Fast extraction of minimal paths in 3D images and applications to virtual endoscopy

Thomas Deschamps

Medical Imaging Systems Group, Laboratoires d'Éléctronique Philips Paris, LEP,
22, avenue Descartes, BP 15, 94453 Limeil-Brévannes Cedex, France
Tel : 33-1-45 10 68 56 Fax : 33-1-44 05 69 59 Email : deschamps@lep-philips.fr

Laurent D. Cohen

CEREMADE UMR CNRS 7534, Université Paris IX Dauphine,
Place du Marechal de Lattre de Tassigny, 75775 Paris Cedex 16, France
Tel : 33-1-44 05 46 78 Fax : 33-1-44 05 45 99 Email : cohen@ceremade.dauphine.fr

June 26, 2000

Abstract

The aim of this article is to build trajectories for virtual endoscopy inside 3D medical images, using the most automatic way. Usually the construction of this trajectory is left to the clinician who must define some points on the path manually using three orthogonal views. But for a complex structure such as the colon, those views give little information on the shape of the object of interest. The path construction in 3D images becomes a very tedious task and precise a priori knowledge of the structure is needed to determine a suitable trajectory. We propose a more automatic path tracking method to overcome those drawbacks: we are able to build a path, given only one or two end points and the 3D image as inputs. This work is based on previous work in (Cohen and Kimmel, 1997) for extracting paths in 2D images using *Fast Marching* algorithm.

Our original contribution is to extend this technique to 3D and give new improvements of the approach that are relevant in 2D as well as in 3D. We also introduce several methods to reduce the computation cost and the user interaction. We also closely examine a fast implementation of the path extraction. Synthetic and real medical images are used to illustrate each contribution.

We show that this technique can be efficiently applied to the problem of finding a centered path in tubular anatomical structures with minimum interactivity, and we apply it to path construction for virtual endoscopy. Virtual endoscopy results are shown in various anatomical regions (colon, brain vessels, arteries) with different 3D imaging protocols (CT, MR).

keywords : Deformable Models, Minimal paths, Level Set methods, Medical image understanding, Eikonal Equation, Fast Marching, Virtual Endoscopy.

1 Introduction

Once a path is obtained in a CT or MR image, it can be used as input for virtual endoscopy inside an anatomical object. This process consists in creating perspective views of the inside of tubular structures of human anatomy along a user-defined path. Clinicians are then provided with an alternative to the uncomfortable and invasive diagnostic procedures of real endoscopy. Ordinarily, the examination of a patient pathology would require threading a camera inside his body. This new method skips the camera and can give views of regions of the body difficult or impossible to reach physically (e.g. brain vessels), the only requirement being X-ray exposure for CT and sometimes the injection of a contrast product (dye or air) in the anatomical objects, for better detection.

A major drawback in general remains when the user must define all path points manually. For a complex structure (small vessels, colon,...) the required interactivity can be very tedious. If the path is not correctly build, it can cross an anatomical wall during the virtual fly-through. Path construction is thus a very critical task and precise anatomical knowledge of the structure is needed to set a suitable trajectory. Our work focuses on the automation of the path construction, reducing the need for interaction and improving performance, in a robust way, given only one or two end points and the image as inputs.

We derived an automatic path tracking routine in 3D images by mapping this path tracking problem into a minimal path problem between two fixed end points. Defining a cost function inside an image, the minimal path becomes the path for which the integral of the cost between the two end points is minimal. This minimal path problem has been studied for ages by mathematicians, and has been solved using graph theory and dynamic programming (Dijkstra, 1959). (Cohen and Kimmel, 1997) solved the minimal path problem in 2D with a front propagation equation between the two fixed end points, using the *Eikonal* equation (that physically models wavelight propagation), with a given initial front. Their approach has much in common with Dijkstra's but it has advantage of being consistent with the continuous formulation of the problem and it avoids metrication error. Therefore, the first step is to build an image-based measure that defines the minimality property in the studied image, and to introduce it in the *Eikonal* equation. The second step is to propagate the front on the entire image domain, starting from an initial front restricted to one of the fixed points.

The propagation is done using techniques developed in (Sethian, 1996b): the author proposed a method to propagate this front in a quick and efficient way. It first considers the initial front implicitly defined as the zero level set of a higher-dimension function, which evolves. This formulation, called level-sets method, allows to manage front propagation problems due to complex curves and topological changes. Then it uses an algorithm called *Fast Marching*, to quickly solve this new front propagation problem.

The original contribution of this work is twofold. Firstly, we extend the minimal path technique developed in (Cohen and Kimmel, 1997) to 3D images. We also propose various improvements for this technique that are useful for image analysis in 2D as well as in 3D. Secondly, we adapt this technique to the particular problem of tubular anatomical structure extraction. This is applied to virtual endoscopy through 3D medical images.

We show that the level set method can be efficiently applied to the problem of finding a path in virtual endoscopy with minimum interactivity. A wide range of application areas are envisaged from colon to brain vessels. We also propose a range of choices for finding the right input measure to the minimal path tracking.

This paper is organized as follows: In section 2, we summarize the method detailed in (Cohen and Kimmel, 1997) for 2D images, and we extend this method to 3D. In section 3 we give details about our improvement made on the front propagation technique, including faster path extraction schemes, reduction of the user interaction, and we explain how to extract centered paths in tubular structures. Finally in section 4, we show how to apply our method to virtual endoscopy for several anatomical objects.

2 Finding minimal paths in 3D images

2.1 The Cohen-Kimmel Method in 2D

2.1.1 Global minimum for Active Contours

We present in this section the basic ideas of the method introduced by (Cohen and Kimmel, 1997) to find the global minimum of the active contour energy using minimal paths. The energy to minimize is similar to classical deformable models (see (Kass *et al.*, 1988)) where it combines smoothing terms and image features attraction term (Potential P):

$$E(C) = \int_{\Omega} \left\{ w_1 \|C'(s)\|^2 + w_2 \|C''(s)\|^2 + P(C(s)) \right\} ds \tag{1}$$

where $C(s)$ represents a curve drawn on a 2D image, $\Omega = [0, L]$ is its domain of definition, and L is the length of the curve. It reduces the user initialization to giving the two end points of the contour C. In (Cohen and Kimmel, 1997), the authors have related this problem with the recently introduced paradigm of the level-set formulation. In particular, its Euler equation is equivalent to the geodesic active contours (Caselles *et al.*, 1995). They introduced a model which improves energy minimization because the problem is transformed in a way to find the global minimum. It avoids the solution being sticked in local minima. Let us explain each step of this method.

2.1.2 Problem formulation

Most of the classical deformable contours have no constraint on the parameterization s, thus allowing different parameterization of the contour C to lead to different results. In (Cohen and Kimmel, 1997), contrary to the classical snake model (but similarly to geodesic active contours), s represents the arc-length parameter, which means that $\|C'(s)\| = 1$, leading to a new energy form. Considering a simplified energy model without any second derivative term leads to the expression $E(C) = \int \{w\|C'\|^2 + P(C)\} ds$. Assuming that $\|C'(s)\| = 1$ leads to the formulation

$$E(C) = \int_{\Omega} \{w + P(C(s))\} ds \tag{2}$$

We now have an expression in which the internal forces are included in the external potential. The regularization of this model is now achieved by the constant $w > 0$. This term integrates as $\int_{\Omega} w ds = w \times \text{length}(C)$ and allows us to control the

smoothness of the contour (see (Cohen and Kimmel, 1997) for details).
Given a potential $P > 0$ that takes lower values near desired features, we are looking for paths along which the integral of $\tilde{P} = P + w$ is minimal. The surface of minimal action U is defined as the minimal energy integrated along a path between a starting point p_0 and any point p:

$$U(p) = \inf_{\mathcal{A}_{p_0,p}} E(C) = \inf_{\mathcal{A}_{p_0,p}} \left\{ \int_\Omega \tilde{P}(C(s))ds \right\} \tag{3}$$

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. Assuming that potential P is always positive, the action map will have only one local minimum which is the starting point p_0, and the minimal path will be found by a simple back-propagation on the energy map. Thus, contour initialization is reduced to the selection of the two extremities of the path.

2.1.3 Fast Marching Resolution

In order to compute this map U, a front-propagation equation related to equation (3) is solved : $\frac{\partial C}{\partial t} = \frac{1}{\tilde{P}} \vec{n}$. It evolves a front starting from an infinitesimal circle shape around p_0 until each point inside the image domain is assigned a value for U. The value of $U(p)$ is the time t at which the front passes over the point p.

The *Fast Marching* technique, introduced by (Sethian, 1996b), was used by (Cohen and Kimmel, 1997) noticing that the map U satisfies the Eikonal equation:

$$\|\nabla U\| = \tilde{P} \tag{4}$$

Classic finite difference schemes for this equation tend to overshoot and are unstable. (Sethian, 1996b) has proposed a method which relies on a one-sided derivative that looks in the up-wind direction of the moving front, and thereby avoids the overshooting associated with finite differences :

$$\begin{aligned} (\max\{u - U_{i-1,j}, u - U_{i+1,j}, 0\})^2 &\;+ \\ (\max\{u - U_{i,j-1}, u - U_{i,j+1}, 0\})^2 &\;= \;\tilde{P}^2_{i,j} \end{aligned} \tag{5}$$

giving the correct viscosity-solution u for $U_{i,j}$. 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 action can only grow due to the quadratic equation (5).
The algorithm is detailed in 3D in next section in table 1. The *Fast Marching* technique selects at each iteration the *Trial* point with minimum action value. This technique of considering at each step only the necessary set of grid points was originally introduced for the construction of minimum length paths in a graph between two given nodes in (Dijkstra, 1959).
Thus it needs only one pass over the image. To perform efficiently these operations in minimum time, the *Trial* points are stored in a min-heap data structure (see details in (Sethian, 1996b)). Since the complexity of the operation of changing the value of one element of the heap is bounded by a worst-case bottom-to-top proceeding of the tree in $O(\log_2 N)$, the total work is about $O(N \log_2 N)$ for the *Fast Marching*

on a N points grid.
Finding the shortest path between any point p and the starting point p_0 is then simply done by back-propagation on the computed minimal action map. It consists in gradient descent on U starting from p until p_0 is reached, p_0 being its global minimum.

2.2 Extension to 3D minimal paths

We are interested in this paper in finding a curve in a 3D image. The application that motivates this problem is detailed in section 4. It can also have many other applications. Our approach is to extend the minimal path method of previous section to finding a path $C(s)$ in a 3D image minimizing the energy:

$$\int_\Omega \tilde{P}(C(s))ds \tag{6}$$

where $\Omega = [0, L]$, L being the length of the curve. An important advantage of level-set methods is to naturally extend to 3D. We first extend the *Fast Marching* method to 3D to compute the minimal action U. We then introduce different improvements for finding the path of minimal action between two points in 3D. In the examples that illustrate the approach, we see various ways of defining the potential P. Our new improvements on the method are valid as well for the 2D problem.

Similarly to previous section, the minimal action U is defined as

$$U(p) = \inf_{\mathcal{A}_{p_0,p}} \left\{ \int_\Omega \tilde{P}(C(s))ds \right\} \tag{7}$$

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 U we start from an initial infinitesimal front around p_0. The 2D scheme equation (5) is extended to 3D, leading to the scheme

$$\begin{array}{rcl}(\max\{u - U_{i-1,j,k}, u - U_{i+1,j,k}, 0\})^2 & + & \\ (\max\{u - U_{i,j-1,k}, u - U_{i,j+1,k}, 0\})^2 & + & \\ (\max\{u - U_{i,j,k-1}, u - U_{i,j,k+1}, 0\})^2 & = & \tilde{P}^2_{i,j,k}\end{array} \tag{8}$$

giving the correct viscosity-solution u for $U_{i,j,k}$. The algorithm which gives the order of selection of the points in the image is detailed in table 1.

Considering the neighbors of grid point (i, j, k) in 6-connexity, we study the solution of the equation (8). We note $\{A_1, A_2\}$, $\{B_1, B_2\}$ and $\{C_1, C_2\}$ the three couples of opposite neighbors such that we get the ordering $U_{A_1} \leq U_{A_2}$, $U_{B_1} \leq U_{B_2}$, $U_{C_1} \leq U_{C_2}$, and $U_{A_1} \leq U_{B_1} \leq U_{C_1}$. To solve the equation, three different cases are to be examined sequentially in table 2. We extend the *Fast Marching* method, introduced by (Sethian, 1996b) and used by (Cohen and Kimmel, 1997) to our 3D problem.

3 Several minimal path extraction techniques

In this section, different minimal path extraction procedures are detailed. We present new back-propagation techniques for speeding up extraction, a one end-point path extraction method to reduce the need for interaction, and a centering

```
Algorithm for 3D Fast Marching
• Definition:
    – Alive is the set of all grid points at which the action value has been reached and will not be changed;
    – Trial is the set of next grid points (6-connexity neighbors) to be examined and for which an estimate of $U$ has been computed using equation 8;
    – Far is the set of all other grid points, for which there is not yet an estimate for $U$;
• Initialization:
    – Alive set is confined to the starting point $p_0$, with $U_{p_0} = 0$;
    – Trial is confined to the six neighbors $p$ of $p_0$ with initial value $U_p = \tilde{P}(p)$;
    – Far is the set of all other grid points with $U_{\mathit{Far}} = \infty$;
• Loop:
    – Let $(i_{min}, j_{min}, k_{min})$ be the Trial point with the smallest action $U$;
    – Move it from the Trial to the Alive set (i.e. $U_{i_{min},j_{min},k_{min}}$ is frozen);
    – For each neighbor $(i, j, k)$ (6-connexity in 3D) of $(i_{min}, j_{min}, k_{min})$:
        ∗ If $(i, j, k)$ is Far, add it to the Trial set and compute $U$ using table 2;
        ∗ If $(i, j, k)$ is Trial, recompute the action $U_{i,j,k}$, and update it.
```

Table 1: *Fast Marching* algorithm

path extraction method adapted to the problem of tubular structures in images. The methods presented in this section are valid in 2D as well as in 3D and this is an important contribution that can be useful for image analysis in general.

Examples in 2D are used to make the following ideas easier to understand. We also illustrate the ideas of this section on two synthetic examples of 3D front propagation in figures 1 and 3. Examples of minimal paths in 3D real images are presented for the application described in Section 4.

3.1 Back-propagation methods

After discussing the previous back-propagation method, we show how we can limit the front propagation to a subset of the image domain, for speeding-up execution.

3.1.1 Examples of front propagation

The minimal action map U computed according to the discretization scheme of equation (7) is similar to convex, in the sense that its only local minimum is the global minimum found at the front propagation start point p_0 where $U(p_0) = 0$. The gradient of U is orthogonal to the propagating fronts since these are its level sets. Therefore, the minimal action path between any point p and the start point p_0 is found by sliding back the map U until it converges to p_0. It can be done with a simple steepest gradient descent, with a predefined descent step, on the minimal action map U, choosing $p_{n+1} = p_n - \text{step} \times \nabla U(p_n)$. More precise gradient descent methods like Runge-Kutta midpoint algorithm or Heun's method can be used for this path extraction. A simpler descent can be choosing $p_{n+1} = \min_{\{\text{neighbors of } p_n\}} U(p)$,

Algorithm for 3D Up-Wind Scheme

1. Considering that we have $u \geq U_{C_1} \geq U_{B_1} \geq U_{A_1}$, the equation derived is
   $$(u - U_{A_1})^2 + (u - U_{B_1})^2 + (u - U_{C_1})^2 = \tilde{P}^2 \tag{9}$$
   Computing the discriminant Δ_1 of equation (9) we have two possibilities
   - If $\Delta_1 \geq 0$, u should be the largest solution of equation (9);
     - If the hypothesis $u > U_{C_1}$ is wrong, go to 2;
     - If this value is larger than U_{C_1}, go to 4;
   - If $\Delta_1 < 0$, at least one of the neighbors A_1, B_1 or C_1, has an action too large to influence the solution. It means that the hypothesis $u > U_{C_1}$ is false. Go to 2;
2. Considering that we have $u \geq U_{B_1} \geq U_{A_1}$ and $u < U_{C_1}$, the new equation derived is
   $$(u - U_{A_1})^2 + (u - U_{B_1})^2 = P^2 \tag{10}$$
   Computing the discriminant Δ_2 of equation (10) we have two possibilities
   - If $\Delta_2 \geq 0$, u should be the largest solution of equation (10);
     - If the hypothesis $u > U_{B_1}$ is wrong, go to 3;
     - If this value is larger than U_{B_1}, go to 4;
   - If $\Delta_2 < 0$, B_1 has an action too large to influence the solution. It means that $u > U_{B_1}$ is false. Go to 3;
3. Considering that we have $u < U_{B_1}$ and $u \geq U_{A_1}$, we finally have $u = U_{A_1} + P$. Go to 4;
4. Return u.

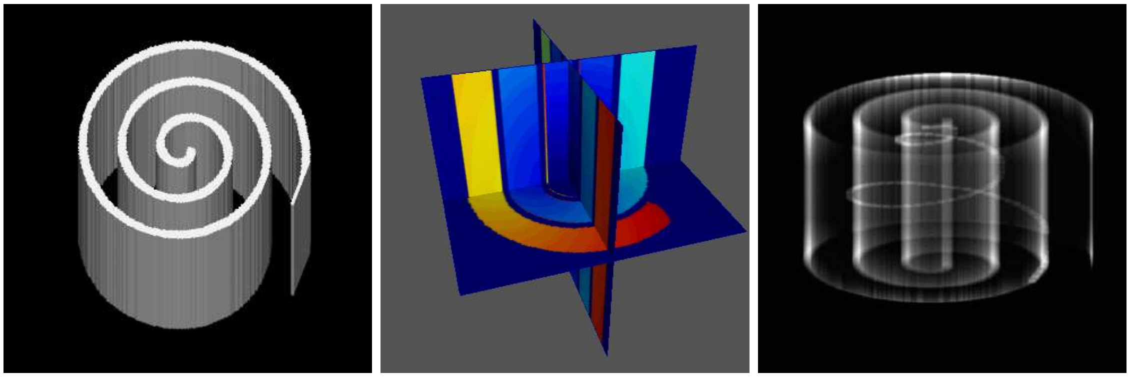
Table 2: Solving locally the upwind scheme

but it gives an approximated path in the L_1 metric. Such a descent has no more the property of being consistent. As an example, see in figure 3 the computed minimal action map for a 3D potential defined by $P(i,j,k) = 1 \ \forall (i,j,k)$.

See in figure 1-middle the action map corresponding to a binarized potential defined by high values in a spiral rendered in figure 1-left. The path found between a point in the center of the spiral and another point outside is shown in figure 1-right by transparency.

3.1.2 Partial front propagation

An important issue concerning the back-propagation technique is to constrain the computations to the necessary set of pixels for one path construction. Finding several paths inside an image from the same seed point is an interesting task, but in the case we have two fixed extremities as input for the path construction, it is not necessary to propagate the front on all the image domain, thus saving computing time. In figure 2 is shown a test on an angiographic image of brain vessels. We can see that there is no need to propagate further the points examined in figure 2-right, the path found being exactly the same as in figure 2-middle where front propagation is done on all the image domain. We used a potential $P(\mathbf{x}) = |\nabla G_\sigma * I(\mathbf{x})| + w$, where I is the original image (512^2 pixels, displayed in figure 2-left), G_σ a Gaussian filter of variance $\sigma = 2$, and $w = 1$ the weight of the model. In figure 2-right, the



Potential $P =$ spire action map with $P =$ spire 3D path in the spire

Figure 1: Examples on synthetic potentials

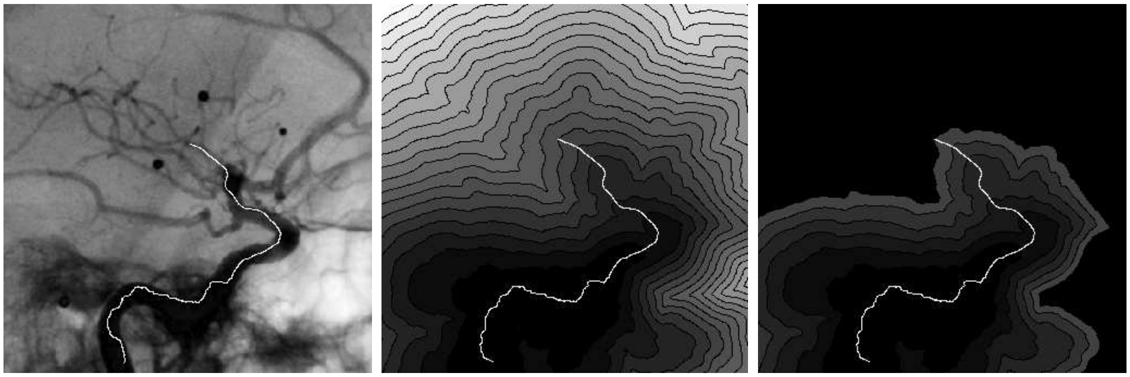


Figure 2: Comparing complete front propagation with partial front propagation method on a digital subtracted angiography (DSA) image

partial front propagation has visited less than 35% of the image. This ratio depends mainly on the length of the path tracked.

3.1.3 Simultaneous partial front propagation

The idea is to propagate simultaneously a front from each end point p_0 and p_1. Lets consider the first grid point p where those front collide. Since during propagation the action can only grow, propagation can be stopped at this step. Adjoining the two paths, respectively between p_0 and p, and p_1 and p, gives an approximation of the exact 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. Basically, it exists a real point p^*, which nearest neighbor on the Cartesian grid is p which belongs to the minimal path. Therefore, the approximation done is sub-pixel and there is no need to propagate further. This *colliding fronts* method is described in table 3.

It has two interesting benefits for front propagation:

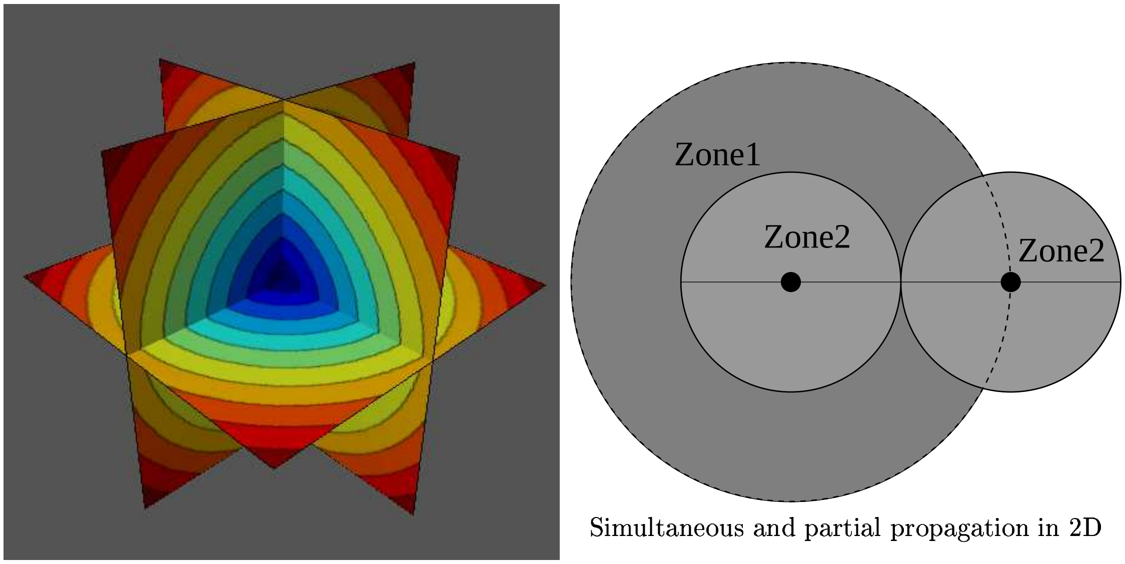
- It allows a parallel implementation of the algorithm, dedicating a processor to each propagation;

Algorithm

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

Table 3: Minimal Path as intersection of two action maps

- It decreases the number of pixels examined during a partial propagation. With a potential defined by $P = 1$, the action map is the Euclidean distance.
  - In 2D (figure 3-right), this number is divided by $\frac{(2R)^2}{2 \times R^2} = 2$;
  - In 3D (figure 3-left), this number is divided by $\frac{(2R)^3}{2 \times R^3} = 4$.



3D action map with $P = 1$

Simultaneous and partial propagation in 2D

Figure 3: 2D and 3D front propagation examples

In figure 4 is displayed a test on a digital subtracted angiography (DSA) of brain vessels. The potential used is $P(\mathbf{x}) = |I(\mathbf{x}) - C| + w$, where I is the original image (256×256 pixels, displayed in figure 4-(a)), C a constant term (mean value of the start and end points gray levels), and $w = 10$ the weight of the model. In figure 4-(b), the partial front propagation has visited up to 60% of the image. With a colliding fronts method, only 30% of the image is visited (see figure 4-(c)), and the difference between both paths found is sub-pixel (see figure 4-(d) the paths superimposed on the data).

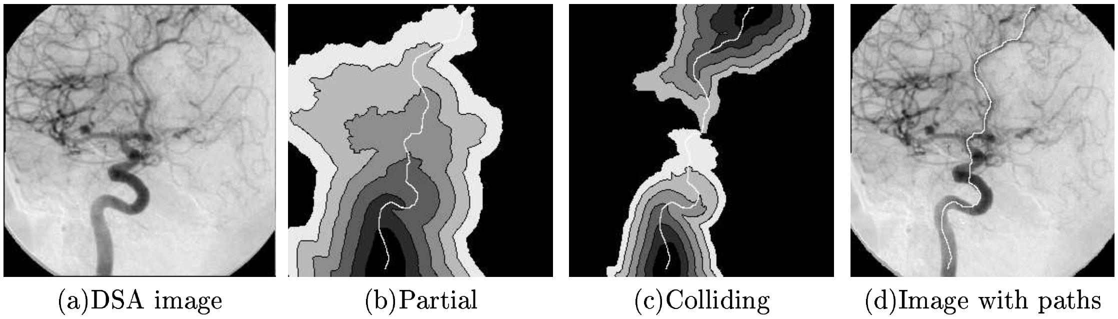

(a)DSA image (b)Partial (c)Colliding (d)Image with paths

Figure 4: Comparing the partial front propagation with the colliding fronts method on a DSA image

3.2 One end point propagation

We have shown the ability of the front propagation techniques to compute the minimal path between two fixed points. In some cases, only one point should be necessary, or the needed user interaction for setting a second point is too tedious in a 3D image. Here we derive a method that builds a path given only one end point and a maximum path length. The technique is similar to that of section 3.1.2, but the new condition will be to stop propagation when a path corresponding to a chosen Euclidean distance is extracted.

As we explain below, we can compute simultaneously at each point the energy U of the minimal path and its length. We choose as end point the first point for which the length of the minimal path has reached a given value. Since the front propagates faster along lower values of Potential, interesting paths are longer for a given value of U.

An example of this path length condition is shown on figure 5 which is a DSA image of brain vessels. Propagating a front with potential $P = 1$ computes the Euclidean distance to the start point. This is obvious from definition (3), and we can see its illustration with figure 3-left. Therefore, we use simultaneously an image-based potential P_1, for building the minimal path and a potential $P_2 = 1$ for

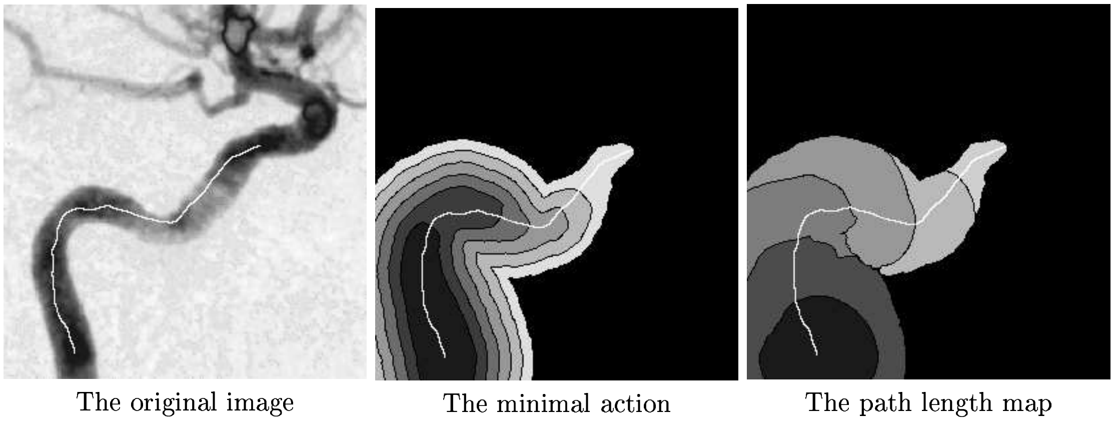

The original image The minimal action The path length map

Figure 5: Computing the Euclidean path length simultaneously

computing the path length.

While we are propagating the front corresponding to P_1 on the image domain, at each point p examined we compute both minimal actions for P_1 (shown in figure 5-middle) and for P_2 (shown in figure 5-right). In this case the action corresponding to P_2 is an approximate **Euclidean length of the minimal path** between p and p_0. Although this length is an approximation, it is still a good estimation since it makes use of the same Eikonal equation scheme. The main advantage of doing so is that it does not add much computation time to the algorithm.

Note that this Euclidean path length is discontinuous and must be smoothed in order to be used in a robust manner.

3.3 The path centering method

The path is the set of locations that minimize the integral of the potential in equation (2). If the potential is constant in some areas, it will lead to the shortest Euclidean 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 in figure 6, and would not be tuned for a problem of finding the optimal trajectory for virtual endoscopy, which may require a centered path. In this section we derive a technique to track centered paths, in the case we have segmented the considered structure. Using front propagation techniques, we compute the distance to the edges of the region of interest and we use the opposite of this distance in our front propagation in the potential, thus attracting the minimal path in the center of the regions. This technique is general and can be used as well for other problems in Image Analysis where a centered path is needed, like road detection for example.

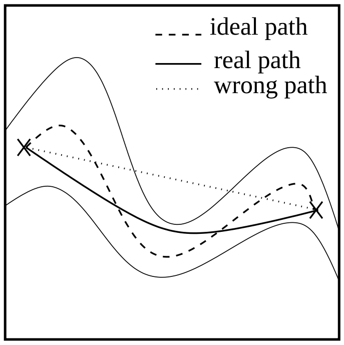


Figure 6: Problem of path centering

3.3.1 Extracting the edges

Several edge detection operator can be used, but in some cases a *Fast Marching* can provide the necessary edge information. It has been already used for segmentation in (Malladi and Sethian, 1998). This allows us to include the segmentation step in the same framework as our minimal path finding.

Having searched for the minimal action path between two given points, using a partial front propagation (see subsection 3.1.2), the algorithm provides different sets of points:

- the points whose action is set and labeled *Alive*;
- the points not examined during the propagation and labeled *Far*;
- the points at the interface between *Alive* and *Far* points, whose actions are not set, and labeled *Trial*.

This last category, the border of the visited points, is a contour in 2D and a surface in 3D which defines a connected set of pixels or voxels. If the potential is a lot higher along edges than it is inside the shape, the edges will act as an obstacle to the propagation of the front. Therefore, the front propagation can be used as a segmentation procedure, recovering the object shapes. In this case the *Trial* points define a surface which can be described as a rough segmentation of a part of the vessel. Once the front has reached the endpoint, we use the front itself to define its shape.

3.3.2 The centering potential

Having obtained this interface of *Trial* points, we can use a new front propagation procedure to compute the distance to the edges inside the object considered. The potential and the initial action for this second front propagation are defined as follows

$$\begin{array}{rl} P(i,j) = 1 & \forall (i,j) \text{ inside the shape} \\ P(i,j) = \infty & \forall (i,j) \text{ outside the shape} \\ U(i,j) = 0 & \forall (i,j) \in \{\mathit{interface}\} \\ U(i,j) = \infty & \text{ elsewhere} \end{array}$$

When the distance map, said $\mathcal{E}$, is computed, it is used to create a potential P which weights the points in order to propagate faster a new front in the center of the desired regions. Choosing a value d to be the minimum acceptable distance to the walls, we propose the following potential:

$$P_1(\mathbf{x}) = \max\left(d - \mathcal{E}(\mathbf{x}); 0\right)^{\gamma} \tag{11}$$

The complete method is described in figure 7. The first step is to propagate the front from the start to the end point. The second step is to store the front position at the end of the first step. The third step is to propagate inward the front towards the center of the object with a uniform potential $P = 1$. This computes the distance to the boundary front. The fourth step is to propagate faster in the center of the object, using the new potential (11) based on the distance computed during the third step. The final step is to make back-propagation from the end point using the last action map. The extracted path is now centered inside the object.

As an illustration, a test is proceeded on a DSA image of the brain vessels shown in figure 8-left. In figure 8-center is shown the result obtained using a potential

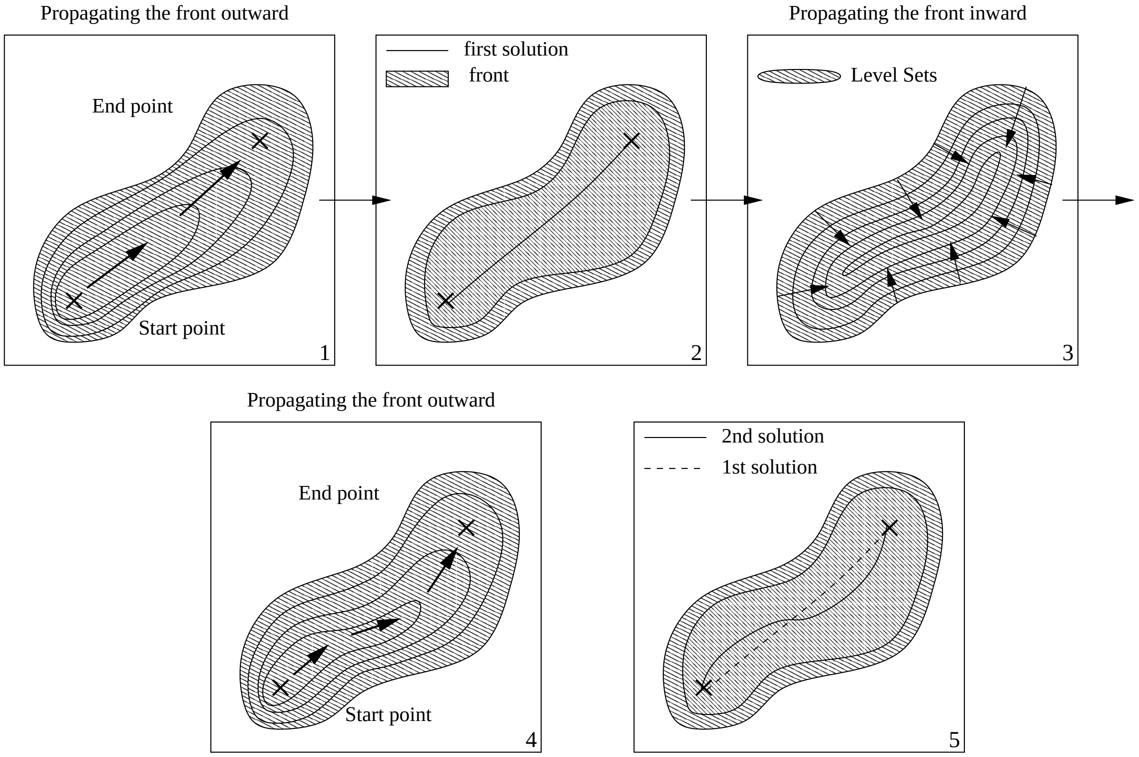
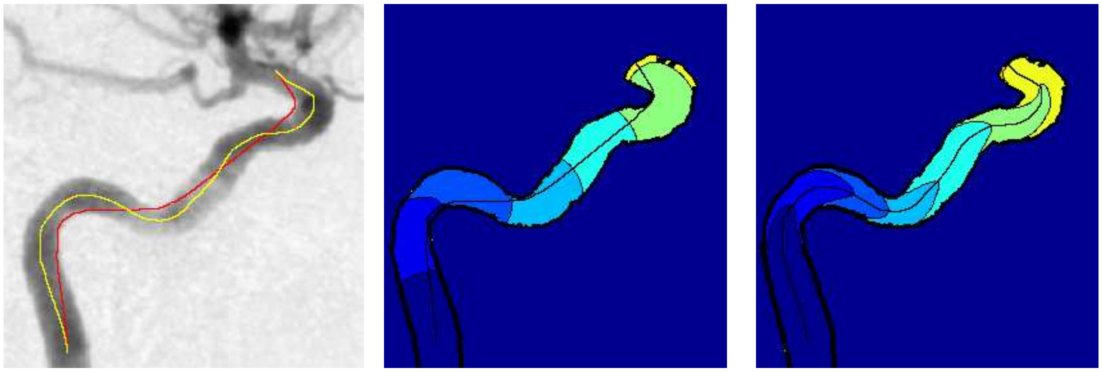


Figure 7: Centering the path inside the object

based on the image, where the shortest path is tangential to edges. But the front propagates only along the vessel direction, and is rapidly stopped transversally, allowing to compute the distance to the walls. Defining a new potential according to equation (11) based on this distance map, the second front propagates faster in the center of the vessel. Due to the shape of the iso-action lines of the centered minimal action shown in figure 8-right, the path avoids the edges and remains in the center of the vessel. We will present results on real 3D data in subsection 4.2 applied to the problem of virtual endoscopy (see figure 16).

3.3.3 Remarks on the method

At this step, it seems important to compare the results obtained with this front propagation with another centered line extraction technique, the skeletonization, and particularly the definition of the medial axis function of (Blum, 1967) which treats all boundary pixels as point sources of a wave front. Considering that the *Fast Marching* computes the Euclidean to an arbitrary set of points using a potential $P = 1$, it is similar to a skeletonization. But an inherent benefit of using the *Eikonal* scheme instead of another method is to be a very fast re-initialization technique: the border points are managed by a min-heap data structure, (as detailed in B). Then at the end of the first partial front propagation, the edge points are already stored in this data structure, and it can immediately compute the distance inside the object by just modifying the potential.



The two paths With a normal potential With a centering potential

Figure 8: Comparing classic and centered paths

The main advantage of our approach is that we find only one path while skeletonization is very unstable and creates a lot of undesired small parts.

Another benefit of our method is to extract more than the exact central line of the tubular structures: the purpose of our application is to have a smooth line which always stays inside the tubular object and which is far from the edges. This is motivated by the application to virtual endoscopy (see next section). Then our method comprises steps of segmentation and path extraction, and achieves them in a very fast way. Therefore, it cannot be replaced by a simple medial-axis transform.

Our method can be seen as a way to compute the skeleton of a tubular shape without the problems encountered in most skeletonization methods.

4 Application to Virtual Endoscopy

In previous sections we have developed a series of issues in front propagation techniques. We study now the particular case of virtual endoscopy, where fast extraction of centered paths in 3D images with minimum user interactivity is required.

4.1 Targets for Virtual Endoscopy

Visualization of volumetric medical image data plays a crucial part for diagnosis and therapy planning. The better the anatomy and the pathology are understood, the more efficiently one can operate with low risk. Different possibilities exist for visualizing 3D data: three 2D orthogonal views (see figure 9), maximum intensity projection (MIP, and its variants), surface and volume rendering. In particular, virtual endoscopy allows by means of surface/volume rendering techniques to visually inspect regions of the body that are dangerous and/or impossible to reach physically with a camera (e.g behind an airway stenosis or obstruction, or too small). A virtual endoscopic system is usually composed of two parts:

1. A Path construction part, which provides the successive locations of the fly-through in the tubular structure of interest (see figure 10-left);

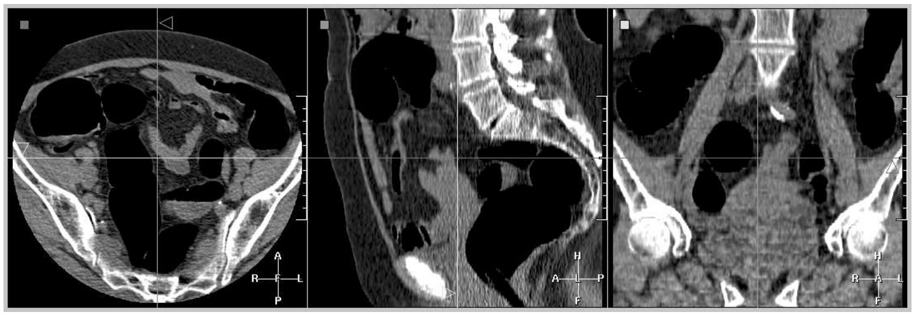
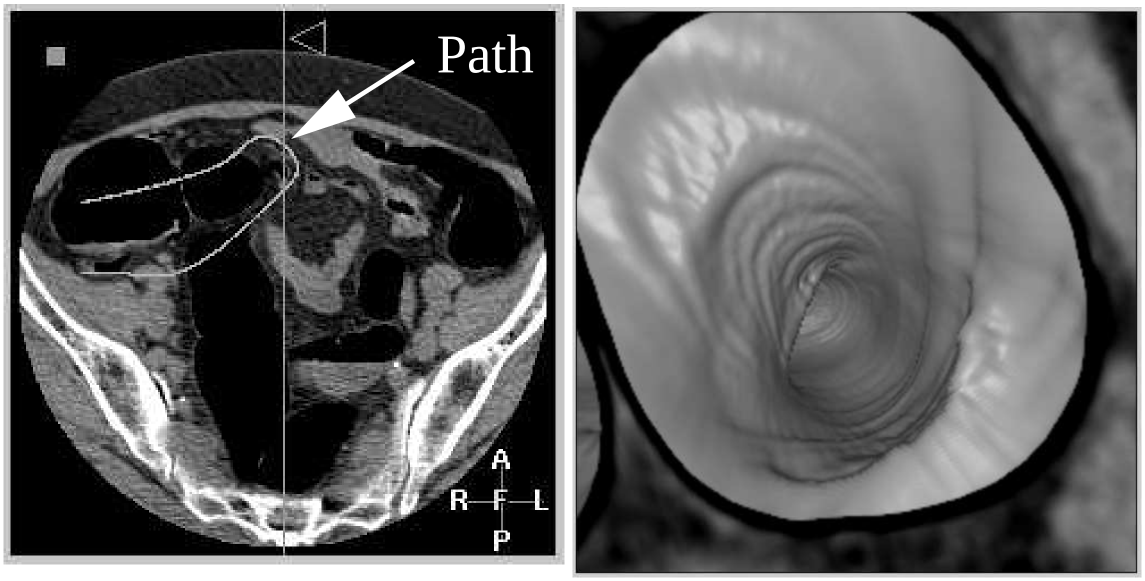


Figure 9: Three orthogonal views of a volumetric CT data set of the colon

2. Three dimensional interior viewing along the endoscopic path. Those views are adjoined creating an animation which simulates a virtual fly-through through them (see figure 10-right).



Original CT slice + path

Endoscopic view

Figure 10: Interior view of a colon, reconstructed from a defined path

A major drawback in general remains when the path construction is left to the user who manually has to "guide" the virtual endoscope/camera. The required interactivity can be very tedious for complex structures such as the colon for example (see figure 11). Actually, on most clinical platforms the user must define all path points manually, using for example three 2D orthogonal views, as shown in figure 9, leading to problems as the following:

- Since the anatomical objects have often complex shapes, they tend to pass in and out of the three orthogonal planes. Consequently the right location is accomplished by successively entering the projection of the desired point in each of the three planes;

- The path is approximated between the user defined points by lines or Bezier splines. If the number of points is not enough, it can easily cross an anatomical wall.

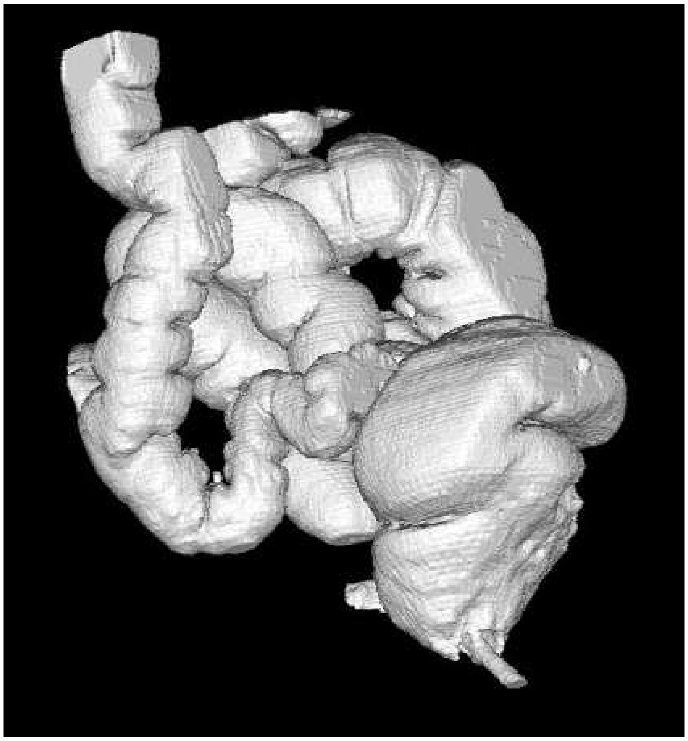


Figure 11: The complex shape of the colon

Path construction in 3D images is thus a very critical task and precise anatomical knowledge of the structure is needed to set a suitable trajectory, with the minimum required interactivity.

Numerous techniques try to automate this path construction process by using a skeletonization technique as a pre-processing (Yeorong *et al.*, 1993). It requires first to segment the object in order to binarize the image, then it extracts the skeleton of this volume. The skeleton often consists in lots of discontinuous trajectories, and post-processing is necessary to isolate and smooth the final path. But those methods can lead to critical cases: if there is a stenosis in the tubular structure, the binarization can produce two separate objects, where a skeletonization is inefficient. The front propagation techniques studied in this paper propose an alternative to the tedious manual path construction by building paths in 3D images with minimum interactivity. In contrast to other methods, it does not require any pre- or post-processing.

It is sometimes necessary to smooth the path extracted by the front propagation. The point of view in the volume rendering of the tubular structure is very important, because it constrains the result of the examination. Thus, during the virtual fly through, the point of view of the camera must change smoothly. Traditionally, the position of the virtual camera frame at a particular path point is orthogonal to the path. If the path is not smooth, the point of view of the virtual camera will change in an abrupt manner. There are two ways to achieve this regularization:

- by modifying the view angle of the virtual camera, being non more orthogonal to the path, but looking in the direction of a path point which is located far from its current position (see figure 12), or using a running average of the local direction of the camera;
- by increasing the weight w in equation (2) since it has a smoothing effect on the minimal path (see (Cohen and Kimmel, 1997) for details). We preferred to use this technique in the following examples, since it is efficient and very simple to add.

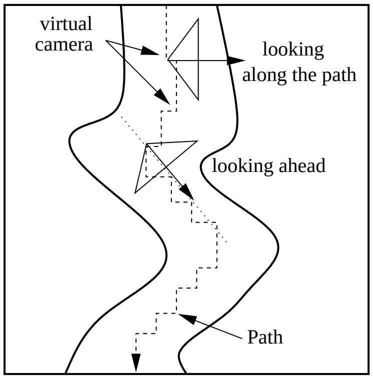


Figure 12: Examples on synthetic potentials

We first apply the minimal path construction to the case of virtual endoscopy in the colon in section 4.2, then we extend this technique to other anatomical shapes in section 4.3.

4.2 Building a potential for virtual colonoscopy

All tests are performed on a volumetric CT scan of size 512 × 512 × 140 voxels, shown in figure 9. The grey level range is between 0 and 1500. The target is to build a potential P with the 3D data set allowing paths to stay inside the anatomical shapes where end points are located. We thus define the potential by a general model $\tilde{P}(x) = |I(x) - I_{mean}|^{\alpha} + w$.

Firstly, the potential must be lower inside the colon in order to propagate the front faster, and to avoid problems with crossing the edges of the anatomical object. In a colon CT scan, an average position I_{mean} of the grey level colon in the histogram can be defined (see figure 13) as a peak in the histogram where $I_{mean} = 200$. Secondly, if the path to be extracted is very long, the situation can lead to pathological cases, and the front can go through potential *walls*. This is frequent for large objects that have complex shapes and very thin edges, as colon. Then, edges should be enhanced

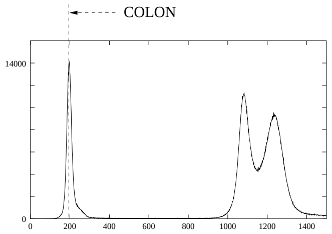
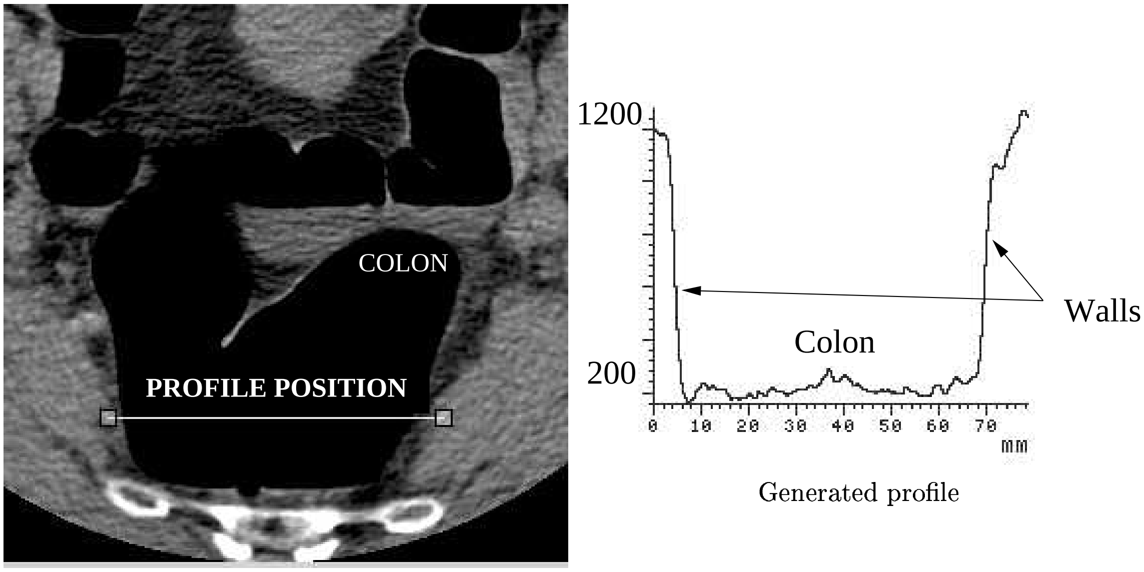


Figure 13: Localization of the colon in the histogram

to enable long trajectories, with a non-linear function. We thus take $\alpha = 2$ in order to enhance the dynamic of the image with a quadratic function.

However, this potential does not produce paths relevant for virtual endoscopy. Indeed, paths should remain not only in the anatomical object of interest but as far as possible from its edges. In order to achieve this target, we use the centering potential method as detailed in section 3.3. We first need to obtain a *shape* information. In fact, a CT scan of the colon contains already a shape information sufficient to constrain a front propagation. In figure 14-left is shown a slice of a colon volumetric data set. Figure 14-right shows the profile along the line drawn in figure 14-left. Air fills the colon and is represented in our CT image by a grey intensity around 200 (see figure 14-right), while edges are defined by a grey intensity around 1200. Then, using the potential $\tilde{P}(x) = |I(x) - I_{mean}|^{\alpha} + w$, the front obtained through *Fast Marching* is stopped by the anatomical shapes, as seen in figure 15. It illustrates the fact that the *Fast Marching* can act also as a segmentation tool, as noted in section 3.3.



Generated profile

Slice of a colon volumetric CT scan

Figure 14: Profile of the colon volume

In figure 16 we show the result of applying this new method to colonoscopy. The

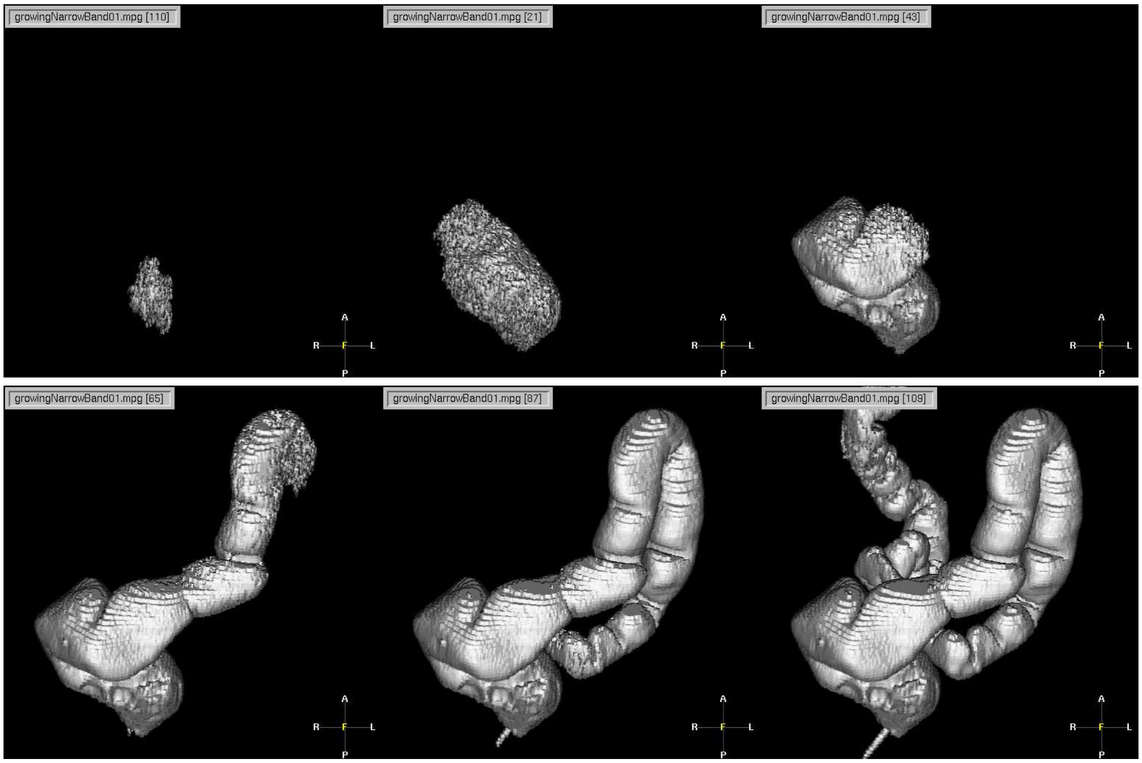
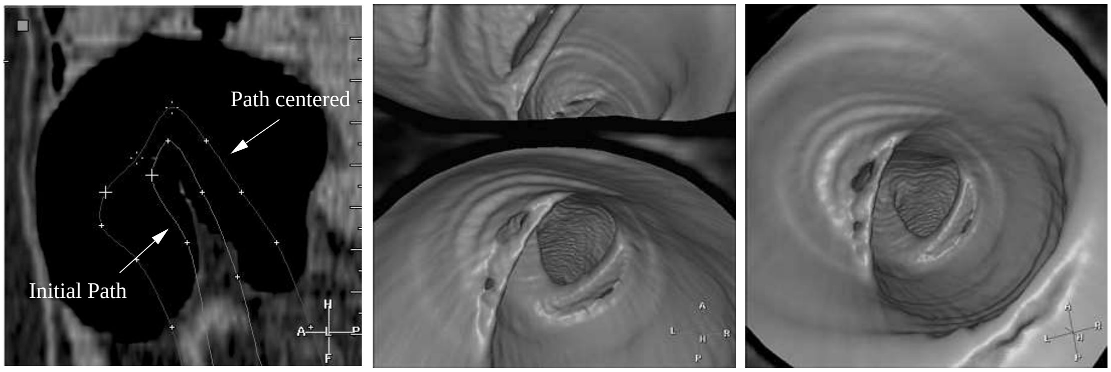

growingNarrowBand01.mpg [110]
growingNarrowBand01.mpg [21]
growingNarrowBand01.mpg [43]
growingNarrowBand01.mpg [65]
growingNarrowBand01.mpg [87]
growingNarrowBand01.mpg [109]
A
R
F
L
P

Figure 15: Propagating inside the colon volume

edges are obtained via a first propagation: in figure 15 we can see the evolution of the *narrow band* during propagation. It gives a rough segmentation of the colon and provides a good information and a fast-reinitialization technique to compute the distance to the edges.

Using this distance map as a potential (from equation (11)) that indicates the distance to the walls, we can correct the initial path as shown in figure 16-left: the new path remains more in the middle of the colon. The two different figures 16-middle and 16-right display the view of the interior of the colon from both paths at the u-turn shown in figure 16-left. With the initial potential, the path is near the wall, and we see the u-turn, whereas with the new path, the view is centered into the colon, giving a more correct view of the inside of the colon. The new centered path is smooth because this final propagation is done on a synthetic potential (the distance to the walls) where noise has been removed.

Therefore, the two end points can be connected correctly, giving a path staying inside the anatomical object. But for virtual colonoscopy, it is often not necessary to set the two end points within the anatomical object. The colon being a closed object with two extremities, we can use the Euclidean path length stopping criterion as explained in subsection 3.2. The figure 15 shows the front propagation in the *Fast Marching* technique with a starting point belonging to the colon and an Euclidean path length criterion of 500 mm. The image resolution is 1mm for x and y axes and 4mm for the z axis. Figure 17 shows the minimal path obtained. Figure 18 shows



The two different paths with a normal potential with a centering potential

Figure 16: Centering the path in the colon

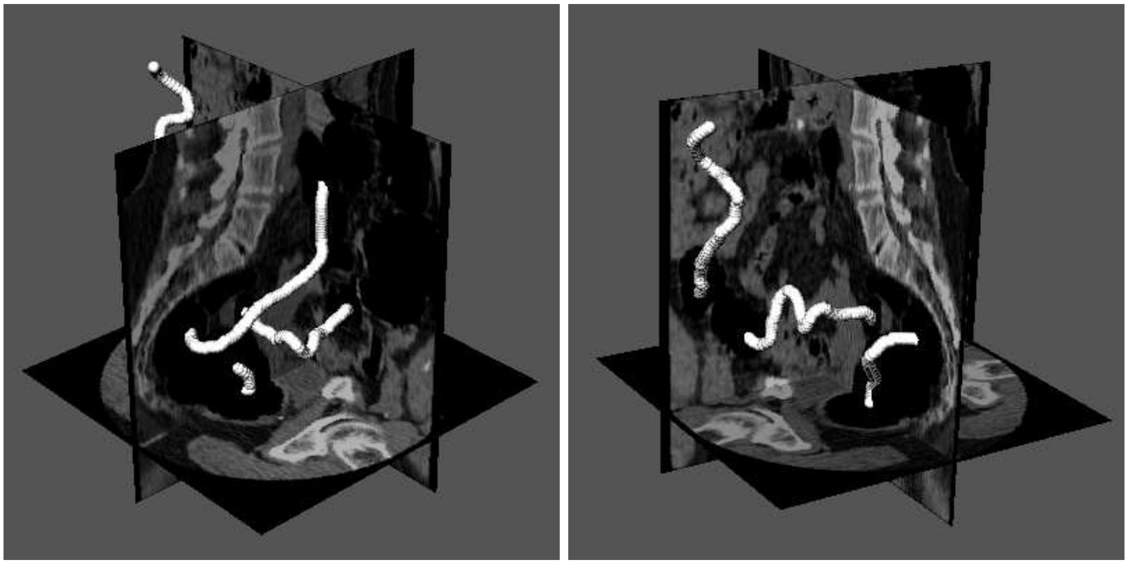


Figure 17: Views of the path inside the colon volume

rendered views from a few points along the path[1].

Interaction is limited to setting the start point for front propagation and choosing the minimal path length (as explained in subsection 3.2). It takes 30 seconds of computing time for building the complete path on an Ultra 30 (with 300 MHz CPU and 1 Go RAM), comprising steps of segmentation of the colon and calculation of the distance to the walls in order to center the path as detailed in subsection 3.3. The complete virtual fly-through renderings (300 images) are computed in approximately 10 minutes (the rendering is a tool included in the EasyVision workstation developed by Philips Medical Systems).

[1] A video of this virtual fly through is available at http://www.ceremade.dauphine.fr/~cohen/MPEG/colon.mpg

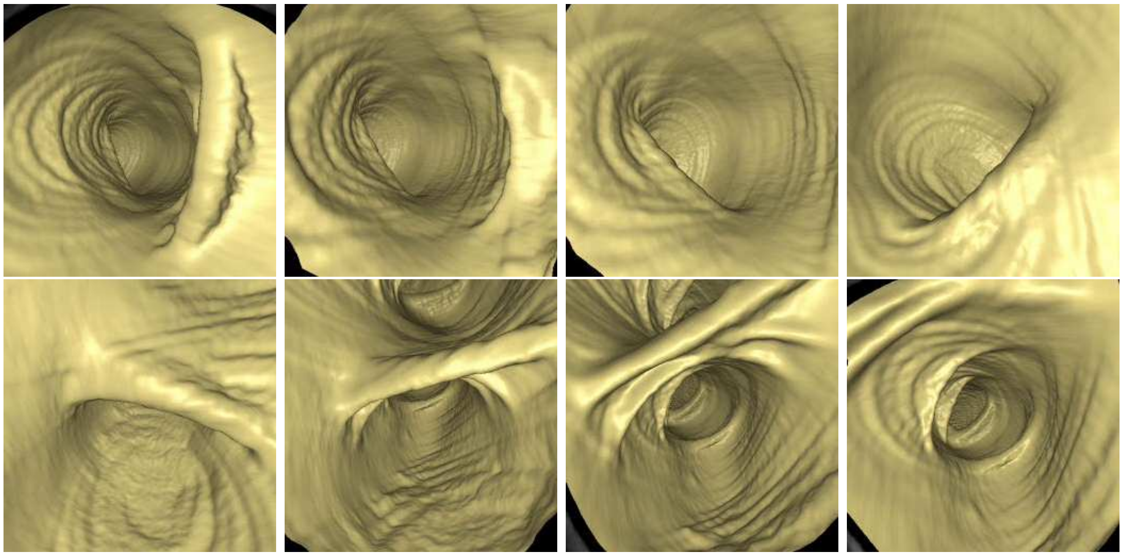


Figure 18: Virtual fly-through in the colon dataset

4.3 Results on Other Anatomical Objects

Trachea CT scan

Extracting paths inside the trachea is the same problem as in the colon. The dataset used is shown in figure 19 by means of three orthogonal slices of the volume displayed together with a path extracted. Air fills the object and give a shape information all along from throat to lungs. Therefore, the anatomical object having a very simple shape, the path construction with one or two fixed points is easier than in the colon case. One example path tracks the trachea, using a nonlinear function of the image

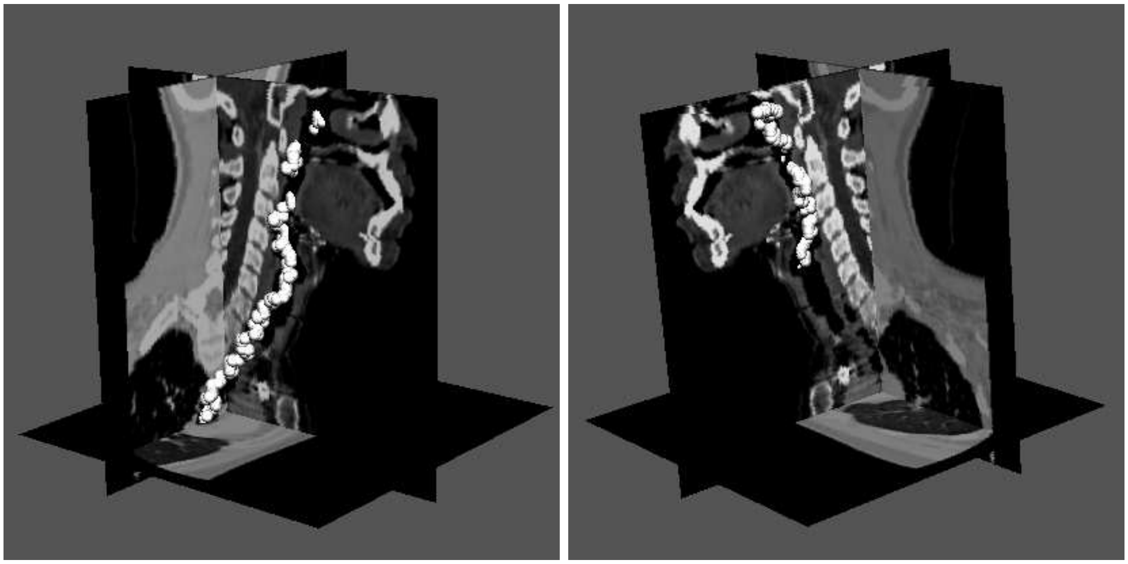


Figure 19: Views of the path inside the trachea

grey levels ($\tilde{P}(x) = |I(x) - 200|^2 + 1$). Two views of an extracted path in 3D are displayed in figure 19 together with 3 orthogonal slices of the dataset. An endoscopic view along the path[2] is displayed in figure 20.

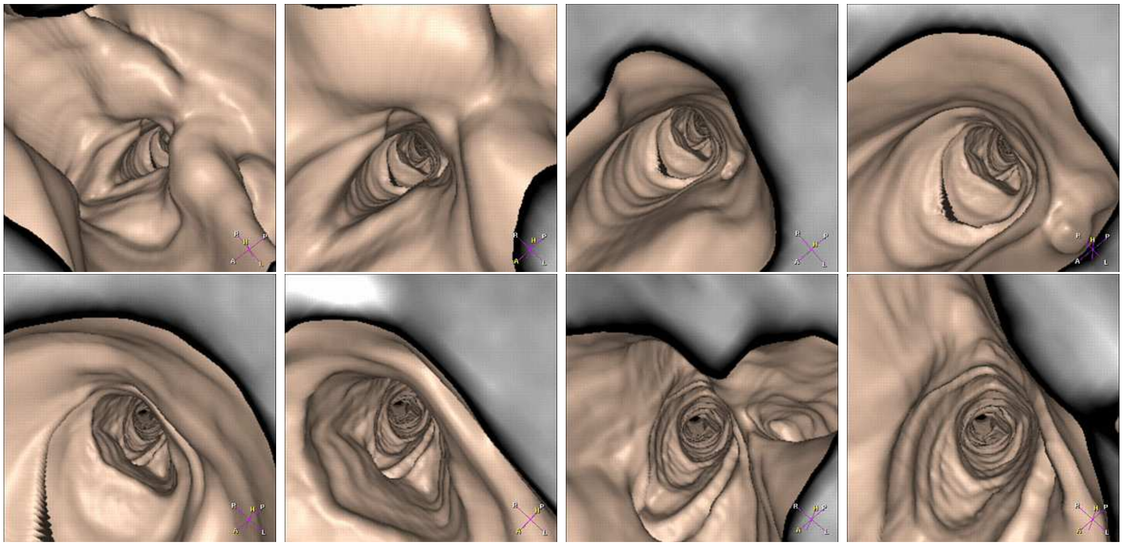


Figure 20: Virtual fly-through in the trachea dataset

Brain magnetic resonance angiography (MRA) image

Tests were performed on brain vessels in a MRA scan. Three orthogonal slices of this dataset are shown in figure 21 together with a path extracted.

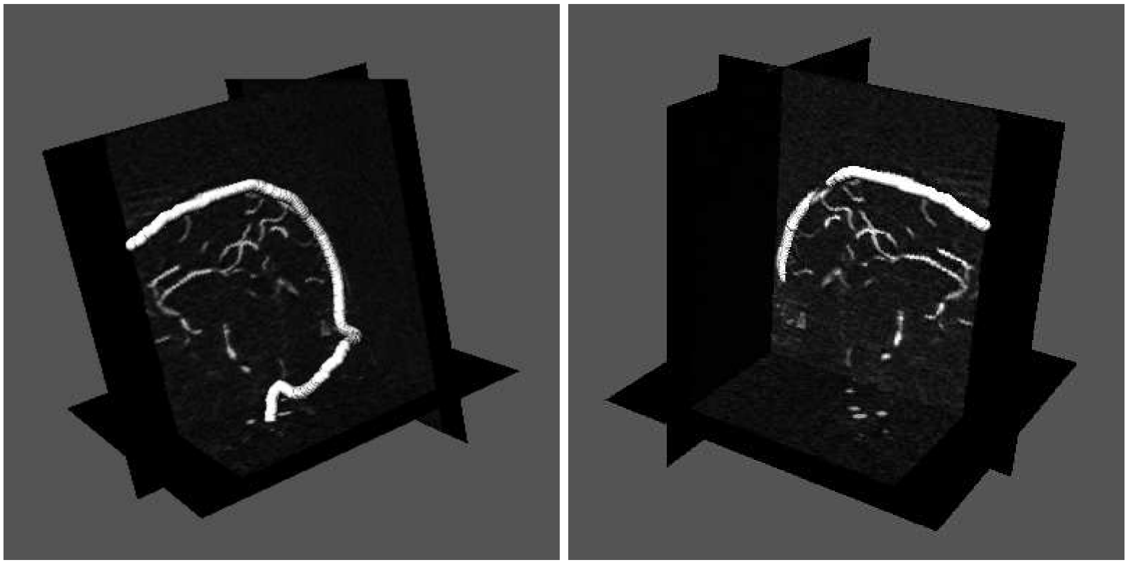


Figure 21: Views of the path inside a brain vessel

[2] A video of this virtual fly through is available at http://www.ceremade.dauphine.fr/~cohen/MPEG/trachea.mpg

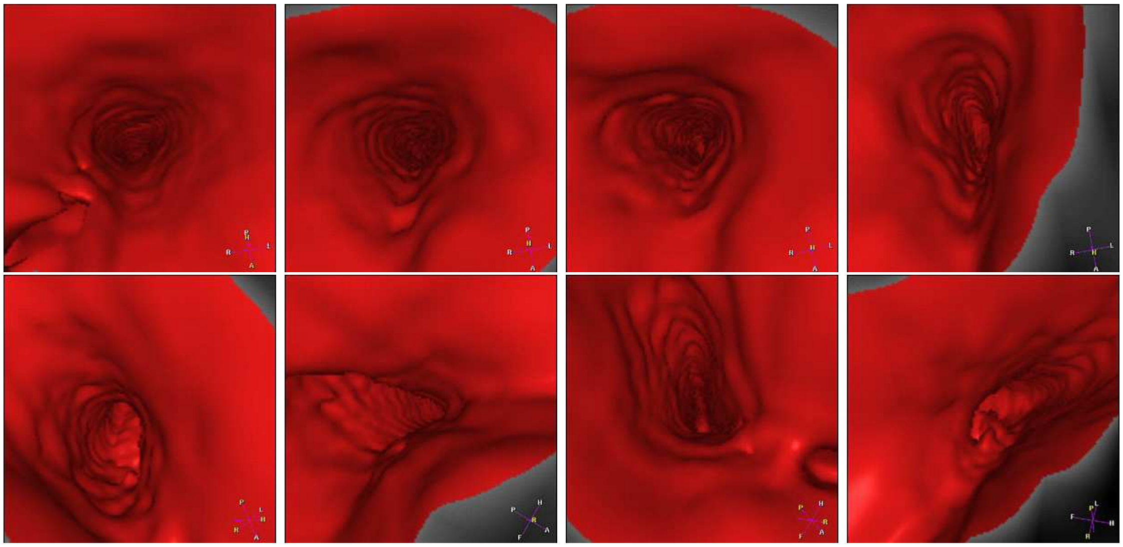


Figure 22: Virtual fly-through in the brain dataset

The problem is different, because there is only signal from the dye in the cerebral blood vessels. All other structures have been removed. The main difficulty here lies in the variations of the dye intensity. The example path tracks the superior sagittal venous canal, using a nonlinear function of the image grey levels ($\tilde{P}(x) = |I(x) - 100|^2 + 1$). Two views of the extracted paths in 3D are displayed in figure 21 together with 3 orthogonal slices of the dataset. A sample of the virtual fly-through[3] along the brain vessel is displayed in figure 22.

Aorta MR scans

A test was made on an aorta MR dataset, shown in figure 23. The propagation measure is based on a nonlinear function of the intensity of the contrast solution that fills the aorta. This data set is difficult since the intensity of the contrast product will vary along the aorta (the contrast bolus dilutes during the acquisition time). Due to this non-uniformity, paths can cross other anatomical structures with similar intensities, if the mean value inside the aorta is not set correctly by the user.

Our example path tracks one illiaca, using the potential $\tilde{P}(x) = |I(x) - 1000|^2 + 10$ in the MR scan. The dataset contains noise, and we must use an important weight to smooth the extracted paths. We have displayed a sample of the endoscopic views[4] of the aorta along the path in figure 24.

5 Conclusion

In this paper we presented a fast and efficient algorithm that computes a path useful for guiding endoscopic viewing that only depends on a start and end point. This work was the extension to 3D of a level-set technique developed in (Cohen and Kimmel,

[3] A video of this virtual fly through is available at http://www.ceremade.dauphine.fr/˜cohen/MPEG/brain.mpg

[4] A video of this virtual fly through is available at http://www.ceremade.dauphine.fr/˜cohen/MPEG/aorta.mpg

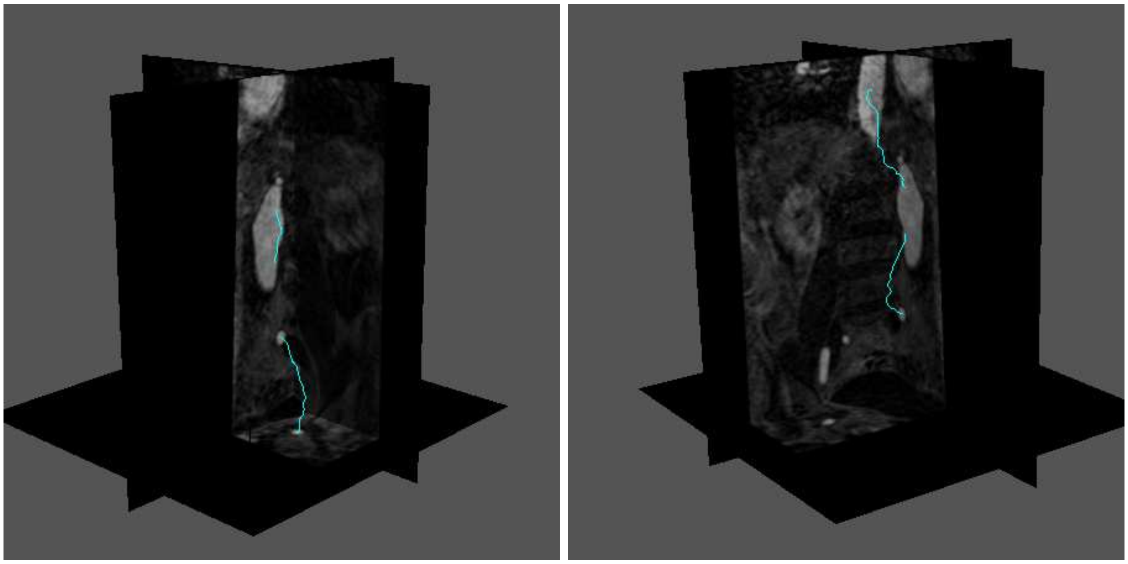


Figure 23: Views of the path inside the MR dataset of the aorta

1997) for extracting paths in 2D images, given only the two extremities of the path and the image as inputs, with a front propagation equation. We improved this front propagation method by creating new algorithms which decrease the minimal path extraction computing cost, and reduce user interaction in the case of path tracking inside tubular structures.

The results are promising for several clinical applications, including those with very complex shapes. The success of the tracking approach for objects with non-uniform grey-level contrast critically depends on the design of the cost function. We have proved the benefit of our method towards manual path construction, and skeletonization techniques, showing that only a few seconds are necessary to build a complete trajectory inside the body, giving only one or two end points and the image as input.

Acknowledgments

We thank Drs Jean-Michel Létang and Sherif Makram-Ebeid for fruitfull collaborations, and very interesting discussions. And we thank Jean Pergrale, group leader of the Medical Imaging Systems Group at Laboratoires d'Éléctronique Philips, for constant support.

A Level Sets Methods in Active Contouring

In this appendix, we discuss the problems of the classical approach in active contouring, introduced in (Kass *et al.*, 1988), towards the improvements brought by the level-set formalism. We examine the theory of level-sets methods introduced in (Sethian, 1996a). A link is made with the theory of deformable models and the

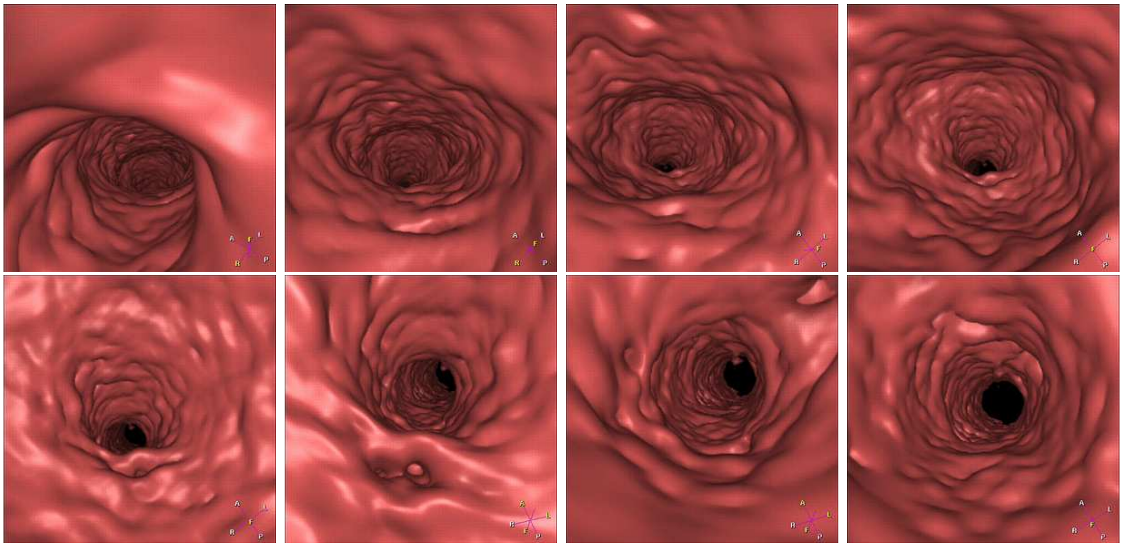


Figure 24: Virtual fly-through in the aorta MR dataset

problem of finding a minimal path in an image.

A.1 Classical Deformable Model

A.1.1 Definition

For a general overview on deformable models, see (McInerney and Terzopoulos, 1996). See also a description and references in (Cohen, 1995) and (Cohen and Kimmel, 1997).

The deformable model C is a mapping

$$\begin{array}{rcl} C:\Omega & \rightarrow & I\!\!R^3 \\ s & \rightarrow & \mathbf{x}(s)=(x(s),y(s),z(s)) \end{array}$$

where $\Omega=[0,1]$ is the parameterization interval. We can define the energy E of the model as a functional which will be minimized on the space $\mathcal{A}$ of all possible curves. A general model can be written for $C\in\mathcal{A}$ and $E:\mathcal{A}\rightarrow I\!\!R$:

$$E(C)=\int_\Omega\left\{\frac{w_1}{2}\|C'(s)\|^2+\frac{w_2}{2}\|C''(s)\|^2+P(C(s))\right\}ds \tag{12}$$

In this formulation, each term appears as a potential acting on the shape. Thus the mechanical properties of the deformable model are controlled by two kinds of constraints:

- The internal potential: C' and C'' are the smoothing terms on the curve. They enable to control its regularity by means of w_1 which quantifies its rigidity and w_2 its elasticity;

- The external potential term P represents the likelihood. This “image” potential traps the curve towards the regions with desired attributes.

A.1.2 Drawbacks

The main drawbacks of the classical deformable model approach are:

- Minimization: The functional is non-convex, and one difficulty is to find a good local minimum. Spurious edges generated by noise may stop the evolution of the surface, giving an insignificant local minimum of the energy;
- Initialization: The user must specify an initial shape that is close to the goal, like a very precise polygon approximation, which may be tedious to draw;
- Topology changes: this method is unable to segment several objects simultaneously, and to merge different shapes;

One of the main issue in using deformable models is their initialization and minimization. The solution introduced in (Sethian, 1996b) allows to solve the global minimization of a problem. His approach considers a slightly modified problem: a curve is considered as an interface between two media, following a particular evolution equation. We will see that under precise assumptions, this front evolution efficiently builds a path between two fixed points, as detailed in (Cohen and Kimmel, 1997).

A.2 Front propagation model

This new formulation of the active deformable model into a front propagating in an image will allow to work with objects of arbitrary topologies. The surface is considered as an interface, often called front, between two media, which is a formalism often used in physics. Therefore, from the variational formulation of the classic deformable models, we can differentiate a new variational expression related to the interface model.

A.2.1 Front propagation approach

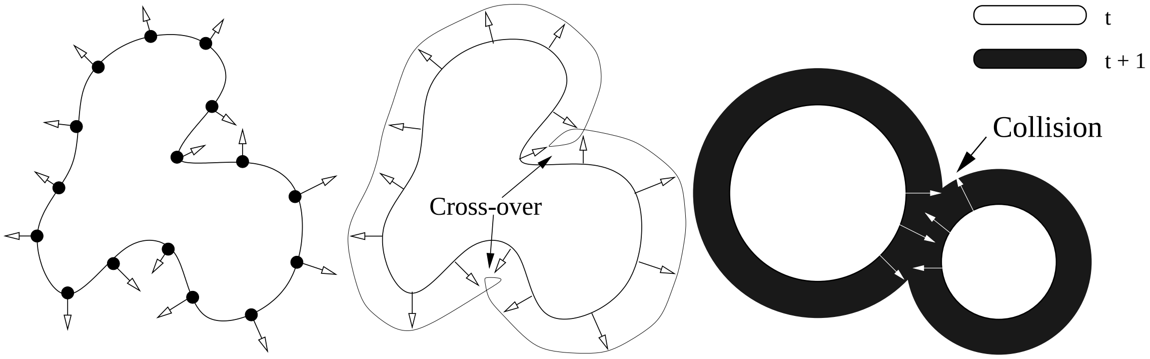
Considering a surface $\mathcal{L}$ as an interface between two media, we allow this interface to move under the influence of internal and external normal forces. The sum of this forces being F, the expression of the evolution of this interface is given by

$$\frac{\partial \mathcal{L}(v,t)}{\partial t} = F \vec{n}(v,t) \tag{13}$$

starting from an initial front, v being the position parameter and t being time. This evolution equation can be derived from an energy formulation in terms of potentials as in equation (12). The equivalence between the potential terms and the forces is recovered via the associated Euler-Lagrange equation of the energy.

Defining a potential P for this interface in the image domain, we propagate with a velocity F proportional to the inverse of the potential, to have high velocities in low cost areas, and vice-versa.

But this formulation has several drawbacks. When tracking the motion of an interface $\mathcal{L}$ propagating along its normal direction with velocity $F = \frac{1}{P}$, most numerical techniques rely on markers, breaking it up into points connected by segments, and moving each point with speed F (see figure 25-(a)). The solution is supposed to gain in accuracy if the number of points is increased. Problems arise if different parts of the front cross each other, or if the shape tries to break into two pieces (see figure 25-(b) and reference (Sethian, 1996b) for details) or if two shapes try to merge into one (see figure 25-(c)).



(a) Markers technique (b) Cross over (c) Cannot merge two fronts

Figure 25: Markers methods and related problems in 2D

A.2.2 Level-sets method to propagate interface

We can reformulate the interface evolution of equation (13) by embedding the front in a higher dimensional function. This allows to overcome numerical difficulties and to handle topology changes.

Rather than evolving the interface $\mathcal{L}$, the level set method embeds this propagating interface in a higher-dimension interface ϕ which can be defined as follows: $\phi(\mathbf{x}, t = 0) = d$ the signed distance from $\mathbf{x}$ to $\mathcal{L}$ (negative inside $\mathcal{L}$, positive outside). See Figure 26 for the representation of the embedding in a synthetic 2D example.

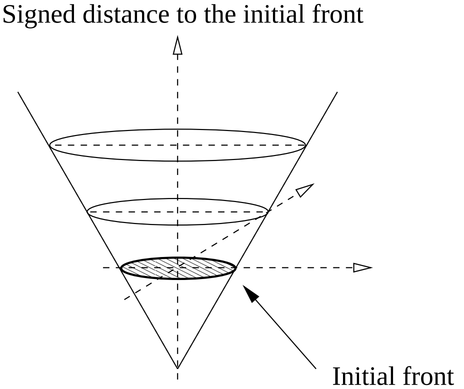


Figure 26: Embedding a 2D interface in the signed distance function

Therefore, $\mathcal{L}$ is the set of point for which ϕ is zero: it is the zero level set of this higher-dimension interface. Assuming that the zero level-set of the evolving surface

ϕ always matches the propagating hypersurface $\mathcal{L}$ means that

$$\phi(\mathcal{L}(t), t) = 0$$

We can differentiate this expression, leading to:

$$\frac{\partial \phi(\mathcal{L}, t)}{\partial t} + \nabla \phi(\mathcal{L}, t) . \frac{\partial \mathcal{L}}{\partial t} = 0 \tag{14}$$

Then, the surface $\mathcal{L}$ evolution is implicitly described by that of the evolving surface ϕ. Having $\overrightarrow{n} = \nabla \phi / \|\nabla \phi\|$, equation (12) and equation (14) lead to

$$\frac{\partial \phi}{\partial t} + F \|\nabla \phi\| = 0 \tag{15}$$
$$\text{given initial } \phi(\mathbf{x}, t = 0)$$

The evolution of the level set equation is a special case of the so-called Hamilton-Jacobi Equation (see (Sethian, 1996a)). In this setting, curvatures and normals may be easily evaluated and topological changes occur in a natural manner. This equation is solved using entropy-satisfying schemes borrowed from the numerical solution of hyperbolic conservation laws which produce the correct viscosity solutions. For details, see (Sethian, 1996b).

A.3 Paths of Minimal Action

We give here some remarks and comments on the minimal path approach described in section 2 and introduced for 2D in (Cohen and Kimmel, 1997).

A.3.1 Relation with front propagation approach

The special case of a monotonically advancing front, i.e. moving with speed F always positive (which is the hypothesis $\tilde{P} > 0$) leads from equation (15) to a particular stationary level set equation for the crossing time T:

$$\|\nabla T\| \, F = 1 \tag{16}$$

The gradient of arrival time $\|\nabla T\|$ is inversely proportional to the interface speed.

We study the particular case where $F = F(x, y, z)$ is a function of the pixels position only. Then equation (16) is the well-known *Eikonal equation*. The speed does not depend on the position of the interface.

Assuming that the isolines $T = U$ and $F = 1/\tilde{P}$ (the speed being high in low cost areas and vice-versa), we can relate equation (3) and equation (16):

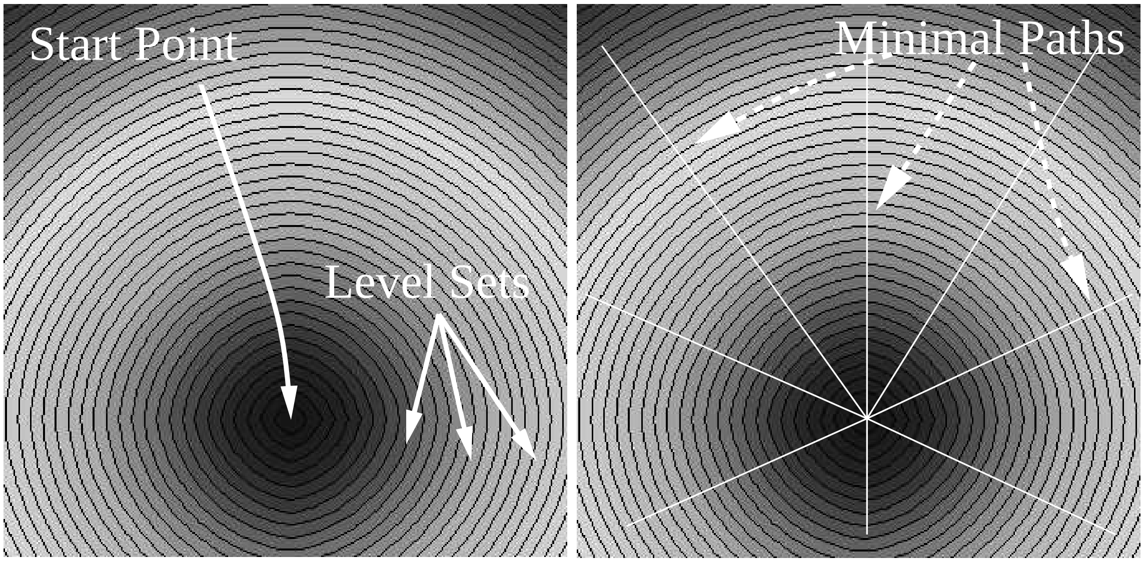
$$\|\nabla U\| = \tilde{P} \tag{17}$$

A rigorous proof is cited in (Cohen and Kimmel, 1997). Thus, we now have related the minimal cost map and the *Eikonal equation*. This equation will propagate a front and compute the values of U on the image domain.

A.3.2 Understand the Role of the Potential Map

The aim of the potential used in equation (17) is to propagate the front in the desired regions, in order to extract a minimal path corresponding to the wanted features.

In figure 27-(a) one can see the iso-action lines of the surface of minimal action provided by a front propagation on a synthetic 2D example: a univalued potential ($\forall(i,j)\ P_{i,j} = 1$). Visualization is focused on the lines of iso-action. Without any obstacle, the front is propagating in every direction at the same speed. The corresponding iso-action lines are circles with the start point as center, and their radius being the Euclidean distance to this start point.



(a) Cost map

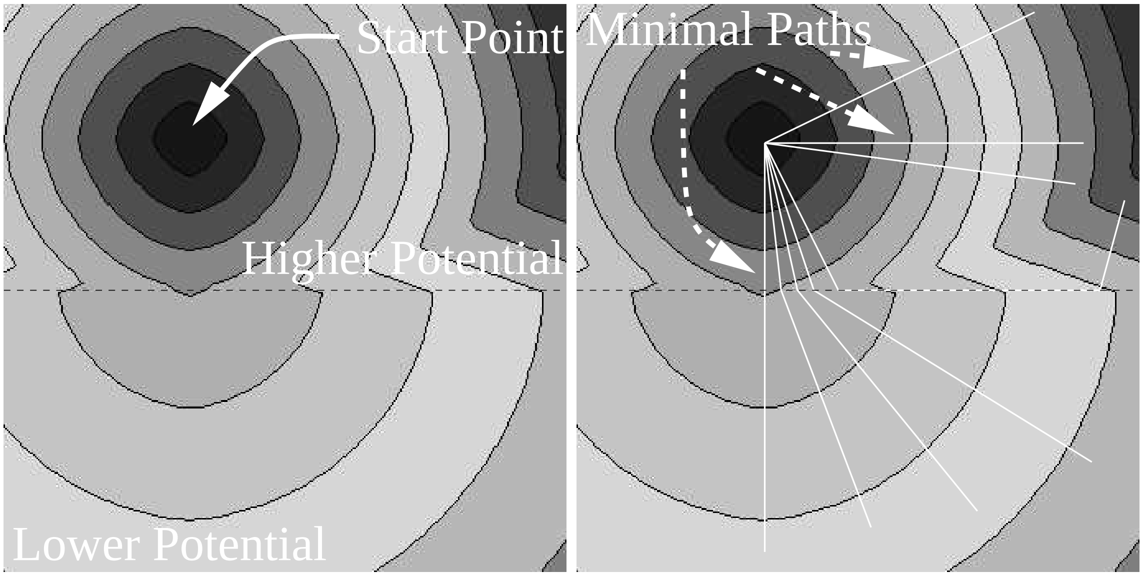
(b) Random minimal paths

Figure 27: Propagation and minimal paths for a univalued potential

In order to understand what occurs if the potential is multi-valued, one can observe figures 28-(a) and 28-(b). Figure 28-(a) represents the iso-action lines of the surface of minimal action provided by a front propagation on a bivalued synthetic potential - the higher value being the upper-half part of the image. One can easily see that the front propagation speed is quicker in the lower half part, because the space between the iso-action lines (level sets of the surface) is bigger. This is similar to *Fermat*'s principle on the minimality of the light path: we can observe on figure 27-(b) that paths are straight lines in homogeneous media, and that paths are deviated at the junction between two different homogeneous media on figure 28-(b).

We can also observe that equation (17) enables to compute distance maps. In figure 27, the propagation over the entire image has given the Euclidean distance to the start point. This technique can be used to compute the distance from an arbitrary set of points in 2D and 3D.

In particular, the initial condition of the higher-dimensional front ϕ is given by the signed distance computed from the original surface $\mathcal{L}(.,0)$. In the case of the front propagation from a starting point, we consider that the original front



(a) Cost map (b) Random minimal paths

Figure 28: Front propagation and minimal paths for a bivalued potential

is an infinitesimal sphere around this start point. In a general case, initialization ($\phi(\mathbf{x}, t = 0)$ in equation (15)) may not be computed from just a single point, but from a whole set of points (disconnected or not). Then, this initialization will be computed using the *Eikonal* equation (17). And the propagation is done "forward" or "outward" from the initial shape $\mathcal{L}$, depending on the sign of the normal $\overrightarrow{n}$.

A.3.3 The Regularity of the Path

In (Cohen and Kimmel, 1997), it is proven that weight w in equation 2 can influence curvature and be used as a smoothing term. An upper bound for the curvature magnitude $|\kappa|$ along the minimal path is found, $\mathcal{I}$ being the image domain:

$$|\kappa| \leq \frac{\sup_{\mathcal{I}} \|\nabla P\|}{w} \tag{18}$$

Length constraint The exact minimal path is obtained with a gradient descent. But care must be paid on the choice of the gradient step to avoid oscillations.

If the weight w is set to a small value ϵ the extracted path length is not limited at all, nor the curvature magnitude in equation (18). Therefore in zones where the action map is flattened, the slope being as small as ϵ, the path can have a spaggethi-like trajectory. The minimal path being obtained by steepest gradient descent, directions are evaluated by interpolation based on nearest neighbors on the Cartesian grid. If the discrete gradient step $\Delta\mathbf{x}$ is too large, the approximation of this trajectory will produce oscillations between relative positions. Those oscillations can lead to a huge number of path points larger than forecasted allocations.

In figure 29, we have displayed a test where the steepest gradient fails (with a number of path points limited). The test is performed on a region of the data

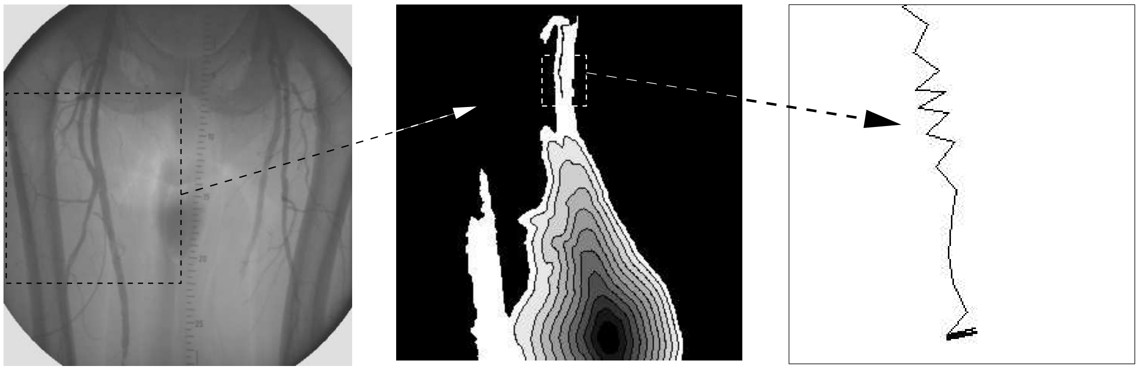


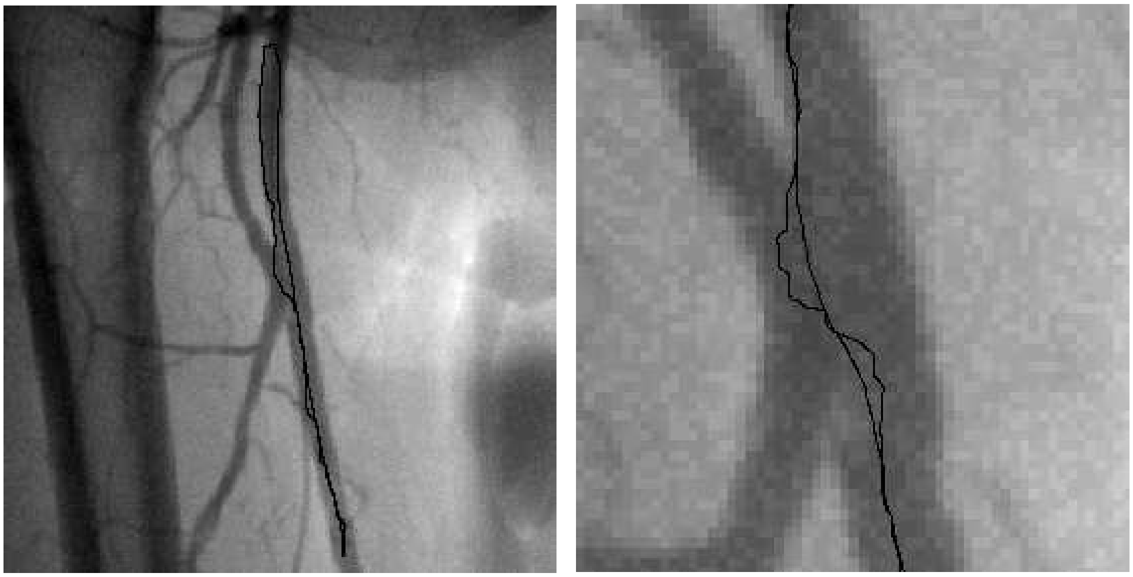
Figure 29: Fail of the steepest gradient descent on a bolus chase reconstruction data

shown in figure 29-left. The cost map for path tracking a vessel is displayed in figure 29-middle. For this problem the upper bound curvature has been estimated to be $\sup_{\mathcal{I}} \|\nabla P\| = 130.$, with $\Delta\mathbf{x} = .5$ and $w = .1$. Those values lead to a curvature radius $R_\kappa = 7.10^{-4}$. The steepest gradient scheme oscillates and stops as shown in figure 29-right.

A solution lies in increasing the cost of the path length. It increases the slope on the action map, leading to the start point faster. It maintains a lower upper-bound on the curvature magnitude and makes the steepest gradient descent scheme robust. Alternatives are to decrease the space step, or take the steepest value neighbor descent, but it might not correspond to the exact minimal path.

Curvature constraint This paragraph illustrates the influence of the weight w of equation (2) on the extracted path regularity. In figures 30 is shown the tracking of a vessel in a X-Ray image of the femoral vessels, using different weights $w_1 = 1$ and $w_2 = 20$. The smoothing done by increasing the weight can be observed in a zoom on the paths shown in figure 30-right. We can also observe the influence of increasing the weight in figure 31 where each path is displayed superimposed on its respective action map. For a small weight $w_1 = 1$, the path is unstable, as shown in figure 31-left. For a weight $w_2 = 20$, leading to the inequality $|\kappa_2| \leq 0.75$, the path is smooth. Differences remain also in the sets of points visited during propagations: it is smaller than with weight $w_2 = 20$. It means that propagation is quicker for small weights. It propagates in every directions for a higher weight (see figure 31)-right), because the tune of w smoothes the image, as it reduces the upper-bound on curvature magnitude in equation (18).

Conclusion on the use of w The estimation of the impact of the upper-bound is not straightforward: in the test shown in figure 31, the value given by the case where $w_2 = 20$ leads to a curvature magnitude $|\kappa| \leq 0.75$. The corresponding lower bound curvature radius is about $R_\kappa = 1.33$, which is not a relevant constraint. Thus, the constraint acts mostly on the path length, and not on curvature. We can observe that the constraint given in equation 18 is based on the value of the potential on all the image domain, while the relevant *a posteriori* curvature constraint is restricted



Femoral vessels + Paths A zoom on the paths

Figure 30: Smoothing the minimal path with the weight w: the paths

to the path domain.

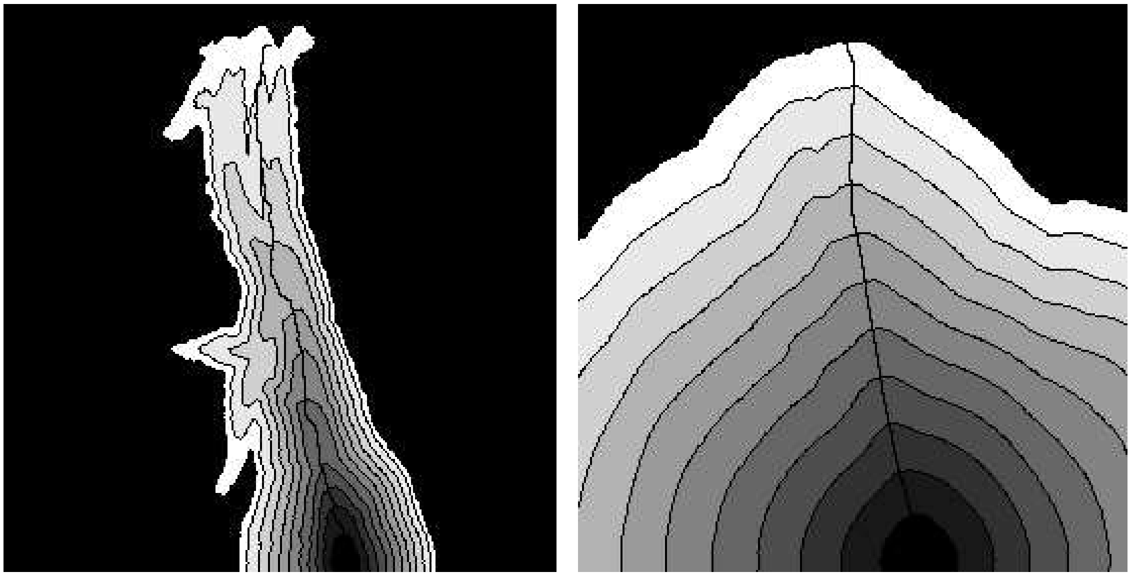
An important issue is to have a more *qualitatively* relevant upper boundary, to be used as a feedback correction. We can estimate *a posteriori* the real constraint on the set of *Alive* points. Having $\{Alive\ points\} \subset \mathcal{I}$ leads to $\sup_{Alive} \|\nabla P\| \leq \sup_{\mathcal{I}} \|\nabla P\|$. Therefore, the zone of the potential that does not affect the minimal path computation nor affect the magnitude curvature, leading to a more precise, but *a posteriori* upper-bound. Even more, we can consider only the set of pixels which the minimal path goes through to build this *a posteriori* upper-bound.

A test was performed on the problem shown in figure 31, and in the case $w_1 = 1$, the *a posteriori* evaluation on the path points of the upper-bound of equation (18) has reduced it from 13.2 to 8.2. This estimation can be used as a feedback correction but a remaining problem relies in the fact that an *a posteriori* value for the weight will lead to different results, and as a consequence to a different set of estimation for $\sup \|\nabla P\|$. In conclusion, the setting of the weight w is not straightforward, and care must be taken with its parameterization.

B The Min-Heap Data Structure

The execution time of Sethian's algorithm for minimal paths strongly depends upon the underlying data structure used to store results of intermediate computations, since most of the algorithm's execution time is spent in sorting the data structure. An efficient scheme is discussed in (Adalsteinsson *et al.*, 1996). Execution time of the algorithm depends upon the time spent for

- locating and removing the grid point in the *narrow band* with the minimum action value;



$w_1 = 1$, and $|\kappa_1| \leq 15.4$ $w_2 = 20$, and $|\kappa_2| \leq 0.75$

Figure 31: Smoothing the minimal path with the weight w: the action maps

- inserting a new grid point in the *narrow band*, and for updating the action value of a *narrow band* point.

Then efficiency is determined by how quickly the data structure can execute the operations of extracting the minimum and decreasing the action.

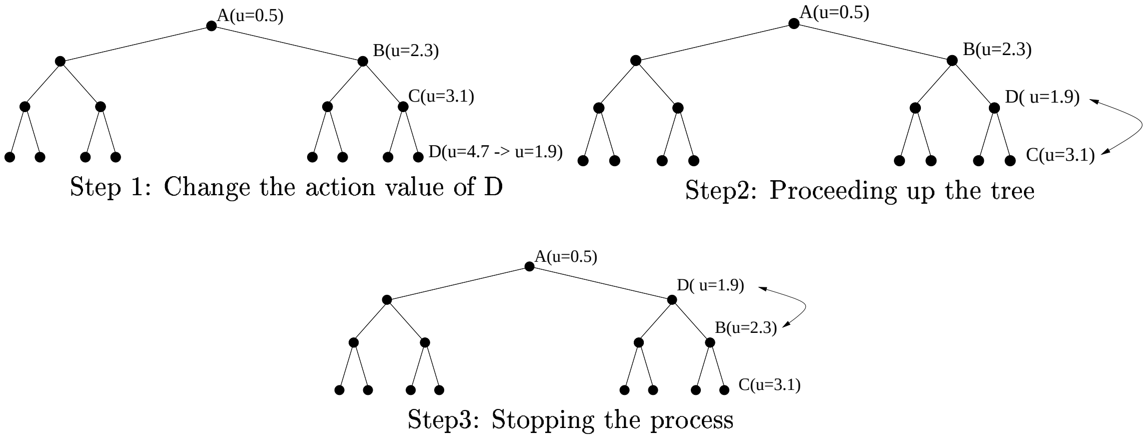
To perform these operations in minimum time, the *narrow band* points are stored in a min-heap data structure. It is defined as a binary tree in which the action of each node is smaller than the action of its children. This partial ordering, which does not include any ordering between two children of the same element, allows all operations to be in logarithmic complexity, provided this ordering is maintained at all times. It means the tree should remain balanced. See (Sedgewick, 1998) for details.

In particular, the tree element with smallest action is always located at the root, thus leading to a fast execution of the *Fast Marching* algorithm.

In practice, a heap can be represented as an array, leading to a more efficient implementation than with a tree structure: the nodes are stored sequentially in an array, assuming that a node at location k has its children at locations $2k$ and $2k+1$. Then the values of U are stored together with the indices which give their location on the grid structure (back pointers).

B.0.4 Introduction of a node and re-estimation

When a grid point labeled *Far* is met, adding it to the heap works by increasing the heap size by one, putting it at the last location and moving it upward the tree to its correct location. For grid points labeled *Trial*, according to the up-wind scheme, the value of U at every node when recomputed under the influence of its 4 neighbors in 2D is lower than or equal to its former value. Then the operation of decreasing the action is involved when inserting or reexamining a grid point.



Step 1: Change the action value of D

Step2: Proceeding up the tree

Step3: Stopping the process

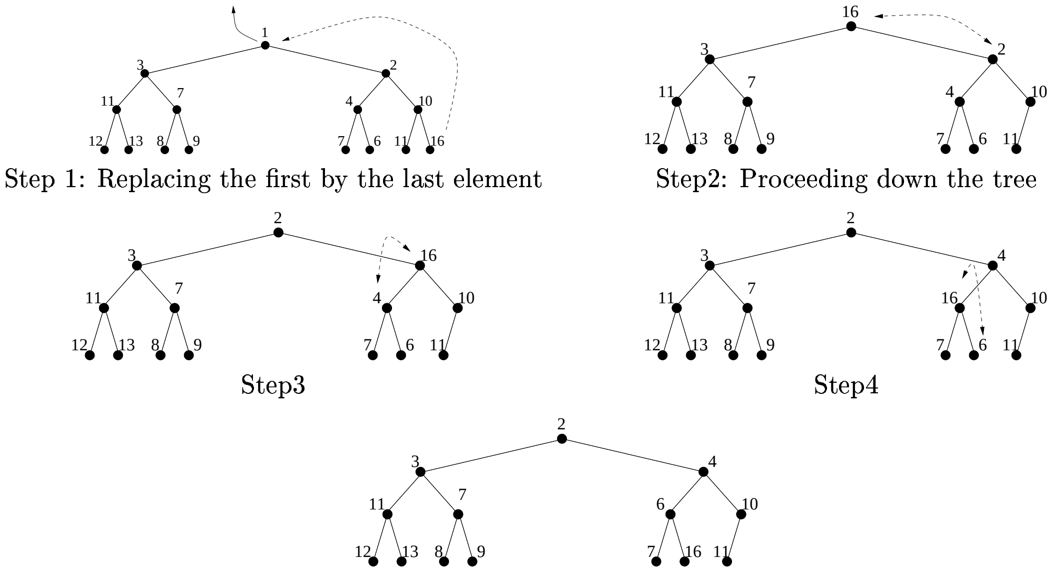
Figure 32: Restoring the heap property when decreasing a node's action

As an example, Figure 32 shows a typical heap structure and an operation of restoring the heap property after the element D gets updated from 4.7 to 1.9. Since the complexity of the operation of changing the value of one element of the heap is bounded by a worst-case bottom-to-top proceeding of the tree in $O(\log_2 N)$, the total work is about $O(N \log_2 N)$ for the *Fast Marching* on a N points grid.

B.0.5 Extract the minimum

The marching algorithm works by first looking for the smallest element in the *narrow band*, and removing it. To ensure that the heap property is not violated, it involves to put the last element of the heap at the first position, then moving down the heap, exchanging the node with its smallest children until we reach a node with both children bigger or until we reach the bottom.

As an example, Figure 33 shows an operation of restoring the heap property after deletion of the minimum action node. With the operation of changing the value of one element of the heap, the total work is bounded here by a worst-case root-to-bottom proceeding of the tree in $O(\log_2 N)$. The total work is about $O(N \log_2 N)$ for the *Fast Marching* on a N points grid, since all the grid points are visited.



Step 1: Replacing the first by the last element

Step2: Proceeding down the tree

Step3

Step4

Step5: The tree is re-balanced

Figure 33: Restoring the heap property when deleting the root

References

Adalsteinsson, D., Kimmel, R., and Sethian, J.A. (1996). Fast marching methods for computing solutions to static hamilton-jacobi equations. Technical report, Center for Pure and Applied Mathematics, Univ. of California, Berkeley.

Blum, H. (1967). A transformation for extracting new descriptors of shape. In Whaten-Dunn, Weiant (ed.), *Symposium Models for Speech and Visual Form*, Cambridge, MA. MIT Press.

Caselles, V., Kimmel, R., and Sapiro, G. (1995). Geodesic active contours. In *ICCV'95*, pp. 694–699, Cambridge, USA.

Cohen, L.D. (1988-1995). Variational methods for image processing (in french). Mémoire d'Habilitation à diriger des recherches. Presented together with 10 main publications (in english).

Cohen, L.D. and Kimmel, R. (1997). Global minimum for active contour models: A minimal path approach. *International Journal of Computer Vision*, 24(1), 57–78.

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

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

Malladi, R. and Sethian, J.A. (1998). A real-time algorithm for medical shape recovery. In *Proceedings of International Conference on Computer Vision*, pp. 304–310.

McInerney, T. and Terzopoulos, D. (1996). Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2).

Sedgewick, R. (1998). *Algorithms in C - Parts 1-4*. Addison-Wesley, Reading, Massachusetts, USA, 3 edition.

Sethian, J.A. (1996a). A fast marching level set method for monotonically advancing fronts. In *Proceedings of the Natural Academy of Sciences*, Vol. 93, pp. 1591–1595, Univ. of California, Berkeley. Center For Pure and Applied Mathematics.

Sethian, J.A. (1996b). *Level set methods:Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences.* Cambridge University Press, University of California, Berkeley.

Yeorong, G., Stelts, D.R., Jie, W., and Vining, D.J. (1993). Computing the centerline of a colon: a robust and efficient method based on 3d skeletons. In *Proceedings of IEEE Nuclear Science Symposium Conference Record.*