

Fast Surface and Tree Structure Extraction of Vascular Objects in 3D Medical Images

Thomas Deschamps and Laurent D. Cohen

Abstract. We present a new fast approach for surface segmentation of thin structures, like vessels and vascular trees, based on *Fast-Marching* and Level Set methods. *Fast-Marching* allows segmentation of tubular structures inflating a “long balloon” from a user-given single point. However, when the tubular shape is rather long, the front propagation may blow up through the boundary of the desired shape close to the starting point. Our contribution is focused on a way to go on front propagation only on the actually moving front and freezing other points. We demonstrate the ability to build a fast and accurate segmentation for those tubular and tree structures. We also develop a useful stopping criterion for the causal front propagation. We illustrate our algorithms by applications to the segmentation of vessels in 3D medical images. The minimal paths obtained from *Fast-Marching* together with the segmentation allow automatic virtual endoscopy through the tree structure.

§1. Introduction

We are interested in this paper in segmentation of tubular surfaces from 3D images, motivated by medical applications related to vessels and the vascular tree. These surfaces can therefore have several branches. Much work has been done on surface segmentation since the introduction of deformable models (see references in [1]). The recent trend of deformable surfaces makes use of Level Set methods (for example [2]). A major drawback of level set methods is their huge computation time, even when using a narrow band. *Fast-Marching*, introduced in [3], allows fast surface segmentation when the evolution is always outwards like a balloon [4,5]. Using the *Fast-Marching* algorithm as a region-growing method [6] to propagate a wave front inside a colon CT scanner, we can extract the surface of the colon, starting from an initial seed point. We have developed an algorithm

that can be the basis of this kind of tubular shape extraction object: a technique to evolve a front inside an object of interest and compute at the same time the geodesic distance to the starting point inside the object. This distance can be used to stop the front propagation inside the desired object. The result of this technique is shown in Figure 1. The top image is the 3D CT scanner, and the bottom images show some steps of the front propagation in the 3D dataset.

However, classical segmentation problems do not provide an excellent contrast like the air-filled colon on a CT scanner, and the propagation cannot stick to the object walls for long and thin objects. We show in this article how the *Fast-Marching* surface segmentation, which is not tuned for this kind of thin and long objects, can be specifically optimized for this target. Another by-product of the method is to extract minimal paths in the image. We extend this technique to extract a trajectory in each branch of the vascular objects, and show applications to virtual endoscopy.

§2. Minimal Paths, Fast-Marching and Surface Segmentation

2.1 Minimal paths extraction

We present in this section the basic ideas of the method introduced by Cohen and Kimmel (see [7,9] for details) to find the global minimum of the active contour energy using minimal paths. The energy to minimize is similar to classical deformable models (see [8]) where it combines smoothing terms and image features attraction term (Potential P):

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

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 . The problem is transformed in a way to find the global minimum, avoiding getting stuck at local minima. We now have an expression in which the internal forces are included in the external potential. The regularization is now achieved by the constant $w > 0$. 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. We can define the surface of minimal action T as the minimal energy integrated along a path between a starting point \mathbf{p}_0 and any point \mathbf{p} :

$$T(\mathbf{p}) = \inf_{\mathcal{A}_{\mathbf{p}_0, \mathbf{p}}} E(C) = \inf_{\mathcal{A}_{\mathbf{p}_0, \mathbf{p}}} \left\{ \int_{\Omega} \tilde{P}(C(s)) ds \right\}, \quad (1)$$

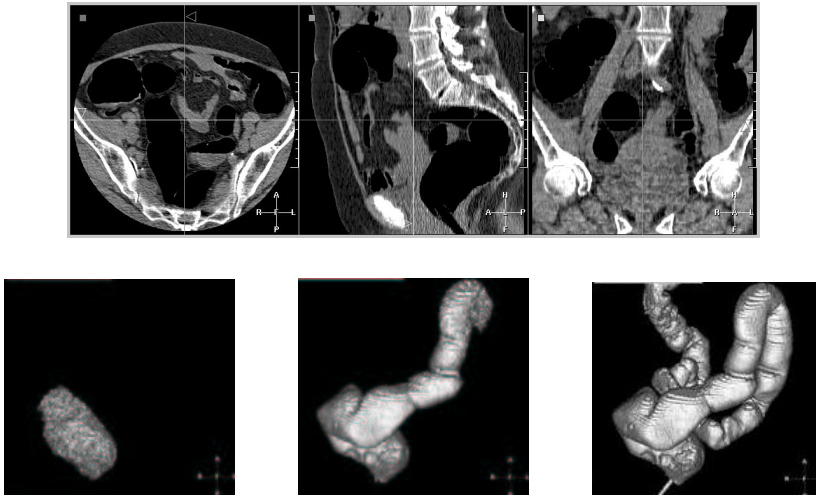


Fig. 1. Segmenting the colon volume with simple front propagation.

where $\mathcal{A}_{\mathbf{p}_0, \mathbf{p}}$ is the set of all paths between \mathbf{p}_0 and \mathbf{p} . The minimal path between \mathbf{p}_0 and any point \mathbf{p}_1 in the image can be easily deduced from this action map by back-propagation (see [7]).

In order to compute this map T , a front-propagation equation related to equation (1) is solved : $\frac{\partial C}{\partial t} = \frac{1}{P} \vec{n}$. It evolves a front starting from an infinitesimal circle shape around \mathbf{p}_0 until each point inside the image domain is assigned a value for T . The value of $T(\mathbf{p})$ is the time t at which the front passes over the point \mathbf{p} . Then it represents the minimal path energy to reach the start point from any point in the image. The *Fast-Marching* technique, introduced by Sethian (see [3]), was used by Cohen and Kimmel [7] noticing that the map T satisfies the Eikonal equation $\|\nabla T\| = \tilde{P}$. Classic finite difference schemes for this equation tend to overshoot and are unstable. Following Sethian [3] we use the up-wind scheme, where at each voxel (i, j, k) , the unknown u satisfies:

$$\begin{aligned} & (\max\{u - T_{i-1,j,k}, u - T_{i+1,j,k}, 0\})^2 + \\ & (\max\{u - T_{i,j-1,k}, u - T_{i,j+1,k}, 0\})^2 + \\ & (\max\{u - T_{i,j,k-1}, u - T_{i,j,k+1}, 0\})^2 = \tilde{P}_{i,j,k}^2, \end{aligned} \quad (3)$$

giving the correct viscosity-solution u for $T_{i,j,k}$. The improvement made by *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. The algorithm is detailed in Table 1.

Considering a 3D surface Γ moving under speed F in its normal direction, in the *Level-Set* formulation, it is embedded as the zero level set of a function ϕ defined in the 3D image space, leading to evolution

$$\phi_t + F|\nabla\phi| = 0.$$

Definitions:

Alive set: grid points at which the values of \mathcal{T} have been reached and will not be changed;

Trial set: next grid points (6-connectivity neighbors) to be examined. An estimate T of \mathcal{T} has been computed using discretized Eikonal Equation from *Alive* points only;

Far set: all other grid points, there is not yet an estimate for T .

Initialization:

Alive set: start point \mathbf{p}_0 , $T(\mathbf{p}_0) = \mathcal{T}(\mathbf{p}_0) = 0$;

Trial set: six neighbors \mathbf{p} of \mathbf{p}_0 with initial value $T(\mathbf{p}) = \tilde{P}(\mathbf{p})$ ($\mathcal{T}(\mathbf{p}) = \infty$);

Far set: all other grid points, with $\mathcal{T} = T = \infty$.

Loop:

1. Let \mathbf{p}_{min} be the *Trial* point with smallest T ;
 2. Move it from the *Trial* to the *Alive* set;
 3. For each neighbor \mathbf{p} of \mathbf{p}_{min} :
 - a. If \mathbf{p} is *Far*, add it to the *Trial* set;
 - b. If \mathbf{p} is *Trial*, update $T_{\mathbf{p}}$.
-

Tab. 1. *Fast-Marching* algorithm.

In the case of Γ moving with a speed $F > 0$, it leads to the same Eikonal equation that determines the evolution of the surface or arrival time $T(\mathbf{p})$ (see [3] for details). This Eikonal equation has been used for surface extraction in [8]. Discretized with an up-wind scheme, it is then solved using *Fast-Marching* (Table 1), given an initial starting point \mathbf{p}_0 . In practice the front is propagated until a fixed time is reached. Figure 1 shows iterations of this front propagation in a 3D image with potential P defined in order to segment the colon. Evolution is stopped when a given geodesic length has been traveled by the front [6].

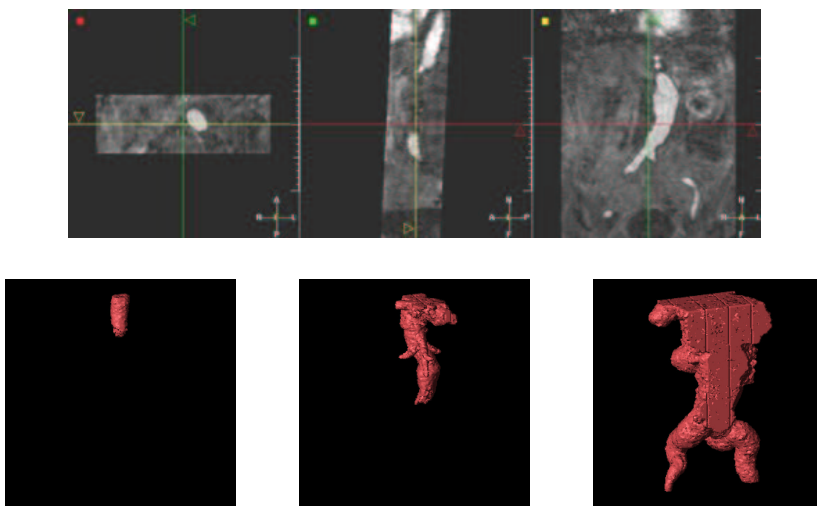


Fig. 2. 3D contrast enhanced MR image of the aorta and front propagation.

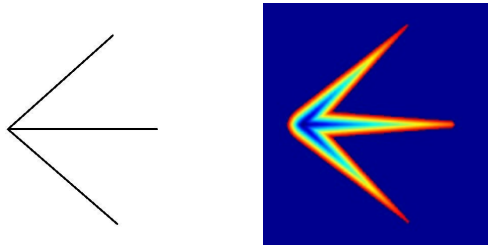


Fig. 3. Synthetic test problem.

§3. Propagation Freezing for Thin Structures

Freezing a voxel during front propagation is to consider that it has reached the boundary of the structure. When the front propagates in a thin 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 2. If we apply *Fast-Marching* in the dataset shown in Figure 2-top, with a potential based on the gray level with contrast enhancement, the corresponding wave propagation looks like Figure 2-bottom. The front floods outside the object and does not give a good segmentation.

For these reasons, it is of no use to make some voxels participate in the computation of the arrival time in the *Eikonal equation*. We thus set their speed to zero, which we call *Freezing*. The first step is to design the appropriate criterion for selecting voxels of the front which need *Freezing*.

A synthetic example of a tree structure is shown in Figure 3. In this case, setting an initial seed point at the left corner point, we would like to extract in a very fast process the multiple branches of the structures. Figure 3-right shows the result of the classical front propagation technique with *Fast-Marching*. The action map \mathcal{T} displayed indicates clearly that the domain visited is a whole “blob-like” structure, where the underlying tubular shape is somehow lost. It emphasizes the drawback of the method in this case, without a clear constraint on the domain of points visited.

3.1 Using Weighted Distance for Freezing

The geodesic weighted distance inside the object between a point and the starting point can be computed in the *Fast-Marching* process without much extra cost as shown for a different application in [6]. This is the Euclidean length of the minimal path (according to P , see [7]) that joins the points. It seems “natural” to use this distance $\mathcal{D}(\mathbf{v})$ between a voxel

\mathbf{v} and the starting point, or relatively to the most far propagating part of the front, since this notion is completely embedded in the topology of the object we are trying to extract: the section of a tube-shaped object must be small with respect to its length. We must discriminate between the points of the front that are near the initializing seed point and other parts of the front that are already far. This will prevent flooding in non-desired areas of the data.

We can fix several criteria for the *Freezing* based on the distance. The geodesic distance to the starting point is a measure which contains information about the geometry of the surface extracted, and in particular its length. Knowing the current maximum geodesic path length d_{max} in the front propagation process we can decide that a voxel \mathbf{v} of the propagating front (i.e. *Trial*) should be removed from the front (i.e. set as *Alive*):

- i) if $\mathcal{D}(\mathbf{v}) < d_{max}/\alpha$, with $\alpha \geq 1$ user-defined; or
- ii) if $\mathcal{D}(\mathbf{v}) < \max(d_{max} - \tilde{d}, 0)$, with $\tilde{d} > 0$ chosen.

A 2D example on the synthetic test is shown in Figure 4. The domain visited by our algorithm is slightly smaller than without freezing (Figure 3-right) and this domain shortens with the distance criterion, when we compare left and middle images in Figure 4 which are action maps with distance criterion of respectively 100 and 50. Figure 4-right is a zoom on the freezing map which clearly demonstrates that the *Freezing* principle discriminates the points located far from the propagating fronts (frozen parts are represented in white).

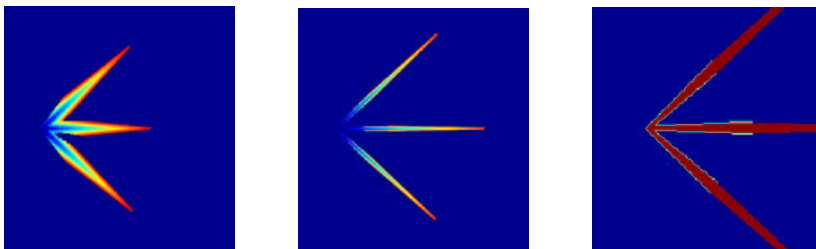


Fig. 4. Distance criterion for *Freezing*.

3.2 Algorithm for Freezing

At each time step we insert our visited points both in the classical action related heap, and in another min-heap data-structure where the ordering key is the distance to the seed point, which means that the element at the top of the heap will still be the point that is the closest *Trial* point to the starting point. At each iteration, we are able to remove all the points whose keys are lower than this criterion, starting from the minimum element in the binary heap.

Initialization:

setting $\mathcal{T}(\mathbf{p}_0) = \mathcal{D}(\mathbf{p}_0) = 0$ and storing the seed point \mathbf{p}_0 in both min-heap structures $\mathcal{H}_{\mathcal{T}}$ and $\mathcal{H}_{\mathcal{D}}$;

$d_{max} = 0$, \tilde{d} and d_{stop} are user defined.

Loop: at each iteration

Let \mathbf{p}_{min} be the *Trial* point with the smallest action \mathcal{T} ;

Fast-Marching algorithm of Table 1, updating min-heaps $\mathcal{H}_{\mathcal{T}}$, $\mathcal{H}_{\mathcal{D}}$ with the new action values for \mathcal{T} , \mathcal{D} computed;

take $d_{max} = \max(d_{max}, \mathcal{D}(\mathbf{p}_{min}))$;

consider \mathbf{q}_{min} , the root of $\mathcal{H}_{\mathcal{D}}$. While $\mathcal{D}(\mathbf{q}_{min}) < \max(d_{max} - \tilde{d}, 0)$ do

1. set $\mathcal{D}(\mathbf{q}_{min}) = \mathcal{T}(\mathbf{q}_{min}) = \infty$;

2. set \mathbf{q}_{min} in *Alive* set and delete it in both $\mathcal{H}_{\mathcal{D}}$ and $\mathcal{H}_{\mathcal{T}}$;

if $d_{max} > d_{stop}$, exit the loop.

Tab. 2. *Freezing* algorithm.

In Table 2 is detailed an algorithmic implementation of the *Freezing* with : Starting point \mathbf{p}_0 located at the root of the tree structure; action map \mathcal{T} , one min-heap structure $\mathcal{H}_{\mathcal{T}}$ and a penalty image P which will drive the front propagation; distance map \mathcal{D} to compute the minimal path Euclidean length [6]; min-heap data structure $\mathcal{H}_{\mathcal{D}}$, where the ordering key for any point \mathbf{p} is the value of $\mathcal{D}(\mathbf{p})$; a counter d_{max} , distances \tilde{d} , d_{stop} .

This heuristic is to discriminate the parts of the front that are propagating slowly, by recording the maximum distance which has been traveled, and comparing it to the distance which has been traveled by these parts. If the ratio between those two distances is superior to a given threshold, we "freeze" those parts by setting their speed artificially to zero. It enables to stay inside the object when it is long and thin like a tubular structure, as shown in Figure 4.

3.3 Illustration of Vascular Tree Segmentation

The method explained previously is very useful when used for vascular segmentation. Segmentation is therefore performed in a very fast manner by just setting a seed point at the top of the tree hierarchy.

In Figure 5 from left-to-right, we show iterations of the segmentation process on the former examples of Figure 2; the frozen voxels are in white and propagating front is darker. Figure 6 displays results of this method on three different objects.

3.4 Stopping criterion

The *Freezing* process provides a criterion which is independent of the number of different branches to recover. If we plot the maximum distance d_{max}

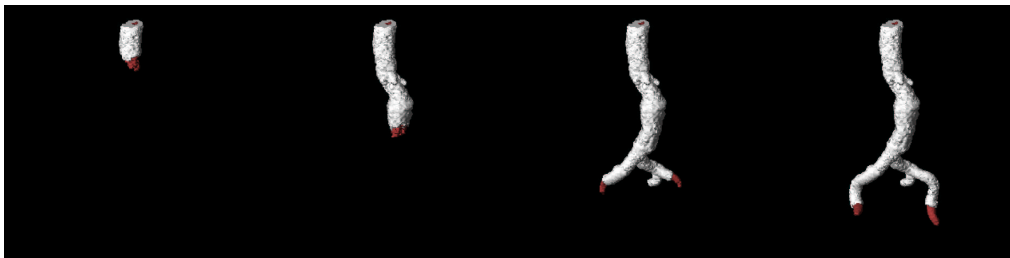


Fig. 5. Using Distance for *Freezing* in the Aorta.

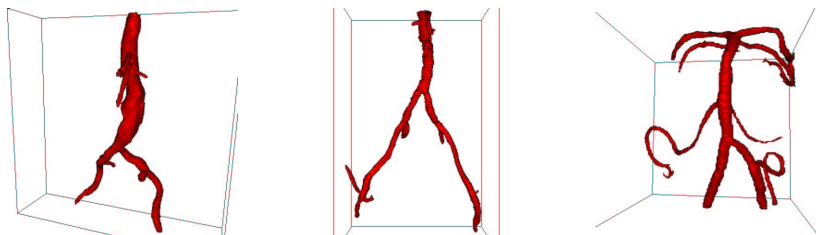


Fig. 6. Segmentation of several tubular objects with the *Freezing* algorithm.

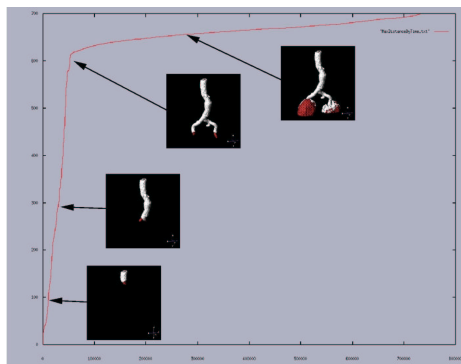


Fig. 7. Using Distance for Stopping propagation in the Aorta.

obtained in the *Freezing* Algorithm as a function of iterations while propagating the front, we observe the profile shown in Figure 7. We clearly see that this distance increases linearly until a big decrease of the slope appears. It is important to notice that this shock indicates when the front flows out of the object at “heads” of the front. We decide to stop front propagation at this particular time. During the first part of the plot, the function is quasi-linear. The slope is directly related to the section area of the tubular object. By definition of *Fast-Marching*, the number of iterations is equal to the number of voxels that are alive, and is close to the volume of the region inside the front. This means that passing through a certain length in the aorta implies visiting a number of voxels proportional to the length. This is the case in general for tubular shapes.

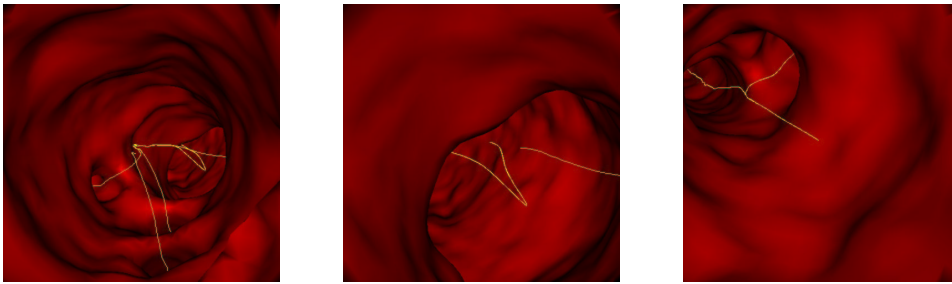


Fig. 8. Endoscopic views along one trajectory showing all minimal paths.

Let us assume that the global section of our aorta is constant in our dataset. This is approximately true in large parts, but no longer holds in the very thin parts of the vessels and arteries. But we can assume that the front propagates at the same speed inside the object. Therefore, the number of voxels visited is proportional to the section area. Then the slope collapse can be easily detected using a simple threshold on the slope, depending on the object we want to extract. Recording the first iteration where the slope decreases gives us the maximum distance where we must stop propagation.

§4. Multiple Path Extraction for the Vascular Tree and Virtual Endoscopy

Once the tubular object has been segmented, we proceed by a few steps in order to get virtual endoscopic views along the different branches of the vascular tree.

- 1) We refine the segmentation by means of a Level-Set deformable surface [2]. Since our freezing method gives a very good estimate of the surface, this step is quite fast.
- 2) The extremities of all branches in the vascular tree are automatically extracted finding the bifurcations with morphological techniques.
- 3) The whole set of trajectories inside the object is extracted using back-propagation from all extremities.
- 4) Rendering views along the different paths in the direction of the curve (see Figure 8). Virtual endoscopy is obtained through the tree structure and the user may fly through the image in the way he decides to go at each bifurcation.

§5. Conclusion

Concerning tree tracking application, the main improvements brought by this method are to accelerate the computations by visiting a very small number of voxels during propagation, and to segment thin tubular structures. Another by-product of our technique is the stopping criterion based

on the maximum distance travelled by the front. The extraction of the multiple trajectories, together with the segmentation step, in a single process, enhances visualization and quantification of pathologies. Those trajectories are then the input to an endoscopic tool, as shown in Figure 8.

Acknowledgments. We would like to thank in particular Sherif Makram-Ebeid and all the Medical Imaging Systems Group of Jean Pergrale, at Philips Research France. T. Deschamps is now with LBL in Berkeley.

References

1. T. McInerney and D. Terzopoulos, Deformable Models in Medical Image Analysis: A Survey, *Medical Image Analysis* **2**(1) (1996).
2. V. Caselles, R. Kimmel, G. Sapiro and C. Sbert, Three dimensional object modeling via minimal surfaces, *Proceedings of the Third European Conference on Computer Vision*, (1996), 97–106.
3. J.A. Sethian, *Level Set methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*, Cambridge University Press, 2nd edition, 1999.
4. Laurent D. Cohen, On active contour models and balloons, *Computer Vision, Graphics, and Image Processing* **53**(2) (1991), 211–218.
5. R. Malladi and J.A. Sethian, A real-time algorithm for medical shape recovery, *Proceedings of the IEEE International Conference on Computer Vision* (1998), 303–310.
6. T. Deschamps and Laurent D. Cohen, Fast extraction of minimal paths in 3D images and applications to virtual endoscopy, *Medical Image Analysis* **5**(4) (2001), 281–299.
7. Laurent D. Cohen and R. Kimmel, Global Minimum for Active Contour Models: A Minimal Path approach, *International Journal of Computer Vision* **24**(1) (1997), 57–78.
8. M. Kass, A. Witkin and D. Terzopoulos, Snakes: Active contour models, *International Journal of Computer Vision* **1**(4) (1988), 321–331.
9. Laurent D. Cohen. Avoiding local minima for deformable curves in image analysis. In *Curves and Surfaces with Applications in CAGD*. A. Le Méhauté, C. Rabut, and L. L. Schumaker (eds.), 1997, 77–84.
10. E.W. Dijkstra, A note on two problems in connection with graphs, *Numerische Mathematic* **1** (1959), 269–271.

Thomas Deschamps and Laurent D. Cohen
CEREMADE, UMR CNRS 7534, Université Paris Dauphine
75775 Paris cedex 16, France
TDeschamps@lbl.gov and Cohen@ceremade.dauphine.fr
<http://www.ceremade.dauphine.fr/~cohen>