Finding a Closed Boundary by Growing Minimal Paths from a Single Point

Fethallah Benmansour & Stéphane Bonneau & Laurent Cohen
CEREMADE, Place du Marchal De Lattre De Tassigny 75775 PARIS CEDEX 16 - FRANCE
`benmansour,bonneau,cohen @ ceremade.dauphine.fr`

In this paper, we present a new method for segmenting closed contours. Our work builds on a variant of the Fast Marching algorithm. 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. Linking pairs of neighboring keypoints with minimal paths allows us to extract a closed contour from a 2D image. Detection of a variety of objects on real images is demonstrated.

1 Introduction

Energy minimization techniques have been applied to a broad variety of problems in image processing and computer vision. Since the original work on snakes (Kass, Witkin, and Terzopoulos 1988), 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. For instance, the geodesic active contour model (Caselles, Kimmel, and Sapiro 1997)relies on the minimization of a geometric energy functional that deforms an initial curve toward local geodesics in a Riemannian metric derived from the image. Whereas the geodesic active contour model presents significant improvements compared to the original snake model, the energy minimization process is still prone to local minima. Consequently, results strongly depend on the model initialization.

To avoid local minima, Cohen and Kimmel (Cohen and Kimmel 1997) 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 γ 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. We can compute simultaneously, and efficiently, the minimal action map and its Euclidean path length with the *Fast Marching Method* as will be detailed in section 2.2.

In section 3, we introduce a novel segmentation approach, based on the Fast Marching Method, to distribute a set of points on a closed curve that is not known a priori. We only assume the user provides a single point (or more if desired) initialized on the desired object boundary. Each newly detected keypoint is immediately defined as a new source of propagation, and keypoints are detected with a criterion based on the Euclidean length of the minimal paths. Since the front propagates faster on the object boundary, the first point for which the length λ is reached, is located in this area (of small values of $\tilde{\mathcal{P}}$) and is a valuable choice as a new keypoint. By using the a-priori knowledge on the topology of the manifold, we devise a relevant criterion for stopping the keypoint detection and front propagation. The criterion is general for any dimension.

In section 4, we explain how to extract a boundary curve using the previous results. The main idea is to link pairs of neighboring keypoints with minimal paths via gradient descent on the minimal action map. Segmentation results on a set of 2D images are presented in section 5. Finally conclusions and perspectives follow in section 6.

2 Background on minimal paths

2.1 Definitions

Given a 2D image $I : \Omega \rightarrow \mathbb{R}^+$ and two points $\mathbf{p}_1$ and $\mathbf{p}_2$, the underlying idea introduced in (Cohen and Kimmel 1997) 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 edges by finding a curve that globally minimizes the energy functional $E : \mathcal{A}_{\mathbf{p}_1,\mathbf{p}_2} \rightarrow \mathbb{R}^+$

$$E(\gamma) = \int_\gamma \Big\{ \mathcal{P}\big(\gamma(s) + w\big) \Big\} \mathrm{d}s = \int_\gamma \tilde{\mathcal{P}}\big(\gamma(s)\big) \mathrm{d}s, \quad (1)$$

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 Ω :

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

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

$$\begin{cases} \|\nabla \mathcal{U}_1(\mathbf{x})\| &= \tilde{\mathcal{P}}(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega, \\ \mathcal{U}_1(\mathbf{p}_1) &= 0. \end{cases} \quad (3)$$

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 $\mathbf{p}_1$ (see Fig. 1), solving the following ordinary differential equation with standard numerical methods like Heun's or Runge-Kutta's :

$$\begin{cases} \dfrac{\mathrm{d}\mathcal{C}_{\mathbf{p}_1,\mathbf{p}_2}(s)}{\mathrm{d}s} &= -\nabla \mathcal{U}_1\big(\mathcal{C}_{\mathbf{p}_1,\mathbf{p}_2}(s)\big), \\ \mathcal{C}_{\mathbf{p}_1,\mathbf{p}_2}(0) &= \mathbf{p}_2. \end{cases} \quad (4)$$

Let us extend the definitions given so far to the case of multiple sources and introduce other definitions which will be useful hereinafter. These definitions hold in dimension 2 and higher. The *minimal action map* associated to the potential $\tilde{\mathcal{P}} : \Omega \rightarrow \mathbb{R}^{*+}$

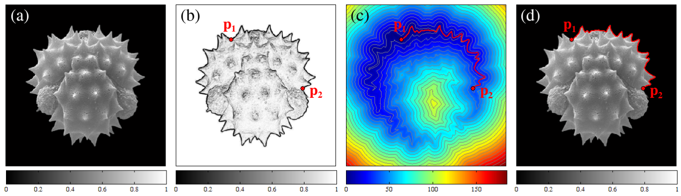

Figure 1: Extraction of an open contour from an electron microscopy image. (**a**) Original image I. (**b**) Potential $\mathcal{P} = (\|\nabla I\| + \varepsilon)^{-3}$, where ε is a small positive constant, and user-supplied points $\mathbf{p}_1$ and $\mathbf{p}_2$. (**c**) Minimal action map $\mathcal{U}_1$ and minimal path $\mathcal{C}_{\mathbf{p}_1,\mathbf{p}_2}$ between $\mathbf{p}_1$ and $\mathbf{p}_2$. (**d**) Image I and minimal path $\mathcal{C}_{\mathbf{p}_1,\mathbf{p}_2}$.

and the set of n sources $\mathcal{S} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ is the function $\mathcal{U} : \Omega \rightarrow \mathbb{R}^+$ defined by

$$\forall\, \mathbf{x} \in \Omega, \quad \mathcal{U}(\mathbf{x}) = \min_{1 \le j \le n} \{\mathcal{U}_j(\mathbf{x})\},$$

$$\text{where} \quad \mathcal{U}_j(\mathbf{x}) = \min_{\gamma \in \mathcal{A}_{\mathbf{p}_j,\mathbf{x}}} \left\{ \int_\gamma \tilde{\mathcal{P}}\big(\gamma(s)\big) \mathrm{d}s \right\}. \quad (5)$$

The map $\mathcal{U}$ is a weighted distance map to the set of sources $\mathcal{S}$, and it satisfies the Eikonal equation

$$\begin{cases} \|\nabla \mathcal{U}(\mathbf{x})\| &= \tilde{\mathcal{P}}(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega, \\ \mathcal{U}(\mathbf{p}_j) &= 0 \quad \text{for } \mathbf{p}_j \in \mathcal{S}. \end{cases} \quad (6)$$

The *Voronoi region* associated to the source $\mathbf{p}_j \in \mathcal{S}$, noted $\mathcal{R}_j$, is the locus of points of the domain Ω which are closer (in the sense of a weighted distance) to $\mathbf{p}_j$ than to any other source of $\mathcal{S}$. The region $\mathcal{R}_j$ is a connected subset of the domain Ω, and its boundary is noted $\partial\mathcal{R}_j$. The union of Voronoi regions and its complementary set, the *Voronoi diagram*, leads to a tessellation of the domain Ω, called the *Voronoi partition*. The *Voronoi index map* is the function $\mathcal{V} : \Omega \rightarrow \{1, \ldots, n\}$ that assigns to any point of the domain Ω the index of its Voronoi region :

$$\forall\, \mathbf{x} \in \mathcal{R}_j,\ \mathcal{V}(\mathbf{x}) = j. \quad (7)$$

If two Voronoi regions $\mathcal{R}_i$ and $\mathcal{R}_j$ are adjacent (i.e. if $\partial\mathcal{R}_i \cap \partial\mathcal{R}_j$ is a non-empty set), then the minimal path $\mathcal{C}_{\mathbf{p}_i,\mathbf{p}_j}$ passes through the point of $\partial\mathcal{R}_i \cap \partial\mathcal{R}_j$ which has the smallest $\mathcal{U}$ value. This point, noted $\mathbf{m}_{i|j}$, is the *midpoint* of the minimal path $\mathcal{C}_{\mathbf{p}_i,\mathbf{p}_j}$ since it is equidistant to $\mathbf{p}_i$ and $\mathbf{p}_j$ in the sense of a weighted distance. This is a saddle point of $\mathcal{U}$.

The *Euclidean path length map* is the function $\mathcal{L} : \Omega \rightarrow \mathbb{R}^+$ that assigns to any point $\mathbf{x}$ of the domain Ω the Euclidean length of the minimal path between $\mathbf{x}$ and the source which is the closest in the sense of a weighted distance :

$$\forall\, \mathbf{x} \in \mathcal{R}_j,\ \mathcal{L}(\mathbf{x}) = \int_{\mathcal{C}_{\mathbf{p}_j,\mathbf{x}}} \mathrm{d}s. \quad (8)$$

Table 1 : *Fast Marching Method for solving equation (6).*

- **Notation.**
  $\mathcal{N}_{\mathrm{M}}(\mathbf{x})$ is the set of M neighbors of a grid point $\mathbf{x}$, where $\mathrm{M} = 4$ in 2D and $\mathrm{M} = 6$ in 3D.
- **Initialization.**
  For each grid point $\mathbf{x}$, do
    Set $\mathcal{U}(\mathbf{x}) := +\infty$, $\mathcal{V}(\mathbf{x}) := 0$ and $\mathcal{L}(\mathbf{x}) := +\infty$.
    Tag $\mathbf{x}$ as *Far*.
  For each source $\mathbf{p}_j \in \mathcal{S}$, do
    Set $\mathcal{U}(\mathbf{p}_j) := 0$, $\mathcal{V}(\mathbf{p}_j) := j$ and $\mathcal{L}(\mathbf{p}_j) := 0$.
    Tag $\mathbf{p}_j$ as *Trial*.
- **Marching loop**.
  While the set of *Trial* points is non-empty, do
    Find $\mathbf{x}_{\min}$, a *Trial* point with the smallest $\mathcal{U}$ value.
    Tag $\mathbf{x}_{\min}$ as *Alive*.
    For each point $\mathbf{x}_{\mathrm{n}} \in \mathcal{N}_{\mathrm{M}}(\mathbf{x}_{\min})$ which is not *Alive*, do
      $\{u, v, \ell\}$:=`UpdateSchemeFMM` $\left(\mathbf{x}_{\mathrm{n}}, \mathcal{N}_{\mathrm{M}}(\mathbf{x}_{\mathrm{n}})\right)$.
      Set $\mathcal{U}(\mathbf{x}_{\mathrm{n}}) := u$, $\mathcal{V}(\mathbf{x}_{\mathrm{n}}) := v$ and $\mathcal{L}(\mathbf{x}_{\mathrm{n}}) := \ell$.
      If $\mathbf{x}_{\mathrm{n}}$ is *Far*, tag $\mathbf{x}_{\mathrm{n}}$ as *Trial*.

Note that if $\tilde{\mathcal{P}}(\mathbf{x}) = 1$ for all $\mathbf{x} \in \Omega$, then the maps $\mathcal{U}$ and $\mathcal{L}$ are equal and both correspond to the Euclidean distance map to the set of sources $\mathcal{S}$.

2.2 Fast Marching Method

The *Fast Marching Method* (FMM) is a numerical method introduced in (Sethian 1999b; Sethian 1999a; Sethian 1996) and (Tsitsiklis 1995) for efficiently solving the isotropic Eikonal equation on a cartesian grid. In equation (6), the values of $\mathcal{U}$ may be regarded as the arrival times of wavefronts propagating from each point of $\mathcal{S}$ with velocity $(1/\tilde{\mathcal{P}})$. The central idea behind the FMM is to visit grid points in an order consistent with the way wavefronts propagate, i.e. with the Huygens principle. It leads to a single-pass algorithm for solving equation (6) and computing the maps $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{L}$ in a common computational framework (see Table 1). 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 (Dijkstra 1959). 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 (see Table 1). 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.

Outputs of the routine `UpdateSchemeFMM` in Table 1 are estimated using a correct first order accurate scheme, for equation 6, given by Rouy an Tourin in (Rouy and Tourin 1992). The scheme is an upwind scheme : the forward and backward differences are chosen to follow the direction of the flow of information. The Euclidian length ℓ is computed in the same manner as the minimal action map by solving the equation $\|\nabla\mathcal{L}\| = 1$ by using the same neighbors as used to solve 6 (see (Deschamps and Cohen 2002)).

3 Distribution of a set of points on a closed curve

First, we consider the case where the domain Ω is a 2D domain. We assume that we are given an initial set $\mathcal{S} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ of points on a closed curve along which a potential $\tilde{\mathcal{P}} : \Omega \to \mathbb{R}^{*+}$ takes lower values. Note that the set $\mathcal{S}$ may contain only one point.

We propose here a variant of the FMM, called the *Fast Marching Method With keypoint Detection* (FMMWKD, see Table 2), to propagate fronts from each point of $\mathcal{S}$ with velocity $(1/\tilde{\mathcal{P}})$ and sequentially detect, during the front propagation, a set of keypoints $\mathcal{S}^* = \{\mathbf{p}^*_{n+1}, \cdots, \mathbf{p}^*_{n+m}\}$ on the closed curve along which $\tilde{\mathcal{P}}$ takes low values. Each newly detected keypoint is immediately defined as a new source of propagation, and keypoints are detected with a criterion based on the Euclidean length of minimal paths. This criterion depends on only one parameter, denoted λ. Front propagation and keypoint detection ceases as soon as the domain visited by the fronts contains the whole curve of interest.

The final domain visited by the fronts, denoted Ω_{F}, correspond to a band surrounding the curve of interest. Furthermore, the FMMWKD also enables the computation of the minimal action map $\mathcal{U} : \Omega_{\mathrm{F}} \to \mathbb{R}^+$, the Voronoi index map $\mathcal{V} : \Omega_{\mathrm{F}} \to \{1, \ldots, n+m\}$ and the Euclidean path length map $\mathcal{L} : \Omega_{\mathrm{F}} \to \mathbb{R}^+$ associated to the potential $\tilde{\mathcal{P}}$ and the set of sources $\mathcal{S} \cup \mathcal{S}^*$.

3.1 Keypoint detection and local correction of maps $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{L}$.

Initially, fronts are propagated from each point of $\mathcal{S}$ with velocity $(1/\tilde{\mathcal{P}})$, until a grid point $\mathbf{x}$ such that $\mathcal{L}(\mathbf{x}) \geq \lambda$ is tagged as *Alive*. This point is then defined as the first keypoint, denoted $\mathbf{p}^*_{n+1}$ (see Fig. 2). Such a criterion has already been used in (Deschamps and Cohen 2001) to find a minimal path given only one endpoint and also to adapt front propagation for segmentation of tubular shapes (Deschamps and Cohen 2002). Assuming that the point $\mathbf{p}^*_{n+1}$ belongs to the Voronoi region $\mathcal{R}_j$ when it is detected, this criterion ensures that the minimal path $\mathcal{C}_{\mathbf{p}_j, \mathbf{p}^*_{n+1}}$ minimizes the integral of $\tilde{\mathcal{P}}$ (along itself) over all open curves with Euclidean lengths greater than or equal to λ and with endpoints in $\mathcal{S}$. Therefore, $\mathbf{p}^*_{n+1}$ is likely to belong to the curve along which the values of $\tilde{\mathcal{P}}$ are low.

Once the first keypoint has been detected, it is defined as a new source of propagation. It is unnecessary to restart the overall algorithm since values of $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{L}$ which have already been estimated would not differ in the vicinity of initial sources (i.e. in the vicinity of points of $\mathcal{S}$). In order to limit the computational cost, one just needs to update $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{L}$ in the following manner :

$$\mathcal{U}(\mathbf{p}^*_{n+1}) := 0, \quad \mathcal{V}(\mathbf{p}^*_{n+1}) := n+1, \quad \mathcal{L}(\mathbf{p}^*_{n+1}) := 0,$$

tag $\mathbf{p}^*_{n+1}$ as *Trial* and continue front propagation. However, without any additional modification of the original FMM, final values of $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{L}$ would be incorrect for grid points which are tagged as *Alive* when $\mathbf{p}^*_{n+1}$ is detected and closer (in the sense of a weighted distance) to $\mathbf{p}^*_{n+1}$ than to the initial sources. These errors would be solely due to the fact that, in the original FMM, values of $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{L}$ are frozen for *Alive* points. An easy way to avoid this problem is just to let an *Alive* point be tagged as *Trial* again if it is closer to the new source of propagation than to initial sources. This algorithmic trick enables the local correction of $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{L}$ in the neighborhood of $\mathbf{p}^*_{n+1}$.

Next, front propagation is continued until a grid point $\mathbf{x}$ such that $\mathcal{L}(\mathbf{x}) \geq \lambda$ is tagged as *Alive*. This point is defined as the second keypoint, denoted $\mathbf{p}^*_{n+2}$, and is added to the set of sources. Afterward, front propagation is continued, and so on. Thus, during the front propagation, keypoints are sequentially detected on the curve along which $\tilde{\mathcal{P}}$ takes low values (see Fig. 2).

3.2 Stopping criterion for keypoint detection and front propagation.

In order to prevent the algorithm from distributing keypoints over the whole domain Ω, one needs to stop the keypoint detection as soon as the domain visited by the fronts contains the curve of interest. Note that even if this curve is unknown, we assume that it is closed. This topological assumption is used to devise a relevant criterion for stopping keypoint detection and front propagation.

One possible strategy is to take into account the Voronoi partition, and to stop keypoint detection as soon as each Voronoi region is adjacent to at least two other Voronoi regions (i.e. as soon as there exists a cycle of Voronoi regions). This strategy, although correct, is limited to the 2D case. To get a scheme which may be extended to higher dimensions, another strategy is employed in the FMMWKD. Let us denote by Ω_{F} the domain visited by the propagating fronts, defined as the set of grid points which are not *Far* (i.e. the set of grid points which are either *Alive* or *Trial*).

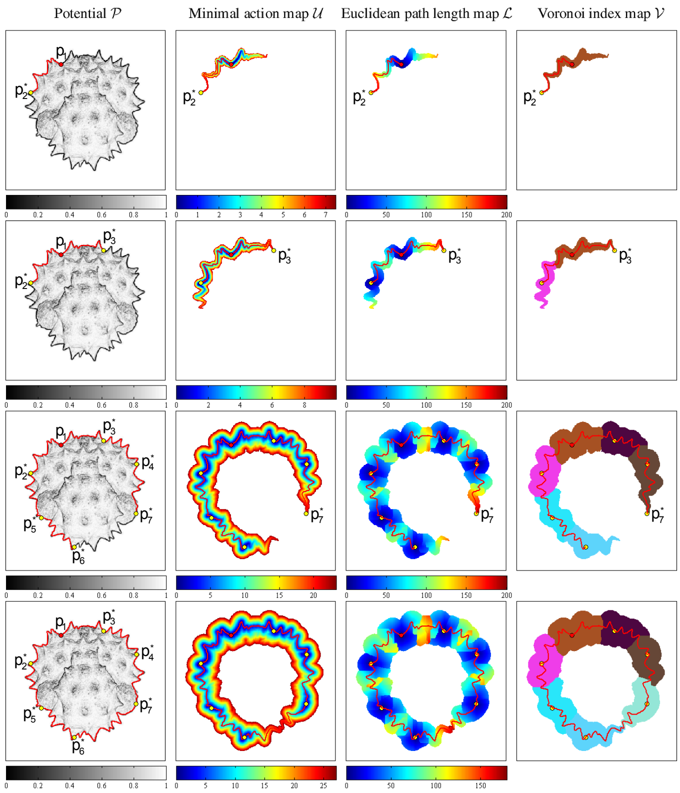

Figure 2: Intermediate and final results for the FMMWKD applied to the 2D potential of the Figure 2.**b**, with $\mathcal{S} = \{\mathbf{p}_1\}$ and $\lambda = 200$. The first, second and third rows show intermediate results obtained when are detected, respectively, $\mathbf{p}^*_2$ (the first keypoint), $\mathbf{p}^*_3$ (the second keypoint) and $\mathbf{p}^*_7$ (the last keypoint). The last row shows final results.

In the FMMWKD, keypoint detection is stopped as soon as Ω_{F} becomes a simply connected subset of Ω delimited by exactly two simply connected boundaries.

The set Ω_{F} may be divided into two subsets : the set of interior points, denoted $\mathrm{int}(\Omega_{\mathrm{F}})$, and the set of boundary points, denoted $\partial\Omega_{\mathrm{F}}$. In the original FMM, $\mathrm{int}(\Omega_{\mathrm{F}})$ and $\partial\Omega_{\mathrm{F}}$ respectively correspond to the set of *Alive* points and the set of *Trial* points. This is no longer true in the FMMWKD because of the local correction of $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{L}$ in the neighborhood of a keypoint. That is why a second labelling is introduced in the FMMWKD : each grid point which is not *Far*, in addition to being tagged as *Alive* or *Trial*, is also tagged as *Interior* or *Boundary* depending on whether it belongs to $\mathrm{int}(\Omega_{\mathrm{F}})$ or $\partial\Omega_{\mathrm{F}}$. Noting that the iteration of the marching loop at which $\mathrm{int}(\Omega_{\mathrm{F}})$ becomes a simply connected subset of Ω is also the iteration at which the number of simply connected components of $\partial\Omega_{\mathrm{F}}$ increases for the first time, we just need to monitor the topological changes of $\partial\Omega_{\mathrm{F}}$.

In the algorithm detailed in Table 2, the stopping criterion for keypoint detection is satisfied as soon

Table 2 : *FMM With keypoint Detection.*

- **Notation.**
  $\mathcal{N}_{\mathrm{M}}(\mathbf{x})$ is the set of M neighbors of a grid point $\mathbf{x}$,
  where $\mathrm{M} = 4$ in 2D and $\mathrm{M} = 6$ in 3D.
  $\mathcal{N}_{\mathrm{M}^+}(\mathbf{x})$ is the set of M^+ neighbors of a point $\mathbf{x}$,
  where $\mathrm{M}^+ = 8$ in 2D and $\mathrm{M}^+ = 26$ in 3D.
- **Initialization.**
  For each grid point $\mathbf{x}$, do
    Set $\mathcal{U}(\mathbf{x}) := +\infty$, $\mathcal{V}(\mathbf{x}) := 0$ and $\mathcal{L}(\mathbf{x}) := +\infty$.
    Tag $\mathbf{x}$ as *Far*.
  For each source $\mathbf{p}_j \in \mathcal{S}$, do
    Set $\mathcal{U}(\mathbf{p}_j) := 0$, $\mathcal{V}(\mathbf{p}_j) := j$ and $\mathcal{L}(\mathbf{p}_j) := 0$.
    Tag $\mathbf{p}_j$ as *Trial* and as *Boundary*.
  $m := 1$, *StopDetection:=FALSE*.
- **Marching loop**.
  While the set of *Trial* points is non-empty, do
    Find $\mathbf{x}_{\min}$, a *Trial* point with the smallest $\mathcal{U}$ value.
    If $\big($*StopDetection=FALSE*$\big)$ and $\big(\mathcal{L}(\mathbf{x}_{\min}) \geq \lambda\big)$, do
      Here, $\mathbf{x}_{\min}$ *is defined as the keypoint* $\mathbf{p}^*_{n+m}$.
      Set $\mathcal{U}(\mathbf{x}_{\min}) := 0$, $\mathcal{V}(\mathbf{x}_{\min}) := n + m$, $\mathcal{L}(\mathbf{x}_{\min}) := 0$.
      $m := m + 1$.
    Else, do
      Tag $\mathbf{x}_{\min}$ as *Alive*.
      For each grid point $\mathbf{x}_{\mathrm{n}} \in \mathcal{N}_{\mathrm{M}}(\mathbf{x}_{\min})$, do
        If $\mathbf{x}_{\mathrm{n}}$ is not *Alive*, do
          $\{u, v, \ell\}$:=`UpdateSchemeFMM` $\big(\mathbf{x}_{\mathrm{n}}, \mathcal{N}_{\mathrm{M}}(\mathbf{x}_{\mathrm{n}})\big)$
          Set $\mathcal{U}(\mathbf{x}_{\mathrm{n}}) := u$, $\mathcal{V}(\mathbf{x}_{\mathrm{n}}) := v$ & $\mathcal{L}(\mathbf{x}_{\mathrm{n}}) := \ell$.
          If $\big($*StopDetection=FALSE*$\big)$ & $\big(\mathbf{x}_{\mathrm{n}}$ is *Far*$\big)$, do
            Tag $\mathbf{x}_{\mathrm{n}}$ as *Trial* and as *Boundary*.
        Else if $\mathcal{V}(\mathbf{x}_{\mathrm{n}}) \neq \mathcal{V}(\mathbf{x}_{\min})$, do
          $\{u, v, \ell\}$:=`UpdateSchemeFMM` $\big(\mathbf{x}_{\mathrm{n}}, \mathcal{N}_{\mathrm{M}}(\mathbf{x}_{\mathrm{n}})\big)$
          If $u < \mathcal{U}(\mathbf{x}_{\mathrm{n}})$, do
            Set $\mathcal{U}(\mathbf{x}_{\mathrm{n}}) := u$, $\mathcal{V}(\mathbf{x}_{\mathrm{n}}) := v$ & $\mathcal{L}(\mathbf{x}_{\mathrm{n}}) := \ell$.
            Tag $\mathbf{x}_{\mathrm{n}}$ as *Trial*.
      If $\mathbf{x}_{\min}$ is *Boundary*, do
        Tag $\mathbf{x}_{\min}$ as *Interior*.
        If *StopDetection=FALSE*, do
          StopDetection:=
          `IsBoundarySplit` $\big(\mathbf{x}_{\min}, \mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{\min})\big)$.

as the routine `IsBoundarySplit` returns *TRUE*. This routine is called after the grid point $\mathbf{x}_{\min}$ is moved from the set of *Trial* points to the set of *Alive* points, once some of the $\mathrm{M} = 4$ neighbors of $\mathbf{x}_{\min}$ have been tagged as *Boundary*. The routine `IsBoundarySplit` returns *TRUE* if both of the following tests are satisfied :

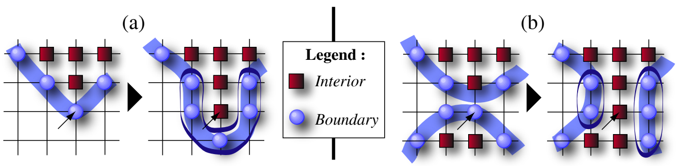

Figure 3: *Local test applied in the vicinity of a grid point* $\mathbf{x}_{min}$ *(the point marked with an arrow) to detect a front collision.* **(a)** $\mathbf{x}_{min}$ *is a simple point of int*(Ω_{F}) *and* $\#C\big(\partial\Omega_{\mathrm{F}} \cap \mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{min})\big) = 1$. **(b)** *Two fronts have collided in the neighborhood of* $\mathbf{x}_{min}$ *and* $\#C\big(\partial\Omega_{\mathrm{F}} \cap \mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{min})\big) = 2$.

- *Local test for detecting a front collision.*
  First, we check if some fronts collide in the vicinity of $\mathbf{x}_{\min}$. Let us denote by $\mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{\min})$ the set of $\mathrm{M}^+ = 8$ neighbors of $\mathbf{x}_{\min}$, and by $\partial\Omega_{\mathrm{F}} \cap \mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{\min})$ the set of points of $\mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{\min})$ which are tagged as *Boundary*. The local test simply relies on the computation of the number of 8-connected components of $\partial\Omega_{\mathrm{F}} \cap \mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{\min})$, denoted $\#C\big(\partial\Omega_{\mathrm{F}} \cap \mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{\min})\big)$. Most of the time, $\mathbf{x}_{\min}$ is a simple point of $\mathrm{int}(\Omega_{\mathrm{F}})$, and $\#C\big(\partial\Omega_{\mathrm{F}} \cap \mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{\min})\big) = 1$ (see Fig. 3.a). The local test is satisfied if $\#C\big(\partial\Omega_{\mathrm{F}} \cap \mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{\min})\big) > 1$, i.e. when there is a shock between some propagating fronts (see Fig. 3.b).

- *Global test for detecting a topological change of* $\partial\Omega_{\mathrm{F}}$.
  When the local test is satisfied, we need to check if the different components of $\partial\Omega_{\mathrm{F}} \cap \mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{\min})$ are also disconnected at a global scale. The global test is satisfied if the front collision has split an 8-connected component of $\partial\Omega_{\mathrm{F}}$ into several 8-connected components. Such a test is easy to implement. For instance, consider the case where $\#C\big(\partial\Omega_{\mathrm{F}} \cap \mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{\min})\big) = 2$. Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be two grid points such that $\mathbf{x}_1$ belongs to the first component of $\partial\Omega_{\mathrm{F}} \cap \mathcal{N}_{\mathrm{M}^+}(\mathbf{x}_{\min})$ and $\mathbf{x}_2$ to the second. We just have to visit all grid points which belong to the same 8-connected component of $\partial\Omega_{\mathrm{F}}$ as $\mathbf{x}_1$, and assign to each visited point a temporary label. Then, the global test is satisfied if $\mathbf{x}_2$ has not been labeled.

Since the scheme used to detect the iteration at which the keypoint detection has to be stopped mainly requires tests at a local scale, it is considerably less computationally expensive than globally counting the number of connected components of $\mathrm{int}(\Omega_{\mathrm{F}})$ and $\partial\Omega_{\mathrm{F}}$ at each iteration of the marching loop. Moreover, note that special care is required to deal with the fact that a propagating front may reach the border of the domain Ω. We suggest adding virtual points along each border of the discrete grid and tagging as *Boundary* every virtual point in the neighborhood of an *Interior* point lying on the border of the grid. This ensures that any connected component of $\mathrm{int}(\Omega_{\mathrm{F}})$ is completely delimited by a connected set of *Boundary* points.

Once the keypoing stopping criterion is satisfied, no more grid points are moved from the set of *Far* points to the set Ω_{F}, and computations are continued until correct values of $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{L}$ have been assigned to each point of Ω_{F}.

4 Building a cyclic sequence of minimal paths to extract a closed contour.

The FMMWKD may be used to extract a closed contour from a 2D image I given a single contour point

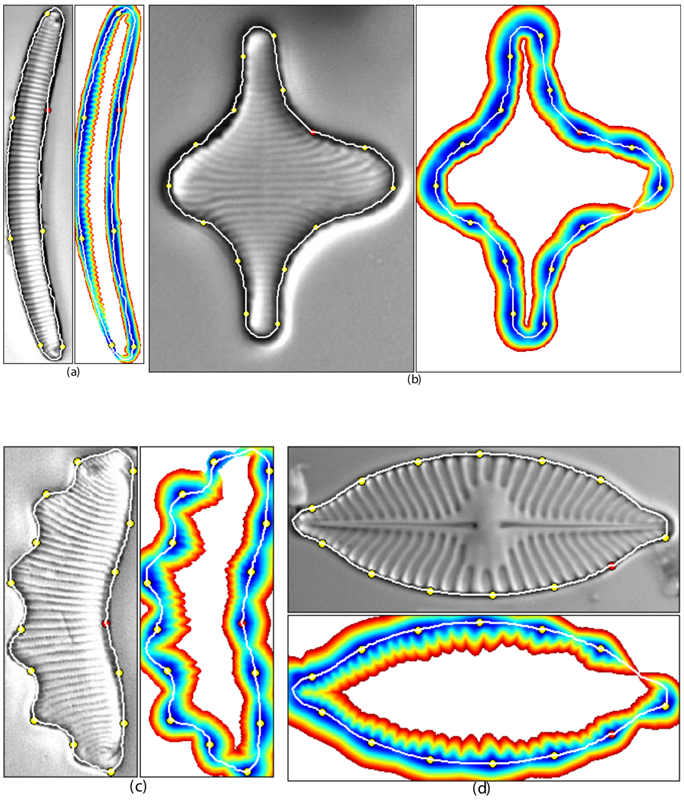

Figure 4: Extraction of a closed contour from a 2D microscopy image. Potential $\mathcal{P}$, set of sources $\mathcal{S} \cup \mathcal{S}^*$, Minimal action map and cyclic sequence of minimal paths. (**a**) Image size 101×521, $\lambda = 180$. (**b**) 385×532, $\lambda = 80$. (**c**) 153×380, $\lambda = 60$. (**d**) 1032×435, $\lambda = 160$.

$\mathbf{p}_1$ in an easy and fast manner. Once a potential $\tilde{\mathcal{P}}$ has been chosen to drive the front propagation, applying the FMMWKD with $\mathcal{S} = \{\mathbf{p}_1\}$ gives a set of points $\mathcal{S} \cup \mathcal{S}^*$, but also the maps $\mathcal{U}$ and $\mathcal{V}$. We exploit the Voronoi diagramm to decide if two sources $\mathbf{p}_i$ and $\mathbf{p}_j$ of $\mathcal{S} \cup \mathcal{S}^*$ are adjacent. Then we look for the associated saddle point $\mathbf{m}_{i|j}$ as described in (Cohen 2001) to make two gradient descent to $\mathbf{p}_i$ and $\mathbf{p}_j$. Linking pairs of neighboring points of $\mathcal{S} \cup \mathcal{S}^*$ by minimal paths enable the extraction of the desired contour.

5 Results and discussion

The way the FMMWKD is built ensures that λ is an upper bound of the Euclidean path length map $\mathcal{L}$ whenever a new keypoint is detected. Thus, the smaller the value given to λ is, the smaller the number of grid points visited during the front propagation is. In a sense, the FMMWKD may be regarded as a way to limit the front propagation to a small neighborhood around the manifold of interest. Better still, FMMWKD speeds up the segmentation process.

In figure 4, we show segmentation results on microscopy images obtained on a commercial computer in under a second.

6 Conclusion

We have presented a new fast front propagation approach for closed contour segmentation. Our method is interactive. At least one keypoint and the Euclidean length parameter λ have to be given by the user. Extending our method to higher dimensions is straightforward, but one may only get a mesh of minimal paths on a closed surface called *geodesic meshing*. Future work will include a new step based on an implicit method, to obtain a complete closed surface.

REFERENCES

Caselles, V., R. Kimmel, and G. Sapiro (1997). Geodesic active contours. *International Journal of Computer Vision* ***22***, 61–79.

Cohen, L. D. (2001). Multiple contour finding and perceptual grouping using minimal paths. *Journal of Mathematical Imaging and Vision* ***14***, 225–236.

Cohen, L. D. and R. Kimmel (1997). Global minimum for active contour models: a minimal path approach. *International Journal of Computer Vision* ***24***, 57–78.

Deschamps, T. and L. Cohen (2002, August). Fast extraction of tubular and tree 3d surfaces with front propagation methods. In *16th International Conference on Pattern Recognition, ICPR'02*, Quebec, Canada.

Deschamps, T. and L. D. Cohen (2001). Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Medical Image Analysis* ***5***, 281–299.

Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematic* ***1***, 269–271.

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

Rouy, E. and A. Tourin (1992). A viscosity solution approach to shape from shading. *SIAM Journal on Numerical Analysis* ***29***, 867–884.

Sethian, J. A. (1996). A fast marching level set for monotonically advancing fronts. *Proceedings of the National Academy of Sciences* ***93***, 1591–1595.

Sethian, J. A. (1999a). Fast marching methods. *SIAM Review* ***41***, 199–235.

Sethian, J. A. (1999b). *Level Set Methods and Fast Marching Methods*. Cambridge University Press.

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