Fast Extraction of Tubular and Tree 3D Surfaces with front propagation methods

Thomas Deschamps[1,2] Laurent D. Cohen[1]

[1]CEREMADE, University Paris-9 Dauphine, 75775 Paris, FRANCE
[2]Philips Research France; now with Lawrence Berkeley Laboratory, Berkeley, USA.
E-mail: tdeschamps@lbl.gov,cohen@ceremade.dauphine.fr

Abstract

We present a new fast approach for surface segmentation of thin structures, like vessels and vascular trees, based on Fast Marching and Level Sets 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 on the segmentation of vessels in 3D medical images.

1 Introduction

We are interested in this paper in segmentation of tubular surfaces from 3D images, motivated by medical applications like vessels and 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 [10]). The recent trend of deformable surfaces makes use of Level Sets methods (for example [1]). A major drawback of level sets methods is their huge computation time, even when using a narrow band. Fast Marching, introduced in [11], allows fast surface segmentation when the evolution is always outwards like a balloon [3, 9]. Using the *Fast-Marching* algorithm to propagate a wave front inside a colon CT scanner, we can extract the surface of the colon, starting from an initial seed point, using the *Fast-Marching* as a region-growing method [5]. 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. Top image is the 3D CT scanner, and 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.

2 Fast-Marching and Surface Segmentation

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

$$\phi_t + F|\nabla\phi| = 0. \tag{1}$$

In the case of Γ moving with a speed $F > 0$, it leads to a new equation that determines the evolution of the surface or arrival time $T(\mathbf{x})$, also called action (see [11] for details):

$$|\nabla T| = \frac{1}{F} = P. \tag{2}$$

This *Eikonal* equation (2) has been used for surface extraction in [9]. 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 [5].

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

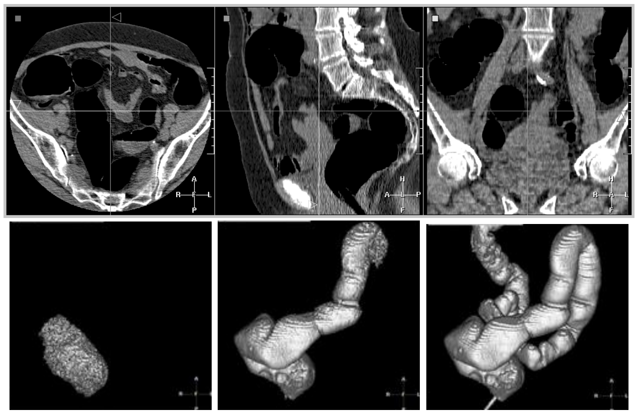

Figure 1. Segmenting the colon volume with simple front propagation

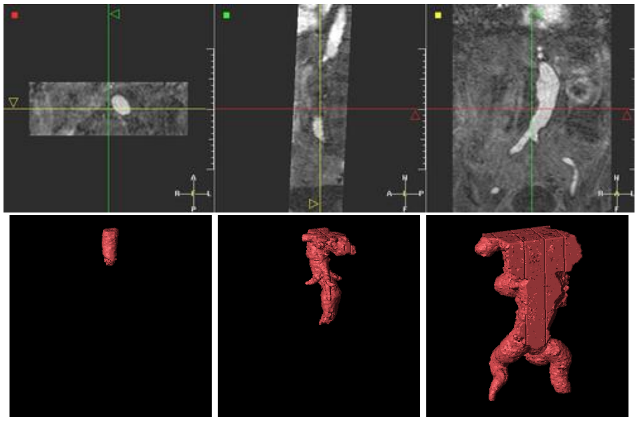

Figure 2. 3D contrast enhanced MR image of the aorta and front propagation

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 *Eikonal equation*. We thus set their speed to zero, which we call *Freezing*. 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 the *Fast-Marching* coupled with a maximum geodesic path length stopping criterion of 300, computed according to the method described in [7]. 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.

```
Algorithm for 3D Fast Marching

• 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-connexity neighbors) to be examined. An estimate $T$ of $\mathcal{T}$ has been computed using discretized Equation (2) from Alive points only (i.e. from $\mathcal{T}$);
    – 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: reduced to the 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:
    – Let $\mathbf{p}_{min}$ be the Trial point with smallest $T$;
    – Move it from the Trial to the Alive set;
    – For each neighbor $\mathbf{p}$ of $\mathbf{p}_{min}$:
        * If $\mathbf{p}$ is Far , add it to the Trial set;
        * If $\mathbf{p}$ is Trial , update $T_{\mathbf{p}}$.
```

Table 1. Fast Marching algorithm

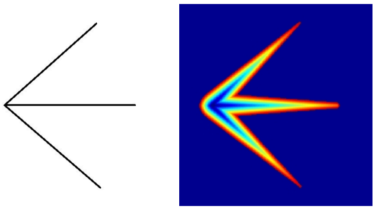

Figure 3. Synthetic test problem

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 [7, 5]. This is the Euclidean length of the minimal path (according to P, see [2]) 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 the points of the front that are near the initializing seed point while other parts of the front are already far. It will prevent from flooding in non-desired areas of the data.

We can fix several criteria for the *Freezing* based on the distance. 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* voxels) should be removed from the front (i.e. set as *Alive* voxel):

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

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. 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. The 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).

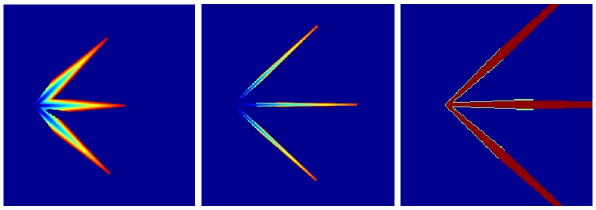

Figure 4. Distance criterion for Freezing

3.2 Algorithm for the 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.

In the following 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 [7, 5]; 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}.

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

This heuristic is to discriminate the parts of the front that are propagating slowly, by recording the maximum distance which has been traveled, and compare 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 there 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 on Vascular Tree Segmentation

The method explained previously is very useful when it is 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. Figure 5 displays results of this method on three different objects. Figure 5-left shows the new result obtained on the dataset of figure 2.

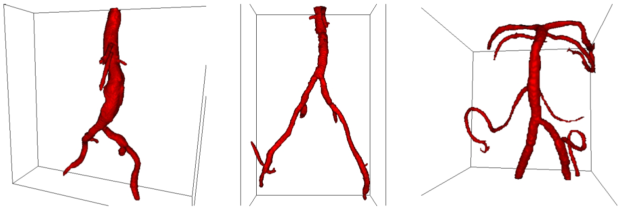

Figure 5. Segmentation of several tubular objects with the Freezing algorithm

The distance threshold is a parameter which is not very sensitive: we generally take a value related to the *a priori* dimensions of the object. This threshold must be larger than the assumed maximum section of the object. It will approximately represent the volume of points bounded by the connected envelope of the front voxels that are not frozen.

3.4 Stopping criterion

Having designed an adequate criterion for *Freezing* the unwanted parts of the front that could lead to “flooding” of the evolving wave in other parts of the image, we now explain our strategy to stop automatically the propagation.

The *Freezing* process will provide a criterion which is independent of the number of different branches to recover. If we plot the maximum distance d_{max} of section 3.2, as a function of iterations while propagating the front, we observe the following profile shown in figure 6. 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 close to the volume of the region inside the front. It means that passing through a certain length in the aorta implies to visit a number of voxels proportional to the length. This is the case in general for tubular shapes.

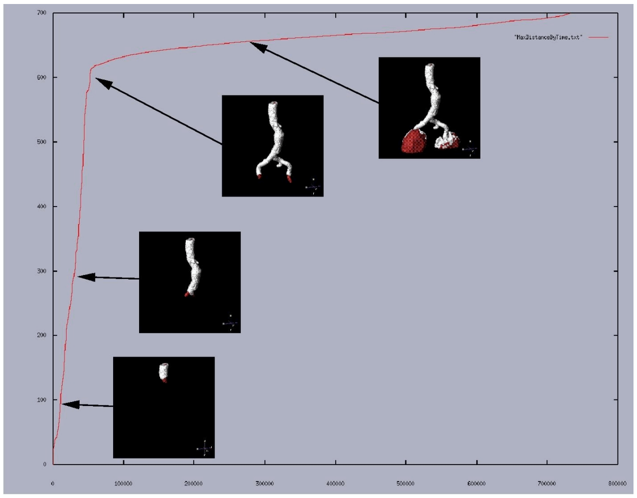

Figure 6. **Using Distance for Stopping propagation in the Aorta**

Let us assume that the global section of our aorta is constant in our dataset. This is approximately true in large parts, but becomes a wrong assumption 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. Even if there are aneurysms in the data set, and even if the mean section of the object increases with the depth, we can assume that we do not want to extract an object which is twice the maximum section. We could then derive a criterion on the maximum section of the object S_{max} which gives a related value of minimal slope. Recording the first iteration where the slope decreases, it gives us the maximum distance where we must stop propagation.

3.5 Discussion

What is the gain of our approach? Toward level-sets implementations of tubular shape extraction, as in [12, 8], solving the Eikonal equation in $o(n \log(n))$ is much faster than any time-dependent scheme, since this is the stationary case of Hamilton-Jacobi equation (See [11] for details). The freezing algorithm reduces the number of points visited to a small portion of the image, leading to accurate segmentation in 10 seconds, for the MR datasets shown in figure 2, on a commercial computer.

3.6 Conclusion

Concerning tree tracking application, 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, therefore enabling the centering of trajectories inside those tubular structures. Further work will concern the extraction of the multiple trajectories, together with the segmentation step, in a single process, for visualization and quantification of pathologies. Those trajectories can be the input to an endoscopic tool, as done in [6].

References

[1] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. 3D object modeling via minimal surfaces. In *ECCV96*.

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

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

[4] L. D. Cohen and T. Deschamps. Grouping connected components using minimal path techniques. In *Proc. CVPR'01*, Kauai, Hawai, December 2001.

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

[6] T. Deschamps and L. Cohen. Minimal paths in 3D images and application to virtual endoscopy. In *Proc. ECCV00*, Dublin, June 2000.

[7] T. Deschamps and L. Cohen. Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Medical Image Analysis*, 5(4):281–299, December 2001.

[8] L. Lorigo, O. Faugeras, W. Grimson, R. Keriven, R. Kikinis, A. Nabavi, and C. Westin. Curves: Curve evolution for vessel segmentation. *Medical Image Analysis*, 5(3):195–206, September 2001.

[9] R. Malladi and J. Sethian. A real-time algorithm for medical shape recovery. In *ICCV98*, 1998.

[10] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: A survey. *Medical Image Analysis*, 1(2):91–108, 1996.

[11] J. Sethian. *Level set methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge University Press, 2nd edition, 1999.

[12] A. Vasilevskiy and K. Siddiqi. Flux maximizing geometric flows. In *Proc. ICCV'01*, volume 1, pages 149–154, Vancouver, Canada, 2001.