\mathbf{p}$  on  $\mathcal{S}$  and a neighborhood of  $\mathbf{p}$ :  $\Sigma \subset \Omega$ , the required conditions are (see figure 1.)

- the boundary  $\partial\Sigma$  is a connected closed surface.
- $\partial\Sigma \cap \mathcal{S}$  is a simple closed curve.
- $\mathbf{p} \in \mathcal{S} \cap \Sigma$ .

The volume  $\Sigma$  might be a ball or any topology equivalent volume.

Our objective is to find the surface patch  $\mathcal{S} \cap \Sigma$  from the source point  $\mathbf{p}$  and the potential  $\mathcal{P}$ . We proceed in two stages : first, we look for the boundary of the



**Fig. 1.** On the left, one can see the required conditions for the surface patch extraction. The point  $\mathbf{p}$  must be initialized on the surface  $\mathcal{S}$  in the volume  $\Sigma$ .  $\partial\Sigma$ , the boundary of  $\Sigma$ , is a closed surface and  $\partial\Sigma \cap \mathcal{S}$  is a simple closed curve. On the right, we represent the information one has in practice : the surface  $\mathcal{S}$  is unknown but the potential  $\mathcal{P}$  takes lower values along  $\mathcal{S}$  and higher values elsewhere.

surface patch  $\mathcal{S} \cap \partial\Sigma$  and give a good estimate of it  $\Gamma$ ; in fact, running the Fast Marching algorithm (which will be detailed in section 2) from the source point  $\mathbf{p}$  one can see that the *Valley Line*, noted  $\Gamma$ , of the arrival time on the boundary  $\partial\Sigma$  is a good approximation of  $\mathcal{S} \cap \partial\Sigma$ . A detailed definition of the valley line and the way it is extracted is presented in section 3. Next, one can represent the surface of interest as a dense network of minimal paths joining points of the valley line  $\Gamma$  to the source point  $\mathbf{p}$  (section 4). The surface generated by this algorithm is completely composed of globally minimal paths. Indeed, by solving a stationary transport equation of the form  $:\nabla\Psi.\nabla\mathcal{U} = 0$ , where  $\mathcal{U}$  is the action map (defined in section 2), and  $\Psi$  is the unknown, we show that any minimal path between the valley line  $\Gamma$  and the source point  $\mathbf{p}$  is contained in its zero level set  $\Psi^{-1}(\{0\})$ . Important advantages of this approach are that it needs minimal interaction and that it is computationally efficient as explained later. This approach can also be used as computing brick for a complete segmentation from one single point (see section 5). Segmentation results on synthetic and medical images are presented in section 5. Finally conclusions, advantages and drawbacks of our method, and perspectives follow in section 6.

## 2 Background on Minimal Paths

Given a 3D image  $I : \Omega \rightarrow \mathbb{R}^+$  and two points  $\mathbf{p}_1$  and  $\mathbf{p}_2$ , the underlying idea introduced by Cohen and Kimmel [2] is to build a potential  $\mathcal{P} : \Omega \rightarrow \mathbb{R}^{*+}$  which takes lower values near desired features of the image  $I$ . The choice of the potential  $\mathcal{P}$  depends on the application. For example, one can define  $\mathcal{P}$  as a decreasing function of  $\|\nabla I\|$  to extract image edges by finding a curve that globally minimizes the energy functional  $E : \mathcal{A}_{\mathbf{p}_1, \mathbf{p}_2} \rightarrow \mathbb{R}^+$

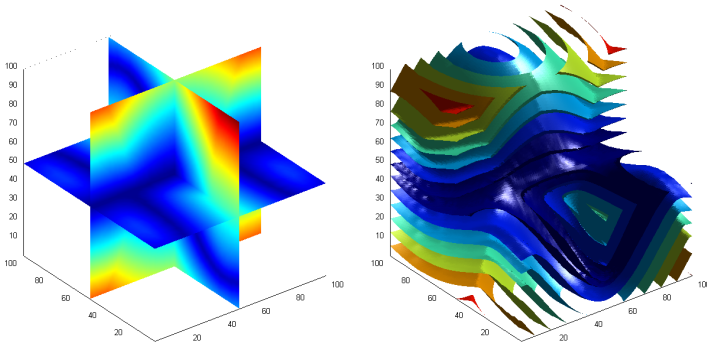
$$E(\gamma) = \int_{\gamma} \{ \mathcal{P}(\gamma(s)) + w \} ds = \int_{\gamma} \tilde{\mathcal{P}}(\gamma(s)) ds, \tag{1}$$

where  $\mathcal{A}_{\mathbf{p}_1, \mathbf{p}_2}$  is the set of all paths connecting  $\mathbf{p}_1$  to  $\mathbf{p}_2$ ,  $s$  is the arc-length parameter,  $w > 0$  is a regularization term and  $\tilde{\mathcal{P}} = (\mathcal{P} + w)$ . A curve connecting  $\mathbf{p}_1$  to  $\mathbf{p}_2$  that globally minimizes the energy (1) is a *minimal path* between  $\mathbf{p}_1$  and  $\mathbf{p}_2$ , noted  $\mathcal{C}_{\mathbf{p}_1, \mathbf{p}_2}$ . The solution of this minimization problem is obtained through the computation of the *minimal action map*  $\mathcal{U}_1 : \Omega \rightarrow \mathbb{R}^+$  associated to  $\mathbf{p}_1$ . The minimal action is the minimal energy integrated along a path between  $\mathbf{p}_1$  and any point  $\mathbf{x}$  of the domain  $\Omega$ :

$$\forall \mathbf{x} \in \Omega, \mathcal{U}_1(\mathbf{x}) = \min_{\gamma \in \mathcal{A}_{\mathbf{p}_1, \mathbf{x}}} \left\{ \int_{\gamma} \tilde{\mathcal{P}}(\gamma(s)) ds \right\}. \tag{2}$$

The values of  $\mathcal{U}_1$  may be regarded as the arrival times of a front propagating from the source  $\mathbf{p}_1$  with velocity  $(1/\tilde{\mathcal{P}})$ .  $\mathcal{U}_1$  satisfies the Eikonal equation

$$\|\nabla\mathcal{U}_1(\mathbf{x})\| = \tilde{\mathcal{P}}(\mathbf{x}) \text{ for } \mathbf{x} \in \Omega, \text{ and } \mathcal{U}_1(\mathbf{p}_1) = 0. \tag{3}$$



**Fig. 2.** Minimal action map  $\mathcal{U}$  from the source  $\mathbf{p}$  using the potential  $\mathcal{P}$  of figure 1 computed using the Fast Marching algorithm. Left: slices through the volume. Right: some equi-distant surfaces (level sets) of  $\mathcal{U}$ .

The map  $\mathcal{U}_1$  has only one local minimum, the point  $\mathbf{p}_1$ , and its flow lines satisfy the Euler-Lagrange equation of functional (1). Thus, the minimal path  $\mathcal{C}_{\mathbf{p}_1, \mathbf{p}_2}$  can be retrieved with a simple gradient descent on  $\mathcal{U}_1$  from  $\mathbf{p}_2$  to  $\mathbf{p}_1$ , solving the following ordinary differential equation with standard numerical methods like Heun’s or Runge-Kutta’s:

$$\frac{d\mathcal{C}_{\mathbf{p}_1, \mathbf{p}_2}(s)}{ds} = -\nabla\mathcal{U}_1(\mathcal{C}_{\mathbf{p}_1, \mathbf{p}_2}(s)), \text{ and } \mathcal{C}_{\mathbf{p}_1, \mathbf{p}_2}(0) = \mathbf{p}_2. \tag{4}$$

### 2.1 Fast Marching Method

The *Fast Marching Method* (FMM) is a numerical method introduced by Sethian in [6] and Tsitsiklis in [7] for efficiently solving the isotropic Eikonal equation on a cartesian grid. In equation (3), the values of  $\mathcal{U}$  may be regarded as the arrival times of wavefronts propagating from the point of  $\mathcal{S}$  with velocity  $(1/\tilde{P})$ . The central idea behind the FMM is to visit grid points in an order consistent with the way wavefronts propagate. It leads to a single-pass algorithm for solving equation (3) and computing the minimal action map  $\mathcal{U}$ .

The FMM is a front propagation approach that computes the values of  $\mathcal{U}$  in increasing order, and the structure of the algorithm is almost identical to Dijkstra’s algorithm for computing shortest paths on graphs [8]. In the course of the algorithm, each grid point is tagged as either *Alive* (point for which  $\mathcal{U}$  has been computed and frozen), *Trial* (point for which  $\mathcal{U}$  has been estimated but not frozen) or *Far* (point for which  $\mathcal{U}$  is unknown). The set of *Trial* points forms an interface between the set of grid points for which  $\mathcal{U}$  has been frozen (the *Alive* points) and the set of other grid points (the *Far* points). This interface may be regarded as a set of fronts expanding from each source until every grid point has been reached. The key to the speed of the FMM is the use of a priority queue to quickly find the *Trial* point with the smallest  $\mathcal{U}$  value. If *Trial* points are ordered in a min-heap data structure, the computational complexity of the FMM is  $\mathcal{O}(N\log_2 N)$ , where  $N$  is the total number of grid points.

A way to estimate  $\mathcal{U}$ , for a grid point  $\mathbf{x}_n$  is presented here. We limit ourselves to the 3D case. Adopting standard notation, we denote by  $\mathcal{U}_{i,j,k}$  the value of  $\mathcal{U}$  at the grid vertex  $(i, j, k)$  associated to the point  $\mathbf{x}_n$  with coordinates  $(i h_x, j h_y, k h_z)$ , where  $h_x, h_y$  and  $h_z$  are grid spacings in the  $x, y$  and  $z$  directions. A discretized version of (3) is solved in order to compute  $\mathcal{U}_{i,j,k}$ . For the Eikonal equation, classic finite difference schemes tend to overshoot and are unstable. Rouy and Tourin [9] showed that the correct viscosity solution for  $\mathcal{U}_{i,j,k}$  is given by the following first order accurate scheme :

$$\begin{aligned} & \left( \frac{\max\{(\mathcal{U}_{i,j,k} - \mathcal{U}_{i-1,j,k}), (\mathcal{U}_{i,j,k} - \mathcal{U}_{i+1,j,k}), 0\}}{h_x} \right)^2 + \\ & \left( \frac{\max\{(\mathcal{U}_{i,j,k} - \mathcal{U}_{i,j-1,k}), (\mathcal{U}_{i,j,k} - \mathcal{U}_{i,j+1,k}), 0\}}{h_y} \right)^2 + \\ & \left( \frac{\max\{(\mathcal{U}_{i,j,k} - \mathcal{U}_{i,j,k-1}), (\mathcal{U}_{i,j,k+1} - \mathcal{U}_{i,j,k}), 0\}}{h_z} \right)^2 = (\tilde{\mathcal{P}}_{i,j,k})^2. \end{aligned} \tag{5}$$

This is an upwind scheme : the forward and backward differences are chosen to follow the direction of the flow of information.

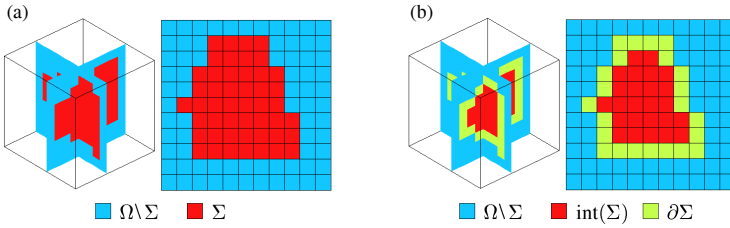
### 3 Valley Line Detection

In this section, we present a method to extract the intersection between the sub-domain boundary and the unknown surface of interest. We propose to use the minimal action map to extract the desired curve, since one can see that it corresponds to a valley line of the minimal action map (without a formal proof). *Ridge* and *valley* lines are concepts used in geomorphology and computer vision [10, 11]. According to Koenderink [12], valley lines are the locus of points on a surface at which the normal curvature assumes a local minimum in the principal direction associated with the largest, negative curvature.

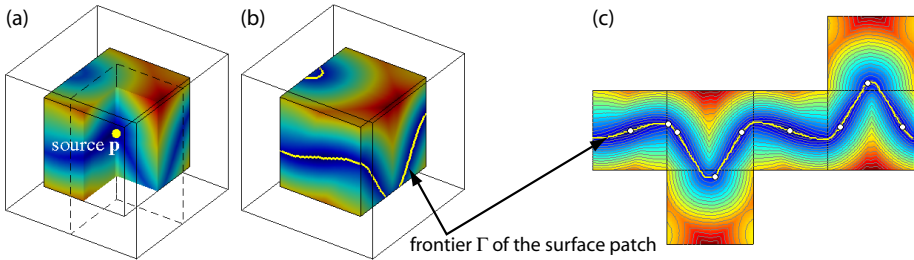
The main drawback of the existing criteria [10, 11] is that thresholding is needed. Hence, the detection is not precise enough, and needs more interaction for real noisy images. Moreover, these approaches are not adapted to our case where we want to extract the valley line of a scalar function defined on a surface topologically equivalent to a sphere. Our approach is heuristic, based on the fact that the fast marching propagates faster along the desired surface and then the minimal action map takes lower values along the curve of intersection between the domain boundary and the surface.

#### *Discrete definition of $\Sigma$ and $\partial\Sigma$ and Minimal action map on $\partial\Sigma$*

In practice, we assume that the volume  $\Sigma$  is defined as a boolean array. Then, we can partition  $\Sigma$  into two subsets,  $\text{int}(\Sigma)$  and  $\partial\Sigma$ , its interior and its boundary. A voxel  $\mathbf{x} \in \Sigma$  is in the interior of the volume if all its 6 neighbors are in  $\Sigma$ , and it is a point of the boundary  $\partial\Sigma$  if  $\mathbf{x} \in \Sigma \setminus \text{int}(\Sigma)$ . Then  $\partial\Sigma$  is also represented by a boolean array (see figure 3).



**Fig. 3.** Discrete representation of the volume  $\Sigma$  and its boundary  $\partial\Sigma$ . (a) The volume  $\Sigma$  is described by a boolean array. (b)  $\Sigma$  is partitioned into two subsets  $\text{int}(\Sigma)$  and  $\partial\Sigma$  such that  $\partial\Sigma$  is connex according to on 26-connectivity.



**Fig. 4.** Minimal action map associated to source point  $\mathbf{p}$  and potential of figure 1. (a) Cut views of the minimal action map  $\mathcal{U}$  on volume  $\Sigma$ . (b) View of  $\mathcal{U}$  on  $\partial\Sigma$ , and its valley line  $\Gamma$ . (c) Unfolded  $\mathcal{U}|_{\partial\Sigma}$ , valley line, and different marked points on  $\Gamma$  correspond to local minima.

Let us note  $\mathcal{U}|_{\partial\Sigma} : \partial\Sigma \rightarrow \mathbb{R}^{+*}$  the restriction of  $\mathcal{U}$  on  $\partial\Sigma$  (see figure 4.) The value  $\mathcal{U}(\mathbf{x})$  for a point  $\mathbf{x}$  in  $\partial\Sigma$  is the arrival time to point  $\mathbf{x}$  of the wavefront propagating from the source point  $\mathbf{p}$  with velocity  $1/\tilde{\mathcal{P}}$ . Since potential  $\tilde{\mathcal{P}}$  takes lower values along the surface  $\mathcal{S}$ , the front propagates faster along it. So, we can reasonably assume that the first point reached by the front on  $\partial\Sigma$  belongs to  $\partial\Sigma \cap \mathcal{S}$ . This point is easy to detect, because it is the global minimum of  $\mathcal{U}|_{\partial\Sigma}$  and is noted  $\mathbf{x}_{\min}$ . In a more general manner, each local minimum  $\mathbf{x}_m$  of  $\mathcal{U}|_{\partial\Sigma}$  has been reached by the front before all points in a small neighborhood of  $\mathbf{x}_m$ . Since, the wavefront propagates faster along  $\mathcal{S}$ , one can expect that the curve  $\partial\Sigma \cap \mathcal{S}$  corresponds to *valley lines* on  $\mathcal{U}|_{\partial\Sigma}$ . For valley line detection, our approach is simple and fast. Using the function  $\mathcal{U}|_{\partial\Sigma}$  and without parametrizing the surface  $\partial\Sigma$ , we find frontier  $\Gamma$  of the surface patch  $\mathcal{S} \cap \partial\Sigma$  by looking for the cyclic sequences of the valley lines of  $\mathcal{U}|_{\partial\Sigma}$  containing  $\mathbf{x}_{\min}$ .

**Finding Valley Lines of  $\mathcal{U}|_{\partial\Sigma}$**

As explained above, valley lines of  $\mathcal{U}|_{\partial\Sigma}$  contain the local minima  $\mathbf{x}_m$  as well as the saddle points. A robust way to link two local minima is to detect the saddle point between them and to make a double gradient descent to each minimum. The difficulty here is that some local minima and saddle points of  $\mathcal{U}|_{\partial\Sigma}$  do not

belong to the curve of interest. To avoid this, saddle points of  $\mathcal{U}|_{\partial\Sigma}$  are detected by increasing order. During this step, we store the information on a graph  $\mathcal{G}$  such that vertices of the graph correspond to local minima of  $\mathcal{U}|_{\partial\Sigma}$ , and an edge corresponds to a pair of valley lines joining two local minima via a saddle point. The valley line detection algorithm stops when a cycle (in the sense of a simple closed path) is detected in the graph  $\mathcal{G}$ .

However, the closed curve  $\Gamma$  tends to have low length, linking between close local minima. In practice, one adds two *ad hoc* constraints which make it possible to extract the border of the surface patch in a more robust manner. The algorithm stops as soon as the global minimum of  $\mathcal{U}|_{\partial\Sigma}$ ,  $\mathbf{x}_{\min}$ , belongs to the closed sequence  $\ell$ , and the subset of  $\text{int}(\Sigma)$  defined by :

$$\mathcal{U}_{|\text{int}(\Sigma)}^{-1}(\left] \max_{\mathbf{x}_\ell \in \ell} \{\mathcal{U}(\mathbf{x}_\ell)\}, +\infty \right]) = \{\mathbf{x} \in \text{int}(\Sigma); \mathcal{U}(\mathbf{x}) > \max_{\mathbf{x}_\ell \in \ell} \{\mathcal{U}(\mathbf{x}_\ell)\}\}$$

includes exactly two connected components for the 26-connectivity, which means that the sequence  $\ell$  cuts the boundary  $\partial\Sigma$  into exactly two connected components (see figure 4).

### 4 Dense Network of Minimal Paths: An Implicit Approach

Once the boundary curve  $\Gamma$  is obtained, it is easy to construct explicitly a network of minimal paths linking points of  $\Gamma$  to the source point  $\mathbf{p}$  by simple gradient descents as in [13]. The network linking  $\Gamma$  to  $\mathbf{p}$  is noted  $\mathcal{N}_\mathbf{p}^\Gamma = \bigcup_{\mathbf{x}_\Gamma \in \Gamma} \mathcal{C}_{\mathbf{x}_\Gamma, \mathbf{p}}$ .

Since this network may have holes, our objective is to find a smooth function  $\Psi : \Sigma \rightarrow \mathbb{R}$ , such that the network  $\mathcal{N}_\mathbf{p}^\Gamma$  is included in the zero level set of  $\Psi$ , i.e  $\mathcal{N}_\mathbf{p}^\Gamma \subset \Psi^{-1}(\{0\})$ , where  $\Psi^{-1}(\{0\}) = \{\mathbf{x} \in \Sigma; \Psi(\mathbf{x}) = 0\}$ . A necessary condition on function  $\Psi$  is

$$\nabla\Psi(\mathbf{x}) \cdot \nabla\mathcal{U}(\mathbf{x}) = 0, \tag{6}$$

for each point  $\mathbf{x}$  belonging to a path  $\mathcal{C}_{\mathbf{x}_\Gamma, \mathbf{p}}$ . Thus, vector  $\nabla\Psi$  is perpendicular to  $\nabla\mathcal{U}$  along the minimal paths of the network  $\mathcal{N}_\mathbf{p}^\Gamma$ . Extending the constraint given by equation (6) to the whole domain  $\Sigma$  gives a sufficient condition on  $\Psi$ . Moreover, adding a linear term on  $\Psi$  smoothes the solution without changing the zero level set of  $\Psi$ .

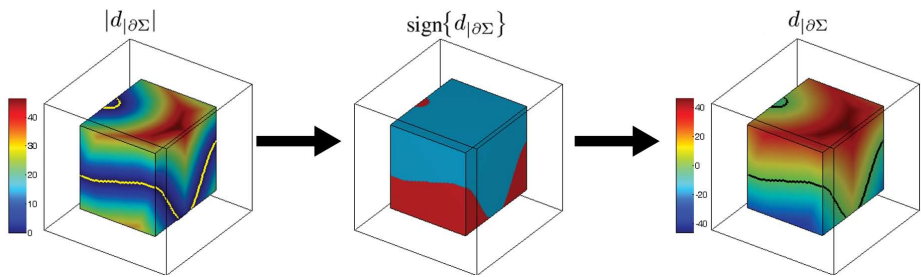
Hence, if  $\Psi$  is a smooth function satisfying the following conditions:

$$\begin{cases} (C_1) \ \forall \mathbf{x} \in \Sigma, \nabla\Psi(\mathbf{x}) \cdot \nabla\mathcal{U}(\mathbf{x}) - \alpha \Psi(\mathbf{x}) = 0, \\ (C_2) \ \forall \mathbf{x} \in \Gamma, \Psi(\mathbf{x}) = 0, \end{cases} \tag{7}$$

where  $\alpha \geq 0$ , then  $\mathcal{N}_\mathbf{p}^\Gamma \subset \Psi^{-1}(\{0\})$ . Finally,  $\Psi^{-1}(\{0\})$  is a dense network of minimal paths. Indeed, if  $\Psi$  satisfies conditions  $(C_1)$  and  $(C_2)$ , then  $\forall \mathbf{x} \in \Psi^{-1}(\{0\})$ , the minimal path  $\mathcal{C}_{\mathbf{x}, \mathbf{p}}$  linking  $\mathbf{x}$  to the source  $\mathbf{p}$  is included in  $\Psi^{-1}(\{0\})$ . Detailed proof of these results can be found in [3]. Using conditions  $(C_1)$  and  $(C_2)$ , we look for a solution  $\Psi$  of the following Dirichlet problem:

$$\begin{cases} \nabla\Psi(\mathbf{x}) \cdot \nabla\mathcal{U}(\mathbf{x}) - \alpha \Psi(\mathbf{x}) = 0 & \text{if } \mathbf{x} \in \text{int}(\Sigma), \\ \Psi(\mathbf{x}) = d_{|\partial\Sigma}(\mathbf{x}) & \text{if } \mathbf{x} \in \partial\Sigma, \end{cases} \quad (8)$$

where  $d_{|\partial\Sigma}$  is a signed Euclidean distance to  $\Gamma$  on  $\partial\Sigma$ . Indeed, that makes the function  $\Psi$  satisfying the second condition ( $C_2$ ). One can propose other boundary conditions satisfying ( $C_2$ ), but empirically, we found that the signed distance is an adequate choice. Since  $\Gamma$  is a simple closed curve on  $\partial\Sigma$  and  $\partial\Sigma$  is topologically equivalent to a sphere,  $\Gamma$  partitions  $\partial\Sigma$  into two distinct open surfaces. That makes the sign choice for  $d_{|\partial\Sigma}$  obvious. First, the unsigned distance from  $\Gamma$  on  $\partial\Sigma$  is calculated using the Fast Marching algorithm (this time using 26-connectivity), then different signs are attributed to the distance on each connected component of  $\partial\Sigma \setminus \Gamma$  of the partition (see figure 5).



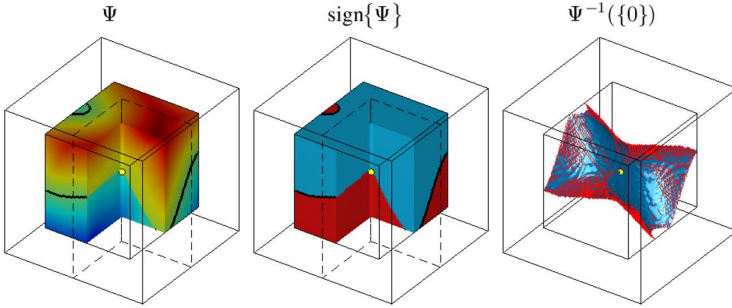
**Fig. 5.** Transport initialization. First, the distance map from the curve  $\Gamma$  is computed. Then using  $\Gamma$ ,  $\partial\Sigma \setminus \Gamma$  is partitioned into exactly two parts. Finally, different signs are attributed to  $d_{|\partial\Sigma}$  on each connected component.

Equation (8) is a stationary transport equation. The associated non stationary PDE models the transport in time and space of material along the vector field  $\nabla\mathcal{U}$ . The stationary transport equation has been studied [3] for surface segmentation, for computing tissue thickness [14] and inpainting [15].

The stationary transport equation (8), as most PDEs for which characteristics intersect are numerically hard to solve. Nevertheless, the direction on which information propagates is known ( $-\nabla\mathcal{U}$ ) thus one can elaborate a single pass algorithm based on an ordered sweeping of the grid points [3, 14, 15]. We propose to find values of  $\Psi$  by exploring points of  $\Sigma$  in decreasing order of  $|\Psi|$ . The algorithm, called *Fast Transport* is similar to the Fast Marching algorithm : only the ordering is different as well as the local update scheme. The complexity of the *Fast Transport* algorithm is  $\mathcal{O}(N \log(N))$ .

The information propagates from  $\partial\Sigma$  to the source point  $\mathbf{p}$  following the direction  $-\nabla\mathcal{U}$ . Thus, it is important to use an upwind scheme that takes into account the direction  $-\nabla\mathcal{U}$  to approximate the derivatives of  $\Psi$ . Let us note  $\Psi_{i,j,k}$  the value of  $\Psi$  at point  $\mathbf{x}$  of coordinate  $(ih_x, jh_y, kh_z)$ ,  $\partial_d\Psi_{i,j,k}$  the derivative of  $\Psi$  along direction  $d$  ( $d$  corresponds to  $x, y$  or  $z$ -direction) and  $\partial_d\mathcal{U}_{i,j,k}$  the derivative of  $\mathcal{U}$  along direction  $d$ . If  $\partial_d\mathcal{U}_{i,j,k} < 0$ , the information is transported increasingly along  $d$  direction. Thus along the  $x$  direction we have:





**Fig. 6.** On the left and on the middle are respectively shown, on a cut view, the function  $\Psi$  and its sign on  $\Sigma$ . On the right is shown the extracted surface pathc, i.e. the isosurface  $\Psi^{-1}(\{0\})$ , as well as the network of minimal paths  $\mathcal{N}_{\mathbf{p}}^T$ .

$$\partial_x \Psi_{i,j,k} = \begin{cases} \frac{\Psi_{i+1,j,k} - \Psi_{i,j,k}}{h_x} & \text{if } \partial_x \mathcal{U}_{i,j,k} \geq 0, \\ \frac{\Psi_{i,j,k} - \Psi_{i-1,j,k}}{h_x} & \text{if } \partial_x \mathcal{U}_{i,j,k} < 0. \end{cases}$$

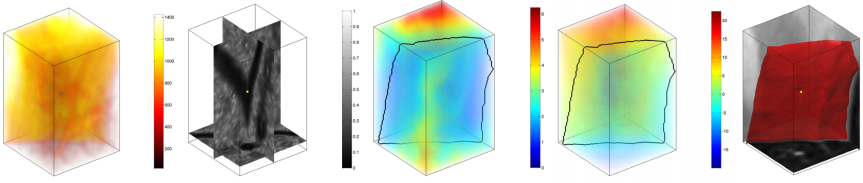
The derivatives along  $y$  and  $z$  direction are similar. The update scheme of the *Fast Transport* algorithm is based on the previous equation, by injecting it in equation (8), see [3] for more details. Lastly, although this scheme is of relatively low precision and dissipative, it gives satisfactory results in our experiments with an acceptable convergence speed. In our implementation  $\alpha$  is a parameter that can be fixed through the maximum discontinuity jump of  $\Psi$  around the source  $\mathbf{p}$ . Indeed, by considering the minimal path  $\mathcal{C}_{\mathbf{x},\mathbf{p}}$ , linking a point  $\mathbf{x} \in \partial\Sigma$  to  $\mathbf{p}$ , parametrized on the interval  $J = [0, \mathcal{L}(\mathbf{x})]$ , where  $\mathcal{L}(\mathbf{x})$  is the Euclidean length of the path, one can prove using equation (8) that

$$\forall s \in J, \Psi(\mathcal{C}_{\mathbf{x},\mathbf{p}}(s)) = d_{|\partial\Sigma}(\mathbf{x}) e^{-\alpha s}.$$

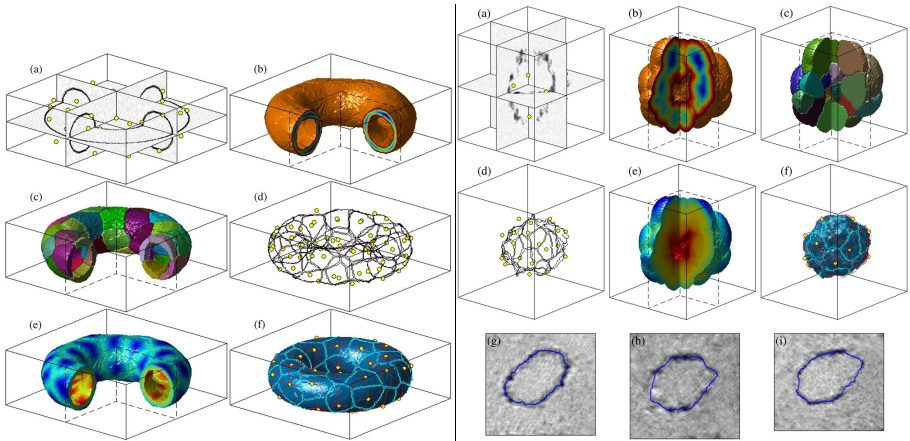
Thus the discontinuity jump occurs around the source point  $\mathbf{p}$  and is as high as  $|d_{|\partial\Sigma}(\mathbf{x})|e^{-\alpha\mathcal{L}(\mathbf{x})}$ . Fixing a maximum discontinuity jump  $\varepsilon$  and

$$\alpha = \frac{\log\left(\max_{\mathbf{x} \in \partial\Sigma} |d_{|\partial\Sigma}(\mathbf{x})|\right) - \log(\varepsilon)}{\min_{\mathbf{x} \in \partial\Sigma} \mathcal{L}(\mathbf{x})},$$

guaranty that the discontinuity jump around the source point  $\mathbf{p}$  is less or equal than  $\varepsilon$ . Imposing this constraint requires the computation of the Euclidean length  $\mathcal{L}$  of the minimal paths. This calculus can be easily done during the Fast Marching propagation as explained in [16, 17]. On figure 6, function  $\Psi$  solution of equation (8), the final segmentation result  $\Psi^{-1}(\{0\})$  as well as the network of minimal paths are shown.



**Fig. 7.** We select a sub-volume from a CT cardiac image. Then an edge detector potential, inversely proportional to the gradient magnitude of the image  $\|\nabla I\|$  is shown. The Fast Marching algorithm is launched from the selected source point to compute the minimal action map  $\mathcal{U}$ . Then the valley line of  $\mathcal{U}$  is calculated. Finally the information is transported from the initialized values of the sub-volume boundary using the fast transport algorithm, and the segmentation result of this patch of surface is found using the marching cube algorithm on the solution of the transport equation.



**Fig. 8.** On the left: segmentation of a synthetic torus. On the right: segmentation of a closed cell from electronic microscopy image. (a) Potential  $\mathcal{P}$  taking lower values on the features of interest on which a single source point is selected. The other points are found automatically using the approach presented in [17]. (b) A cut view of the visited domain  $\Omega^*$  showing the value of the minimal action map  $\mathcal{U}$ . (c) A Cut view of the domain  $\Omega^*$  showing the Voronoi partition. (d) The set of sources and the valley lines detected on each Voronoi cell. (e) A cut view of the domain  $\Omega^*$  showing values of function  $\Psi$  solution of the transport equation (8). (f) Isosurface  $\Psi^{-1}(\{0\})$  on which the detected keypoint points, the valley lines and the geodesic meshing are superimposed. On the right: (g-h-i) Some slices of the original image and the final segmentation  $\Psi^{-1}(\{0\})$  superimposed on it.

## 5 Experimental Results

Using our method, one can extract a surface patch from a single point, see figure 7. The main advantages of our method is that it is minimally interactive

and fast. The important constraint is that the boundary of the selected sub-volume intersects the surface on a single closed curve. One can imagine that by considering a subdivision of the whole domain, and by selection of a few points on the sub-domains that contains the surface of interest, one can extract a full segmentation of the desired object.

Recently, we presented [17] a new method for segmenting closed contours and surfaces. Our work builds on a variant of the minimal path approach. First, an initial point on the desired contour is chosen by the user. Next, new keypoints are detected automatically using a front propagation approach. We assume that the desired object has a closed boundary. This a-priori knowledge on the topology is used to devise a relevant criterion for stopping the keypoint detection and front propagation. The final domain visited by the front will yield a band surrounding the object of interest. Using this method for 3D closed objects, we can extract a networks of minimal paths from a 3D image called Geodesic Meshing. But this segmentation is insufficient. The Voronoi partition of the visited domain gives a good subdivision of it, and by applying the algorithm presented in this paper on each Voronoi cell, one can find a full segmentation of the object of interest, see figure 8.

## 6 Conclusion

In this paper we have proposed a new method to segment a surface patch from a single source point. Our method needs minimal interaction : a single source point. An important condition is that the boundary of the sub-volume that contains the surface patch of interest should intersects the surface on a single closed curve. By remarking that this closed curve corresponds to the valley line of the arrival time from the source point we have proposed a heuristic to extract it automatically. Finally we adapted an existing implicit surface segmentation method to find a complete surface that contains the valley line and the network of minimal paths linking this valley line to the source point. Our approach can be extended to segment a complete surface by subdividing the domain into several sub-domains containing the desired surface patches. Then, a few points can be enough to generate a coherent object boundary segmentation.

## Acknowledgements

We would like to thank Stéphane Bonneau for his contributions, and Professor Anthony J. Yezzi for interesting discussions. This work was partially supported by ANR grant SURF -NT05-2\_45825.

## References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. *International Journal of Computer Vision* 1, 321–331 (1988)
2. Cohen, L.D., Kimmel, R.: Global minimum for active contour models: a minimal path approach. *International Journal of Computer Vision* 24, 57–78 (1997)

3. Ardon, R., Cohen, L.D., Yezzi, A.: Fast surface segmentation guided by user input implicit extension of minimal paths. *Journal of Mathematical Imaging and Vision* 25, 289–305 (2006)
4. Caselles, V., Kimmel, R., Sapiro, G., Sbert, C.: Minimal surfaces based object segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, 394–398 (1997)
5. Cohen, L.D., Cohen, I.: Finite element methods for active contour models and balloons for 2D and 3D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15, 1131–1147 (1993)
6. Sethian, J.A.: *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge (1999)
7. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control* 40, 1528–1538 (1995)
8. Dijkstra, E.W.: A note on two problems in connection with graphs. *Numerische Mathematica* 1, 269–271 (1959)
9. Rouy, E., Tourin, A.: A viscosity solution approach to shape from shading. *SIAM Journal on Numerical Analysis* 29, 867–884 (1992)
10. López, A., Lloret, D.: On ridges and valleys. In: *ICPR 2000: Proceedings of the International Conference on Pattern Recognition*, Washington, DC, USA, p. 4059. IEEE Computer Society, Los Alamitos (2000)
11. Tang, C.K., Medioni, G.G.: Extremal feature extraction from 3-D vector and noisy scalar fields. In: *IEEE Visualization 1998*, October 1998, pp. 95–102 (1998)
12. Koenderink, J.: *Solide Shape*. MIT Press, Cambridge (1990)
13. Ardon, R., Cohen, L.D.: Fast constrained surface extraction by minimal paths. *Int. J. Comput. Vision* 69(1), 127–136 (2006)
14. Yezzi, A., Prince, J.L.: An Eulerian PDE Approach for Computing Tissue Thickness. *IEEE Transactions On Medical Imaging* 22, 1332–1339 (2003)
15. Bornemann, F., Marz, T.: Fast image inpainting based on coherence transport. *JMIV* 28(3), 259–278 (2007)
16. Cohen, L.D., Deschamps, T.: Segmentation of 3D tubular objects with adaptive front propagation and minimal tree extraction for 3D medical imaging. *Computer Methods in Biomechanics and Biomedical Engineering* 10, 289–305 (2007)
17. Benmansour, F., Cohen, L.D.: Fast object segmentation by growing minimal paths from a single point on 2D or 3D images. *Journal of Mathematical Imaging and Vision* 33(2), 209–221 (2009)