How To Deal with Point Correspondences and Tangential Velocities in the Level Set Framework

J.-P. Pons[†], G. Hermosillo[*], R. Keriven[†] and O. Faugeras[*]
[*]INRIA, 2004 Route des Lucioles
Sophia-Antipolis, France
[†]CERMICS, ENPC
Marne-La-Vallée, France
[jppons,ghermosi,faugeras]@sophia.inria.fr, keriven@cermics.enpc.fr

Abstract

In this paper, we overcome a major drawback of the level set framework: the lack of point correspondences. We maintain explicit backward correspondences from the evolving interface to the initial one by advecting the initial point coordinates with the same speed as the level set function. Our method leads to a system of coupled Eulerian partial differential equations. We show in a variety of numerical experiments that it can handle both normal and tangential velocities, large deformations, shocks, rarefactions and topological changes. Applications are many in computer vision and elsewhere since our method can upgrade virtually any level set evolution. We complement our work with the design of non zero tangential velocities that preserve the relative area of interface patches; this feature may be crucial in such applications as computational geometry, grid generation or unfolding of the organs' surfaces, e.g. brain, in medical imaging.

1. Introduction

One major drawback of the level set method [14] is the inability to keep point correspondences during the deformation. As a consequence it is not possible to handle data associated with a moving interface with the straightforward level set approach. Some hybrid Lagrangian-Eulerian methods have been proposed to circumvent this limitation [9, 19, 21]. However, for sake of stability and topology independence, a completely Eulerian approach is desirable. In [5] the authors come to this end in the particular case of region tracking. Recently, [22, 1] address the transport and the diffusion of general data on a moving interface in the level set framework. In Section 2 we go one step further and design a new method that maintains explicit point correspondences during the evolution.

We then turn to the design of non trivial (i.e. non zero) tangential velocities. This problem has been overlooked so far because it is known that tangential velocities have no influence on the geometry of the interface. But they do affect point correspondences. In Section 3 we show that in some cases interface data may be considerably altered without an adequate tangential velocity and we describe a method to build a tangential velocity that preserves the relative area of interface patches, ie the ratio between the area of any patch and the total area of the interface. Our method can be an alternative to the area preserving mappings worked out in [2] for the purpose of cortex unfolding and virtual colonoscopy.

2. Correspondences in the level set framework

2.1. Method 1: general interface data

In [5], region tracking on a deforming interface Γ is achieved in the level set framework by representing the boundary of the region of interest as the intersection of Γ with an auxiliary surface $\hat{\Gamma}$. Γ and $\hat{\Gamma}$ are represented as the zero level sets of the functions ϕ and $\hat{\phi}$ respectively. The latter are evolved according to the following system of coupled PDE's:

$$\left\{ \begin{array}{lcl} \frac{\partial \phi}{\partial t} & = & \beta \, \| \nabla \phi \| \\ \frac{\partial \hat{\phi}}{\partial t} & = & \left(\beta \frac{\nabla \phi}{\|\nabla \phi\|} \cdot \frac{\nabla \hat{\phi}}{\|\nabla \hat{\phi}\|} \right) \left\| \nabla \hat{\phi} \right\| \end{array} \right. , \tag{1}$$

where β is the magnitude of the inward normal speed of the interface Γ. Equation (1) also writes

$$\left\{ \begin{array}{l} \mathbf{v} = -\beta \frac{\nabla \phi}{\|\nabla \phi\|} \\ \frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0 \\ \frac{\partial \hat{\phi}}{\partial t} + \mathbf{v} \cdot \nabla \hat{\phi} = 0 \, . \end{array} \right. \tag{2}$$

In equation (2) we can now clearly see that Γ and $\hat{\Gamma}$ move with the same speed $\mathbf{v}$. In [5] this speed is taken normal to the interface but we could as well consider velocities containing a tangential part. Tangential velocities do not affect

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

IEEE COMPUTER SOCIETY

the evolution of the interface, but they do affect the evolution of $\hat{\Gamma}$ and hence of the region of interest.

Moreover, we notice that the zero level set of $\hat{\phi}$ does not play a particular role. All the level sets of $\hat{\phi}$ evolve according to $\mathbf{v}$. Actually, the evolution equation for $\hat{\phi}$ is nothing but an advection equation with an extrinsic velocity field.

Hence we can safely forget that $\hat{\phi}$ stands for the level set function of a hypersurface and change it to a general scalar or vector-valued function u coding for some data on the interface. For example, if we want to visualize neural activity on the cerebral cortex, u could be the functional MRI signal. We then obtain a straightforward method to associate a quantity to a moving interface in the level set framework. This method is summarized by the following system of coupled PDE's:

$$\left\{ \begin{array}{ll} \frac{\partial \phi}{\partial t} + \mathbf{v} \cdot \nabla \phi = 0 \;, & \text{(3a)} \\ \frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u = 0 \;, & \text{(3b)} \end{array} \right. \qquad (3)$$

where $\mathbf{v}$ is either given or defined from geometric properties of the zero level set of ϕ (e.g. normal, curvature, etc.).

In [22, 1], a variant of this method is worked out to model the evolution of a material quantity along a moving interface in the level set framework: besides being passively advected, interface data are also scaled due to local compression/expansion of the front and diffused along the interface. Although more elaborate than equation (3), this formulation is not relevant to our problem of point correspondences.

Note that the method stated in equation (3) can be plugged into an existing level set interface evolution without any modification. Moreover, while the proper numerical scheme for equation (3a) depends on the chosen velocity, equation (3b) only requires a simple upwinding scheme since $\mathbf{v}$ is extrinsic with respect to u.

In many real problems, both $\mathbf{v}$ and u have natural extensions off the interface. For instance, the fMRI signal is known throughout the scanning volume. Otherwise one may build an extension of $\mathbf{v}$ and/or u that is constant along the normal to the interface by considering the steady state of the following PDE [4]:

$$\frac{\partial u}{\partial t} + sign(\phi)\,(\nabla u \cdot \nabla \phi) = 0.$$

2.2. Method 2: explicit backward correspondences

Method 1 suffers from several limitations: it does not provide explicit point correspondences and requires to solve a PDE for each scalar component of the interface data.

In order to solve these problems we focus on the initial coordinates of the points of the interface. We regard the latter as vector-valued interface data that we can advect as suggested previously. Let us consider a function $\Psi : \mathbb{R}^n \times \mathbb{R}^+ \mapsto \mathbb{R}^n$ such as

$$\left\{ \begin{array}{l} \Psi(\mathbf{x}, 0) = \mathbf{x} \;, \\ \frac{\partial \Psi}{\partial t} + D\Psi\,\mathbf{v} = 0 \;, \end{array} \right. \qquad (4)$$

where $D\Psi$ stands for the jacobian matrix of Ψ.

Then for each point $\mathbf{x}$ of the interface at time t, $\Psi(\mathbf{x}, t)$ holds the position that this point occupied at time $t = 0$. Hence $\Psi(., t)$ provides explicit backward correspondences from the current interface to the initial one.

Moreover, we can ignore equation (3b) and build any interface data by composition of some initial data u_0 with Ψ. Indeed, $u = u_0 \circ \Psi$ formally satisfies equation (3b):

$$\frac{\partial u}{\partial t} + \nabla u \cdot \mathbf{v} = (\nabla u_0 \circ \Psi) \cdot \left(\frac{\partial \Psi}{\partial t} + D\Psi\,\mathbf{v} \right) = 0 \;.$$

We might be tempted to consider forward correspondences too by the means of a function $\hat{\Psi} : \mathbb{R}^n \times \mathbb{R}^+ \mapsto \mathbb{R}^n$ such as

$$\left\{ \begin{array}{l} \hat{\Psi}(\mathbf{x}, 0) = \mathbf{x} \;, \\ \frac{\partial \hat{\Psi}}{\partial t} = \mathbf{v} \circ \hat{\Psi} \;. \end{array} \right. \qquad (5)$$

This approach is used in [19] to construct transverse lines for the purpose of grid generation with level set methods and in [21] for image registration. But the Eulerian PDE (4) has two important advantages over the Lagrangian ODE (5). First, it is numerically more stable since the computations are performed on a fixed grid. More importantly, it performs automatic deleting of merging characteristics, whereas this task requires intricate delooping algorithms in the Lagrangian approach. Moreover, forward correspondences may not exist if the interface evolution forms shocks; the interface may even collapse and mereley disappear. In such cases, equation (5) is not relevant.

For all the reasons discussed above, our method focuses on backward correspondences. In some applications, such as texture mapping or tracking of isolated points, forward correspondences are needed and it is necessary to invert the Ψ map at a postprocessing stage.

If $\mathbf{v}$ is sufficiently smooth [8, 20], equation (4) generates a one-parameter family of diffeomorphisms. So does equation (5) and by the chain rule we get $\Psi(., t) \circ \hat{\Psi}\;(, t) = \hat{\Psi}\;(, t) \circ \Psi(., t) = id \;\; \forall t$. This is not true for general velocities. Typically, Ψ fails to be surjective in the presence of a shock and fails to be injective in the presence of a rarefaction.

2.3. Numerical experiments

For equations (3b) and (4) we use a simple upwind scheme, since the velocity is extrinsic to u and Ψ. The proper scheme for equation (3a) depends on speed properties; all the schemes needed in our experiments are taken from [15, 19]. To estimate one-sided space derivatives we adopt the fifth order weigthed essentially non-oscillatory (WENO) scheme derived in [11, 10]. As regards time differencing, a simple Euler scheme has proven sufficient in our experiments. For sake of efficiency we use the PDE-based fast local level set implementation proposed in [16].

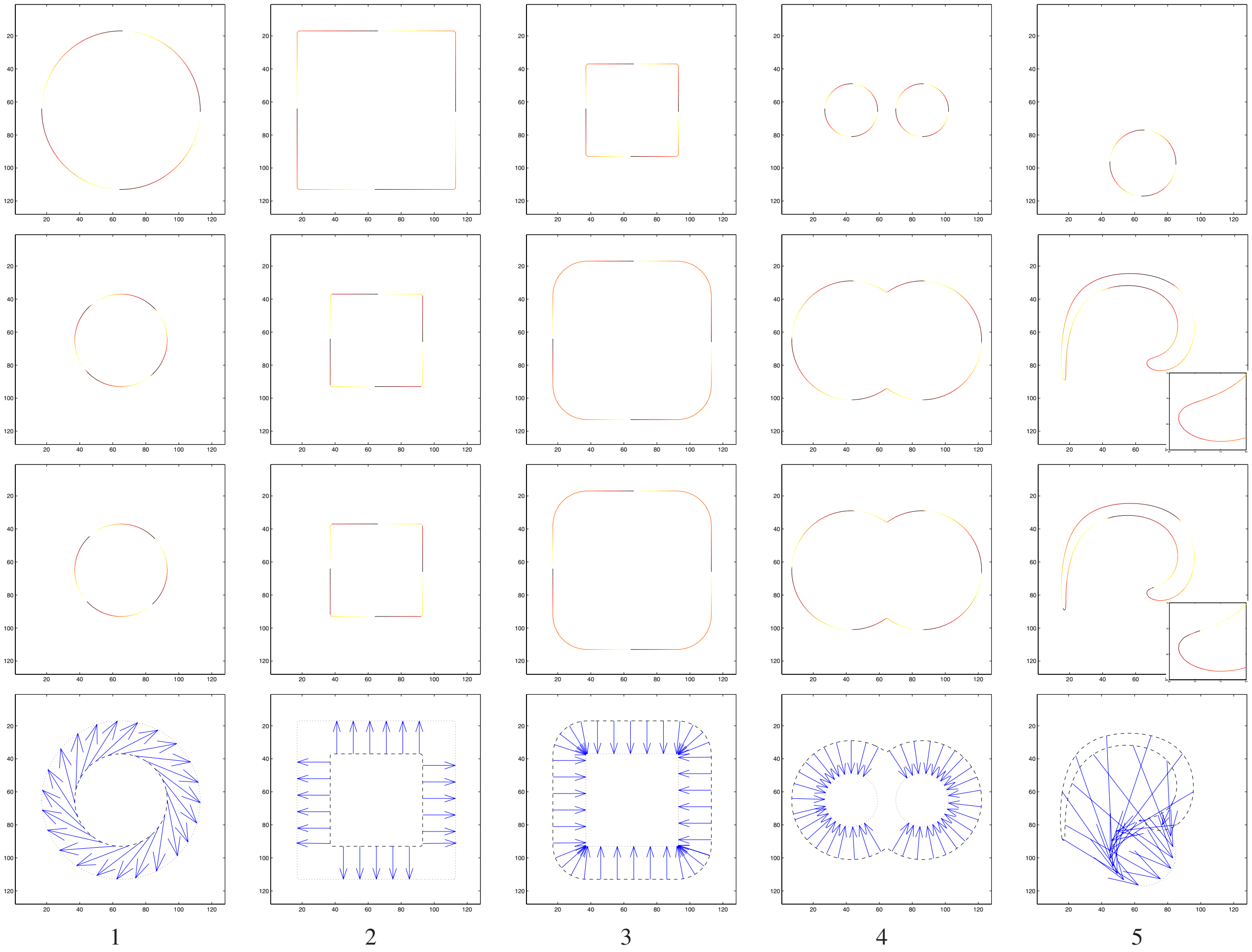

Figure 1. 2D examples of point correspondences in the level set framework (see text).

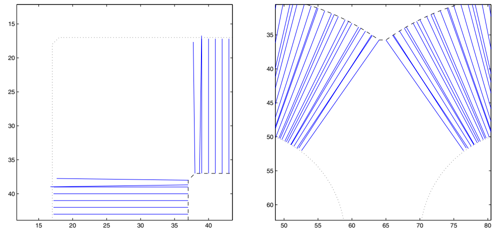

Figure 2. Some detailed views of Figure 1.

Figure 1 demonstrates our method in several 2D test cases. Throughout this series of experiments, we use a 128x128 computational grid. All curves are extracted with a marching-cube algorithm and the data are interpolated at resulting vertices. In all columns, the first row shows the initial curve colored with a texture u; the latter purposely holds both smooth parts and discontinuities. The second row shows the curve after deformation colored with the data tracked with method 1 (§2.1). The two remaining rows shows the results of method 2 (§2.2): the third row shows the final interface colored with the transformed texture $u_0 \circ \Psi$; in the fourth row, Ψ is plotted using cubic spline interpolation at one out of ten vertices. The arrow tails indicate the positions where Ψ is evaluated and the arrow heads point to the corresponding values of Ψ. The initial and the final curves are plotted too, with a dotted line and a dashed line respectively. The first column shows a circle shrinking with a constant speed in its normal direction and simultaneously advected by an extrinsic rigid rotation velocity field; this example demonstrates that both normal and tangential velocities can be implemented. The second and the third columns are dedicated to shocks and rarefactions; they present a shrinking and an expanding square respectively. The fourth column focuses on the merging of two expanding circles; this example demonstrates that our methods naturally handle topological changes. The fifth column depicts the evolution of a circle under a non-constant vorticity velocity field defined by

$$\mathbf{v}(x,y) = \begin{pmatrix} \sin^2(\pi x)\sin(2\pi y)) \\ -\sin(2\pi x)\sin^2(\pi y) \end{pmatrix}.$$

This example is more challenging than the previous ones because the flow considerably stretches the interface. The construction of an extension velocity constant along the normal to the interface has proven necessary here.

Our results are in remarkable agreement with the expected solutions. A comparison between row 2 and row 3 reveals that method 1 and method 2 perform similarly. However, in the last column, we note that method 2 better preserves interface data discontinuities. The latter also has wider applications since it provides explicit point correspondences. Hence method 2 should be generally preferred to method 1. If we examine row 4 we note that the values of ϕ on the final curves point almost exactly to the initial curves. In other words, although not enforced explicitely, the equality

$$\phi = \phi_0 \circ \Psi \tag{6}$$

holds with a good accuracy. For further accuracy, it is possible to postprocess ϕ as follows:

$$\Psi \;\leftarrow\; \Psi + (\phi - \phi_0 \circ \Psi)\; \frac{\nabla\phi_0 \circ \Psi}{\|\nabla\phi_0 \circ \Psi\|^2}\;.$$

This correction helps to enforce equality (6) by projecting each value of Ψ onto the suitable level set of ϕ_0.

Figure 2 provides some detailed views of examples 2 and 4, with a denser representation for Ψ. Arrow heads are hidden for better visualization. We can see that the expected discontinuities of Ψ are successfully recovered.

We now focus on method 2 and three dimensions. In Figure 3 we use our method to unfold a cortex surface, extracted from MRI data of the human brain, with the averaged mean curvature flow; the displayed data are the mean curvature of the initial surface.

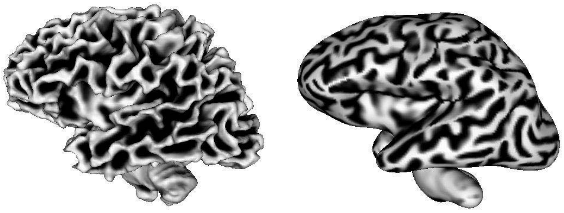

Figure 3. Cortex unfolding with method 2.

In order to test the robustness of our method to large deformations, we consider the following incompressible flow field proposed by LeVeque [12]:

$$\mathbf{v}(x,y,z) = \cos\frac{\pi t}{T}\begin{pmatrix} 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z) \\ -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z) \\ -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z) \end{pmatrix}.$$

This flow is a surimposition of a deformation in the xy plane with a deformation in the xz plane. It reverses at time $T/2$ so that the initial interface and data should be revovered at time T. This trick provides a convenient way to evaluate the correctness of our results. We take $T = 1$ and we chose the plane $x = 0.5$ as the initial interface. For our computations we use a 128x128x128 computational grid.

Figure 4 shows the computed solution at t=0, 0.1, 0.2, 0.5, 0.85 and 1. Rather than plotting point correspondences which are somewhat difficult to visualize in a 3D setting, we color the interface with some texture obtained by composition with Ψ. The mean square error on correspondences at $t = 1$ is 0.16 voxels. Visually, the result is close to perfect.

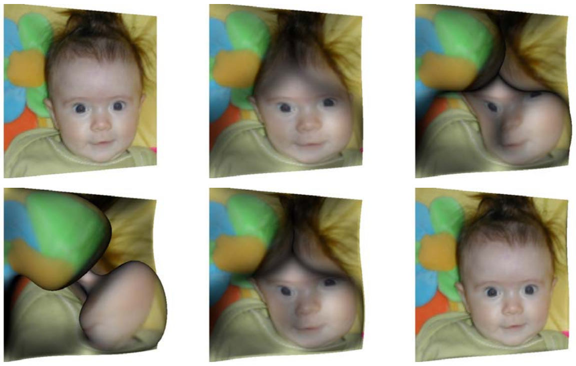

Figure 4. 3D deformation test for method 2.

3. Relative area preserving tangential velocities

Tangential velocities have no effect on the shape of evolving interfaces, but they do affect point correspondences and hence the evolution of interface data. In most problems where interface data are considered, a tangential velocity is unambiguously prescribed. However, one may encounter problems where no such natural tangential velocity exist and where interface data or point correspondence are nonetheless needed. The usual choice in such cases is a null tangential velocity, but we will demonstrate in some numerical experiments that this choice may considerably alter the local properties of interface data.

In this section we present a method to build non-trivial tangential velocities that preserve the relative area of interface patches, i.e. the ratio between the area of any patch and the total area of the interface.

3.1. Differential geometry formulation

In differential geometry terms [7, 6], the condition stated above writes

$$\mathrm{div}_\Gamma\,\mathbf{v} = \overline{\mathrm{div}_\Gamma\,\mathbf{v}}\;, \tag{7}$$

where div_Γ is the intrinsic divergence operator on Γ, and $^-$ denotes the average of a quantity along Γ.

If we decompose $\mathbf{v}$ into its outward normal component v_N and its tangential part $\mathbf{v}_T$, equation (7) becomes

$$\mathrm{div}_\Gamma\,\mathbf{v}_T + (n-1)\,H\,v_N = (n-1)\,\overline{H\,v_N}\;, \tag{8}$$

where H denotes the mean curvature of Γ. Note that if the normal velocity is such that the evolution preserves the total area of the interface, then the right-hand sides of equations (7) and (8) vanish.

For a plane curve C embedded in $\mathbb{R}^2$ deforming with speed $\mathbf{v} = \alpha \mathbf{T} + \beta \mathbf{N}$, the relative area preservation condition boils down to $\frac{\partial \alpha}{\partial s} + \kappa\beta = \overline{\kappa\beta}$, where κ denotes the curvature of C. This particular case has been used (with the inward normal convention) in [13] in the Lagrangian curve evolution framework: the authors build a tangential velocity that achieves a uniform redistribution of grid points along the curve, in order to overcome numerical instabilities caused by merging of grid points or by formation of the so-called swallow tails.

We now outline our method. Given a normal velocity field v_N, let us consider the solution η of the following intrinsic Poisson equation on Γ:

$$\Delta_\Gamma \, \eta = (n-1) \left(H \, v_N - \overline{H \, v_N} \right) \ , \qquad (9)$$

where Δ_Γ denotes the Laplace-Beltrami operator on Γ. Finding a solution of equation (9) is possible because the right-hand side is of zero average [18]. Moreover, the solution η is only defined up to a constant. Then, clearly,

$$\mathbf{v} = v_N \mathbf{N} - \nabla_\Gamma \, \eta \ , \qquad (10)$$

where ∇_Γ denotes the intrinsic gradient on Γ, verifies equation (8). Note that the normal velocity is not altered since $\nabla_\Gamma \, \eta$ is purely tangential, and that the resulting speed is non local: it depends on the whole shape and motion of the interface.

3.2. Numerical experiments

For each time step we have to solve an intrinsic Poisson equation. This can be done with a finite element technique on a triangulation of the interface as in [3]. Equation (9) then translates into a linear system with a sparse symmetric positive semi-definite matrix suited for numerical iterative methods such as the conjugate gradient (CG) method.

However, this method is not natural in the level set framework: it requires to compute a new triangulation and to extend the computed tangential velocity off the interface for each time step. Hence we adopt a cartesian grid approach to solve the Poisson equation, based on the discretization of the Laplace-Beltrami operator proposed in [4]. We have computed explicitly a discretized Laplace-Beltrami operator in 2 and 3 space dimensions by averaging complementary schemes which alternate backward and forward gradient and divergence operators. [17]. Following [16], we compute the average $\overline{H \, v_N}$ required in the right-hand side of the Poisson equation on the cartesian grid using a smoothed version of the Dirac function. Equation (9) then translates into a linear system with a sparse symmetric indefinite matrix which we solve with a minimum residual (MINRES) method. We use the solution η of time step $t-1$ as the initial guess for the MINRES method at time step t. This way, solving the Poisson equation with the desired accuracy only requires a handful of iterations.

In Figure 5 we show the results of some basic 2D evolutions with a null tangential velocity *(top)* and with a relative area preserving tangential velocity computed with our method *(bottom)*. The first two columns represent the case of a shrinking or expanding square respectively. The remaining columns detail the case of a free-hand folded figure evolving under the averaged mean curvature flow. This latter case illustrates that interface data may be considerably altered without an adequate tangential velocity. Indeed, note how the large interior part with high curvature of the initial curve *(third column)* turns into a much smaller patch on the final curve *(columns 4 and 5, top row)*. The corresponding data are unacceptably distorted. We insist on the fact that this phenomenon is not an artefact of our point tracking method: it is intrinsic to the evolution. In contrast, in the second row, the data are uniformly redistributed along the curve as expected.

We finish with a cortex unfolding example in Figure 6. We simulate a tumor on the left temporal gyrus of a subsampled human brain *(left)*. The area of the tumor is considerably underestimated if the cortex is unfolded with a regular mean curvature flow *(middle)*. This does not occur if we use a relative area preserving tangential velocity *(right)*.

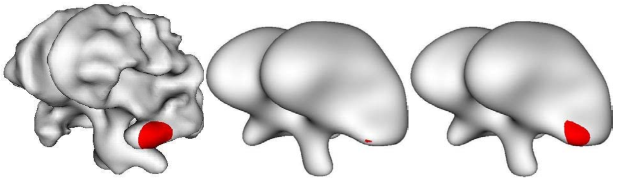

Figure 6. Cortex unfolding with (right) or without (middle) a relative area preserving tangential velocity.

4. Conclusion

We have described a completely Eulerian method to maintain explicit backward point correspondences in the level set framework. Our method is capable of handling both normal and tangential velocities, large deformations, shocks, rarefactions and topological changes.

We have shown that a zero tangential velocity may unacceptably distort interface data in some cases. We have solved this problem by designing a relative area preserving tangential velocity which uniformly redistributes the data along the interface.

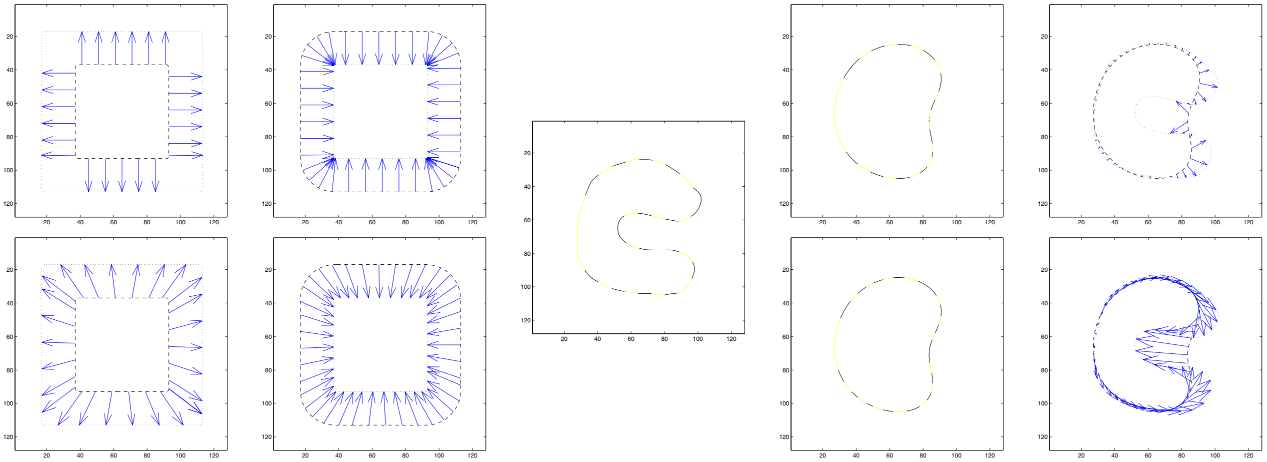

Figure 5. Some 2D evolutions with (bottom) and without (top) an area preserving tangential velocity.

References

[1] D. Adalsteinsson and J. Sethian. Transport and diffusion of material quantities on propagating interfaces via level set methods. *Journal of Computational Physics*, 185(1), Feb. 2003.

[2] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis. On area preserving mappings of minimal distorsion. Preprint.

[3] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis. Laplace-Beltrami operator and brain surface flattening. *IEEE Transactions on Medical Imaging*, 18:700–711, 1999.

[4] M. Bertalmio, L. Cheng, S. Osher, and G. Sapiro. Variational problems and partial differential equations on implicit surfaces. *Journal of Computational Physics*, 174:759–780, Dec. 2001.

[5] M. Bertalmio, G. Sapiro, and G. Randall. Region Tracking on Surfaces Deforming via Level-Sets Methods. In M. Nielsen, P. Johansen, O. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 58–69. Springer, Sept. 1999.

[6] M. Delfour and J.-P. Zolésio. *Intrinsic differential geometry and theory of thin shells*. Quaderni, Scuola Normale Superiore (Pisa, Italy), to appear, 2000.

[7] M. P. DoCarmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, 1976.

[8] P. Dupuis, U. Grenander, and M. Miller. Variational problems on flows of diffeomorphisms for image matching. *Quarterly of Applied Math.*, 1998.

[9] G. Hermosillo, O. Faugeras, and J. Gomes. Unfolding the cerebral cortex using level set methods. In M. Nielsen, P. Johansen, O. Olsen, and J. Weickert, editors, *Scale-Space Theories in Computer Vision*, volume 1682 of *Lecture Notes in Computer Science*, pages 58–69. Springer, Sept. 1999.

[10] G.-S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM Journal of Scientific Computing*, 21(6):2126–2143, 2000.

[11] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202–228, 1996.

[12] R. LeVeque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal of Numerical Analysis*, 33:627–665, 1996.

[13] K. Mikula and D. Sevcovic. Evolution of plane curves driven by a nonlinear function of curvature and anisotropy. *SIAM Journal of Applied Mathematics*, 61(5):1473–1501, 2001.

[14] S. Osher and R. P. Fedkiw. *Level set methods : overview and recent results*. Tutorials on Geometrically Based Motion, IPAM, Ucla, Los Angeles, 2001.

[15] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed: algorithms based on the Hamilton–Jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[16] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *Journal on Computational Physics*, 155(2):410–438, 1999.

[17] J.-P. Pons, G. Hermosillo, R. Keriven, and O. Faugeras. How to deal with point correspondences and tangential velocities in the level set framework. Technical Report 4857, INRIA, June 2003.

[18] J. Rauch. *Partial Differential Equations*. Springer-Verlag, New York, 1991.

[19] J. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Sciences*. Cambridge Monograph on Applied and Computational Mathematics. Cambridge University Press, 1999.

[20] A. Trouvé. Diffeomorphisms groups and pattern matching in image analysis. *International Journal of Computer Vision*, 28(3):213–21, 1998.

[21] B. Vemuri, J. Ye, Y. Chen, and L. C.M. A level-set based approach to image registration. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 86–93, Hilton Head, South Carolina, June 2000.

[22] J. Xu and H. Zhao. An Eulerian formulation for solving partial differential equations along a moving interface. Technical Report 02-27, UCLA Computational and Applied Mathematics Reports, May 2002.