Nicolas Forcadel, Carole Le Guyader and Christian Gout

Generalized Fast Marching Method: Applications to Image Segmentation

Numerical Algorithms, in press 2008

DOI : 10.1007/s11075-008-9183-x

LINK to ARTICLE

Generalized Fast Marching Method: Applications to Image Segmentation.

[1,2]Nicolas Forcadel, [3,4]Carole Le Guyader, [5,6,7]Christian Gout

Running title: **GFMM: Applications to Image Segmentation.**
Affiliations:
[1] : Projet Commands, CMAP-INRIA Futurs, Ecole Polytechnique, 91128 Palaiseau, France
[2] : ENSTA, UMA, 32 Bd Victor, 75739 Paris Cedex 15, France.
[3] : INSA de Rennes, Centre de Mathématiques de l'INSA, 20 Avenue des Buttes de Coësmes, CS 14315, 35043 Rennes cedex, France
[4] (summer 2007): University of California, Los Angeles, Department of Mathematics, 405, Hilgard Avenue, Los Angeles, CA 90095-1555, U.S.A.
[5] : Université de Valenciennes, LAMAV-ISTV2, Le Mont Houy, 59313 Valenciennes Cedex 9, France
[6] : INSA de Rouen, Laboratoire de Mathématiques de l'INSA, Place Emile Blondel, BP 08, 76 131 Mont-Saint-Aignan Cedex, France.
[7] : INRIA Futurs Research Center Team-Project Magique3D, Pau, France.
E-mail addresses: forcadel@cemrics.enpc.fr, chris_gout@cal.berkeley.edu, carole.le-guyader@insa-rennes.fr

Abstract. *In this paper, we propose a segmentation method based on the Generalized Fast Marching Method (GFMM) developed by Carlini et al.([6]). The classical Fast Marching Method (FMM) is a very efficient method for front evolution problems with normal velocity (see also [13]) of constant sign. The GFMM is an extension of the FMM and removes this sign constraint by authorizing time-dependent velocity with no restriction on the sign. In our modelling, the velocity is borrowed from the Chan-Vese model for segmentation ([9]). The algorithm is presented and analyzed and some numerical experiments are given, showing in particular that the constraints in the initialization stage can be weakened and that the GFMM offers a powerful and computationally efficient algorithm.*

Keywords: Fast marching method, Level Set methods, Chan-Vese model for segmentation.
Mathematics Subject Classification: 65M06, 68U10, 49Lxx

1 Introduction

The main goal of this paper is to propose a segmentation model based on the Generalized Fast Marching Method (GFMM). Let us recall that segmentation is the process that consists in detecting and visualizing the boundaries of the objects contained in an image.

While the Fast Marching Method (FMM) is useful when tracking a propagating front that evolves with a constant sign speed, the GFMM offers more general assumptions on the velocity and in particular no sign restrictions. This last element motivates the proposed work in the sense that many of the well-known variational segmentation methods require a careful choice of the initial condition. One of the most famous variational methods to process this partition of the image is the Active Contour model introduced by Kass, Witkin and Terzopoulos ([18]). It consists in evolving a parameterized curve so that it matches the object boundary. The shape taken by the curve through the process is related to an energy minimization. This energy comprises a regularization term and a data-fitting term and is non-convex. Thereof, we can only expect local minimizers. In practice, it means that the contour to be deformed must be initialized near the object boundary. Cohen ([12]) has proposed a way to alleviate this constraint by adding an inflating/deflating force in the modelling defined by $k\vec{n}$, $\vec{n}$ denoting the unit outward normal to the curve and k being a constant. According to the sign of the constant k, the curve inflates or deflates. Thereby, in practice, the contour to be deformed is either initialized inside the object or it encloses the object of interest.

In [37], Xu and Prince address both the problem of initialization and the one of slow/poor convergence near boundaries with strong concavities by introducing a new static external force called gradient vector flow. The initialization constraint is removed, that is, initialization can be made inside, outside or across the object boundaries and the front evolution is easily handled even in boundary concavities.

The main idea behind this model is to increase the capture range of the external edge-map-related force field and to make the contour progress toward the desired boundaries, here where classical methods would fail to. Unlike classical active contours, the introduced external force does not derive from a potential function and cannot be computed straightforwardly from the image edge map. More precisely, the model cannot be phrased in terms of a unique functional minimization problem but is defined in two steps. In a first step, the external force (*gradient vector flow*) v is obtained by minimizing an energy functional in a variational framework. Denoting by ∇f the gradient field of an edge map, the energy is designed such that when $|\nabla f|$ is large, it is minimized by setting $v = \nabla f$ and when $|\nabla f|$ is small, the resulting v is smooth and slowly varying. The Euler-Lagrange equations are computed and lead to solve a linear partial differential equation decoupled system, each equation traducing an equilibrium state. A gradient descent method is then applied and the discretization is made using finite difference schemes.

The second step consists thus in replacing, in the dynamic snake equation, the classical potential force by the newly computed external force v.

In the proposed model, we aim, as done in [37], at alleviating the choice of the initial curve but contrary to [37], the algorithm we propose is not a two-step one. Moreover, our model authorizes a larger choice of initializations.

Coming back to the classical active contour model, another drawback is that it is non-intrinsic. It means that a reparameterization of the curve modifies the expression of the energy. In [8], Caselles *et al.* address this issue and prove that with certain assumptions, the classical active contour model is equivalent to searching for a geodesic curve in a Riemann space whose metric is linked to the image content (see also [19], [10] or [4] for fast global minimization of the geodesic active contour model, and [16], [15] or [21] for applications of this method with geometrical constraints). More precisely, if the considered curve is defined by $\mathcal{C} : \left\{ \begin{array}{l} [0,1] \rightarrow \mathbb{R}^2 \\ q \mapsto \mathcal{C}(q) \end{array} \right.$, Caselles *et al.* prove that it boils down to minimize the following functional:

$$\int_0^1 g(|\nabla I(\mathcal{C}(q))|)|\mathcal{C}'(q)|dq, \tag{1}$$

with I the image and where g is an edge-detector function satisfying the following properties:

$$g : [0, +\infty[\rightarrow [0, +\infty[, \; g \text{ positive, strictly decreasing such that } g(0) = 1 \text{ and } \lim_{r \rightarrow +\infty} g(r) = 0. \tag{2}$$

Let us stress that in this case, the problem is intrinsic since a reparameterization of the curve leaves the energy unchanged.

The evolution equation of the curve $\mathcal{C}$ to deform the initial contour $\mathcal{C}(\cdot, 0)$ towards a local minimum of (1) is established (see [8] for further details) and one obtains:

$$\frac{\partial \mathcal{C}}{\partial t}(q,t) = g(|\nabla I(\mathcal{C}(q,t))|)\kappa(q,t)\vec{n}(q,t) - < \nabla g(|\nabla I|)(\mathcal{C}(q,t)), \vec{n}(q,t) > \vec{n}(q,t), \tag{3}$$

with $\kappa(q,t)$ the curvature at point $\mathcal{C}(q,t)$, $\vec{n}(q,t)$ the unit inward normal vector to the curve at point $\mathcal{C}(q,t)$ and $< \cdot, \cdot >$ the euclidean scalar product in $\mathbb{R}^2$. The model is then cast in the level set setting to allow topological changes.

Here again, in this modelling, the initial contour generally encloses the objects one aims at detecting. By adding a constant evolution term of the kind $kg(|\nabla I(\mathcal{C}(q,t))|)\vec{n}(q,t)$ in (3) however, it can be initialized inside the object.

In [22], Malladi and Sethian propose a shape-detection method that combines Fast Marching and Level Set methods. We will give more details in the section dedicated to FMM. But in this method as well, the contour is initialized inside the desired region and grows.

In [9], Chan and Vese introduce a segmentation method that aims at establishing the best partition of the image as a set of two regions with approximatively piecewise-constant intensities. The model is related to the Mumford-Shah functional (see [25]), is stated in the level set framework and contrary to the methods presented above, boundaries are not defined by the image gradient. In practice, the initial contour can be anywhere in the image and the method allows simultaneous detection of interior and exterior contours. Extensions have been proposed to partition the image in more than two piecewise-constant intensity regions using several level set functions (see [36]) or several isocontours of a same level set function (see [11]).

The model we propose is grounded on the region-based approach developed by Chan and Vese [9]. But as will be seen later, our presented approach is different in several aspects: first, our model does not stem from an energy-minimization problem but is directly stated in terms of a propagating front with a suitable normal velocity. Besides, the mathematical formulation of our model leads to consider a problem whose unknown is the passing time of the front over the nodes of the grid. The front will stop moving as soon as the passing time on nodes becomes infinite. In [9], the energy-minimization problem is cast in the level set framework. The unknown is thus a lipschitz continuous function Φ defined on the image domain $\bar{\Omega}$, whose zero level-line is the contour of interest. The Euler-Lagrange equation satisfied by the unknown is established and a gradient-descent method is applied. The discretization, in our case, is new.

Lastly, the initialization step in our method is straightforward and maybe more flexible than in [9]. Indeed, as previoulsy mentioned, in [9], the initial contour is implicitly represented via an higher-dimensional level set function Φ. Usually, this function is preferred to be a signed distance function for the stability of numerical computations, which implies that one may periodically reinitialize the level set function as the signed distance function to the set $\{x \in \Omega \mid \Phi(x) = 0\}$. In our case, the initialization can be made, for instance, of several shapes, no particular procedure being required.

These elements motivated the work proposed below. The outline of the paper is as follows: in Section 2, we describe the classical Fast Marching Method, and we then introduce the Generalized Fast Marching Method. Section 3 is devoted to the algorithm of the model. We conclude the paper with experimental results on various synthetic and real images in Section 4.

2 The Generalized Fast Marching Method (GFMM)

2.1 A Brief Review of the Fast Marching Method

The Fast Marching Method (Tsitsiklis [35], Sethian [32], [31], [33]) has proved to be very efficient when tracking hypersurfaces moving according to their normal with a constant-sign velocity. Indeed, the evolution problem is rephrased in such a way that one considers the stationary problem whose unknown is the arrival time at the grid points. It means more precisely that we switch from an initial value problem to a boundary value one as shown in [32], both methods being formulated in a Eulerian way as stressed by Sethian .

Let us consider, for the purpose of illustration, an open bounded subset Ω of $\mathbb{R}^2$ and an oriented curve $\mathcal{C} \subset \Omega,\ \mathcal{C} : \left\{ \begin{array}{l} [0,1] \times \mathbb{R}^+ \to \mathbb{R}^2 \\ (q,t) \mapsto \mathcal{C}(q,t) \end{array} \right.$ in two dimensions, separating two regions, that is evolving according to its normal with a positive sign velocity F.

In the boundary value formulation prospect, the unknown is the arrival time of the front $T(x)$ when it crosses the point $x \in \Omega$. T is obviously a single-valued function since the front crosses once and only once a grid point while evolving, no time step is required since the time is itself the unknown (see Carlini *et al.* [6]) and consequently no CFL conditions. Thus the boundary value problem can be formulated as follows:

$$|\nabla T| F = 1 \tag{4}$$

$$\text{with } T = 0 \text{ on } \mathcal{C}(\cdot, 0).$$

The front is therefore characterized at each time t as the set of points for which the arrival time T is equal to t that is

$$\mathcal{C}(t) = \big\{x \in \Omega \mid T(x) = t\big\}.$$

As for the initial value problem, it is linked to the level set formulation of the model (cf. Osher and Sethian [28], Osher and Fedkiw [26], Osher and Paragios [27], Sethian [32],[33], [30]). The evolving contour is embedded in a higher-dimensional lipschitz continuous function Φ and at each time t, is defined as the zero isocontour of the function Φ. (We assume $\Phi < 0$ on the interior w of $\mathcal{C}$ and $\Phi > 0$ on $\Omega - \bar{w}$). The evolution equation of Φ is straightforwardly derived. Since $\forall q \in [0,1]$, $\forall t \in \mathbb{R}^+$, one has $\Phi(\mathcal{C}(q,t),t) = 0$, by differentiating this last relation with respect to t, one gets the following PDE:

$$\frac{\partial \Phi}{\partial t}(\mathcal{C}(q,t),t) + < \nabla\Phi(\mathcal{C}(q,t),t), \frac{\partial \mathcal{C}}{\partial t}(q,t) >= 0,$$

$< \cdot, \cdot >$ denoting the euclidean scalar product in $\mathbb{R}^2$.

The front evolving in the outward normal direction with a speed F (no tangential component), one has $\frac{\partial \mathcal{C}}{\partial t}(q,t) = F\vec{n}(q,t)$ where $\vec{n}(q,t)$ is the unit outward normal vector to the curve at point $\mathcal{C}(q,t)$. Introducing this component into the previous equation and replacing $\vec{n}(q,t)$ by its level set counterpart, we finally get the following PDE:

$$\frac{\partial \Phi}{\partial t}(\mathcal{C}(q,t),t) + F|\nabla\Phi(\mathcal{C}(q,t),t)| = 0.$$

The advantage of the Level Set method is that this equation can be considered as defined on the whole domain $\Omega \times \mathbb{R}^+$. It implies in particular that the velocity F should be known not only on the zero isocontour but also off this level line. This extrapolation is sometimes natural in the considered problem or requires the use of numerical devices (see chapter 8 of [26] or chapter 11 of [32]). This finally yields:

$$\frac{\partial \Phi}{\partial t} + F|\nabla\Phi| = 0, \text{ on } \Omega \times \mathbb{R}^+. \tag{5}$$

equipped with the initial condition $\Phi(x,0) = \Phi_0$. The relation between equations (4) and (5) stems from the following change of variable

$$\Phi(x,t) = T(x) - t.$$

In fact, it is equivalent to searching for the zero level set of $\Phi(x,t)$ or the level set t of $T(x)$.

Coming back to the boundary value problem (4), a powerful discrete algorithm can be run (see Sethian [32], [31], Bærentzen [2]), using numerical schemes borrowed from hyperbolic conservation laws. Let us notice that information flows outward (if we consider $F > 0$) from smaller values of T to bigger ones. To match this property, Sethian ([32],[31]), Rouy and Tourin [29] devised upwind numerical schemes. Considering a uniform mesh, denoting by h the space step in both directions and denoting by $T_{i,j}$ an approximation of T at the node $(x_i, y_j) = (ih, jh)$, Rouy and Tourin [29] propose the following scheme to approximate the gradient of T:

$$|\nabla T|_{ij} \simeq \left[\max\ \left(D_{ij}^{-x}T, -D_{ij}^{+x}T, 0\right)^2 + \max\ \left(D_{ij}^{-y}T, -D_{ij}^{+y}T, 0\right)^2\right]^{\frac{1}{2}}, \tag{6}$$

with $D_{ij}^{-x}T = \dfrac{T_{i,j} - T_{i-1,j}}{h}$ and $D_{ij}^{+x}T = \dfrac{T_{i+1,j} - T_{i,j}}{h}$.

Thus starting from an initial band made of grid points that have been assigned the value zero, the idea is to march out in the normal direction and update the values of the grid points that are crossed over. We briefly describe the algorithm: a more detailed depiction can be found in Bærentzen's technical report [2] or in [32].

In order to alleviate the computational cost, a classification of the grid points is introduced. The idea is to concentrate the computational effort on the points contained in a narrow band around the front. The grid points belonging to the initial band (it can be either one grid point or several ones) are declared frozen and are assigned the value zero. The 4-neighborhood grid points of the frozen points which are not yet frozen are said to belong to the narrow band. For each of them, a tentative value of the arrival time T is computed.

To compute this tentative value, one only uses the values of the frozen grid points, it means that the obtained value does not depend on the other tentative values (see chapter 7 from [26] or chapter 8 from [32]). The first grid point from the narrow band to be updated is the one first crossed over by the front, that is the one for which the tentative value is the smallest. This point is tagged frozen and its value is no longer changed. The process is then repeated: the narrow band has almost not changed. Only the 4-neighborhood grid points of the newly accepted grid point must be looked into. Either their tentative value had not been computed and consequently they must be tagged 'narrow band' and assigned a value, or their tentative value must be recomputed to take into account the value of the newly frozen grid point. As stressed by Bærentzen, this method consists therefore in propagating the narrow band grid points. At each step of the algorithm, one needs to extract the narrow band grid point with the smallest value. A convenient data structure to store the different tentative values is the binary heap (see Sethian ([32]), Bærentzen ([2])). It can be seen as a particular case of a binary tree, which is either a balanced tree (all the leaves are at the same depth) or at least with all leaves at the depth l or $l-1$ (all the children on the last level occupying the leftmost spots) and such that one node is always greater than the nodes of its parents. Therefore, to access the element with the smallest tentative value, one just needs to extract the root of the tree.

Adding an element to the binary heap is a $\mathcal{O}(\log n)$ procedure with n the number of grid points in the narrow band. (This is linked to the height of the binary heap). The grid point to be inserted is first added to the lower level of the binary heap. If the heap property is violated, that is, if one of its parents is greater, one just swaps these two elements and repeat the process if needed. In the same way, the deleting operation is a $\mathcal{O}(\log n)$ procedure. From the above, we can infer that the algorithm complexity is thus in $\mathcal{O}(N \log N)$ where N is the number of grid points. Many extensions of this method have been proposed to reduce the overall complexity of the algorithm such as fast sweeping methods [34], [39]. We also would like to point out the recent work by Yatziv *et al.* [38]. They have developed an implementation of the FMM that reduces the complexity to $\mathcal{O}(N)$ using a data structure named untidy priority queue. The main idea is that errors can be allowed in the computation of the tentative values when establishing the priority of each grid point. In practice, it means that a quantization of the priorities is introduced which implies some possible ordering errors. But as stressed by Yatziv *et al.*, the misordering generated by this technique creates few and small errors and thereby does not affect the overall accuracy of the process.

If we now go back to the segmentation problem, as previously mentioned, Malladi and Sethian (see [22] and [32]) have developed an algorithm that both uses fast marching and level set methods. They consider the following flow, the evolving contour being embedded in the level set function Φ:

$$\frac{\partial \Phi}{\partial t} + g_I(1-\epsilon\kappa)|\nabla\Phi| - \beta < \nabla P, \nabla\Phi >= 0. \tag{7}$$

Three components appear in this equation:

- An inflating force $g_I|\nabla\Phi|$ which is obtained from the image gradient. The speed is defined by $g_I(x) = \dfrac{1}{1+|\nabla(G_\sigma * I)(x)|^p}$, $p \geq 1$, ($G_\sigma * I$, smoother version of the image I, is the convolution of the image I with a Gaussian filter) so that the front stops propagating when localized on an edge.

- a curvature-depending term $-g_I \epsilon \kappa |\nabla \Phi|$ (see [23]).
- A force that guarantees a good attraction of the evolving contour towards the desired boundary with stabilizing effects (see [8]): $-\beta < \nabla P, \nabla \Phi >$.

The algorithm is divided into two steps: Malladi and Sethian [22] propose to first solve the flow $\frac{\partial \Phi}{\partial t} + g_I |\nabla \Phi| = 0$ using the FMM, that is, to solve the problem $|\nabla T| = \frac{1}{g_I}$. It is done until the contour has almost reached the boundary. It makes sense insofar as the time becomes very large when the desired boundary is nearly reached.

The last stage consists in solving the problem (7) taking as initial condition the one provided by the obtained contour in the previous step.

2.2 Introducing the GFMM

In this section, we will recall the Generalized Fast Marching Method developed by Carlini *et al.* in [6] (see also [5]) for tracking hypersurfaces moving according to their normal with a sign-changing velocity. More precisely, the initial front is the boundary of an open set Ω_0, which is represented by a characteristic function $1_{\Omega_0} - 1_{\Omega_0^c}$, equal to 1 on Ω_0 and -1 on its complementary set. Mathematically, we are interested in the discontinuous viscosity solution $\theta(x,t)$ of the following equation

$$\begin{cases} \theta_t(x,t) = c\,(x,t)\,|\nabla \theta(x,t)| \quad \text{on} \quad \mathbb{R}^N \times (0,T) \\ \theta(\cdot, 0) = 1_{\Omega_0} - 1_{\Omega_0^c} \end{cases} \tag{8}$$

where $c(x,t)$ is a given velocity with a changing sign. The only assumptions on c are that it is Lipschitz continuous in space and time (this allows in particular to have a comparison principle for (8), see Barles [3]). Here the support of the discontinuities of the function θ localizes the front we are interested in.

The first algorithm developed in [6] has been improved by the first author in [14] where he has proposed a monotone scheme (the one proposed in [6] was not completely monotone). In [7], Carlini *et al.* have also proposed a simpler scheme which is monotone if the velocity does not change sign in time. This is the algorithm we will describe and that we will use for our problem.

The main idea of this generalized algorithm is to make two different Fast Marching: one for which the velocity is positive and another one for which the velocity is negative. To do so, we need to introduce a band where the numerical speed is zero, in order to separate the sets where the velocity is respectively positive and negative (see Definition 2 in the algorithm).

At a discrete time t_n, the front will be represented by the boundary of an open set Ω_n and by a phase field θ^n defined equal to 1 on Ω_n and -1 on its complementary set. The field θ^n can be seen as a discretization of the solution θ of equation (8). In view of equation (8), where the velocity is positive, this will be the region $\{\theta^n = 1\}$ that will grow whereas where the velocity is negative, this will be the region $\{\theta^n = -1\}$ (see Figure 1).

As in the classical FMM, we define the Narrow Band NB^n as the points that can be reached by the front. More precisely, the Narrow Band is defined by

$$NB^n = \{J \in \mathbb{Z}^N,\ \exists I \in V(J),\ \theta^n_J = -\theta^n_I \text{ and } \theta^n_J c^n_J < 0\},$$

where θ^n_J denotes the value of the field θ^n taken at point J at time t_n, c^n_J the value of the velocity at point J at time t_n and $V(J)$ denotes the neighborhood of the point J (see Definition 1).

We also define the set of useful points

$$\mathcal{U}^n = \{I \in \mathbb{Z}^N,\ \exists J \in V(I) \cap NB^n,\ \theta^n_J = -\theta^n_I\}.$$

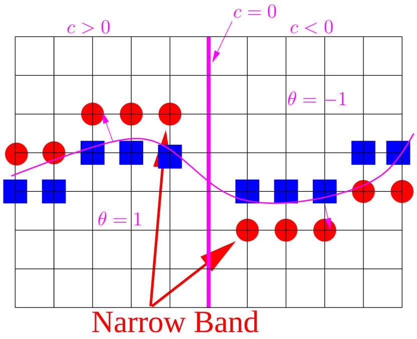

Figure 1: Schematic representation of the GFMM. The red balls represent the points of the Narrow Band, while the blue squares represent the useful points.

In fact, this set consists more or less of the points that the front has already crossed and for which we know the crossing time. This corresponds to the "frozen" region of the classical FMM. These are the values of the points that we will use in the computation of the candidate time.

With this new definition, the algorithm is now essentially the same as the classical FMM: for all point J of the Narrow Band NB^n, we compute a candidate time $(\tilde{u}_J^n)$ by solving (6) using only the values of the useful points. We denote by $\tilde{t}_{n+1}$ the value of the minimum of the candidate times and we accept ($i.e$ we change θ) at time t_{n+1} all the points that realize the minimum. In most cases, we have $t_{n+1}=\tilde{t}_{n+1}$, except if $\tilde{t}_{n+1}<t_n$ or $\tilde{t}_{n+1}\gg t_n$ (see Step 6 of the algorithm).

In [6, 14], the authors have proved the convergence of the algorithm in the framework of discontinuous viscosity solutions. To explain their result, we need to introduce some notations.

Let us denote by $\{t_{n_k}, k\in\mathbb{N}\}$ a strictly increasing subsequence of the physical sequence of time $\{t_n, n\in\mathbb{N}\}$ given by the algorithm (and which is only non-decreasing), such that

$$t_{n_k}=t_{n_k+1}=\ldots=t_{n_{k+1}-1}<t_{n_{k+1}}.$$

We denote by S_I^k the square cell $S_I^k=[x_I,x_I+\Delta x[\times[t_{n_k},t_{n_{k+1}}[$ with

$$[x_I,x_I+\Delta x[=\Pi_{\alpha=1}^N[x_{i_\alpha},x_{i_\alpha}+\Delta x[.$$

We denote by θ^ϵ an extension of θ^n on the whole space:

$$\theta^\epsilon(x,t)=\theta_I^{n_{k+1}-1} \text{ if } (x,t)\in S_I^k. \tag{9}$$

Finally, we define the half-relaxed limits of θ^ϵ by

$$\overline{\theta}^0(x,t)=\limsup_{\epsilon\to0,y\to x,s\to t}\theta^\epsilon(y,s),\quad \underline{\theta}^0(x,t)=\liminf_{\epsilon\to0,y\to x,s\to t}\theta^\epsilon(y,s). \tag{10}$$

The convergence result for this scheme is the following one: $\overline{\theta}^0$ is a sub-solution of (8) and $\underline{\theta}^0$ is a super-solution. In particular, if the solution θ of (8) is unique and if the Lebesgue measure of $\partial\{\theta=1\}$ is zero, then we have the convergence in the L^1 sense (see [7] for an exact setting). The key fact to obtain the convergence is that the algorithm is monotone (in [6], it is monotone on the region where the velocity is non zero, which is sufficient). We also refer to [7] for the convergence of the scheme for a non-local motion in which the velocity depends on all the front.

3 Image Segmentation using the GFMM

With the mathematical background previously outlined, the question that remains to be dealt with is how to build a normal velocity suitable for the segmentation problem which should allow to simultaneously have parts moving inward and others moving outward. As previously mentioned, the way the velocity is designed is related to the work by Chan-Vese. We think here that a region-based approach, as they do, should be preferred insofar as classical methods which use local detection criteria (geodesic active contours, etc...) require the computation of geometrical properties of the front (normal vectors, ...). This, without regularization constraint, may lead to artefacts or inappropriate behaviour of the front.
In subsection 3.1, we recall the main ideas of the Chan-Vese model and give the modelling of our approch. Its corresponding algorithm is detailed in subsection 3.2.

3.1 Modelling

Let Ω be a bounded open subset of $\mathbb{R}^n$, $\partial\Omega$ its boundary and let I be a given bounded image function defined by $I : \bar{\Omega} \to \mathbb{R}$. For the purpose of illustration, we consider $n = 2$ although the model can be easily extended to the case $n = 3$. As mentioned by Aubert and Kornprobst in [1], the Chan-Vese model aims at finding the best approximation of the image as a function taking only two values. It is a particular case of the minimal partition problem and it is based on the piecewise-constant Mumford-Shah functional (see [25]). Contrary to classical methods which involve a stopping criterion based on the image gradient (that is, a local criterion), the Chan-Vese model is region-based and does not require the computation of an edge-detector function. The evolving contour $\mathcal{C}$ is embedded in a higher-dimensional lipschitz continuous function Φ. More precisely, it is implicitly represented as the zero level line of the function Φ such that:

$$\begin{cases} \mathcal{C} = \partial w = \big\{(x,y) \in \Omega \mid \Phi(x,y) = 0\big\}, \\ \mathit{inside}(\mathcal{C}) = w = \big\{(x,y) \in \Omega \mid \Phi(x,y) > 0\big\}, \\ \mathit{outside}(\mathcal{C}) = \Omega - \bar{w} = \big\{(x,y) \in \Omega \mid \Phi(x,y) < 0\big\}. \end{cases}$$

An energy minimization problem is then introduced including both a fidelity term that ensures that the curve will match the edge(s) of the object(s) to be detected and a regularizing term based on the length of the curve (in order to penalize oscillations). Chan and Vese consider the following problem:

$$\inf_{c_1,c_2,\mathcal{C}} F(c_1, c_2, \mathcal{C}),$$

with

$$F(c_1, c_2, \mathcal{C}) = \mu \int_\Omega \delta(\Phi)|\nabla\Phi| dxdy + \lambda_1 \int_\Omega (I(x,y) - c_1)^2 H(\Phi(x,y)) dxdy +$$

$$\lambda_2 \int_\Omega (I(x,y) - c_2)^2 (1 - H(\Phi(x,y))) dxdy,$$

c_1 and c_2 being two unknown constants depending on $\mathcal{C}$, H and δ being respectively the Heaviside function and Dirac delta function and $\mu, \lambda_1, \lambda_2 > 0$ fixed tuning parameters. Note that $\int_\Omega \delta(\Phi)|\nabla\Phi| dxdy$ is the level set representation of the length of the curve $\mathcal{C}$. The constants c_1 and c_2 are expressed keeping Φ fixed and minimizing F with respect to c_1 and c_2, it yields:

$$c_1 = \frac{\displaystyle\int_\Omega I(x,y)H(\Phi(x,y))dxdy}{\displaystyle\int_\Omega H(\Phi(x,y))dxdy},$$

$$c_2 = \frac{\displaystyle\int_\Omega I(x,y)(1-H(\Phi(x,y)))dxdy}{\displaystyle\int_\Omega (1-H(\Phi(x,y)))dxdy}$$

and the Euler-Lagrange equation for the unknown Φ is computed using regularized versions of H and δ denoted by H_ϵ and δ_ϵ. A gradient descent method is applied and the evolution equation of Φ is eventually defined by:

$$\frac{\partial \Phi}{\partial t} = \delta_\epsilon(\Phi)\Big(\mu \operatorname{div}(\frac{\nabla\Phi}{|\nabla\Phi|}) - \lambda_1\big(I-c_1\big)^2 + \lambda_2\big(I-c_2\big)^2\Big).$$

The segmentation criterion in this modelling can be easily understood. Let us consider the following synthetic image (Figure 2) and the evolving contour in red. Let us also consider the green point of the curve. It is obvious that in this case $(I-c_2)^2 \simeq 0$ and $(I-c_1)^2$ is bigger. As a consequence, this part of the curve will move according to the inward normal vector to the curve. On the contrary, the yellow point is such that $(I-c_2)^2$ is bigger thus this part of the curve will move outward.

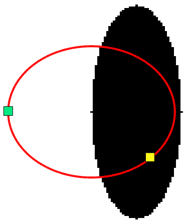

Figure 2: Illustration of the segmentation criterion in the modelling.

Our velocity is based on the fitting component of the Chan-Vese model and is defined as follows (we use the notation x as space variable): at time t_n, with the previous notations,

$$c(x,t_n) = (I(x)-c_2)^2 - (I(x)-c_1)^2,$$

with $c_1 = \dfrac{\displaystyle\int_\Omega I(x)1_{\Omega_n}dx}{\displaystyle\int_\Omega 1_{\Omega_n}dx}$ and $c_2 = \dfrac{\displaystyle\int_\Omega I(x)1_{\Omega_n^c}dx}{\displaystyle\int_\Omega 1_{\Omega_n^c}dx}$, c depending on time through c_1 and c_2 that are both time-varying. (A weighted combination of $(I(x)-c_2)^2$ and $(I(x)-c_1)^2$ could be used).

3.2 Algorithm

In this subsection, we want to propose a numerical scheme of Fast Marching Method type (see Sethian [32, 31]) to compute the evolution of a front driven by its normal velocity $c(x,t)$ given by

$$c(x,t) = (I(x) - c_2)^2 - (I(x) - c_1)^2, \tag{11}$$

with c_1 and c_2 defined above. This velocity is easily computed at each time t_n with the characteristic functions of the sets Ω_n and Ω_n^c. In particular, we do not have a constant sign for c during the evolution. The goal is to use the generalization of the Fast Marching Method proposed by Carlini *et al.* in [6, 7] for the eikonal equation with changing-sign velocities. More precisely, the initial front is the boundary of an open set Ω_0, which is represented by a characteristic function $1_{\Omega_0} - 1_{\Omega_0^c}$, defined equal to 1 on Ω_0 and -1 on its complementary set. Mathematically, we are interested in the discontinuous viscosity solution $\theta(x,t)$ of the following equation

$$\begin{cases} \theta_t(x,t) = c\,(x,t)\,|\nabla\theta(x,t)| & \text{on} \quad \mathbb{R}^N \times (0,T) \\ \theta(\cdot,0) = 1_{\Omega_0} - 1_{\Omega_0^c}. \end{cases} \tag{12}$$

Here the support of the discontinuities of the function θ localizes the front we are interested in.

3.2.1 Notations and preliminary definitions

Let N be the dimension. Let us consider a lattice $Q \equiv \{x_I = (x_{i_1}, .., x_{i_N}) = (i_1\Delta x, .., i_N\Delta x),$
$I = (i_1, .., i_N) \in \mathbb{Z}^N\}$ with space step $\Delta x > 0$. We will also use a time step $\Delta t > 0$.

The following definitions will be useful in the sequel.

Definition 1. *The* **neighborhood of the node** $I \in \mathbb{Z}^N$ *is the set* $V(I) \equiv \{J \in \mathbb{Z}^N : |J - I| \leq 1\}$.

As mentioned above, we have to somehow regularize the velocity in space by introducing a band of zero to separate the region where the velocity is positive from the one where it is negative. This prevents the case where two neighboring points are respectively in NB^+ and NB^- from occurring.

Definition 2. *We define the speed* c_I^n *by:*

$$c_I^n = c(x_I, t_n)$$

We then define the numerical speed by:

$$\widehat{c}_I^n \equiv \begin{cases} 0 & \textit{if there exists } J \in V(I) \textit{ such that } (c_I^n c_J^n < 0 \textit{ and } |c_I^n| \leq |c_J^n|), \\ c_I^n & \textit{otherwise.} \end{cases}$$

Definition 3. *The* **numerical boundary** ∂E *of a set* $E \subset \mathbb{Z}^N$ *is*

$$\partial E \equiv V(E) \backslash E$$

with

$$V(E) = \left\{ J \in \mathbb{Z}^N, \quad \exists I \in E, \quad J \in V(I) \right\}$$

Definition 4. *Given a field* θ^n *with values* $+1$ *and* -1*, we define the two phases*

$$\Theta_\pm^n \equiv \{I : \theta_I^n = \pm 1\},$$

and the fronts

$$F_\pm^n \equiv \partial\Theta_\mp^n, \quad F^n \equiv F_+^n \cup F_-^n.$$

In the description of the algorithm, we will use the notations:

$$\pm g \geq 0 \text{ for } I \in F_\pm$$

means

$$+g \geq 0 \text{ for } I \in F_+ \text{ and } -g \geq 0 \text{ for } I \in F_-.$$

Moreover,

$$\min_\pm\{0, g_\pm\} \equiv \min\{0, g_+, g_-\} \text{ and } \max_\pm\{0, g_\pm\} \equiv \max\{0, g_+, g_-\}$$

and

$$I^{k,\pm} = (i_1, .., i_{k-1}, i_k \pm 1, i_{k+1}, .., i_N). \tag{13}$$

To present the algorithm, we need to define, for each point $K \in \mathbb{Z}^N$, the set of points I that are useful to compute the candidate time $\tilde{u}^n_K$, namely, the set:

$$\mathcal{U}^n(K) = \{I \in V(K),\ \theta^n_I = -\theta^n_K \text{ and } \theta^n_I \hat{c}^n_K > 0\}, \quad \mathcal{U}^n = \cup_K \mathcal{U}^n(K).$$

We also define the Narrow Band:

$$NB^n = \{I, \exists\ J\ s.t.\ J \in \mathcal{U}^n(I)\}, \quad NB^n_\pm = NB^n \cap \{I, \theta^n_I = \pm 1\}.$$

We observe that the Narrow Bands $NB^n_\pm$ are related to the previous definitions of front sets by:

$$NB^n_+ = F^n_+ \cap \{I, \hat{c}^n_I < 0\}, \quad NB^n_- = F^n_- \cap \{I, \hat{c}^n_I > 0\}.$$

3.2.2 The algorithm step-by-step

We now describe our algorithm. As one can see, to correctly track the evolution, we need to introduce a discrete function $u^n_I \in \mathbb{R}^+$ defined for $I \in \mathcal{U}^n$ to represent the approximated physical time for the front propagation at nodes $I = (i_1, .., i_N)$ of the fronts and at the n-th iteration of the algorithm.

Initialization

1. Set $n = 1$
2. *Initialize the field* θ^0 as
   $\theta^0_I = \begin{cases} 1 & \text{for } x_I \in \Omega_0 \\ -1 & \text{elsewhere} \end{cases}$
3. *Initialize the time on* F^0
   $u^0_I = 0$ for all $I \in \mathcal{U}^0$

Main cycle

4. *Compute* $\tilde{u}^{n-1}$ *on* NB^{n-1} *as follows:*
   Let $I \in NB^{n-1}$, then we define $\widehat{u}^{n-1}_{J\to I}$ as

$$\widehat{u}^{n-1}_{J\to I} = \begin{cases} u^{n-1}_J & \text{if } J \in \mathcal{U}^{n-1}(I) \\ \infty & \text{otherwise.} \end{cases}$$

   We compute $\tilde{u}^{n-1}_I$ as the solution of the following second order equation:

$$\sum_{k=1}^N \left(\max_\pm \left(0, \tilde{u}^{n-1}_I - \widehat{u}^{n-1}_{I^{k,\pm}\to I}\right)\right)^2 = \frac{(\Delta x)^2}{|\widehat{c}^{n-1}_I|^2}, \tag{14}$$

   where $I^{k,\pm}$ is defined in (13) (see [32] for numerical details).

5. $\tilde{t}_n = \min\left\{\tilde{u}_I^{n-1},\ I \in NB^{n-1}\right\}$.

6. $t_n = \max(t_{n-1}, \tilde{t}_n)$

7. if $t_n = t_{n-1} + \Delta t$ and $t_n < \tilde{t}_n$ go to 4 with $n := n+1$, $\theta^n = \theta^{n-1}$ and $u^n = u^{n-1}$.

8. *Initialize the new accepted point*
$NA_\pm^n = \{I \in F_\pm^{n-1},\ \tilde{u}_I^{n-1} = \tilde{t}_n\}$, $NA^n = NA_+^n \cup NA_-^n$

9. *Reinitialize* θ^n
$$\theta_I^n = \begin{cases} \mp\theta_I^{n-1} & \text{if } I \in NA_\pm^n \\ \theta_I^{n-1} & \text{otherwise} \end{cases}$$

10. *Reinitialize* u^n *on* $\mathcal{U}^n$

    (a) If $I \notin \mathcal{U}^{n-1}$, then $u_I^n = t_n$,

    (b) If $I \in \mathcal{U}^{n-1}$, then $u_I^n = u_I^{n-1}$.

11. Set $n := n+1$ and go to 4

Remark 1.

- In *step 4*, we use the regularized velocity $\hat{c}$ and not c in order to stabilize the front. Indeed, if we do not do that, this typically leads to a duplication of the front.

- The sequence of time $(\tilde{t}_n)$ is not necessary non-decreasing in time. Indeed, if the velocity increases in time, we can have $\tilde{t}_n < \tilde{t}_{n-1}$ and so we have to do something to have an increasing sequence of time. In fact, *step 8* guarantees that the physical time t_n does not decrease.

- In *step 10*, for the reinitialization of u_I^n, we change its value only if a point of the neighborhood of the point I has been accepted. Moreover when u_I^n is updated, we use the physical time t_n and not $\tilde{t}_n$.

3.2.3 Implementation and computational complexity

In this subsection, we give some rough asymptotic bounds on the computational complexity of our GFMM algorithm for image segmentation and we give some remarks on the implementation.

From a complexity point of view, it is not interesting to recompute the velocity at each iteration. Indeed, in the implementation, we will fix a time step ΔT and we will recompute the velocity only at each $k\Delta T$ for $k \in \mathbb{Z}$.

To fix the ideas, we work on a (spatial) grid box of width $\Delta x \simeq \dfrac{1}{M^{\frac{1}{2}}}$, and therefore with a total number of grid points equal to M. We assume that the velocity is normalized $|c| \simeq 1$ and then the time T necessary for the front to pass once on the whole grid box is roughly $T \simeq 1$. The typical size (as a number of grid points) of the front is $M^{\frac{1}{2}}$ in the box.

Since the time T necessary for the front to pass once on the whole grid box is roughly $T \simeq 1$, we deduce that the number of time intervals $[k\Delta T, (k+1)\Delta T)$ is $\frac{T}{\Delta T} = \frac{1}{\Delta T}$. On each interval $[k\Delta T, (k+1)\Delta T)$, we will use a binary heap (as in the classical FMM). Because the velocity is independent of time, we only need to recompute the values of the time at the points I whose neighbors have been accepted (*i.e.*, $I \in V(NA^n)$). This implies that our GFMM is equivalent to two FMM algorithms and so we recover the complexity in $M_k \log M_k$ where M_k is the number of points crossed by the front during this interval of time (with $\sum_k M_k = M$). This gives a complexity
$$\sum_{k=1}^{\frac{1}{\Delta T}} M_k \log M_k.$$

Moreover, at each $k\Delta T$, we have to recompute the velocity (only for the points of the Narrow Band), the candidate times and the binary heap (since the velocity changes). The complexity for these operations is $O(M^{\frac{1}{2}} \log M)$. Therefore, the total complexity is:

$$\sum_{k=1}^{\frac{1}{\Delta T}} M_k \log M_k + \sum_{k=1}^{\frac{1}{\Delta T}} M^{\frac{1}{2}} \log M \leq M \log M + \frac{1}{\Delta T} M^{\frac{1}{2}} \log M.$$

If $\Delta T \geq \frac{1}{M^{\frac{1}{2}}}$, we then get a complexity in $O(M \log M)$ (as in the classical case).

4 Numerical results

We conclude this paper by presenting several numerical results. These experiments have been performed on a 2.21GHz Athlon with 1.00 GB of RAM. As previously stressed, the computation cost of this method is comparable to the one of the classical Fast Marching Method. For the presented experimentations, it is of the order of the second.

We first give a basic example that emphasizes the robustness of the GFMM with regards to initialization. It requires a non-constant sign of the normal velocity for the last two tests. Three kinds of initialization are given:

- A first one (figure 3) which shows that the GFMM handles splits.
- A second one (figure 4) which stresses that the initial set Ω_0 can be made of several shapes without additional procedure. The classical FMM would fail to detect both shapes. The shape below would just collapse and disappear.
- A last one (figure 5), showing that the initial condition can be anywhere in the image.

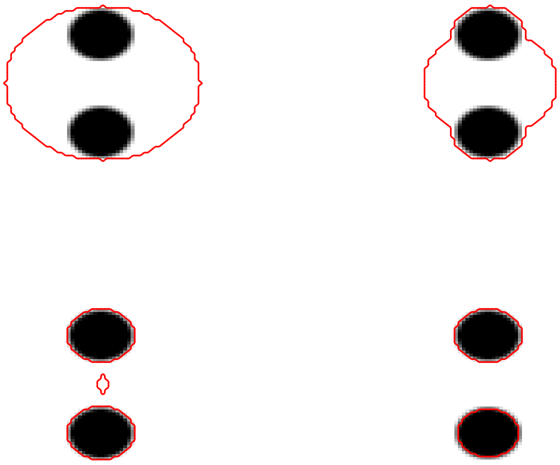

Figure 3: Segmentation of two balls. Different steps of the process. The GFMM handles topological changes

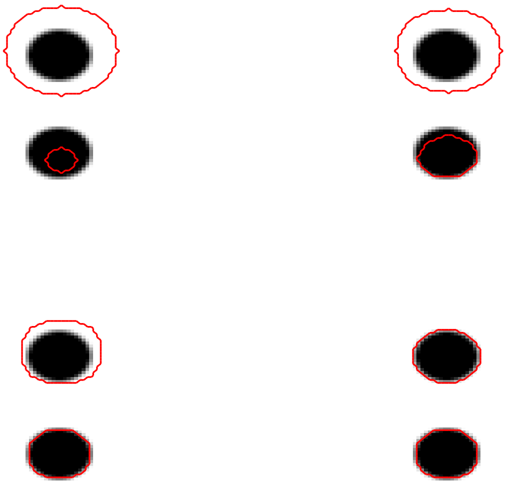

Figure 4: Segmentation of two balls. Different steps of the process. The initial set Ω_0 can be made of several shapes.

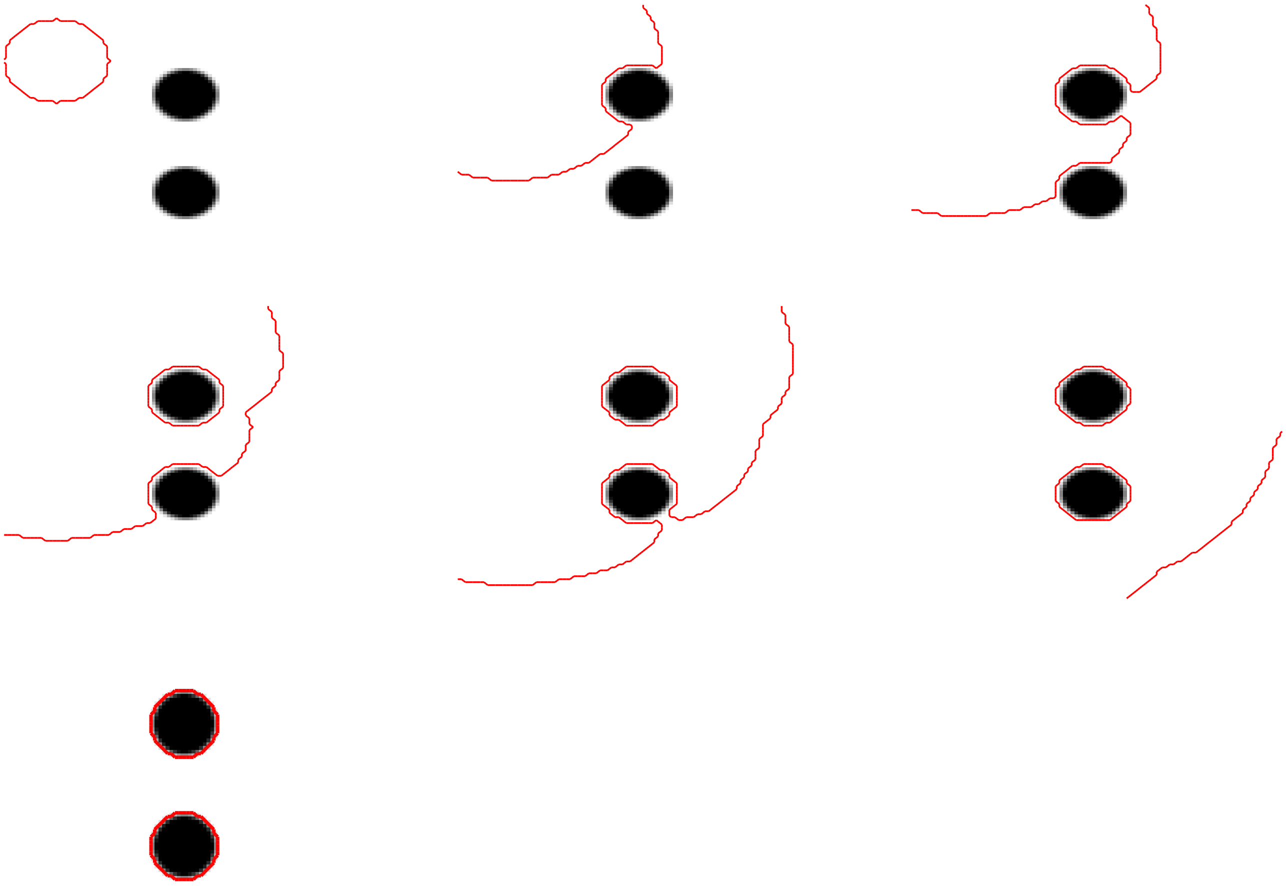

Figure 5: Segmentation of two balls. Different steps of the process. The initial set Ω_0 can be anywhere in the image.

The following examples (figures 6 and 7) show that the initial condition can intersect the object to be detected and that the method can handle intricate shapes. The proposed method qualitatively performs in a way similar to the Chan-Vese model in this case. This, again, would not be possible with the classical FMM.

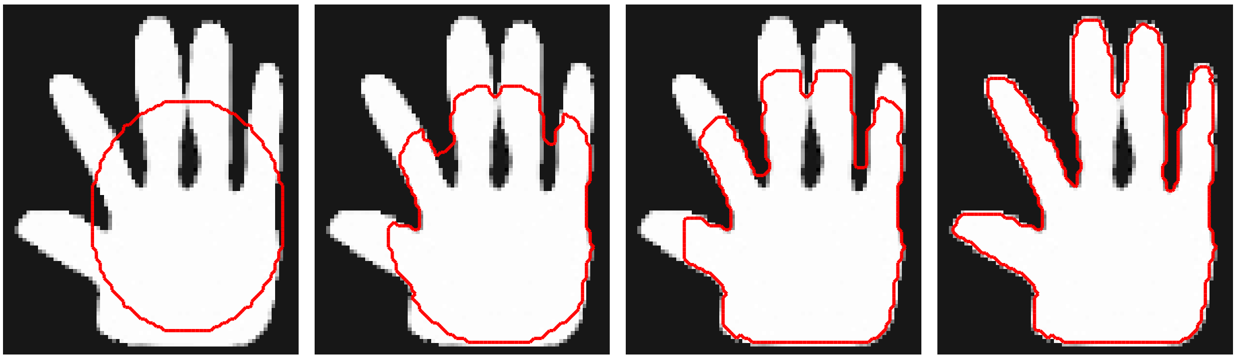

Figure 6: Segmentation of a hand. Different steps of the process. The initial front intersects the object.

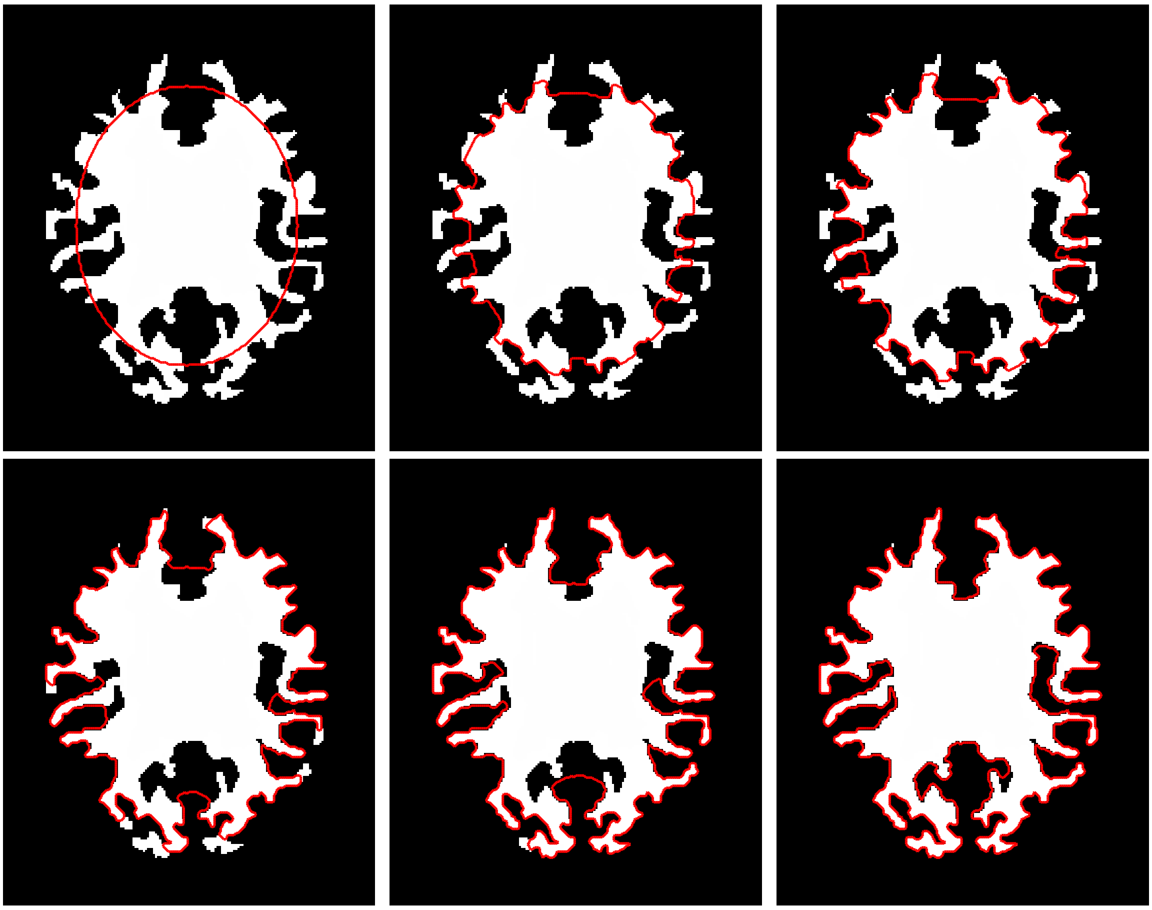

Figure 7: Segmentation of a brain. Different steps of the process. The initial front intersects the object.

Of course, the proposed method cannot automatically detect interior contours but this drawback is overcome by the flexibility in the initialization step induced by the modelling. We illustrate this remark with the following examples. In figure 8 (Courtesy of the Laboratory of Neuro Imaging, UCLA, School of Medicine), the initial set Ω_0 is a deformed tore.

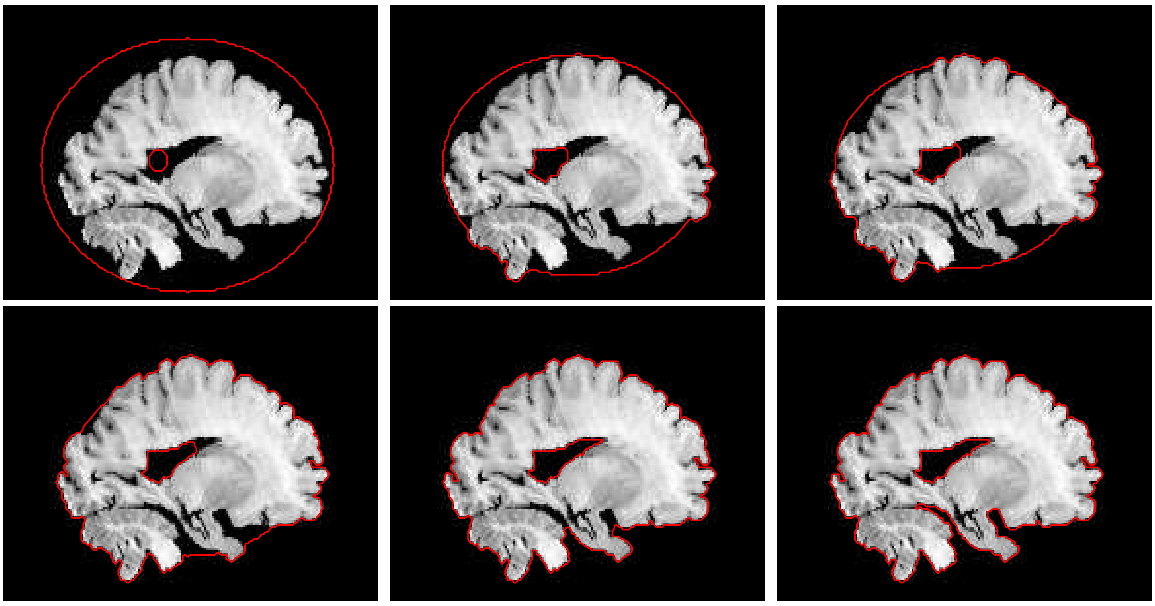

Figure 8: Different steps of the process. The initial set Ω_0 is a deformed tore.

In figure 9, the initial shape contains three holes. This stresses the ability of the GFMM to handle any kind of initialization.

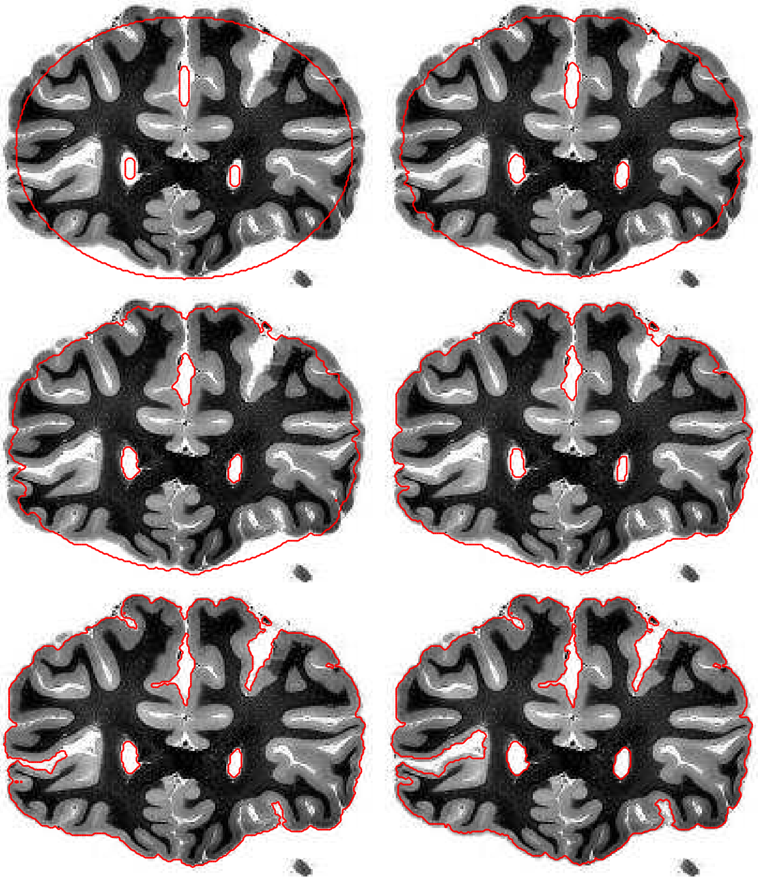

Figure 9: Different steps of the process. The initial set Ω_0 is a shape with three holes.

5 Conclusion

We have proposed a new approach based on the FMM and on the Chan-Vese model for segmentation. This new model keeps the advantages (fast, efficient, robust) of the well-known FMM. The GFMM also offers the possibility of having a velocity with a non constant sign. Theoretical results, modelling and applications have been given, and they demonstrated the potentiality of this method. Many other applications (in combustion, fluid simulations, interface evolution, 3D sculpting in CAGD...) are in progress.

We now expect to extend this new approach to more complicated images with textured components for which the region-based velocity alone would be inaccurate, or simply noisy images (this last point requiring a regularization term to penalize oscillations). We believe that, in this prospect, the expression of the velocity should include both a non-local term and a local term. This local term could use geometrical properties of the propagating front such as normal vectors or mean curvature (for regularizing properties). In [24], Merriman, Bence and Osher address the issue of evolving the boundary of a region (with characteristic function χ) with a normal speed equal to the mean curvature. The procedure is simple since it consists in convolving the characteristic function with a Gaussian kernel, which can be done by solving the heat equation.

An area to think about would then be to propose an alternating framework: our proposed GFMM algorithm with some periodic regularizing steps using the approach proposed by Merriman *et al.*. This is a work in progress.

Aknowledgement. The first author was supported by the ACI JC 1025 and the ANR MICA (2006-2009).

References

[1] G. Aubert and P. Kornprobst. *Mathematical Problems in Image processing: Partial Differential Equations and the calculus of Variations.* Springer Verlag, 2002.

[2] J. A. Baerentzen. On the Implementation of Fast Marching Methods for 3d lattices. *Technical Report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU*, 2001.

[3] G. Barles. *Solutions de viscosité des équations de Hamilton-Jacobi*, volume 17 of *Mathématiques & Applications (Berlin) [Mathematics & Applications].* Springer-Verlag, Paris, 1994.

[4] X. Bresson, S. Esedoglu, P. Vandergheynst, J. Thiran, and S. Osher. Fast global minimization of the active contour/snake models. *J. Math. Imaging. and Vision*, 28(2):151–167, 2007.

[5] E. Carlini, E. Cristiani, and N. Forcadel. A non-monotone fast marching scheme for a hamilton-jacobi equation modeling dislocation dynamics. *ENUMATH 2005, Santiago de Compostela (Spain)*, 2007.

[6] E. Carlini, M. Falcone, N. Forcadel, and R. Monneau. Convergence of a Generalized Fast Marching Method for a non-convex eikonal equation. *submitted*, 2007.

[7] E. Carlini, N. Forcadel, and R. Monneau. Generalized fast marching method for dislocation dynamics. *work in progress*, 2007.

[8] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. *International Journal of computer Vision*, 22(1):61–87, 1993.

[9] T. Chan and L. Vese. Active Contours Without Edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[10] T.F. Chan, S. Esedoglu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.*, 66(5):1632–1648, 2006.

[11] J. T. Chung and L. Vese. Image Segmentation Using a Multilayer Level-Set Approach. *EMMCVPR 2005: 439-455*, 2005.

[12] L.D. Cohen. On active contour models and balloons. *CVGIP:IU*, 52(2), 1991.

[13] C.L. Epstein and M. Gage. The curve shortening flow. *Wave motion: Theory , Modelling and Computation*, A. Chorin and A. Majda Eds, 1997.

[14] N. Forcadel. Comparison principle for the generalized fast marching method. work in progress.

[15] C. Gout and C. Le Guyader. Segmentation of complex geophysical structures with well data. *Comput. Geosci.*, 10(4):361–372, 2006.

[16] C. Gout, C. Le Guyader, and L. Vese. Segmentation under geometrical conditions using geodesic active contours and interpolation using level set methods. *Numerical Algoirthms*, 39(1,3):155–173, 2005.

[17] A. Herbulot, S. Jehan-Besson, M. Barlaud, and G. Aubert. Shape gradient for image segmentation using information theory. *IEEE ICASSP*, pages 21–24, 2004.

[18] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models in Proceeding. *First international conference on computer vision, London, England, IEEE, Piscataway NJ*, pages 259–268, 1987.

[19] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Gradient flows and geometric active contour models. *Proceedings, Fifth International Conference on Computer Vision*, pages 810–815, 1995.

[20] J. Kim, J.W. Fisher III, A. Jr Yezzi, M. Cetin, and A.S. Willsky. Nonparametric methods for image segmentation using information theory and curve evolution. *IEEE ICIP*, pages 797–800, 2002.

[21] C. Le Guyader, D. Apprato, and Gout C. The level set methods and image segmentation under interpolation conditions. *Numerical Algorithms*, 39((1-3)):221–235, 2005.

[22] R. Malladi and J.A. Sethian. An $\mathcal{O}(n \log n)$ Algorithm for Shape Modeling. *Proceedings of the National Academy of Sciences*, 93:9389–9392, 1996.

[23] R. Malladi, J.A. Sethian, and B. Vemuri. Evolutionary Fronts for Topology-independent Shape Modeling and Recovery. *Proceedings of Third European Conference on Computer Vision*, pages 3–13, 1994.

[24] B. Merriman, J. Bence, and S. Osher. Diffusion Generated Motion by Mean Curvature. *In J.E. Taylor, editor, Proceedings of the Computational Crystal Growers Workshop, American Mathematical Society, Providence, Rhode Island*, pages 73–83, 1992.

[25] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.*, 42:577–685, 1989.

[26] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces.* Springer Verlag, 2003.

[27] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision, and Graphics.* Springer Verlag, 2003.

[28] S. Osher and J.A. Sethian. Fronts propagation with curvature dependent speed : Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79, 1988.

[29] V. Rouy and F. Tourin. A Viscosity Solutions Approach to Shape-From-Shading. *SIAM J. Num. Anal.*, 29(3):867–884, 1992.

[30] J.A. Sethian. A review of recent numerical algorithms for hypersurfaces moving with curvature dependent flows. *J.Differential Geometry*, 31:131–161, 1989.

[31] J.A. Sethian. A Fast Marching Level Set Method for Monotonically Advancing fronts. *submitted for publication: Proceedings of the National Academy of Sciences*, 1995.

[32] J.A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Material Science.* Cambridge University Press, Londres, 1999.

[33] J.A. Sethian. Evolution, Implementation and Application of Level Set and Fast Marching Methods for Advancing Fronts. *Journal of Computational Physics*, 169(2):503–555, 2001.

[34] Y.R. Tsai, L. T. Cheng, S. Osher, and H. K. Zhao. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *SIAM Journal on Numerical Analysis*, 41(2):673–694, 2002.

[35] N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Tran. Automatic. Control*, 40:1528–1538, 1995.

[36] L. Vese and T. Chan. A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[37] C. Xu and J.L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, 7(3):359–369, 1998.

[38] L. Yatziv, A. Bartesaghi, and G. Sapiro. $\mathcal{O}(n)$ Implementation of the Fast Marching Algorithm. *Journal of Computational Physics*, 212(2):393–399, 2006.

[39] H.-K. Zhao. Fast sweeping method for Eikonal equations. *Mathematics of Computation*, 74:603–627, 2004.

[40] H.-K. Zhao, T. Chan, B. Merriman, and S Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, 127:179–195, 1996.