A Recursive Anisotropic Fast Marching Approach to Reaction Diffusion Equation: Application to Tumor Growth Modeling

Ender Konukoglu**, Maxime Sermesant, Olivier Clatz, Jean-Marc Peyrat, Hervé Delingette, and Nicholas Ayache

Asclepios Research Project, INRIA Sophia Antipolis, France,
ender.konukoglu@sophia.inria.fr

Abstract. Bridging the gap between clinical applications and mathematical models is one of the new challenges of medical image analysis. In this paper, we propose an efficient and accurate algorithm to solve anisotropic Eikonal equations, in order to link biological models using reaction-diffusion equations to clinical observations, such as medical images. The example application we use to demonstrate our methodology is tumor growth modeling. We simulate the motion of the tumor front visible in images and give preliminary results by solving the derived anisotropic Eikonal equation with the recursive fast marching algorithm.

1 Introduction

One of the main gaps between mathematical models explaining biological phenomena and medical applications is due to the inconsistency between the number of observations available clinically and needed mathematically. While building more realistic models is very important to improve our insight on the general phenomena, creating reduced models is essential in using mathematical models in specific clinical situations.

Reaction-diffusion equations like:

$$u_t = \nabla \cdot (D \nabla u) + f(u) \tag{1}$$

arise in many different biological models, where one describes the change of a density u in time with an anisotropic diffusion characterized by the tensor D and a reaction term $f(u)$. We can give examples to such situations like tumor growth, electrophysiology and wound healing. However, available observations are often sparse and incomplete and these models are often computationally costly. Thus, making the adjustment of complete models to a specific case is difficult. This is why reduced models are of great interest. In the case of Equation 1, one can

** This work has been partly supported by the European Health-e-Child project (IST-2004-027749) and by the CompuTumor project (http://www-sop.inria.fr/asclepios/projects/boston/).

approximate the motion of a single iso-contour of u in terms of its arrival times T using the anisotropic Eikonal equation

$$F\sqrt{\nabla T^t D \nabla T} = 1 \tag{2}$$

based on the fact that reaction-diffusion equations admit traveling wave solutions in certain cases [1, 2]. This form is preferable because it does not require the whole distribution of u to compute the motion of a single iso-contour, it can reduce the number of unknown parameters based on the form of F and it can be solved faster. Aiming at clinical applications, it is essential to solve this *front tracking approximation* given in Equation 2 efficiently and accurately.

In this paper, we propose a new fast algorithm to solve the anisotropic Eikonal equation and improve the current front tracking approximation by taking into account the convergence properties of the reaction-diffusion equations. For demonstration, we apply these methods to tumor growth modeling, more specifically, to modeling glial based ones. These tumors account for approximately 40-45% of all primary intracranial tumors, forming the largest class in this pathology, [3]. Characteristics of this type of tumors vary a lot within the group, and when faced with one, understanding its aggressiveness and correct grading is very crucial in therapy planning and might improve prognosis. Although medical imaging is not the sole source of information used for this, it plays an important role in understanding the pattern and speed of invasion of healthy tissue by cancerous cells. One of the most important hints that can be obtained from images is the progression of the visible tumor front. Therefore mathematically describing and simulating the motion of this front would help the grading process and therapy planning.

2 Recursive Anisotropic Fast Marching

Anisotropic Eikonal equations, given as Equation 2, poses extra difficulties for fast numerical schemes compared to its isotropic counterpart, $F|\nabla T| = 1$. There have been different ways proposed to solve such equations or in general convex, static Hamilton-Jacobi equations using single-pass methods [4], or iterative methods [5, 6]. Single-pass methods start from points where time (T) values are already known and follow the characteristic direction of the PDE to compute T at other points. This approach is based on the fact that in equations such as Eqn. 2, the value of T at a point is only determined by a subset of its neighboring points, which lie along the characteristic direction [7]. In isotropic case these methods are very efficient because they follow gradient direction, which coincides the characteristic direction [8]. In other words, they only use immediate neighbors of a point with lower values of T to compute the new arrival time at that point using an upwind scheme. In the anisotropic case, characteristic direction does not necessarily coincide with gradient direction and the same idea used for isotropic case yields false results,see Fig. 1. In order to deal with this, Sethian and Vladimirsky enlarged the neighborhood around a point used to compute the new arrival time such that characteristic direction remains within

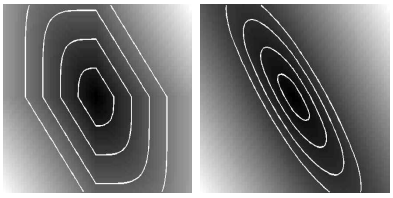

Fig. 1. $F\sqrt{\nabla T^t D \nabla T}$ with a constant anisotropic D solved using isotropic Fast Marching Method (left) and the solution solved by anisotropic methods (right).

the neighborhood [4]. But size of the enlarged neighborhood increases with increasing anisotropy of D. This would result in large number of points used to calculate new values and a high computational load in case of high anisotropies. Iterative methods start from an initial distribution of T, use an upwind, monotone and consistent discretization of the equation and iterate over the domain until convergence [5]. However, depending on the spatial variation of D these methods might need high number of iterations to converge and they need an ordering of the mesh to sweep the domain, which might not be trivial to obtain for general meshes.

2.1 - Algorithm: Recursive anisotropic fast marching, proposed here, is based on the single-pass idea and it uses immediate neighborhood to compute arrival times. As a novel step, it includes a recursive correction scheme taking into account the fact that due to anisotropy the immediate neighborhood used for computation may not always contain the characteristic direction. This algorithm works efficiently under general meshes, very high anisotropies and highly varying D fields. Moreover, it can be applied to more general forms of static, convex Hamilton-Jacobi equations, which is beyond the scope of this article.

Algorithm 1 Anisotropic Fast Marching: Initialization

```
for all X ∈ KNOWN do
  for all Y_i ∈ N(X) and Y_i ∈ FAR do
    compute T(Y_i) ← UPDATE(Y_i, X)
    remove Y_i from FAR and add Y_i to TRIAL
  end for
end for
```

The overall algorithm is similar to the original fast marching method proposed for the isotropic Eikonal equation. The main differences are the recursive correction scheme and the computation of T values. The initialization steps are the same: we go over points whose value are already known (kept in the $KNOWN$ list), compute their unknown neighbors' values using only known points (kept in the FAR list) trial T values and add these neighbors to the

$TRIAL$ list while removing them from the FAR list, see Algorithm 1. By neighborhood $\mathcal{N}(X)$ we mean all points directly connected to the point X in some preferred connectivity sense (e.g. 4-8 in 2D and 6-18-26 in 3D cartesian grid). The isotropic fast marching algorithm follows same operations throughout its main loop. The $TRIAL$ point with the minimum value of T, Y, is removed from the $TRIAL$ list, added to the $KNOWN$ list, trial values of unknown neighbors of Y are computed, if they are in the FAR list they are added to the $TRIAL$ list and removed from the FAR one, and if they are already in the $TRIAL$ list their values are updated.

In order to take into account the anisotropy, we insert the recursive correction in the main loop. Before trial values of unknown neighbors of Y are computed, we recompute its known neighbors' values. The reason for this is that when values of these points were computed Y was not used since it was not known. Hence, in their computation the characteristic direction may not have been contained in their known neighborhood, which was used to compute their T value. If we obtain a lower value of T during the recomputation we update the value and add the point to the $CHANGED$ list, which holds known points whose values have been changed. This correction is based on the fact that the lowest T value for a point is obtained when the characteristic direction is contained in the neighborhood used for the computation [6]. Every time the main loop restarts it checks if the $CHANGED$ list is empty, if this is not the case then instead of taking a point from the $TRIAL$ list it takes from the $CHANGED$ list. In other words the main loop tries to empty the $CHANGED$ list first. The pseudo code for the algorithm gives a clear summary in Algorithm 2.

2.2 - Local Solver: Up to now we have not detailed the computation of $T(X)$ value using $\mathcal{N}(X)$. We have defined $\mathcal{N}(X)$ as the set of immediate neighbors of X and naturally there exists a set of elements corresponding to this neighborhood, set of triangles ($\triangle_X$) in 2D or set of tetrahedras (TET_X) in 3D. $T(X)$ is calculated inside every element using linear interpolation between nodes and solving a minimization problem, $T(X) = f_{1D}(X,Y)$ for 1D, $T(X) = f_{2D}(X,Y,Z)$ for 2D and $T(X) = f_{3D}(X,Y,Z,W)$ for 3D, using nodes of the element neighboring X.

$$f_{1D}(X,Y) = T(Y) + \frac{[\mathbf{v}_{1D}^t D^{-1} \mathbf{v}_{1D}]^{1/2}}{F} \quad (3)$$

$$f_{2D}(X,Y,Z) = \min_{p\in[0,1]} \{T(Y)p + T(Z)(1-p) + \frac{[\mathbf{v}_{2D}(p)^t D^{-1} \mathbf{v}_{2D}(p)]^{1/2}}{F}\} \quad (4)$$

$$f_{3D}(X,Y,Z,W) = \min_{p,q\in[0,1]\times[0,1]} \{[T(Y)p + T(Z)(1-p)]q + T(W)(1-q) + \frac{[\mathbf{v}_{3D}(p,q)^t D^{-1} \mathbf{v}_{3D}(p,q)]^{1/2}}{F}\} \quad (5)$$

where $\mathbf{v}_{1D} = \overrightarrow{YX}$, $\mathbf{v}_{2D}(p) = \overrightarrow{YX}p + \overrightarrow{ZX}(1-p)$ and $\mathbf{v}_{3D}(p,q) = [\overrightarrow{YX}p + \overrightarrow{ZX}(1-p)]q + \overrightarrow{WX}(1-q)$. The common term $[\mathbf{v}^t D^{-1} \mathbf{v}]^{1/2}/F$ visible in all these equations is the time difference between a point connected to X with vector $\mathbf{v}$ and $T(X)$

Algorithm 2 Anisotropic Fast Marching: Main Loop with Recursive Correction

```
while TRIAL or CHANGED lists are not empty do
  if CHANGED list is not empty then
    X ← argmin_{X∈CHANGED} CHANGED
    remove X from CHANGED
  else
    X ← argmin_{X∈TRIAL} TRIAL
    remove X from TRIAL and add X to KNOWN
  end if
  for all X_i ∈ N(X) and X_i ∈ KNOWN do
    compute T̄(X_i) ← UPDATE(X_i, X)
    if T̄(X_i) < T(X_i) then
      T(X_i) ← T̄(X_i)
      add X_i to CHANGED list
    end if
  end for
  for all Y_i ∈ N(X) and Y_i ∈ TRIAL ∪ FAR do
    compute T̄(Y_i) ← UPDATE(Y_i, X)
    if Y_i ∈ TRIAL and T̄(Y_i) < T(Y_i) then
      T(Y_i) ← T̄(Y_i)
    else if Y_i ∈ FAR then
      T(Y_i) ← T̄(Y_i)
      remove Y_i from FAR and add Y_i to TRIAL
    end if
  end for
end while
```

under the effect of the diffusion tensor D. It is derived from the group velocity idea for which the details can be found in [9]. As in the original fast marching algorithm we only use known points in $\mathcal{N}(X)$ to compute the value at X. In other words when in the case of a tetrahedral element we use Equation 5 when all nodes are known, Equation 4 when 2 nodes are known and Equation 3 when only 1 node is known, see Algorithm 3. The minimization of Equation 4 has an analytical solution however, the one in Equation 5 is not trivial. Instead of solving it with a minimization algorithm, which would increase the computational load, we use the quadratic equation in $T(X)$ obtained by discretizing equation $F\sqrt{\nabla T^t D \nabla T} = 1$ on the nodes of the tetrahedral element. We check if this computed value of $T(X)$ satisfies the causality condition, which is that the characteristic direction should lie inside the element used. Practically this is just computing ∇T using the new computed $T(X)$ on the element and checking if $D\nabla T$ vector resides within the tetrahedra. If this is the case, the minimum lies inside the tetrahedra and it is approximated with the computed $T(X)$. If this is not the case we search the minimum on the triangular sides of the tetrahedra using f_{2D}. This method was proposed by Qian *et al.* [6] and it speeds up the overall algorithm greatly.

We have tested the proposed algorithm by solving $F\sqrt{\nabla T^t D \nabla T} = 1$ in 2D, 3D cartesian grid and on surfaces using triangulation. These results are shown in

Algorithm 3 Computation of $\overline{T}(X_i) = UPDATE(X_i, X)$

IN 2D
$\overline{T}(X_i) \leftarrow \infty$
for all $\triangle(XX_iY) \in \triangle_{X_i}^X = \{\triangle(XX_iY) | Y \in \mathcal{N}(X_i)\}$ **do**
  if $Y \in KNOWN$ **then**
    $\overline{T}(X_i) \leftarrow \min(\overline{T}(X_i), f_{2D}(X, X_i, Y))$
  else
    $\overline{T}(X_i) \leftarrow \min(\overline{T}(X_i), f_{1D}(X, X_i))$
  end if
end for
IN 3D
$\overline{T}(X_i) \leftarrow \infty$
for all $TET(XX_iYZ) \in TET_{X_i}^X = \{TET(XX_iYZ) | Y, Z \in \mathcal{N}(X_i)\}$ **do**
  if $Y, Z \in KNOWN$ **then**
    $\overline{T}(X_i) \leftarrow \min(\overline{T}(X_i), f_{3D}(X, X_i, Y, Z))$
  else if $Y \in KNOWN$ **then**
    $\overline{T}(X_i) \leftarrow \min(\overline{T}(X_i), f_{2D}(X, X_i, Y))$
  else if $Z \in KNOWN$ **then**
    $\overline{T}(X_i) \leftarrow \min(\overline{T}(X_i), f_{2D}(X, X_i, Z))$
  else
    $\overline{T}(X_i) \leftarrow \min(\overline{T}(X_i), f_{1D}(X, X_i))$
  end if
end for

Fig. 2. Computation times for these results can be found in Table 1, where we also compare our algorithm with the sweeping algorithm proposed in [6], for which we used our own implementation done in the best way possible. Comparison is only done for cases in 2D cartesian grid based on the examples provided in the mentioned reference. The sweeping method has been iterated until convergence, where the maximum number of iterations was 12 in the variable D case. In the recursive anisotropic fast marching algorithm the size of the $CHANGED$ list did not exceed 3 for these cases. The following computational times were obtained with Matlab7.1 for 2D cases and C++ for 3D cases on a 2.4GHz Intel Pentium machine with 1Gb of RAM. Cases given in Table 1 correspond to images shown in Fig. 2. The proposed algorithm is fast and visually accurate even in the case of very high and variable anisotropy. Moreover, applying the explained method to general meshes bears no difficulty. In our experiments with triangular meshes on 2D and on surfaces, the algorithm was appearently much faster.

3 Approximating the Front Motion: Time Varying Speed

Reaction-diffusion models explain the change of the distribution of densities by combination of diffusion and reaction processes. Usually, one is interested in the motion of an iso-contour (front) of such distributions, which can be attained by an anisotropic Eikonal equation $F\sqrt{\nabla T^t D \nabla T} = 1$. In this approximation, the speed term $F = F(x)$ is normally set to a function constant in time [10]. But

Case (D is anisotropic in all cases)	Sweeping Method [6] (seconds)	Fast Marching (seconds)
2D: constant D, 64×64 grid	24.43	16.15
2D: constant D, 128×128 grid: Fig. 2(a)	91.06	63.39
2D: spirally varying D, 64×64 grid: Fig. 2(c)	80.6076	13.56
2D: spirally varying D, 128×128 grid	319.34	49.48
3D: constant D, $64 \times 64 \times 18$ grid: Fig. 2(g)		26
3D: helix D, $64 \times 64 \times 64$ grid: Fig. 2(h)		65
3D: constant D, 13000 node mesh: Fig. 2(e)		2

Table 1. Computational times

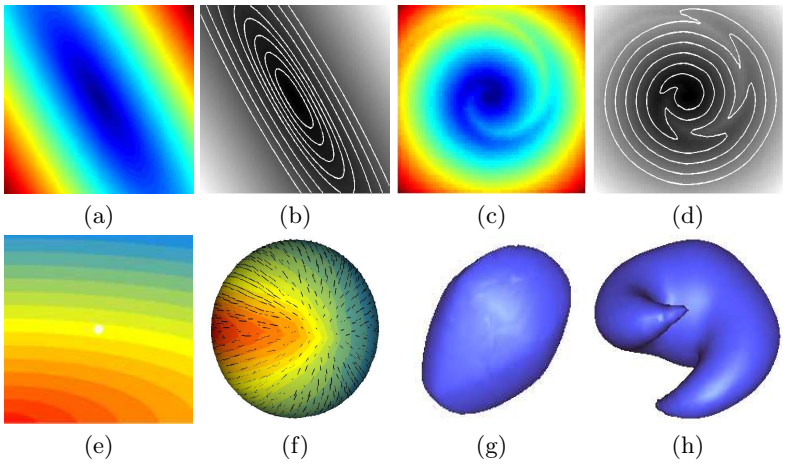

Fig. 2. a) 2D cartesian grid, high anisotropy in 120° increasing distance from blue to red, b) iso-contours of a, c) 2D cartesian grid, D is highly anisotropic inside a spiral following it, isotropic in other regions, d) iso-contours of c, e) 2D triangular mesh with 13000 nodes anisotropy in x direction, colors represent iso-contours, f) 2D triangular mesh on a surface D is anisotropic and principle eigenvector is shown in black lines, colors represent iso-contours, g) 3D cartesian grid, anisotropic D h) 3D cartesian grid, D is highly anisotropic inside a helix following it, isotropic in other regions.

the convergence properties of the reaction-diffusion system is then neglected, which can lead to important errors. Taking this into account, we propose to use a time varying function $F = F(x, T)$ to have a better approximation of the front motion.

In order to obtain the approximation for the motion of the front we start from the reaction-diffusion equation in 1-D with constant coefficients:

$$\begin{aligned} u_t &= du_{xx} + \rho f(u) \\ u(x, 0) &= U \end{aligned} \tag{6}$$

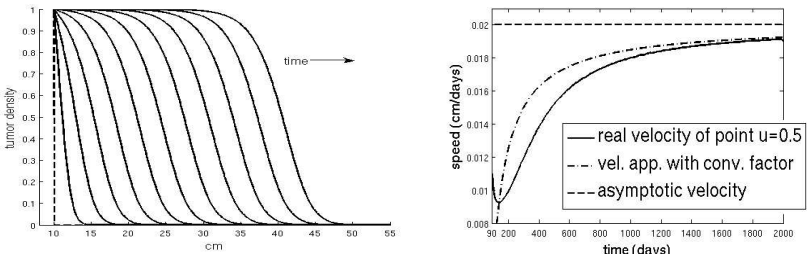

Fig. 3. Left: Front evolution starting from a step function under Equation 6. Right: Speed of the point $u = 0.5$ on the front is plotted along with the asymptotic speed and the speed approximation including convergence term.

where $f(u)$ has a homogeneous stable state at $u = 1$ and a homogeneous unstable state at $u = 0$. Equation 6, which is also called the Fisher-Kolmogorov equation, admits traveling wave solutions in the form of $u(x,t) = u(x - vt) = u(\xi)$, where v is the asymptotic speed of the wavefront, [11, 12]. In other words, initial conditions evolve in time to uniformly translating front shapes, see Fig. 3. This allows us to formulate the motion of a single point on the wavefront $u = u_0$ by simply describing the arrival time of the point $u = u_0$ as $T_x = 1/v$, where T is the arrival time function. The value of the asymptotic speed depends on the initial condition U. All initial conditions that are steep enough, meaning $\lim_{x\to\infty} u(x,0) = 0$, converge to the travelling wave moving with an asymptotic speed: $2\sqrt{df'(0)}$, where $f'(0)$ denotes the derivative at $u = 0$ [1]. This suggests that for all practical modeling problems we can use this asymptotic value in the arrival time formulation to approximate the motion of the front. However, it was shown that convergence of the front speed to the asymptotic one is

$$v(t) = 2\sqrt{df'(0)} - \frac{3}{2t}\sqrt{\frac{d}{f'(0)}} \tag{7}$$

independent of the initial condition and the u_0 value being tracked, [13, 2]. This convergence rate is algebraic and rather slow. Thus, using a time varying $v(t)$ as given in Equation 7 is a better approximation of the actual speed than using the asymptotic one, see Fig. 3. The convergence to a traveling wave behavior of the Equation 6 applies to higher dimensions when coefficients are constant and the initial condition has non-curved iso-contours. In this case the initial condition converges to a traveling plane having a cross section looking as a 1D wavefront with the time varying velocity given as $v(t) = 2\sqrt{f'(0)\mathbf{n}^t D\mathbf{n}} - 3/(2t)\sqrt{\mathbf{n}^t D\mathbf{n}/f'(0)}$, where $\mathbf{n}$ is the normal of the moving plane.

Convergence properties of the reaction-diffusion equation is only valid when the front is not curved and parameters of the equation are constants. Models using Equation 1 does not necessarily have constant parameters and the front they describe can be curved. In order to formulate the motion of the general front

using the construction explained above, we make the assumption that **within a voxel** front is planar and parameters of the reaction-diffusion equation are constants, which are taken as values at that voxel. Under these assumptions, we derive the arrival time formulation for the front in 3D as:

$$|\nabla T| = 1/v(t) = \left[2\sqrt{f'(0)\mathbf{n}^t D\mathbf{n}} - \frac{3}{2T}\sqrt{\frac{\mathbf{n}^t D\mathbf{n}}{f'(0)}} \right]^{-1} \tag{8}$$

where $\mathbf{n}$ can be replaced by $\nabla T/|\nabla T|$. This simply leads to the anisotropic Eikonal equation given as:

$$\sqrt{\nabla T^t D \nabla T} = \frac{2\sqrt{f'(0)}T}{4f'(0)T - 3} \tag{9}$$

4 Application: Predicting the Motion of Tumor Front

There has been a large amount of mathematical models proposed to describe the growth dynamics of glial tumors, e.g. [14, 15]. Those trying to explain growth and invasion dynamics based on observations from medical images, describe these processes using cell densities and average behavior [15], consisting of fewer equations and parameters. Such models are based on reaction-diffusion formalism introduced in [11], which uses reaction-diffusion type equations. Although these models are successful in explaining underlying dynamics of the tumor growth, they encounter some problems in adapting to patient data. Given images, in order to compute the growth they require tumor cell density values at every point. What is available in conventional modalities like MR and CT is not cell densities, but an enhanced homogeneous looking region and its boundary with the brain tissue, the tumor front, as seen in Fig. 4(a). Moreover, numerical load of such simulations, depending on the mesh size used, can be very heavy.

In this paper for demonstration, we use the front approximation given in Section 3 to describe the motion of the tumor front and use the recursive anisotropic fast marching algorithm to simulate its motion. The specific reaction-diffusion model we base our approximation on is proposed by Clatz *et al.* [16]. Their model for tumor growth can be given by the following reaction-diffusion equation:

$$\frac{\partial u}{\partial t} = \nabla \cdot (D(\mathbf{x})\nabla u) + \rho u(1-u) \tag{10}$$

$$D(\mathbf{x})\nabla u \cdot \overrightarrow{n}_{\Sigma} = 0 \tag{11}$$

where u is the tumor cell density, $D(\mathbf{x})$ is the diffusion tensors, ρ is the proliferation rate of tumor cells and $\overrightarrow{n}_{\Sigma}$ is the normal direction at the boundaries, which in the case of the brain are skull and ventricles. The first term in Equation 10, $\nabla \cdot (D(\mathbf{x})\nabla u)$, defines invasion of brain tissue by tumor cells using a diffusion process. The second term in the same equation, $\rho u(1-u)$ describes proliferation of tumor cells using logistic growth. Based on the experiments done by Giese *et al.* [17], which shows that tumor cells move faster on myelin sheath, $D(\mathbf{x})$ is set

as a spatially varying tensor, becoming isotropic on grey matter and anisotropic on white matter following fiber tracts as defined in Equation 12.

$$D(\mathbf{x}) = \begin{cases} d_g\mathbf{I} & \text{if } \mathbf{x} \text{ is in grey matter} \\ \mathbf{V}(Diag(d_w\lambda_1,\, 0,\, 0) + d_g\mathbf{I})\mathbf{V}^t & \text{if } \mathbf{x} \text{ is in white matter} \end{cases} \tag{12}$$

where d_g and d_w are speed of diffusion in grey matter and white matter respectively, $\mathbf{I}$ is the 3x3 identity matrix, λ_1 is the principal eigenvalue and $\mathbf{V}$ is the eigenvector matrix of the water diffusion tensor obtained from DT-MRI. By construction d_w/d_g can have a large value resulting in a high anisotropy.

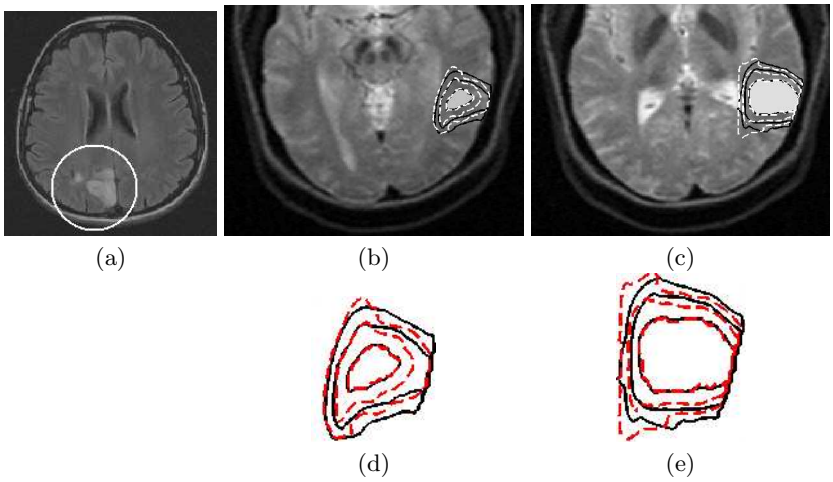

Fig. 4. a) A low grade glioma (inside the white circle) showing a single contour around a homogeneous prolongation of Flair MRI. b,c) Different axial slices of an artificially grown tumor. Grey region shows the visible part of the tumor at day 90. Black contours shows the location of $u = 0.1$ front of the artificial tumor at days 180 and 240. White contours show same locations computed by the front approximation. d,e) Same contours as shown in b,c. Black solid is the actual location and red dashed are approximated locations. Discrepancies between contours is a result of curvature and boundary effects, which were not taken into account in the current state of the front approximation.

The front approximation for this model can be given as:

$$\sqrt{\nabla T^t D \nabla T} = \frac{2\sqrt{\rho}T}{4\rho T - 3} \tag{13}$$

In order to have a realistic simulation and to compare the results of the front approximation and the reaction-diffusion model, we have grown an artificial tumor using real boundary conditions, white-grey matter segmentation and diffusion tensor images (DTI) taken from a healthy subject. We set the diffusion coefficients as $d_g = 0.001$ and $d_w = 0.1$ and the proliferation rate as an average rate of

$\rho = 0.012$ [15]. The artificial tumor is grown for 240 days using the formulation given in Equation 10 and 11. We took the time of diagnosis as day 90, where we obtain the first image as in Fig. 4(b) and observe the tumor front, taken as $\gamma = \{\mathbf{x}|u(\mathbf{x}, 90) = 0.1\}$. We set this contour as the boundary condition of Equation 13 ($T(\gamma) = 90$) and we solved it using the recursive anisotropic fast marching algorithm. Fig. 4(c) and (d) compare the motion of the front computed by the reaction-diffusion model (solid blue curves) and computed by the front approximation (dashed red curves). Observe that the speed of invasion is well captured by the front approximation formulation given in Equation 13. The pattern of invasion on the other hand shows some differences visible in Fig. 4. There are two reasons for these differences: (1) the front approximation is based on un-curved tumor fronts, however, curvature plays a role in smoothing and slowing down the front, (2) the Neumann boundary conditions are not captured by the approximation which creates differences near the boundary. The computation time for reaction-diffusion model to grow the tumor for 240 days was 4500 seconds while fast marching algorithm computed the motion of the γ front throughout the whole brain in 250 seconds, which corresponds to a growth simulation of 2500 days. This yields a speed up of nearly 180 folds. Implementations were done in C++ using GMM library on a 2.4GHz Intel Pentium processor with a 1Gb memory.

5 Discussions

In this paper, we proposed the recursive anisotropic fast marching algorithm and to use the time varying speed term in front approximation formulation for reaction-diffusion models in order to create the link between mathematical models and clinical observations. The fast marching algorithm is successful in handling high anisotropies, which are often encountered in biological modeling, on general meshes. We demonstrated the usage of proposed tools by simulating the motion of the tumor front, visible in medical images.

Formulating the motion of the tumor front sets a link between observations in images and the mathematical models explaining the growth dynamics. Fast and efficient algorithms like the recursive fast marching method explained here gives us an easy tool to adapt growth models to specific patient cases and do simulations. It can also help us quantify the speed of growth and invasion by solving an inverse problem to estimate the parameters like diffusion coefficients. We have seen that the formulation is successful in capturing the speed of invasion, however, the patterns have differences. In the future, we would like to explore new ways to integrate the effect of curvature and boundaries in the formulation to get better approximation.

Although we have concentrated on tumor growth modeling in this paper, front approximations for reaction-diffusion equations and the proposed algorithm can be useful in other modeling problems as well. Such an example is the electrophysiological model of the heart, where one is interested in computing the excitation times throughout the organ. This problem is similar to tumor

growth, in the sense that it requires following the motion of a front. Moreover, this application requires fast computations, which is possible with the method presented in this work.

The recursive fast marching method explained here is a general tool and can be used for lots of different applications than simulating the motion of a wave-front, such as fiber tracking or geophysics. Moreover, the algorithm can also be used for solving general static, convex Hamilton-Jacobi equations encountered in computer vision and material science. The future work consists in characterizing convergence properties of the recursive fast marching algorithm.

References

1. Aronson, D., Weinberger, H.: Multidimensional nonlinear diffusion arising in population genetics. Advances in Mathematics **30** (1978)
2. U. Ebert, W.S.: Front propagation into unstable states: universal algebraic convergence towards uniformly translating pulled fronts. Physica D: Nonlinear Phenomena **146** (2000)
3. Tovi, M.: Mr imaging in cerebral gliomas analysis of tumour tissue components. Acta Radiol. Suppl. (1993)
4. Sethian, J., Vladimirsky, A.: Ordered upwind methods for static hamilton-jacobi equations: theory and algorithms. SIAM J. Numer. Anal. **41** (2003)
5. Kao, C., Osher, S., Tsai, Y.: Fast sweeping methods for static hamilton-jacobi equations. SIAM J. Numer. Anal. **42** (2005)
6. Qian, J., Zhang, Y., Zhao, H.: A fast sweeping method for static convex hamilton-jacobi equations. UCLA Comp. and App. Math. Reports **06-37** (2006)
7. Kevorkian, J.: Partial differential equations: Analytical solution techniques. Springer (2000)
8. Sethian, J.: Level set methods and fast marching methods: Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge University Press (1999)
9. Qian, J., Symes, W.: Paraxial eikonal solvers for anisotropic quasi-p travel times. J. Comp. Physics **173** (2001)
10. Keener, J., Sneyd, J.: Mathematical physiology. Springer (1998)
11. Murray, J.: Mathematical Biology. Springer-Verlag (2002)
12. Maini, P., McElwain, D., Leavesley, D.: Traveling wave model to interpret a wound-healing cell migration assay for human peritoneal mesothelial cells. Tissue Eng. **10** (2004)
13. Bramson, M.: Convergence of solutions of the kolmogoroff equations to traveling waves. Mem. Am. Math. Soc. (1983)
14. Cristini, V., Lowengrub, J., Nie, Q.: Nonlinear simulation of tumor growth. Journal of Math. Biol. **46** (2003)
15. Swanson, K., Alvord, E., Murray, J.: Virtual brain tumours (gliomas) enhance the reality of medical imaging and highlight inadequacies of current therapy. British Journal of Cancer **86** (2002)
16. Clatz, O., Sermesant, M., Bondiau, P., Delingette, H., Warfield, S., Malandain, G., Ayache, N.: Realistic simulation of the 3d growth of brain tumors in mr images coupling diffusion with biomechanical deformation. IEEE T.M.I. **24**(10) (2005)
17. Giese, A., Kluwe, L., Laube, B., Meissner, H., Berens, M., Westphal, M.: Migration of human glioma cells on myelin. Neurosurgery **38**(4) (1996)