

# Fast Evolution of Image Manifolds for Segmentation

Thomas Deschamps  
Mathematics Department  
University of California, Berkeley  
Berkeley, CA 94720, USA  
*e-mail: TDeschamps@lbl.gov*

Ravi Malladi  
Computer Science Department  
Lawrence Berkeley National Laboratory  
Berkeley CA 94720, USA  
*e-mail: R\_Malladi@lbl.gov*

## Abstract

*We present a new fast approach for segmentation of object with missing boundaries based on the Subjective Surface [15] combined with the Fast-Marching algorithm [17]. The Subjective-Surface [15] deals with constructing perceptually meaningful interpretation from partial image data by mimicking the human visual system. However, initialization of the surface is critical for the final result, and its main drawback is very slow convergence and a huge number of iterations required. In this paper, we address those two problems. We first show that the governing equation for the subjective surface flow can be re-arranged in an AOS implementation, providing a near real time solution to the shape completion problem in 2D and 3D. Then we devise a new initialization paradigm based on the Fast-Marching algorithm [17]. We compare the original method with our new algorithm on several examples of real 3D medical images, thus revealing the improvement achieved.*

## 1 Introduction

The Subjective-Surface [15] deals with constructing perceptually meaningful interpretation from partial image data.

One advantage of this method toward Level-Sets [13] is its ability to complete missing boundaries. The approach takes a more general view of the segmentation problem: rather than following a particular interface which one attempts to steer to desired image features, as in [2, 12, 3], it evolves a initial “point-of-view” surface toward a piecewise constant surface, by sharpening it around edges and connecting segmented boundaries across the missing information, thus mimicking the human visual system.

Another advantage is that it does not need re-initialization of a distance function, as detailed in [6]. However, the final segmentation relies on the quality of the initialization of the “point-of-view” surface, and the convergence process is very slow, with lots of iterations required.

In this paper, our goal is to provide a solution to both problems.

Actually, evolving an interface toward image features - as in the Level-Sets formalism - usually presents a numerical integration of a parabolic PDE with one dimension in scale

and two/three dimensions in space. In many instances, numerical integration of these PDEs is the most time consuming operation. This is because the scale step is limited by conditional stability of explicit schemes.

One technique to overcome this computing cost, the Additive Operator Splitting (AOS), has been introduced by *Weickert et al.* [19] for the nonlinear diffusion flow and later applied by *Goldenberg et al.* [5] and *Kuhne et al.* [8] to implement a fast version of the geodesic contour model. A similar work has been done by *Malladi and Ravve* [10], where the authors applied the AOS method to anisotropic diffusion of gray level, and vector-valued imagery with the so-called *Beltrami* flow [18].

In this paper, we first show that the governing equation for the subjective surface flow can be re-arranged in an AOS implementation, providing a near real time solution to the shape completion problem in 2D and 3D.

Then we devise a new initialization paradigm based on the Fast-Marching algorithm [17]. The “point-of-view” surface is now a function of the crossing times of a front propagated in the image domain with the Eikonal equation, based on a speed function related to the grey-level information in this domain. This surface has already been sharpened near the image features, and thus needs less iterations to be evolved to the final piecewise constant surface. We compare the original method with our new algorithm on several examples of real 3D medical images, thus revealing the improvement achieved.

In Section 2, we present and rearrange the governing equation for the subjective surfaces. In Section 3, we introduce the new initialization method, based on the *Fast-Marching* algorithm. Before concluding the article, we show in Section 4 some 2D and 3D results of numerical simulation for completing missing boundaries.

## 2 The Subjective Manifolds

### 2.1 Principle

In this Section, we consider the Subjective Manifolds method for segmentation of images with missing boundaries introduced in [15]. As in [7], we view segmentation as the evolution of an initial reference manifold under the influ-

ence of local image features. Let us consider a 3D manifold  $V : (x, y, z) \rightarrow (x, y, z, \phi)$  defined over the domain  $\Omega$  of a volumetric image  $I(x, y, z)$ . The differential area of the graph  $V$  in the Euclidean space is given by

$$dV = \sqrt{1 + \phi_x^2 + \phi_y^2 + \phi_z^2} \quad (1)$$

We begin by defining a simple edge indicator function acting on  $I$

$$g(x, y, z) = \frac{1}{1 + \left(\frac{\|\nabla I_\sigma\|}{\beta}\right)^2} \quad (2)$$

where  $I_\sigma$  is the initial image with missing boundaries  $I_0$  convolved with a Gaussian kernel of scale  $\sigma$ , and  $\beta$  in equation (2) is a gradient scaling factor.

This edge indicator  $g$  will stretch and shrink a metric appropriately chosen so that the edges act as attractors under particular flow. With the metric  $g$  applied to the space, we obtain a definition of the volume of the manifold as

$$V_g = \int_{\Omega} g \sqrt{1 + \phi_x^2 + \phi_y^2 + \phi_z^2} dx dy dz \quad (3)$$

Considering the internal gradient function

$$h(x, y, z) = \frac{1}{1 + \phi_x^2 + \phi_y^2 + \phi_z^2} \quad (4)$$

the minimizing flow for the volume functional is then given by the steepest descent of equation (3), namely

$$\frac{\partial \phi}{\partial t} = g \frac{\nabla \cdot (h \nabla \phi)}{h} + \nabla \phi \cdot \nabla g \quad (5)$$

which can be expressed in terms of partial derivatives as

$$\begin{aligned} \frac{\partial \phi}{\partial t} = & g \frac{\phi_{xx}(\phi_y^2 + \phi_z^2 + 1) + \phi_{yy}(\phi_x^2 + \phi_z^2 + 1) + \phi_{zz}(\phi_x^2 + \phi_y^2 + 1)}{1 + \phi_x^2 + \phi_y^2 + \phi_z^2} \\ & + \frac{-2\phi_x \phi_y \phi_{xy} - 2\phi_x \phi_z \phi_{xz} - 2\phi_y \phi_z \phi_{yz}}{1 + \phi_x^2 + \phi_y^2 + \phi_z^2} \\ & + \frac{g_x \phi_x + g_y \phi_y + g_z \phi_z}{1 + \phi_x^2 + \phi_y^2 + \phi_z^2} \end{aligned} \quad (6)$$

Initially,  $\phi(x, y, z, t = 0)$  is an inverse distance function to a point or to a finite sector of a straight line:

$$\phi_o = \frac{\alpha}{\sqrt{D^2(x, y, z) + \Delta_S^2}} \quad (7)$$

where  $D$  is the distance function,  $\alpha$  is the initial scaling factor, and  $\Delta_S$  is the smoothing parameter. The value on the boundary is equal to the minimum of  $\phi_o$ .

The process of segmentation is realized in several steps:

1. we select a fixation point and build an initial volume;
2. this ‘‘point-of-view’’ volume is evolved according to equation (6);
3. we pick the level-set that describes the desired object.

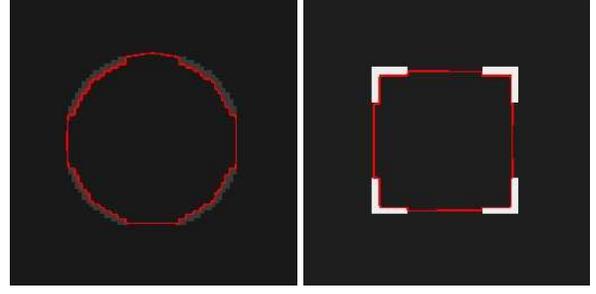


Figure 1: Subjective contours results on synthetic images with missing boundaries.

This flow is quite effective in segmenting objects with missing boundaries in 2D and 3D, as shown in figure 1. Since it deals with complex topologies and is implemented with a PDE, it is natural to compare this flow to the Classical Level-Sets implementation [13] of the Active Contours Model [12, 3]. In particular, on the problem of completing missing boundaries, we can focus on the work of Paragios [14] adapting the Gradient-Vector-Flow method of [20] to the Level-Sets method. The subjective manifolds does not need to filter the gradient vector field, and does not need any re-initialization of the distance function [6, 1]. But its main drawback is (very) slow convergence and the number of iteration required. We are going to derive a semi-implicit scheme similarly to that used for the *Beltrami* flow in [10].

## 2.2 The AOS method

As suggested by *Weickert* in [19], with the help of the internal gradient function of equation (4) the first term of equation (5) can be split into the additive form:

$$\frac{\nabla \cdot (h \nabla \phi)}{h} = \frac{1}{h} \left[ \frac{\partial(h\phi_x)}{\partial x} + \frac{\partial(h\phi_y)}{\partial y} + \frac{\partial(h\phi_z)}{\partial z} \right]$$

The completion flow of equation (6) becomes:

$$\phi_t = (\mathbf{A}_x + \mathbf{A}_y + \mathbf{A}_z)\phi$$

where

$$\begin{cases} \mathbf{A}_x \phi = \frac{g}{h} \frac{\partial(h\phi_x)}{\partial x} + \frac{\partial(g\phi_x)}{\partial x} - g \frac{\partial^2 \phi}{\partial x^2} \\ \mathbf{A}_y \phi = \frac{g}{h} \frac{\partial(h\phi_y)}{\partial y} + \frac{\partial(g\phi_y)}{\partial y} - g \frac{\partial^2 \phi}{\partial y^2} \\ \mathbf{A}_z \phi = \frac{g}{h} \frac{\partial(h\phi_z)}{\partial z} + \frac{\partial(g\phi_z)}{\partial z} - g \frac{\partial^2 \phi}{\partial z^2} \end{cases} \quad (8)$$

where  $g$  is the edge indicator function (2), and we replace  $\frac{\partial}{\partial x} \frac{1}{g} \frac{\partial u}{\partial x}$  by  $\frac{\partial}{\partial x} \left( \frac{1}{g} \frac{\partial u}{\partial x} \right) - \frac{1}{g} \frac{\partial^2 u}{\partial x^2}$  in equations (5). Therefore the resulting linear system will be a tri-diagonal matrix, thus more stable than a system with no value on the diagonal.

This new form of equation (6) hides the mixed derivatives and leads to the AOS approach.

Applying the backward difference formula to the equation (8) we get

$$\frac{u^{n+1} - u^n}{\Delta t} = (\mathbf{A}_x + \mathbf{A}_y + \mathbf{A}_z)u^{n+1} \quad (9)$$

Using  $u^{n+1}$  on the right side of equation (9) makes the integration scheme implicit and unconditionally stable, namely

$$[\mathbf{I} - \Delta t(\mathbf{A}_x + \mathbf{A}_y + \mathbf{A}_z)]u^{n+1} = u^n \quad (10)$$

where  $\mathbf{I}$  is the identity matrix. Before proceeding in time, we calculate the values of the edge indicator function  $h$  in equation (4), using the known values of  $u^n$ . Thus, the scheme is only semi-implicit. Although  $h$  depends on the gradient of  $u$ , we treat it like a given function of  $(x, y)$ , making the governing PDE ‘‘quasi-linear’’.

Note that equation (10) includes a large bandwidth matrix, because all equations, related to new pixel values  $u^{n+1}$  are coupled. Our aim is to decouple the set in equation (10) so that each row and each column of pixels can be handled separately. For this, we re-arrange the equations into the following form:

$$u^{n+1} = [\mathbf{I} - \Delta t(\mathbf{A}_x + \mathbf{A}_y + \mathbf{A}_z)]^{-1}u^n \quad (11)$$

Of course, we do not intend to invert the matrix to solve the linear set. This is only a symbolic form used for further derivation. For a small value of  $\Delta t$ , the matrix in the brackets on the right side of Equation (11) is close to the identity  $\mathbf{I}$ . Thus, its inverse can be expanded into the Taylor series in the proximity of  $\mathbf{I}$ :  $[\mathbf{I} - \Delta t(\mathbf{A}_x + \mathbf{A}_y + \mathbf{A}_z)]^{-1} \approx \mathbf{I} + \Delta t(\mathbf{A}_x + \mathbf{A}_y + \mathbf{A}_z)$ , where the linear term is retained and the high order terms are neglected. Introducing this form into equation (11), we get

$$u^{n+1} = \frac{1}{3} \sum_{l \in \{x, y, z\}} (\mathbf{I} + 3\Delta t \mathbf{A}_l) u^n \quad (12)$$

Introducing the notations  $V = (\mathbf{I} + 3\Delta t \mathbf{A}_x)u^n$ ,  $W = (\mathbf{I} + 3\Delta t \mathbf{A}_y)u^n$  and  $Z = (\mathbf{I} + 3\Delta t \mathbf{A}_z)u^n$  the solution is simply

$$u^{n+1} = \frac{V + W + Z}{3} \quad (13)$$

In order to get an implicit scheme, we apply the differential matrix operators  $\mathbf{A}_x$ ,  $\mathbf{A}_y$  and  $\mathbf{A}_z$  to  $u^{n+1}$ , namely for  $x$ :  $(\mathbf{I} + 3\Delta t \mathbf{A}_x)^{-1}V = u^n$ . Following the procedure of expanding the matrix inverses into Taylor series and applying the linearization for small  $\Delta t$ , we finally obtain the equation sets for  $V$  as follows:  $(\mathbf{I} - 3\Delta t \mathbf{A}_x)V = u^n$ .

This leads, for the differential operator in  $x$  to the semi-implicit linearized numerical scheme described by the following equation

$$-\alpha_m V_{i-1} + [1 + (\alpha_m + \alpha_p)]V_i - \alpha_p V_{i+1} = u_i^n$$

with

$$\begin{cases} \alpha_m = \frac{3\Delta t}{4\Delta_x^2} \left( g_{i-1} + g_i \frac{h_{i-1}}{h_i} \right) \\ \alpha_p = \frac{3\Delta t}{4\Delta_x^2} \left( g_{i+1} + g_i \frac{h_{i+1}}{h_i} \right) \end{cases}$$

These equations can be solved with the Dirichlet boundary conditions along the contour of the computational box; these and other details are described in [10].

Considering the *Courant-Friedrich-Levy* (CFL) condition [9] for the explicit scheme, in the case of an anisotropic cubic grid the maximum time step allowed should be  $\frac{1}{2} \left( \frac{1}{\Delta x^2 + \Delta y^2 + \Delta z^2} \right)$ . The unconditionally stable scheme that we are using here enables us to multiply this time step by an acceleration factor  $f$ . As we see, the implicit scheme is always stable, but for  $f \gg 50$ , the resulting accuracy may be insufficient for certain applications.

To achieve the optimum accuracy for minimum computation time, we apply non-uniform scale step, assuming that more coarse steps are allowed at the initial stage of marching and finer steps are required toward the end. The steps in scale make a decreasing geometric progression. The last scale step is kept the same for all problems considered, and it corresponds to the acceleration factor 10. The average acceleration factor  $f$  changes at every step, and is constrained to be greater than 10. The average acceleration factor yields the same total number of steps  $N$  as if this factor was kept constant:  $N = \frac{4S}{f}$  where  $S$  is the total scale of marching.

### 3 Initializing with the *Fast-Marching*

In this section, we want to build a new initialization framework for the point-of-view volume. This manifold deforms slowly under the action of the flow (6).

#### 3.1 The *Fast-Marching* algorithm

Considering  $P > 0$ , the *Eikonal* equation:

$$\|\nabla T\| = P \quad (14)$$

is the stationary case of the Hamilton-Jacobi equation. It computes the front propagation in a media  $P$  defined over the image domain, with a speed  $F = \frac{1}{P}$ . Classic finite difference schemes for this equation tend to overshoot and are unstable. *Sethian* [16] 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 over-shooting of finite differences. At each voxel  $(i, j, k)$ , the unknown crossing-time  $t$  of the front satisfies:

$$\begin{aligned} & (\max\{t - T_{i-1,j,k}, t - T_{i+1,j,k}, 0\})^2 + \\ & (\max\{t - T_{i,j-1,k}, t - T_{i,j+1,k}, 0\})^2 + \\ & (\max\{t - T_{i,j,k-1}, t - T_{i,j,k+1}, 0\})^2 = \tilde{P}_{i,j,k}^2. \end{aligned} \quad (15)$$

giving the correct viscosity-solution  $t$  for  $T_{i,j,k}$ . The improvement made by the *Fast-Marching* is to introduce order in the selection of the grid points, leading to an algorithm in  $o(M \log M)$  for  $M$  grid points. This order is based on the fact that information is propagating *outward*, because the time  $t$  can only grow due to the quadratic equation (15).

The *Fast-Marching* algorithm has been used for segmentation purposes, in particular its good properties contributed already to initialize and therefore accelerate the *Level-Sets* methods in [11]. For a detailed study on the algorithm itself, see [17], and for its applications to medical imaging see [4].

### 3.2 New point-of-view surface

The reason why we want to use the *Fast-Marching* algorithm here is to replace the “point-of-view” volume initialization. The *Fast-Marching* propagates a front in the image domain, at a speed inversely proportional to a chosen “potential” function  $P(x, y, z)$ . If this potential has low values in a desired region, the crossing-time of voxels in this region are going to be small, while in the neighborhood of this region they will greatly increase, thus creating a “point-of-view” which already contains a lot of information. The slope of the crossing-time of the *Eikonal* equation will help drive the initial surface to the final segmentation in a rapid manner. In other words, using this method, we are providing a better initial condition to the subjective surface flow. For this purpose, we take a potential function defined by the edge indicator function of equation (2)

$$P(x, y, z) = \frac{1}{1 + \left(\frac{\|\nabla I_\sigma\|}{\beta}\right)^2} \quad (16)$$

The new “point-of-view” manifold is computed using the crossing-times  $T$  computed with the *Fast-Marching* algorithm.  $\phi$  is now the inverse of a *weighted* distance function

$$\phi_o = \frac{\alpha}{\sqrt{T^2(x, y) + \Delta_S^2}} \quad (17)$$

where  $T$  is the solution to the *Eikonal* equation (14),  $\alpha$  is the initial scaling factor, and  $\Delta_S$  is the smoothing parameter. The value on the boundary is equal to the minimum of  $\phi_o$ .

The new segmentation framework is now devised in the following steps:

1. The user select a fixation point and build an initial volume;
2. the crossing-time of the *Eikonal* equation (14) are computed on the image domain  $\Omega$ ;
3. the “point-of-view” volume is derived from this time map;
4. the user picks an iso-surface that describes roughly the desired object;
5. the subjective Manifold flow is solved for a small number of iterations.

In figure 2, we can see the difference between the normal initialization of equation (7) on the first row, and using the *Fast-Marching* as initialization with equation (17) in the second row. In this test, we want to delineate the contour

of the brain displayed in the last column of figure 2. The image has a lack of contrast near the sinus and in the hole which is located on the skull. Therefore classical segmentation paradigm will fail to segment the contour of the brain. For example the classic formulation of the geodesic active contour [3] will probably lead to a bad segmentation in these regions where there is absence of evidence. Subjective contours are well-suited for this problem with missing boundaries, as illustrated in figure reffig:synthetic2D.

But the framework using the *Fast-Marching* is even better suited to solving this problem: when the user has computed the crossing-times with equation 14, he can initialize the point-of-view surface as shown in figure 2 on the second row. This point-of-view surface contains already a lot of information which enables to quickly converge and extract the contour of the brain.

Considering an acceleration factor of 100 and the classical CFL condition, four iterations are now necessary to obtain the results shown in the second row of figure 2. While several other time-steps are necessary for the classical subjective surface to match a possible result, the new framework is already providing a correct result.

The final contour extracted is represented in both cases of initialization on the last column of figure 2.

## 4 Completing Missing Boundaries

### 4.1 Results on 2D images

Ultrasound images are an interesting examples: despite the improvements in terms of probe, reconstruction and speed of acquisition, images are still very noisy. But the non-invasiveness and low-cost specificities of this imagery is much likely to make its importance and use continue to grow.

The subjective manifold technique is especially well-suited for the segmentation of ultrasound images where missing boundaries are very important since the signal is attenuated in all except the probe direction. Two simple results of segmentation obtained with this technique are displayed in figure 3.

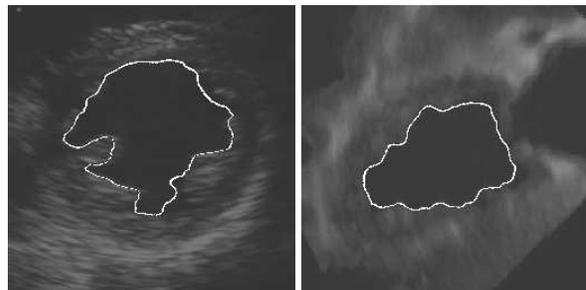


Figure 3: Subjective contours results on ultrasound images

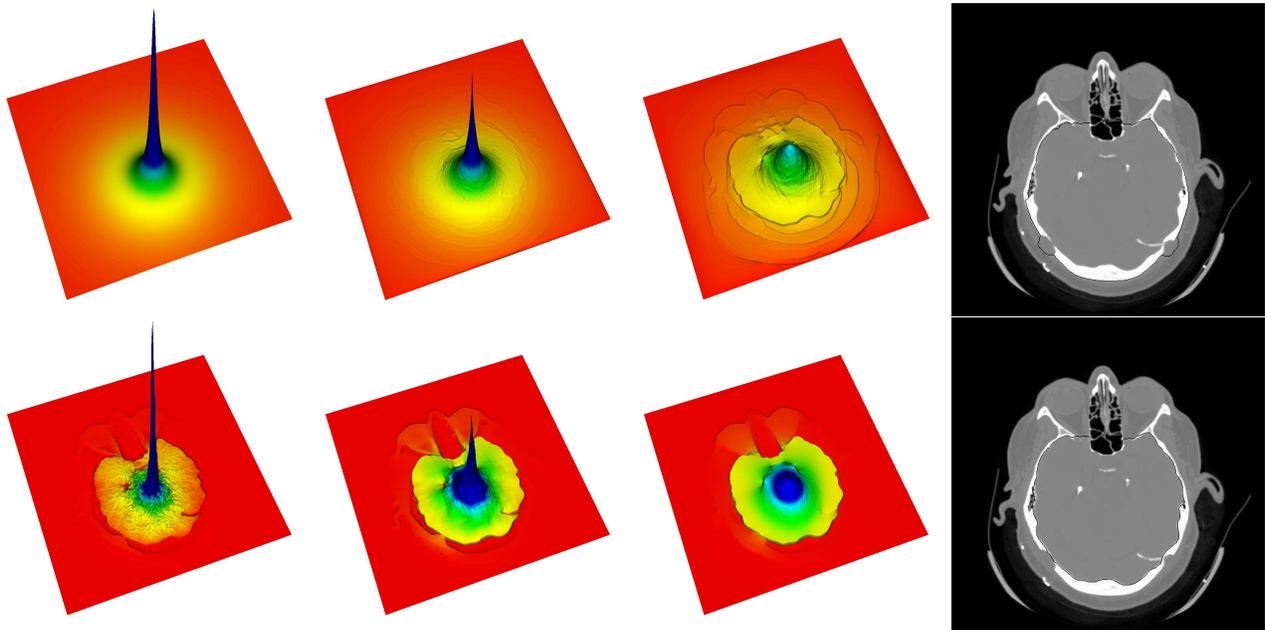


Figure 2: Comparing the Extraction of the contour of the brain in a CT slice, using the Subjective Surfaces with different initializations: first row is initialized with equation (7) and second row is the initialization with the *Eikonal* equation (17). The first column represent the initialization step; the second and third columns are the same representations at times 100 and 1000.

## 4.2 Results on 3D medical images

In figure 4 is displayed the segmentation result of our new framework on a 3D CT scanner of the head, from which a slice was extracted to study the impact of the *Fast-Marching* initialization in figure 2. The cerebral tissues are not depicted in the grey-level information of the dataset. However, the target here is to reconstruct the envelope of the brain, avoiding leakage into the holes of the patient skull which result from previous surgery. This envelope could be used to do registration with other acquisition of the patient head.

The initialization of this dataset is obtained by computing the *Eikonal* equation on the whole domain, starting from any point centered inside the dataset. This process takes approximately 20 seconds on a laptop with a 1.7MHz processor and 1Go of RAM. The first row of figure 4 represents 3 views of the propagation front at crossing time 330 (manually chosen). Notice that this manual selection is as simple as clicking in one of the slices of the dataset, near the skull.

The second row represents the result of 100 iterations of the Subjective Manifolds model, starting from the initialization given by the *Fast-Marching*.

Figure 5 contains three different axial slices of the 3D medical dataset, represented together with the superimposed contour of the segmentation obtained (in blue). The contour clearly avoids the holes in the sinuses in the left axial slice, and the holes in the skull in the middle and right slices, thus depicting the boundaries of the brain while there is an absence of evidence on those boundaries.

This process takes 12 seconds on the same computer. The

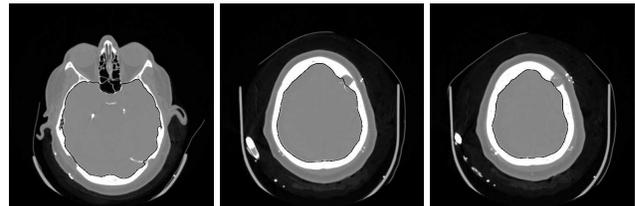


Figure 5: Intersection of the final isosurface with three different axial slices where boundaries are missing in the skull.

problem of *leakage*, due to the holes in the skull is overcome by the Subjective Manifolds algorithm, by completing the missing boundaries in interactive time.

The same methodology can be applied to extract the contour of the heart in an ultrasound image, as shown in figure 6. From left to right, the first image represents the front initialized by the *Fast-Marching* algorithm, and then progressively refined by the Subjective manifolds at time  $t = 100, 200, 300$ . The total process does not exceed the minute, for this example where the image dimensions are  $151 \times 151 \times 101$ .

The first column of figure 7 represents two views of a volume of interest of a CT dataset of the brain. The first row comprises segmentation results of the brain ventricles in this dataset, using the *Fast-Marching* algorithm starting from a seed point in each ventricle, where the speed of the front is computed as a function of the image gray-levels, similarly

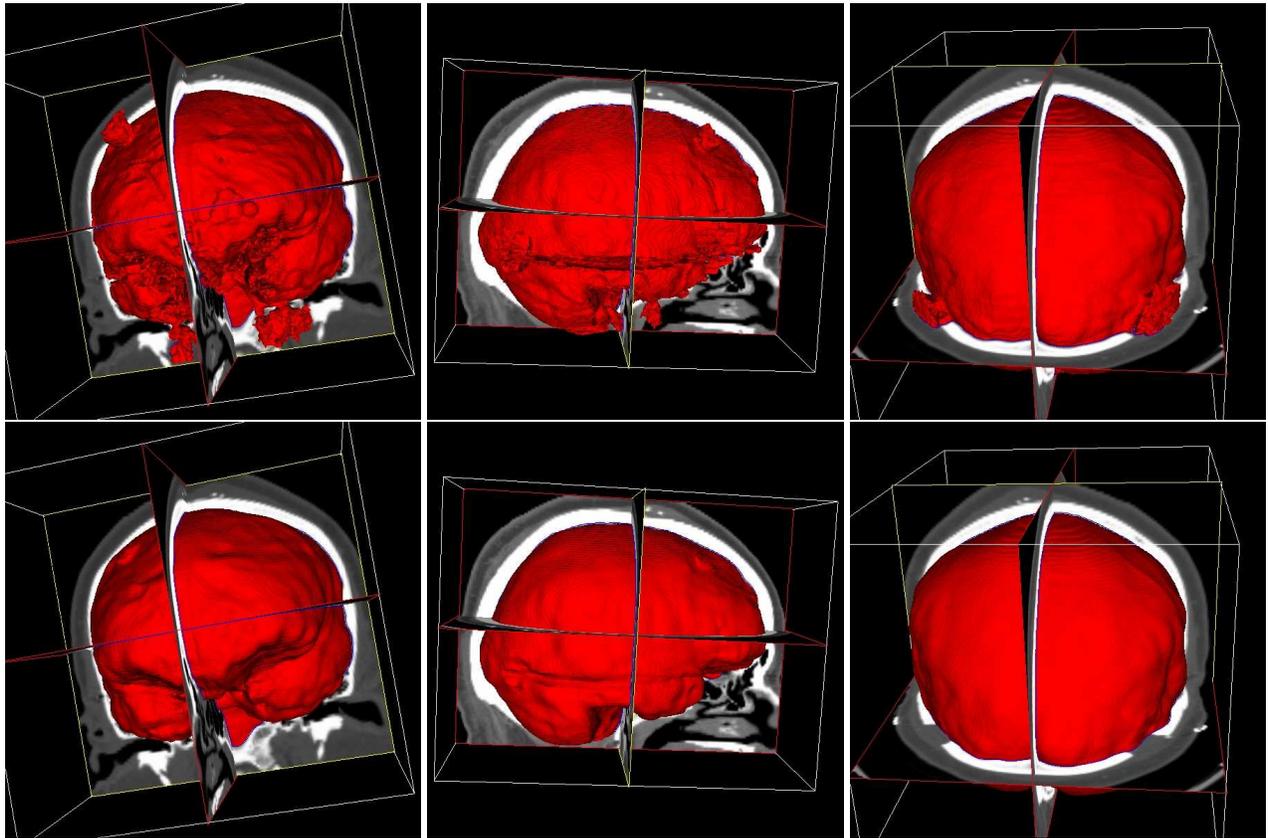


Figure 4: Segmentation of the Brain in a CT scanner: The 3d images in the first row are different views of the initial guess obtained with the *Fast-Marching* algorithm; the corresponding images in the second row represent the final segmentation after 100 iterations of the Subjective Manifolds;

to the model used in [11]. Following previous examples, the second row depicts the final segmentation obtained using the Subjective manifolds. It is important to notice the difficulty of the problem in figure 7: there is no evidence of a clear separation between the ventricles, explaining why our method results in a single segmented object where they are joined.

## 5 Conclusion

Based on the approach introduced by *Weickert et al.* [19], the unconditionally stable difference scheme is developed for the computation of subjective manifolds, which is a very efficient segmentation algorithm for completing missing boundaries.

The implicit scheme leads to considerable saving of the computational time as compared to the explicit scheme: up to ten times and even more, depending on the value of the scale step, with no visible loss of accuracy. This overcomes the main drawback of the Subjective Surface method.

The second most important drawback being the initialization of the "point-of-view" surface, we introduce a new framework based on the *Fast-Marching* equation [17]. Initializing this surface with the crossing time of the *Eikonal*

equation both enhances the result and reduce the computing time, by giving an initial surface very close to the final result.

The new framework obtained is very efficient, since it does not need to re-initialize a distance function [1], unlike the classical Level-Sets implementation [13]. It completes in real-time missing boundaries, as we show it on several examples like synthetic images and real medical images.

## References

- [1] D. Adalsteinsson and J.A. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148:2–22, 1999.
- [2] V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours. *Numerische Mathematik*, 66:1–31, 1993.
- [3] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.
- [4] T. Deschamps. *Curve and Shape Extraction with Minimal Path and Level-Sets techniques - Applications to 3D Medical Imaging*. PhD thesis, Université Paris-IX Dauphine, Place du maréchal de Lattre de Tassigny, 75775 Paris Cedex, December 2001.
- [5] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *IEEE Transactions on Image Processing*, 10(10):1467–1475, October 2001.
- [6] R. Keriven. *Equations aux Dérivées Partielles, Evolutions de Courbes et de Surfaces et Espaces d'Echelle: Application à la Vision par Ordinateur*. PhD thesis, Ecole Nationale des Ponts et Chaussées, 1997.

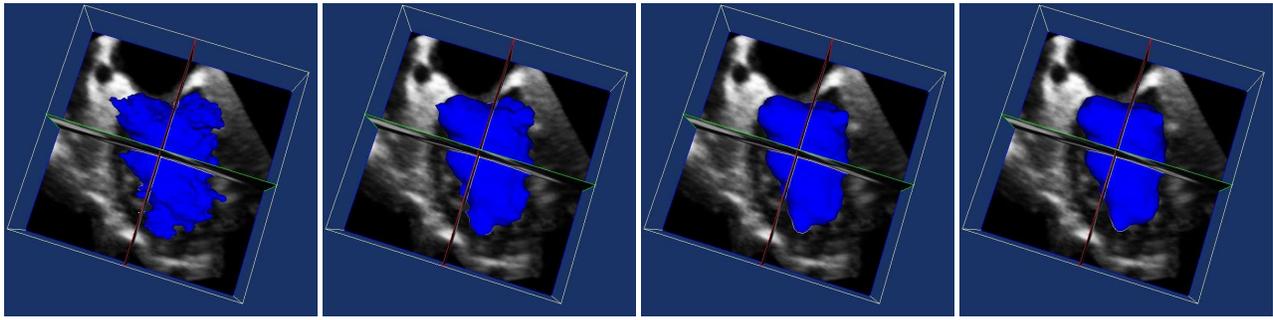


Figure 6: Segmentation of the heart left-ventricle in an ultrasound image: starting from the *Fast-Marching* initialization, the process converges quickly to the final surface.

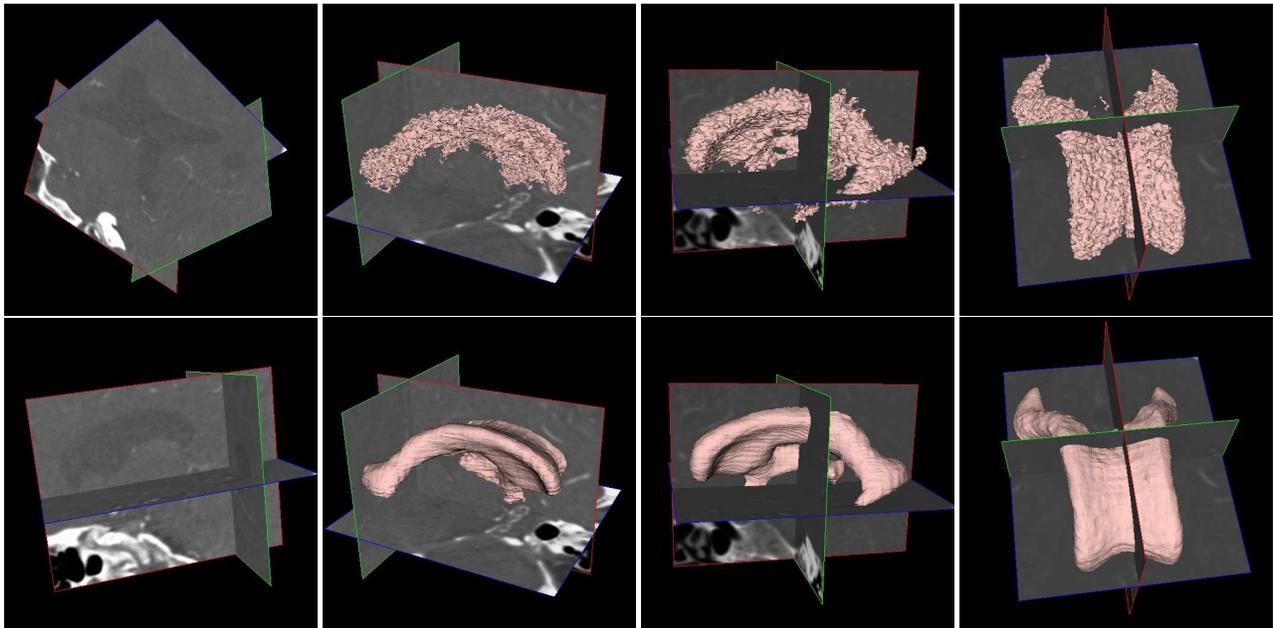


Figure 7: Segmentation of the brain ventricles.

- [7] R. Kimmel, R. Malladi, and N. Sochen. Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images. *International Journal of Computer Vision*, 1999.
- [8] G. Kuhne, J. Weickert, M. Beier, and W. Effelsberg. Fast implicit active contour models. In L. Van Gool, editor, *DAGM 2002*, volume 2449 of *Pattern Recognition, Lecture Notes in Computer Science*, pages 133–140. Springer, Berlin, 2002.
- [9] R.J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhauser, 1992.
- [10] R. Malladi and I. Ravve. Fast difference schemes for edge-enhancing beltrami flow. In *Proceedings of the Seventh European Conference on Computer Vision (ECCV'02)*, Copenhagen, Denmark, May 2002.
- [11] R. Malladi and J.A. Sethian. A real-time algorithm for medical shape recovery. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'98)*, pages 304–310, January 1998.
- [12] R. Malladi, J.A. Sethian, and B.C. Vemuri. Shape modelling with front propagation: A level set approach. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 17(2):158–175, February 1995.
- [13] S. Osher and J.A. Sethian. Fronts propagating with curvature dependent speed: algorithms based on the hamilton-jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.
- [14] N. Paragios, O. Mellina-Gotardo, and V. Ramesh. Gradient vector flow fast geodesic active contours. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV'01)*, volume 1, pages 67–73, Vancouver, Canada, 2001.
- [15] A. Sarti, R. Malladi, and J.A. Sethian. Subjective surfaces: A method for completing missing boundaries. *International Journal of Computer Vision*, 46(3):201–221, 2002.
- [16] J.A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences of the United States of America*, 93(4):1591–1595, February 1996.
- [17] J.A. Sethian. *Level set methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge University Press, University of California, Berkeley, 2nd edition, 1999.
- [18] N. Sochen, R. Kimmel, and R. Malladi. A general framework for low level vision. *IEEE Transactions on Image Processing*, 7(3):310–318, march 1998. Special Issue on Geometry Driven Diffusion.
- [19] J. Weickert, B.M. ter Haar Romeny, and M.A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, March 1998.
- [20] C.Y. Xu and J.L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, March 1998.