Minimal Paths and Fast Marching Methods for Image Analysis

Laurent D. Cohen

ABSTRACT We present an overview of part of our work on minimal paths. Introduced first in order to find the global minimum of active contours' energy using Fast Marching [13], we have then used minimal paths for finding multiple contours for contour completion from points or curves in 2D or 3D images. Some variations allow to decrease computation time, make easier initialization and centering a path in a tubular structure. Fast Marching is also an efficient way to solve balloon model evolution using level sets. We show applications like for road and vessel segmentation and for virtual endoscopy.

Keywords: Minimal paths, active contours, deformable models, fast marching, Eikonal Equation, level sets, weighted distance, energy minimization, Perceptual grouping, salient curve detection, medical imaging, aerial images.

1 Introduction

Deformable models have been the object of considerable studies and variations since their introduction in [20]. Most of the approaches that were introduced since then tried to overcome the main drawbacks of this model: initialization, minimization and topology changes. The model requires the user to input an initial curve close to the goal. Using the balloon model [7] allows a less demanding initialization. Level sets approaches have the same property [4, 24, 5]. A region-based approach (for example [10, 8]) also makes the solution less sensitive to local minima and initialization. Also, a priori knowledge included in a parametric deformable model (for example [3, 6]) allows to be more robust.

However, for images like the one in figure 4, a very precise initialization is needed to avoid the active contour being trapped by an insignificant local minimum of the energy [8, 7]. In order to find a global minimum for the energy, authors of [13] have introduced a minimal path approach. This is based on previous work by [22, 21] in a different framework. Curve initialization is replaced by just giving two endpoints. The numerical method has the advantages of being consistent (see [13]), fast and efficient, using the *Fast-Marching* algorithm introduced in [26].

This chapter contains various improvements of the original method, relevant in 2D or 3D. Some of the problems we dealt with for segmentation and contour extraction, finding trajectories and perceptual grouping are presented in this paper as follows:

- Minimal path between two points: The solution proposed in [12, 13] with Fast Marching is reviewed in Section 2.
- Minimal paths between an ordered list of points or a given set of pairs of points is a simple application of the previous case.
- Minimal paths for a given unstructured set of points: we propose a way to find pairs of linked neighbors and paths between them [9] (Section 3).
- Minimal paths between an unknown set of key points to be determined from a larger set of admissible points [9].
- Minimal paths for an unstructured set of connected components, by extending the previous approaches to determine pairs of regions to be linked. [17] (Section 4).
- Segmentation of 2D and 3D tubular and tree structures [15, 16] (sections 4.2 et 5).
- Finding a centered path inside a tubular structure and application to virtual endoscopy [15] (section 6).

2 Minimal Paths

2.1 Geometrical optics

In order to understand *Fermat* Principle which is the physical interpretation of minimal paths described afterward, we illustrate light propagation in two simple cases.

According to *Fermat* Principle, the path followed by monochromatic light to go from a point p_0 to a point p_1 is the path which takes least time. In the case of an homogeneous medium, light speed is constant, and thus light follows a straight line, since shortest time is proportional to distance, as seen on figure 1-left. Sets of points that are reached at a given time are circles. Let us now consider a non homogeneous medium composed of two homogeneous regions separated by a horizontal line in the middle, like in Figure 1-middle. Assuming that light speed is larger in the bottom rectangle, the trajectory will "prefer" to remain in this rectangle as much as possible. As a consequence, trajectories are submitted to a refraction effect, as seen on a few trajectories shown in the figure. Angles between the two lines and the normal to the interface between the two media satisfy Snell-Descartes'law (ratio of their sines is equal to the ratio of refraction indices). The refraction index $n > 1$ is the ratio between light speed in emptiness c and its speed in the considered medium v. From this definition, travel time

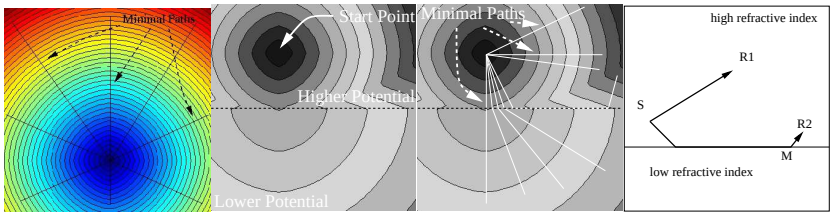

FIGURE 1. Cost function by front propagation and minimal paths for a potential with one or two values. See text.

T between two points is the integral along the followed path of the inverse of the speed $\frac{1}{v} = \frac{n}{c}$. The followed path is a minimum for $T = \frac{1}{c}\int_{p_0}^{p_1} n ds$. The Eikonal equation (see section 2.4) was obtained for this minimization by Hamilton, as a special case of Hamilton-Jacobi equations.

One of the trajectories shown again on figure 1-right illustrates the well known mirage effect. Light source S is visible from points R_1 et R_2. But the path followed between S and R_2 is not a straight line, since light "prefers" going through the smaller refraction index area to go faster. This is a common phenomenon when temperature variations are large enough between the ground and atmosphere, making believe an observer at R_2 there is an oasis in the desert. Similarity will be obvious in the following sections where active contours potential P takes the same place as refraction index n.

2.2 Global Minimum for active contours

We present in this section the basic ideas of the method introduced in [13] to find the global minimum of the active contour energy using minimal paths. The energy to minimize is similar to classical deformable models (see [20]) where it combines smoothing terms and image features attraction term:

$$E(C) = \int_{\Omega} \left\{ w_1 \|C'(s)\|^2 + w_2 \|C''(s)\|^2 + P(C(s)) \right\} ds \tag{1.1}$$

where $C(s)$ represents a curve drawn on a 2D image and Ω is its domain of definition. The method of [13] improves energy minimization since the problem is transformed in a way allowing to find the global minimum.

2.3 Problem formulation

As explained in [13], skipping second order term, we are lead to minimize

$$E(C) = \int_{\Omega=[0,L]} \{w + P(C(s))\} ds, \tag{1.2}$$

where s is the arclength parameter ($\|C'(s)\| = 1$). The regularization of this model is now achieved by the constant $w > 0$ (see [13] for details). Given a

potential $P \geq 0$, the energy is like a distance weighted by $\tilde{P} = P + w$. The minimal action $\mathcal{U}$ is defined as the minimal energy integrated along a path between starting point p_0 and any point p:

$$\mathcal{U}(p) = \inf_{\mathcal{A}_{p_0,p}} E(C) = \inf_{\mathcal{A}_{p_0,p}} \left\{ \int_\Omega \tilde{P}(C(s))ds \right\} \tag{1.3}$$

where $\mathcal{A}_{p_0,p}$ is the set of all paths between p_0 and p. The minimal path between p_0 and any point p_1 in the image can be easily deduced from this action map by a simple back-propagation (gradient descent on $\mathcal{U}$) starting from p_1 until p_0 is reached. This backpropagation step is made possible

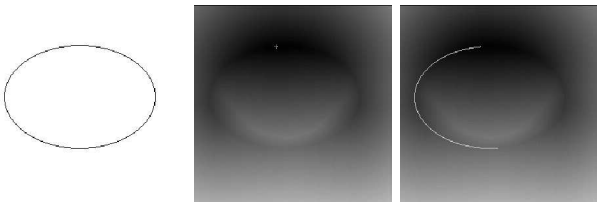

FIGURE 2. On the left, the potential is defined to be minimal on the ellipse. In the middle, the minimal action or weighted distance to the marked point. On the right, minimal path using backpropagation from the second point.

due to the fact that $\mathcal{U}$ has no local minimum except point p_0, therefore the descent converges to p_0 for any p_1. More accurate gradient descent methods like Runge-Kutta midpoint algorithm or Heun's method can be used.

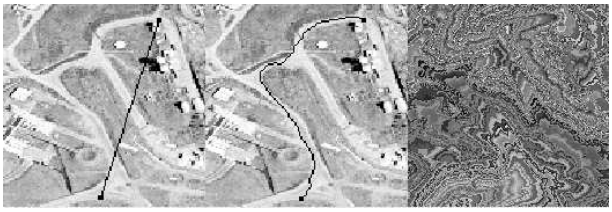

FIGURE 3. Global minimum of active contour model. After giving two points on the left, the minimal path between them is found in the middle image. On the right we show the cost function from the start point. Notice faster propagation along the roads. Potential is defined as a decreasing function of the gray level.

2.4 Fast Marching Resolution

In order to compute $\mathcal{U}$, a front-propagation equation related to Eqn. (1.3) is solved: $\frac{\partial \mathcal{C}}{\partial t} = \frac{1}{P}\vec{n}$. It evolves a front $\mathcal{C}$ starting from an infinitesimal circle shape around p_0 until each point inside the image domain is assigned a value

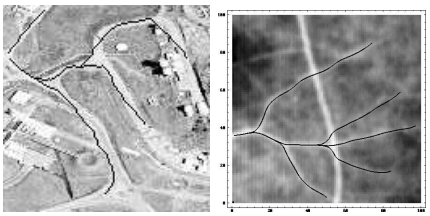

FIGURE 4. Many minimal paths are obtained from a same start point and many end points. This allows extracting the set of roads in the aerial image on the left and vessels in the eye fundus image on the right.

for $\mathcal{U}$. The value of $\mathcal{U}(p)$ is the time t at which the front passes over p. The *Fast Marching* technique, introduced in [26], was used in [12, 13] noticing that the map $\mathcal{U}$ satisfies the *Eikonal* equation $\|\nabla \mathcal{U}\| = \tilde{P}$ and $\mathcal{U}(p_0) = 0$. The relation with this equation will be explained in section 5. Since classic finite difference schemes for this equation are unstable, an up-wind scheme was proposed by [26]:

$$(\max\{u - \mathcal{U}_{i-1,j}, u - \mathcal{U}_{i+1,j}, 0\})^2 + (\max\{u - \mathcal{U}_{i,j-1}, u - \mathcal{U}_{i,j+1}, 0\})^2 = \tilde{P}^2_{i,j}. \quad (1.4)$$

The improvement made by the *Fast Marching* is to introduce order in the selection of the grid points. This order is based on the fact that information is propagating *outward*, because the action can only grow due to the quadratic Eqn. (1.4). The main idea is similar to the construction of minimum length paths in a graph between two given nodes introduced in [18] (see discussion in [13]). Complexity of *Fast Marching* on a grid with N nodes is bounded by $O(N \log_2 N)$ for the *Fast Marching* on a grid with N nodes. The algorithm is detailed in Table 1.1. Examples are shown in Fig. 2 to 4. Solving Eqn. (1.4) is detailed next.

2.5 2D Up-Wind Scheme

Notice that for solving Eqn. (1.4), only values of alive neighbor points are considered (Table 1.1). Considering the neighbors of grid point (i, j) in 4-connexity, we note $\{A_1, A_2\}$ and $\{B_1, B_2\}$ the two couples of opposite neighbors such that we get the ordering $\mathcal{U}(A_1) \leq \mathcal{U}(A_2)$, $\mathcal{U}(B_1) \leq \mathcal{U}(B_2)$, and $\mathcal{U}(A_1) \leq \mathcal{U}(B_1)$. Considering that we have $u \geq \mathcal{U}(B_1) \geq \mathcal{U}(A_1)$, the equation derived is

$$(u - \mathcal{U}(A_1))^2 + (u - \mathcal{U}(B_1))^2 = \tilde{P}^2_{i,j} \quad (1.5)$$

Based on testing the discriminant Δ of Eqn. (1.5), one or two neighbors are used to solve it:

1. If $\tilde{P}_{i,j} > \mathcal{U}(B_1) - \mathcal{U}(A_1)$, solution of Eqn. (1.5) is $u = \frac{\mathcal{U}(B_1) + \mathcal{U}(A_1) + \sqrt{2\tilde{P}^2_{i,j} - (\mathcal{U}(B_1) - \mathcal{U}(A_1))^2}}{2}$.

Algorithm for 2D Fast Marching for minimal action $\mathcal{U}$
Definitions:

- *Alive* set: grid points at which values of $\mathcal{U}$ have been reached and will not be changed;
- *Trial* set: next grid points (4-connexity neighbors) to be examined. An estimate U of $\mathcal{U}$ is computed using Eqn. (1.4) from alive neighbors only;
- *Far* set: all other grid points, there is not yet an estimate for U;

Initialization:

- *Alive* set: start point $p_0, U(p_0) = \mathcal{U}(p_0) = 0$;
- *Trial* set: four neighbors p of p_0 with initial value $U(p) = \tilde{P}(p)$ ($\mathcal{U}(p) = \infty$);
- *Far* set: all other grid points, $\mathcal{U} = U = \infty$;

Loop:

- Let $p = (i_{min}, j_{min})$ be the *Trial* point with the smallest action U;
- Move it from the *Trial* to the *Alive* set;
- For each neighbor (i, j) of (i_{min}, j_{min}):
  - If (i, j) is *Far*, add it to the *Trial* set;
  - If (i, j) is *Trial*, update $U_{i,j}$ with Eqn. (1.4).

TABLE 1.1. *Fast Marching* algorithm

2. else $u = \mathcal{U}(A_1) + \tilde{P}_{i,j}$.

2.6 Minimal Paths in 3D

A 3D extension of the *Fast Marching* algorithm was presented in [15]. Similarly to previous section, the minimal action $\mathcal{U}$ is defined as $\mathcal{U}(p) = \inf_{\mathcal{A}_{p_0,p}} \left\{ \int_\Omega \tilde{P}(C(s))ds \right\}$ where $\mathcal{A}_{p_0,p}$ is now the set of all 3D paths between p_0 and p. Given a start point p_0, in order to compute $\mathcal{U}$ we start from an initial infinitesimal spherical front around p_0. The 2D scheme of equation (1.4) is extended to 3D, leading to:

$$(\max\{u - \mathcal{U}_{i-1,j,k}, u - \mathcal{U}_{i+1,j,k}, 0\})^2 + (\max\{u - \mathcal{U}_{i,j-1,k}, u - \mathcal{U}_{i,j+1,k}, 0\})^2 + (\max\{u - \mathcal{U}_{i,j,k-1}, u - \mathcal{U}_{i,j,k+1}, 0\})^2 = \tilde{P}^2_{i,j,k} \quad (1.6)$$

giving the correct viscosity-solution u for $\mathcal{U}_{i,j,k}$. An example is given in figure 14 of section 6.

2.7 Simultaneous Front Propagation

The idea is to propagate simultaneously a front from each end point p_0 and p_1 [15]. Let us consider the first grid point p where those fronts meet. This point has to be on the minimal path between p_0 and p_1. Since during propagation the action can only grow, propagation can be stopped at this step in order to make backpropagation. Adjoining the two paths, respectively between p_0 and p, and p_1 and p, gives an approximation of the exact

FIGURE 5. Simultaneous propagation: The left image is the data set, used as potential to extract a path in a vessel. In the middle, the action map is obtained from the first point till second point is reached. The right image shows the action map resulting from a simultaneous propagation from both extremities points, and the two paths from the intersection point.

minimal action path between p_0 and p_1. Since p is a grid point, the exact minimal path might not go through it, but in its neighborhood. Precise location can be obtained through interpolation between grid points like in [25]. This algorithm is described in table 1.2. This approach allows a par-

Algorithm

- Compute the minimal action maps U_0 and U_1 to respectively p_0 and p_1 until the two fronts have an *Alive* point p_2 in common;
- Compute the minimal path between p_0 and p_2 by back-propagation on U_0 from p_2;
- Compute the minimal path between p_1 and p_2 by back-propagation on U_1 from p_2;
- Join the two paths found.

TABLE 1.2. Minimal Path from two action maps

allel implementation of the two propagations. Also, the region covered by Fast Marching is greatly reduced (see Figure 5).

2.8 Simultaneous estimate of the path length

In some cases, like for giving extremities in a 3D image, it is easier for the user to give only one start point and the second should be found automatically. We now describe an approach which builds a path given a starting point and a given path length to reach [15]. We are able to compute simultaneously at each point of the front energy U of the minimal path and its length. The end point is then chosen as the first point that reach the expected length. Propagation is stopped when this point is reached and minimal path is computed from it. Since the front propagates faster along small values of the potential, the interesting paths are longer among all paths which have same minimal action U. When the front propagates in a tubular structure, all points who reach first the given length are in a same

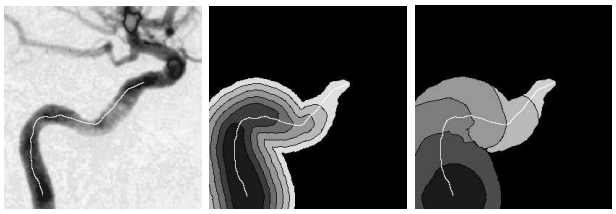

FIGURE 6. Simultaneous estimate of the path length. On the left, potential; In the middle, minimal action map; on the right, length of the minimal path. These maps are computed only until a given length is reached.

Notations

- a start point p_0 is manually set;
- the minimal energy map U, a min-heap $\mathcal{H}_U$ and a potential image P;
- a distance map D to compute the Euclidean length of the minimal path ;
- a min-heap $\mathcal{H}_D$, where the ordering key for any point p is the value of $D(p)$ (the first element of this heap will be the *T*rial point with smallest D);

Initialization

- initialize the front propagation method, by setting $U(p_0) = D(p_0) = 0$ and storing p_0 in both min-heaps $\mathcal{H}_U$ and $\mathcal{H}_D$;

Loop: at each iteration, consider p_{min} the *T*rial point with smallest U

- Move it to *A*live set, and remove it from both $\mathcal{H}_U$ and $\mathcal{H}_D$
- for each neighbor p of p_{min}:
  - proceed according to the classical Fast Marching algorithm: update $U(p)$ and re-balance $\mathcal{H}_U$;
  - update $D(p)$ according to $\|\nabla D\| = 1$ using the same neighbors of p that were involved in updating $U(p)$ and re-balance $\mathcal{H}_D$

TABLE 1.3. Computing the Euclidean Distance traveled by the front.

region of the image, far from the starting point and inside the tubular shape. This gives a justification for this choice of end point (see Figure 6).

Once the path is extracted by gradient descent, we can easily compute its length. But this is a very time consuming process to systematically do this at each point visited. Therefore we proposed to compute on-the-fly an approximation of the distance traveled by the front. We use the property that when propagating a front with a constant speed equal to one, the minimal energy obtained at each point represents the *Euclidean* distance D to the starting point. The Euclidean length of the path is found solving $\|\nabla D\| = 1$ using with the same neighbors involved for P in Eqn. (1.5). The corresponding algorithm is described in table 1.3. This algorithm was used for reducing user-intervention in the *V*irtual Endoscopy process presented in section 6 by giving only one point [15].

3 Minimal paths from a set of endpoints p_k

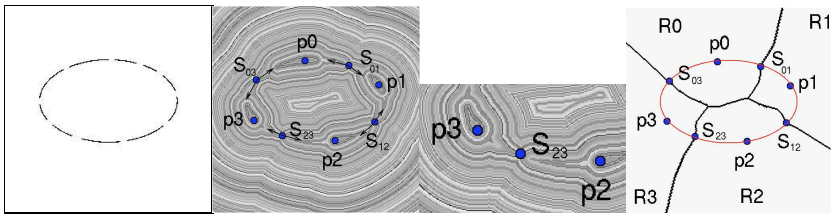

FIGURE 7. Ellipse. From left to right: potential is an incomplete ellipse and points p_k are given; level sets of minimal action U from p_k's; zoom on a *saddle point*; backpropagation from selected *saddle points* to their two source points give the set of paths and voronoi diagram.

Minimal paths between points p_k, minimal action $V = \mathcal{U}_{\{p_k, 0 \leq k \leq N\}}$

- Initialization:
  - p_k's are given; $\forall k, V(p_k) = 0; l(p_k) = k$ is the front index, p_k *alive*.
  - $\forall p \notin \{p_k\}, V(p) = \infty; l(p) = -1$; p is *far* except 4-connexity neighbors of p_k's that are *trial* with estimate U using Eqn. (1.4).
- Loop for computing $V = \mathcal{U}_{\{p_k, 0 \leq k \leq N\}}$:
  - Let $p = (i_{min}, j_{min})$ be the *Trial* point with the smallest action U;
  - Move it from the *Trial* set to the *Alive* set with V(p) = U(p);
  - Update $l(p)$ with the same index as point A_1 in formula (1.5). If $l(A_1) \neq l(B_1)$ and we are in case 1 of section 2.5 where both points are used and if this is the first time regions of labels $l(A_1)$ and $l(B_1)$ meet, $S(p_{l(A_1)}, p_{l(B_1)}) = p$ is set as the *saddle point* between $p_{l(A_1)}$ and $p_{l(B_1)}$. If these points have not yet two *linked neighbors*, they are put as *linked neighbors* and $S(p_{l(A_1)}, p_{l(B_1)}) = p$ is selected,
    For each neighbor (i, j) of (i_{min}, j_{min}):
    - If (i, j) is *Far*, add it to the *Trial* set;
    - If (i, j) is *Trial*, update action $U_{i,j}$.
- Obtain all paths between selected *linked neighbors* by backpropagation each way from their *saddle point*.

TABLE 1.4. Algorithm for unstructured set of points.

3.1 Multiple minimal paths

We propose to use the minimal path approach to extract a set of contours from an unstructured set of points given on an image. In order to find the set of most representative contours on the image, we are looking for minimal paths between pairs of points. We describe briefly the method when points

p_k are already known. An approach to automatically find points p_k that are most representative among a larger set of admissible points was introduced in [9], based on an iterative farthest point strategy relative to the weighted distance. Such a strategy was used later on to find adaptive or uniform remeshing of a surface using fast marching [25].

We assume here that points p_k are known. If we knew as well which pairs of points have to be linked among p_k's, finding all contours is a trivial application of section 2. The problem we are interested in here is also to find out which pairs of points have to be connected by a contour. Since the set of points p_k's is assumed to be given unstructured, we do not know in advance how the points connect. This is the key problem that is solved here using a minimal action map.

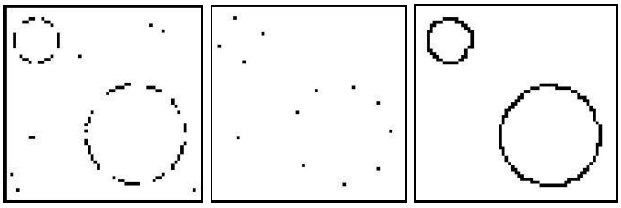

FIGURE 8. Two circles: from left to right: incomplete noisy data set; the set of found p_k's; multiple minimal paths between p_k's.

The main goal of our method is to obtain all significant paths joining the given points. However, each point should not be connected to all other points, but only to those that are closer to them in the energy sense. There are many possibilities to decide which pairs of points have to be linked. It depends on data and on the application in view. In some cases, it is necessary to detect closed curves and avoid bifurcation, or T-junctions. The criterion is then to constrain a point p_k to be linked to at most two other points among $p'_k s$, in order to generate a closed curve. In case we are looking for tree structures, the criterion is different, as in section 4.2.

For perceptual grouping, potential P to be minimized along the paths is often a binary image of edge points, that form incomplete contours, as on figure 7-left. Attraction potential to the set of edge points can be defined (see [11]) in order to have lower values along edge points and higher values in the background.

3.2 Main ideas of the approach

Our approach is similar to computing the distance map to a set of points and their Voronoi diagram. However, we use here a weighted distance defined through the potential P. This distance is obtained as the minimal action with respect to P with zero value at all points p_k. Instead of computing a minimal action map for each pair of points, as in Section 2.3, we

only need to compute one minimal action map in order to find all paths. At the same time the action map is computed we determine the pairs of points that have to be linked together by finding meeting points of the propagation fronts. These are *saddle points* of the minimal action $\mathcal{U}$.

Although the minimal action is computed using fast marching, the level sets of $\mathcal{U}$ give the evolution of the front. During the fast marching algorithm, the boundary of the set of alive points also gives the position of the front.

Figure 7 illustrates the steps of the algorithm. Figure 8 shows the result with points p_k found automatically. More details can be found in [9].

4 Multiple minimal paths between regions R_k

We consider perceptual grouping and contour completion from an unstructured set of regions in a 2D or 3D image. As an extension of previous section 3, complete curves are obtained as minimal paths between pairs of regions [17]. This approach is extended to finding a set of minimal paths that connect a set of 3D regions in 3D images. This makes use of *Fast-Marching* in a 3D image, as in section 2.6 [15, 14].

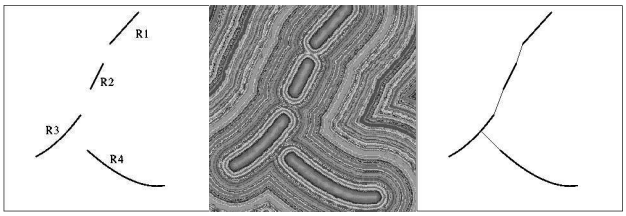

FIGURE 9. From left to right: examples of regions to link; level sets of the minimal action from the 4 regions; minimal paths obtained from the 3 selected *saddle points*.

4.1 Minimal path between 2 regions

Defining a minimal path between two regions is an easy extension of [13]. Consider two connected regions, the start region R_0 and the set of end points R_1. The problem is finding a minimal path among all paths starting from a point in R_0 and ending on R_1. Minimal action is then defined as:

$$\mathcal{U}(p) = \inf_{\mathcal{A}_{R_0,p}} E(C) = \inf_{p_0 \in R_0} \inf_{\mathcal{A}_{p_0,p}} E(C) \tag{1.7}$$

where $\mathcal{A}_{R_0,p}$ is the set of paths starting from a point in R_0 and ending at p. This is computed using Fast Marching as in table 1.1, with initial set of *Alive* points being R_0, with $\mathcal{U} = 0$. In order to find a minimal path between

R_1 and R_0, we determine a point $p_1 \in R_1$ such that $\mathcal{U}(p_1) = \min_{p \in R_1} \mathcal{U}(p)$. We then make backpropagation from p_1 to R_0.

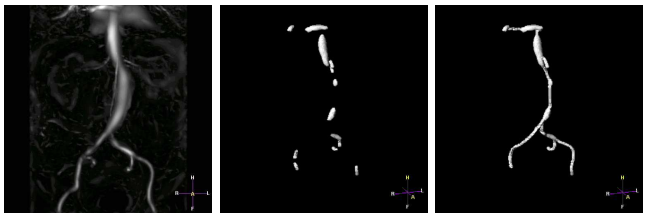

FIGURE 10. Perceptual grouping in the 3D aorta image: MIP view of vascularity potential; detection of regions in the aorta; vascular tree completion by minimal paths relatively to vascularity potential.

4.2 Tubular structures

Linking regions can be useful when these regions are for example connected components obtained after edge detection. In the example of Fig. 9, which represent a tree structure, regions are selected in a way that they do not form together a closed curve. In medical imaging, finding vessels is a very important problem. Regions can then be defined from thresholding a vascularity criterion of [19] to detect tubular regions in a vessel image. In Figure 10, we show a MIP view of the vascularity potential [19] obtained from 3D MRI of the aorta with contrast product. We obtain a set of regions by thresholding the multiscale criterion. Our method helps completing these region and finding the structure of the vascular tree.

5 Segmentation by Fast Marching

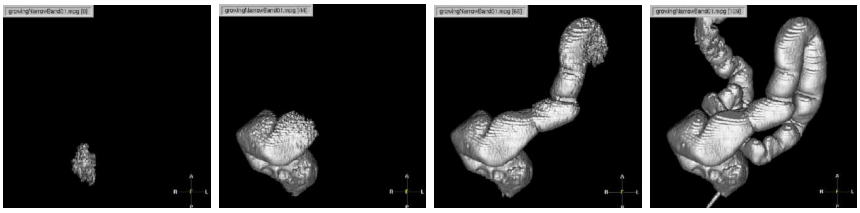

FIGURE 11. Propagation inside colon using Fast Marching.

Several approaches are possible to segment the boundary surface of an object starting from points inside. We can use for example a balloon model [7] or its level-sets implementation, as in [24]. In fact, this kind of region

growing method can also be solved fast using the *Fast Marching* algorithm [23]. This allows to make a segmentation step in the same framework as minimal path finding. Having searched for the minimal action from one point, the algorithm provides the following regions:

- Inside : the points whose action is set, labeled *Alive*;
- Outside : the points not yet examined, labeled *Far*;
- the points at the interface between *Alive* and *Far* points, whose actions are not set, labeled *Trial*.

This last region, on the boundary of the visited points, is a contour in 2D and a surface in 3D. If the potential is a lot higher along edges than it is inside the shape, the edges will act as an obstacle to the front propagation. In this case the *Trial* points define a surface which segments the object.

In order to see the precise relation between fast marching propagation and active contours, consider the usual evolution equation of an interface (2D curve or 3D surface) that appears in level sets methods $\frac{\partial C}{\partial t}(p) = F(\kappa)\mathbf{n}$ and $C(p,0) = C_0(p)$. Assume the speed $F = \frac{1}{P} > 0$, and thus the front moves always outwards in the normal direction $\mathbf{n}$, like an inflating balloon [7], but with a speed which is not necessarily constant. A way to characterize the interface is to compute at each point $\mathbf{x}$ of the image the arrival time $T(\mathbf{x})$ of the interface $C(t)$ when it sweeps the domain. Using the classical properties of a level set of T that its normal is in the direction of ∇T, the following equation is obtained from the evolution of interface $C(t)$:

$$T(C(\mathbf{x},t)) = t \Rightarrow \nabla T \cdot C_t = 1 \Rightarrow \nabla T \cdot \left(F \frac{\nabla T}{|\nabla T|} \right) = 1 \Rightarrow F \cdot |\nabla T| = 1$$

where we recognize the Eikonal equation seen above in section 2.4. This equation was thus solved by fast marching in [23] for surface segmentation since it has the same advantages as the level set formulation, but is much faster. This equation is solved using 3D Fast Marching (see section 2.6) in the example for segmentation of the colon shown in figure 11, [15].

When the front propagates in a long and thin structure for which the potential contrast between inside and outside is not sufficient, the front will likely flood out of the object during propagation. Indeed, when the front propagates in the tubular structure, there is only a small part of the front, which we could call the "head" of the front, that really moves. Most of the front is located close to the boundary of the structure and moves very slowly. For example voxels that are close to the starting point, the "tail" of the front, are moving very slowly. However, since the structure may be very long, in order for the "head" voxels to reach the end of the structure, the "tail" voxels may flow out of the boundary since their speed is always positive, and integrated over a long time. This is illustrated in the example of Figure 12.

We introduced in [16] an approach where points of the front are "frozen" when a distance criterion is satisfied. This makes use of the length of the

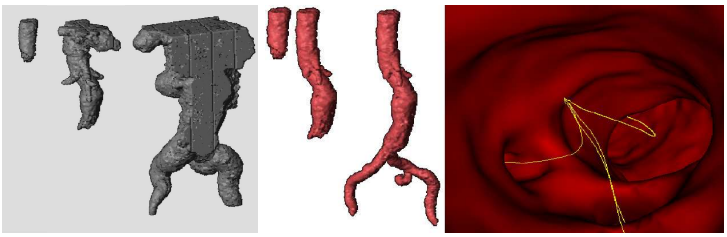

FIGURE 12. Front Propagation in a 3D MR image of the aorta. On the left it floods, in the middle, freezing prevents flooding. On the right, virtual endoscopy in the tree structure, with visible paths.

minimal paths computed as in section 2.8. Figure 12 shows the result with freezing which gives a correct segmentation.

6 Centered Minimal Paths and virtual endoscopy

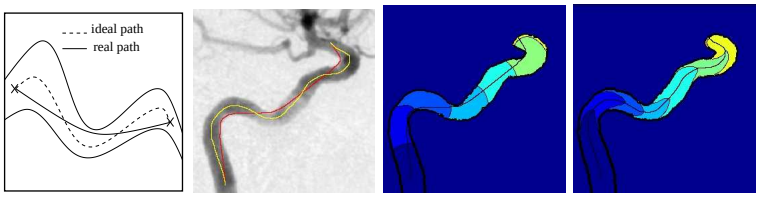

FIGURE 13. Centered path in a vessel: Two images on the left show both paths on a sketch and original image.Two images on the right show propagation and path for classical potential and centering distance potential obtained.

A minimal path minimizes the integral of the potential in equation (1.2). If the potential is constant in some areas, like inside a tubular object, it will lead to a shortest geodesic path. The same thing happens when the potential does not vary much inside a tubular shape. The minimal path extracted is often tangential to the edges, as shown on the left of figure 13, and this is a problem when looking for a trajectory for virtual endoscopy [15]. A centered path is more relevant. The method we proposed to obtain a centered path in a tubular shape first segments the tubular region and then looks for a path inside as far as possible from the walls, using a distance map. The complete method is detailed in [15], here are the main steps:

1. Segmentation: compute the weighted distance map by front propagation from the given start point till reaching the end point, which can be found automatically using a length criterion of section 2.8.
2. Segmentation: set of *trial* points, as described in section 5.

3. Centering Potential : compute inside the tubular object the distance map $\mathcal{D}$ to the surface previously obtained (fast marching with $P = 1$).
4. Centered path : this is the minimal path between start and end points relatively to a *decreasing* function of the distance $\mathcal{D}$. The path locates as far as possible from the walls, which means in the center where distance to the boundary is larger. The final step is to make back-propagation from the end point using the last action map.

Figure 13 compares the resulting path with classical potential and centering potential on brain vessels. Figure 14 shows an example of the centered minimal path obtained in a 3D colon image. This path is used as a trajectory for a virtual camera by image rendering at each point of the path from the 3D image data giving a virtual endoscopy. Movies are available on the website http://www.ceremade.dauphine.fr/~cohen/MPEG

This approach can be extended [16] to extraction of a set of paths in a tree structure and the possibility of virtual endoscopy where the user can choose at each bifurcation the path he wishes to follow (figure 12).

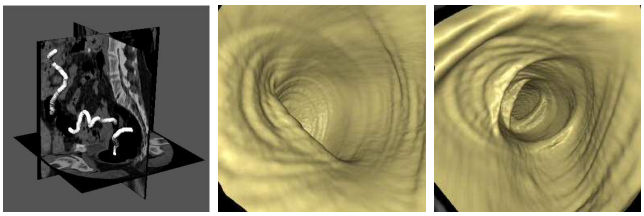

FIGURE 14. On the left, example of a minimal path on a 3D image of colon. On the right, virtual endoscopy through the colon (coloscopy).

Acknowledgments. A large part of the presented work was done in collaboration with R. Kimmel or during PhD supervision of T.Deschamps and I thank both of them greatly.

Conclusion

We have presented various aspects of minimal paths methods and their applications, in particular for medical imaging. These approaches allow to extract a contour or a set of contours in a 2D image, as well as tubular structures, or tree structures in 2D and 3D images. The Fast marching algorithm makes the task much easier and also allows to segment curves or surfaces in an image very fast. Let us quote some of our more recent related work : surface segmentation defined as a set of minimal paths, [2], image segmentation from a set of source points using an extension of the definition of minimal action [1] and fast marching on a triangulated surface used for adaptive remeshing [25].

7 References

[1] P. Arbelaez and L. D. Cohen. Energy partitions and image segmentation. *Journal of Mathematical Imaging and Vision*, 20(1-2):43–57, January - March 2004.

[2] R. Ardon and L. D. Cohen. Fast constrained surface extraction by minimal paths. In *Proc. IEEE Workshop on Variational and Level Set Methods in Computer Vision*, Nice, France, September 2003.

[3] E. Bardinet, L. D. Cohen, and N. Ayache. A parametric deformable model to fit unstructured 3D data. *CVIU*, 71(1):39–54, July 1998.

[4] V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66:1–31, 1993.

[5] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

[6] I. Cohen and L. D. Cohen. A hybrid hyperquadric model for 2-D and 3-D data fitting. *Computer Vision and Image Understanding*, 63(3):527–541, May 1996.

[7] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, March 1991.

[8] L. D. Cohen. Avoiding local minima for deformable curves in image analysis. In *Curves and Surfaces with Applications in CAGD*, pages 77–84. A. Le Méhauté, C. Rabut, and L. L. Schumaker (eds.), 1997.

[9] L. D. Cohen. Multiple contour finding and perceptual grouping using minimal paths. *Journal of Mathematical Imaging and Vision*, 14(3), 2001. CEREMADE TR 0101, Jan 2001.

[10] L. D. Cohen, E. Bardinet, and N. Ayache. Surface reconstruction using active contour models. In *Proceedings SPIE 93 Conference on Geometric Methods in Computer Vision, Vol. 2031*, pages 38–50, San Diego, CA, July 1993. INRIA TR 1824, December 1992.

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

[12] L. D. Cohen and R. Kimmel. Fast marching the global minimum of active contours. In *IEEE International Conference on Image Processing (ICIP'96)*, pages I:473–476, Lausanne, Suisse, Sept. 1996.

[13] 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, Aug. 1997.

[14] T. Deschamps. *Curve and Shape Extraction with Minimal Path and Level-Sets techniques - Applications to 3D Medical Imaging.* PhD thesis, Université Paris-IX Dauphine, Paris, December 2001.

[15] T. Deschamps and L. Cohen. Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Medical Image Analysis*, 5(4):281–299, Dec. 2001. Video in the web version of the journal.

[16] T. Deschamps and L. D. Cohen. Fast extraction of tubular and tree 3D surfaces with front propagation methods. In *Proc. IEEE ICPR'02*, Quebec, Canada, August 2002.

[17] T. Deschamps and L. D. Cohen. Grouping connected components using minimal path techniques. In Springer, editor, *Geometrical Method in Biomedical image processing.* R. Malladi (ed.), 2002.

[18] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematic*, 1:269–271, 1959.

[19] A. Frangi, W. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. In *Proc. MICCAI'98, Cambridge*, pages 130–137, 1998.

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

[21] R. Kimmel, A. Amir, and A. Bruckstein. Finding shortest paths on surfaces using level sets propagation. *IEEE Trans. on PAMI*, PAMI-17(6):635–640, June 1995.

[22] R. Kimmel, N. Kiryati, and A. M. Bruckstein. Distance maps and weighted distance transforms. *Journal of Mathematical Imaging and Vision*, 6:223–233, May 1996. Special Issue on Topology and Geometry in Computer Vision.

[23] R. Malladi and J. Sethian. A real-time algorithm for medical shape recovery. In *Proc. IEEE ICCV'98*, pages 304–310, Jan. 1998.

[24] R. Malladi, J. A. Sethian, and B. C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Trans. on PAMI*, 17(2):158–175, February 1995.

[25] G. Peyre and L. D. Cohen. Geodesic re-meshing and parameterization using front propagation. In *Proc. IEEE Workshop on Variational and Level Set Methods in Computer Vision*, Nice, France, October 2003.

[26] J. A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science.* Cambridge Univ. Press, 1999.