A Physically-Motivated Deformable Model Based on Fluid Dynamics

Jalba, Andrei C.; Roerdink, Jos B.T.M.

Published in:
COMPUTER VISION - ECCV 2006 , PT 1, PROCEEDINGS

A Physically-Motivated Deformable Model Based on Fluid Dynamics

Andrei C. Jalba and Jos B.T.M. Roerdink

Institute for Mathematics and Computing Science,
University of Groningen, P.O. Box 800,
9700 AV, Groningen, The Netherlands
{andrei, roe}@cs.rug.nl

Abstract. A novel deformable model for image segmentation and shape recovery is presented. The model is inspired by fluid dynamics and is based on a flooding simulation similar to the watershed paradigm. Unlike most watershed methods, our model has a continuous formulation, being described by two partial differential equations. In this model, different fluids, added by placing density (dye) sources manually or automatically, are attracted towards the contours of the objects of interest by an image force. In contrast to the watershed method, when different fluids meet they may mix. When the topographical relief of the image is flooded, the interfaces separating homogeneous fluid regions can be traced to yield the object contours. We demonstrate the flexibility and potential of our model in two experimental settings: shape recovery using manual initializations and automated segmentation.

1 Introduction

A central problem in computer vision is image segmentation, the process of partitioning of an image into several constituent components. Among many segmentation techniques, deformable models, introduced in the 2.5-D case by Terzopoulos [1], specialized to the 2-D case by Kass *et al.* [2], and generalized to the 3-D case by Terzopoulos *et al.* [3], found applications in medical imaging (see [4, 5] for recent surveys), geometric modeling, computer animation, texture segmentation and object tracking. More recently, deformable models based on the level set framework [6, 7] have become extremely popular, since they can handle complicated topologies of the underlying shapes, unlike parametric snakes [2, 8].

Fluid models have been previously developed for medical image registration by Christensen *et al.* [9] and by Bro-Nielsen *et al.* [10]. Although Jain *et al.* [11] point out the connections between Christensen's work and active contours, the idea does not seem to have been explored in detail. Therefore, it is one of the goals of this paper to present a novel, physically-motivated deformable model for image segmentation and shape recovery based on a fluid simulation. The segmentation process implied by our method can be regarded as a flooding simulation of the topographical relief model of the gradient-magnitude image, similar to the

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part I, LNCS 3951, pp. 496–507, 2006.

watershed paradigm [12, 13]. Similar to the watershed from markers, several density (dye) sources, placed automatically or manually, can be thought of as locations where the relief was pierced, and different fluids can enter as the relief is flooded. In contrast to the watershed method, when different fluids meet they are allowed to mix; this may happen at locations with weak response of the gradient operator. However, at locations with high gradient magnitudes, the advancing fluid-fronts will not mix. When the relief is completely flooded, the interfaces separating fluid regions with different densities can be traced to yield the contours of the objects present in the input image.

The proposed method needs an initialization step, but this step is less critical than in the active contour model, rendering the method suitable for automated segmentation. Unlike the active contour model, our model allows dye sources to be placed entirely inside an object, outside on one side of the object, or crossing over parts of boundaries. In contrast to attractive forces based on the squared gradient-magnitude image [2] which act only in small vicinities along boundaries of objects, the image force in our model exhibits increased capture range because of its long range attraction, and enhanced robustness against boundary leakage. Unlike the watershed method which needs to address the problem of severe over-segmentation, in our model this problem is dealt with intrinsically.

2 Formulation of the Proposed Deformable Model

2.1 Model Formulation

Mathematically, in the Eulerian (grid based) formulation, fluids are described by a velocity field $\mathbf{u}$, a density field ρ and a pressure field p. The evolution of these quantities is governed by the Navier-Stokes equations [14]

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0\,, \tag{2}$$

where ν is the kinematic viscosity. The velocity causes the fluid to transport (advect) objects, densities, and other quantities along with the flow; in fact, the velocity of a fluid also carries itself; this is represented by the first term on the right-hand side of Eq. (1). The second term of Eq. (1), the pressure term, appears when an external force is applied to a fluid. The viscosity of a fluid measures the resistance of the fluid to the flow. In Eq. (1) viscosity is represented by diffusion of the velocity field. These equations have to be supplemented with boundary conditions, and here we will assume that the fluid lies in some bounded domain.

The Navier-Stokes equations can be adapted for image segmentation by (i) providing suitable external (image) forces, denoted by $\mathbf{F}$, which attract the fluid to the boundaries of the objects of interest; (ii) providing (manually or automatically) density (dye) sources, S_ρ, and (iii) defining appropriate initial and boundary conditions. In addition, we modify the equation for conservation of

momentum by dropping the pressure term and by adding an additional damping term, such that the modified equation becomes

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu \nabla^2 \mathbf{u} + \beta \nabla(\nabla \cdot \mathbf{u}) + \mathbf{F}\,, \tag{3}$$

where β is the dynamic-viscosity coefficient of the fluid (see below). The rationale for removing the pressure term is that it is expensive to compute, since it involves solving a Poisson equation [14,15]. The new term in Eq. (3), the gradient of the divergence of the velocity field, is better suited for our purposes. Since $\nabla \cdot \mathbf{u}$ represents net change in velocity across a small region of space, following the gradient of this change tends to restore the initial velocity. Hence, it does act as a damping term, having a regularizing effect during the flow. Note that in our formulation we drop the incompressibility requirement, *i.e.*, Eq. (2).

The method should be able to represent objects (or boundaries) during and at the end of the simulation. Moreover, it should be possible during initialization to place density (dye) sources manually or automatically (similar to markers in watershed segmentation). Therefore, we supplement Eq. (3) by an additional equation for a density (dye) moving through the velocity field

$$\frac{\partial \rho}{\partial t} = -(\mathbf{u} \cdot \nabla)\rho + S_\rho\,, \tag{4}$$

where S_ρ denotes density sources. Note that these quantities are only carried along (advected) by the fluid, and they do not affect its flow.

One still needs to devise suitable initial and boundary conditions. Analogous to our flooding paradigm, the initial fluid velocity is set to zero, *i.e.*, $\mathbf{u}(\mathbf{x}, t = 0) = \mathbf{u}_0(\mathbf{x}) = 0$, the initial density is set to some small constant ρ_0, $\rho(\mathbf{x}, t = 0) = \rho_0$. The same initialization is used at each grid location. Further, we assume that the resolution of the computational grid equals that of the input image. Then, dye sources S_ρ with equal density ($S_\rho = \rho_1$, with $\rho_1 > \rho_0$) are provided, which stop adding dye to the flow after a few time steps. An obvious choice for the external force $\mathbf{F}$ is some measure of the gradient of the input image I, such that the dye is attracted towards the contours of the objects of interest.

In our model sources can be placed far away from object boundaries, due to *self-advection* of velocities, which results in a *long-range attractive field*. The advantage of long-range attractive fields over attractive fields based on the squared gradient-magnitude image [2], is their increased capture range. Similar attractive fields have already been successfully used within the context of deformable models, see for example [16,17].

External forces. A suitable external force which guides the advancing fronts towards object boundaries is given by

$$\mathbf{F} = \mathbf{F_{img}} + \mathbf{F_{st}} + \mathbf{F_{ct}} = \nabla(|\nabla I_\sigma|) - \kappa \mathbf{n} - \alpha \mathbf{n}\,, \tag{5}$$

where $I_\sigma = I * G_\sigma$ denotes the input image, regularized by convolution with a Gaussian kernel of width σ, κ and $\mathbf{n}$ are the curvature and the unit normal at the

interface S between fluids with different densities, and $\alpha \in (0, 1]$ is a constant weight. The term $\mathbf{F_{img}}$ represents the image force and attracts the dye towards object boundaries. The term $\mathbf{F_{st}}$ represents *surface tension* and ensures that homogeneous dye regions have smooth boundaries, since curvature is minimized. During the flow this term has a stabilizing effect and improves the behaviour of the model with respect to *boundary leakage*, a problem frequently encountered with active contours based on the level set formalism, see [17, 16, 6, 7]. The last term, $\mathbf{F_{ct}}$ plays the role of a pressure term, spreading the dye at constant speed, thus correcting the problem with densities being advected to the nearest object boundaries, see subsection 2.1. Note that this pressure force acts only at the interface between fluids, unlike the pressure force in Eq. (1). Similar pressure terms have been previously proposed both in the contexts of parametric active contours [18, 16] and level sets [6, 7]. All terms in Eq. (5) are scaled and/or normalized, such that they have the same magnitude.

Boundary conditions. For densities, Eq. (4), we simply assume continuity at the boundaries of the computational grid, *i.e.*, $\partial\rho/\partial\mathbf{n} = 0$. For velocities at the boundaries of the computational grid we use the so-called *no-slip* boundary condition [14], *i.e.*, $\mathbf{u} = 0$ at these locations. We could address the problem with the fluid spreading over object boundaries by defining boundary conditions similar to the no-slip condition. However, since the method should be also usable with grey-scale images, we use the rule

$$\mathbf{u} \leftarrow \mathbf{u}\, e^{-\gamma|\nabla I_\sigma|}, \tag{6}$$

where $\gamma > 0$ is a constant parameter controlling the "strength" of the stopping criterion used to update velocities at each time step of the simulation. According to this rule, the velocity is decreased exponentially in the presence of large image gradients, *i.e.*, near object boundaries. This rule along with the damping and viscosity terms of Eq. (3) greatly improves the behaviour of the model with respect to boundary leakage, see section 3.

2.2 Relation to Active Contours and Watersheds

Consider the interface S as a parametric curve that deforms in time, *i.e.*, $\mathbf{X}(s,t) = [x(s,t), y(s,t)]$ with $s \in [0,1]$ is the arc-length parameter and t is the time. The dynamics of the curve is described by Newton's law,

$$\mu\frac{\partial^2\mathbf{X}}{\partial t^2} + \gamma\frac{\partial\mathbf{X}}{\partial t} + \mathbf{F}_{int} = \mathbf{F}_{ext}, \tag{7}$$

where μ is mass density, γ is viscosity (damping) coefficient, and $\mathbf{F}_{int}$ and $\mathbf{F}_{ext}$ are internal and external forces. Since at the interface the forces are $\mathbf{F}_{int} = -(\alpha+\kappa)\,\mathbf{n}$ and $\mathbf{F}_{ext} = \nabla(|\nabla I_\sigma|)$, setting $\mu = 0$ in Eq. (7) as in [19], the equation of motion becomes

$$\gamma\frac{\partial\mathbf{X}}{\partial t} = (\alpha+\kappa)\,\mathbf{n} + \nabla(|\nabla I_\sigma|). \tag{8}$$

Note that since the particles are transported with the fluid, the advection equation, Eq. (4), can be discarded. Embedding the curve $\mathbf{X}$ as the zero level set of the scalar function $\phi(\mathbf{x}, t)$, the evolution becomes

$$\gamma \frac{\partial \phi}{\partial t} = (\alpha + \kappa) \, |\nabla \phi| + \nabla(|\nabla I_\sigma|) \cdot \nabla \phi. \tag{9}$$

The first three force terms from Eq. (3) (convection, diffusion and damping) have been ommited here since they act on the velocity field $\mathbf{u}$ in the proposed method. However, these terms have beneficial effects in our model (*e.g.* long-range attraction due to convection of velocities, regularization, etc.), which are lost in the level-set formulation from Eq. (9). Moreover, the flexibility in initialization will be lost and boundary-leakage problems specific to level sets will appear.

If we neglect the second term from Eq. (9), set $\kappa = 0$ and let $\alpha = \frac{1}{|\nabla I_\sigma|} \geq 0$, the level-set motion equation of the interface becomes

$$\gamma \frac{\partial \phi}{\partial t} = \frac{1}{|\nabla I_\sigma|} \, |\nabla \phi| \, , \tag{10}$$

which is a continuous formulation of the watershed method based on the Eikonal equation. Further, if sources (markers) are placed at regional minima of the gradient-magnitude image, the evolution becomes similar to the watershed from markers.

2.3 Method of Solution

Equations (3) and (4) can be solved using the *stable fluids* technique for solving Navier-Stokes equations developed by Stam [15]. The advantage of his technique is that it is easy to implement, allows the user to interact in real-time with three-dimensional simulations of fluids, and is stable, allowing for large time steps during the numerical integration. Although the model may not be accurate enough for certain engineering applications, its accuracy suffices for our purposes.

Eq. (3) is solved using a time step Δt. Assuming that the field is known at time t and we wish to advance the solution at time step $t + \Delta t$, we resolve Eq. (3) over the time step Δt in five steps. That is, the solution is found by composition of transformations on the state, *i.e.*, each transformation is a step that takes a field as input and produces a new field as output.

Force application. The gradient magnitude of the regularized image (see Eq. (5)) is computed using the Sobel operator. It was chosen because it is a difference-of-averages operator and because its response to diagonal edges is better than that of other operators such as the Prewitt operator. The surface-tension force, $\mathbf{F_{st}} = -\kappa \mathbf{n}$, is computed based on the following result. If the location of the interface at time t is given by a level set function $S(\mathbf{x}, t) = 0$, then its temporal evolution follows from

$$\frac{DS}{Dt} \equiv \frac{\partial S}{\partial t} + (\mathbf{u} \cdot \nabla) S = 0 \, , \tag{11}$$

where DS/Dt is the material (advective) derivative. This equation simply states that the interface propagates with the fluid velocity. Comparing this equation with Eq. (4) it follows that the level sets $S(\mathbf{x}, t)$ can be tracked through the motion of densities by considering the graph $z = \rho(\mathbf{x}, t)$ and defining the interface in terms of level sets S as $S(\mathbf{x}, z, t) \equiv \rho(\mathbf{x}, t) - z = 0$. Then the normal to the interface, $\mathbf{n}$, can be computed by evaluating $\mathbf{n} \equiv \frac{\nabla S}{|\nabla S|}$, whereas the curvature is computed using $\kappa \equiv \nabla \cdot \mathbf{n}$. Finally, the pressure force, $\mathbf{F_{ct}}$, is evaluated using the previously computed normal, and the force application operator is given by

$$\mathbf{u}(\mathbf{x}, t + \Delta t) = \mathbf{u}(\mathbf{x}, t) + \Delta t\, \mathbf{F}(\mathbf{x}, t)\,. \tag{12}$$

Here we assume that the external force does not vary considerably during the time step Δt. Also, since the application of boundary conditions (see Eq. (6)) does not amplify the magnitudes of the velocity vectors, a stability requirement is that the total external force from Eq. (5) is bounded, which holds in our case.

Damping. Instead of solving the following equation $\frac{\partial \mathbf{u}}{\partial t} = \beta\, \nabla(\nabla \cdot \mathbf{u})$, to account for damping in Eq. (3), we simply consider the term $\beta\, \nabla(\nabla \cdot \mathbf{u})$ as a *damping force* and add its contribution to the velocity field as already done for the external force in Eq. (12).

Advection. The trajectory of the particle is traced back in time from each grid cell to its former position. Then the quantity q is copied from this position to the starting grid cell using some interpolation scheme. More formally, to update quantity q, the following equation is used

$$q(\mathbf{x}, t + \Delta t) \leftarrow q(\mathbf{x} - \Delta \mathbf{x}, t) = q(\mathbf{x} - \mathbf{u}(\mathbf{x}, t)\, \Delta t, t)\,. \tag{13}$$

As shown by Stam [15], the advantage of this method is that it results in an *unconditionally stable* advection solver.

Diffusion. The diffusion of velocity is modeled according to $\frac{\partial \mathbf{u}}{\partial t} = \nu \nabla^2 \mathbf{u}$. An obvious approach for solving this equation is to formulate an explicit, discrete form similar to Euler's method for integration of ordinary differential equations, see [20]. This method becomes unstable for large values of Δt and ν. Therefore, we prefer to use Stam's implicit method and solve the following equation

$$(\mathbf{I} - \nu \Delta t \nabla^2)\mathbf{u}(\mathbf{x}, t + \Delta t) = \mathbf{u}(\mathbf{x}, t)\,, \tag{14}$$

where $\mathbf{I}$ is the identity matrix. This formulation is stable for arbitrary time steps and viscosity coefficients.

2.4 Visualization

To steer the computations and to gain insight into the segmentation problem the user can interact with the simulation, for example, by adding or removing dye sources, adjusting parameters, etc., rendering the method suited for interactive segmentation. On the other hand, if the very purpose is automated segmentation, we will show in Section 3 that simple automatic initializations are also

possible. To enable performing interactive segmentation, some method is needed to visualize either the concentration of the dye or the interfaces between different fluids. Since densities are readily available and can be easily visualized, we will only describe a method for tracing the interfaces between different fluids.

Using the density field ρ, for each grid cell $G_i(x_i, y_i)$ a weighted sum is computed at each corner of the current cell, *i.e.*, at locations (x_i, y_i), (x_{i+1}, y_i), (x_i, y_{i+1}), (x_{i+1}, y_{i+1}). At each of these corners, the sum is computed by adding the densities of the (4-connected) neighbouring locations, weighted by forward and backward finite absolute differences between the current location and its neighbours; the resulting density value is normalized by dividing it by the sum of weights. Then, the average of the four density values, obtained by weighted summation at each corner of grid cell $G_i(x_i, y_i)$, is computed to obtain an estimate for the variation of the density $\rho_{a,i}$ inside the current cell. Finally, if the resulting value, $\rho_{a,i}$, is greater than zero, meaning that there is an interface between two fluids inside the current cell, it is traced using a 2-D polygonization method similar to the Marching Cubes algorithm [21]. All contours (fluid interfaces) found are then drawn superimposed on the input image.

3 Results

We will show several results on binary and grey-scale images obtained using the proposed deformable model. In particular, we will show that the new deformable model is (i) robust with respect to boundary leakage, (ii) insensitive to initialization, (iii) robust against noisy conditions, and (iv) can be used to perform automated segmentation.

Boundary leakage. The behaviour with respect to boundary leakage can be controlled by adjusting the viscosity parameter, ν, from Eq. (3), see Fig. 1. When setting $\nu = 1$, the fluid does not penetrate the thinner gaps of the object present in the image. In the remainder, we fix this value to $\nu = 0.1$.

Robustness to initialization. An important advantage of the proposed method over active contours is that it allows simple initializations, see Fig. 2. The initializations in Fig. 2 are very difficult (if not impossible) to handle by most snake methods, because some sources are placed inside objects, while others are outside. Besides, level-set snakes only accept closed contours. These initializations

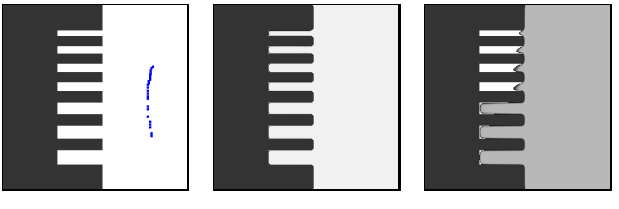

Fig. 1. *Left*: Initialization superimposed on the initial image; *center*: no viscosity, $\nu = 0$; *right*: $\nu = 1.0$

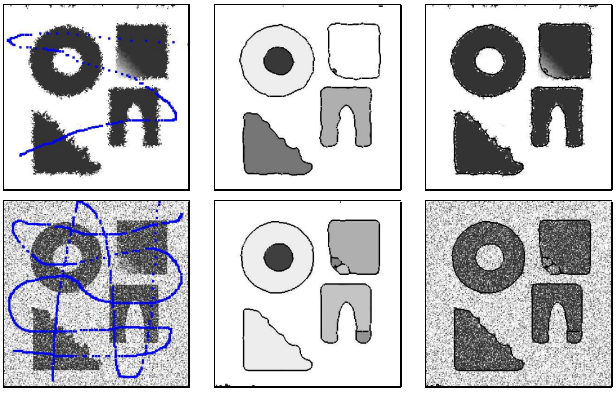

Fig. 2. *Left*: Initializations; segmentation results: *center* – dye regions with superimposed interfaces (boundaries), *right* – original images with superimposed contours

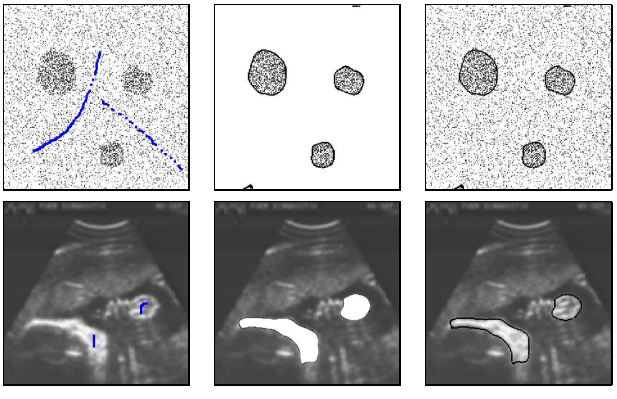

Fig. 3. Segmentation results on noisy images. *Left*: Initializations; segmentation results: *center* – dye regions with superimposed interfaces (boundaries), *right* – original images with superimposed contours.

pose also problems for the watershed method. As the number of markers does not change during the watershed evolution, a marker region lost during marker selection cannot be recovered later. Alternatively, more markers in one object result in over-segmentation.

Robustness to noise. The results in Fig. 3 show that the method copes quite well with respect to noisy images. As observed, the method is able to recover correctly the major shapes of the objects present in these noisy images.

Segmentation of medical images Results on several medical images are shown in Figs. 4 and 5. Although the image shown in the first figure (angiogram) is

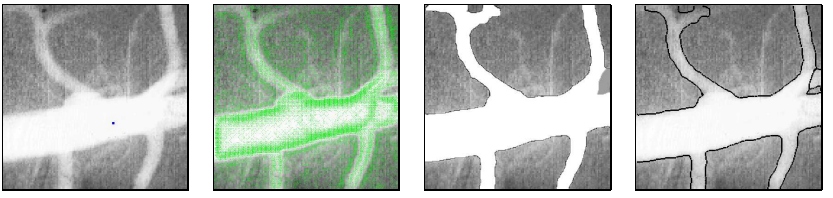

Fig. 4. *Left-to-right*: input image and initialization (a dot), velocity field **u**, detected regions, contours superimposed on the original image

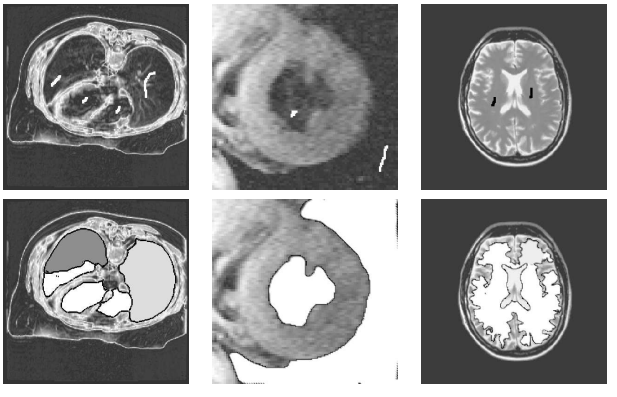

Fig. 5. *First row*: initializations (*left*, *center*: white segments, *right*: black segments); *second row*: results (regions)

quite noisy and difficult because of the thin and elongated structures, the method yields quite a good result, being able to recover the whole artery. Most important structures have been also correctly recovered for the objects in Fig. 5.

Automated segmentation. Our next experiment is automatic segmentation using a trivial automated method for initialization. To contrast our method to the watershed from markers, we placed dye sources at *all local minima* of the gradient magnitude image. This type of initialization usually results in over-segmentation using watershed from markers. However, this initialization poses no problem for our method, see Figs. 6 and 7. Since weak gradients do not stop the fluid from flowing, fluids with different densities will mix, but near equilibrium, different objects/regions reach different yet homogeneous densities. Note that, all major structures present in these images were correctly identified.

Missing and fuzzy boundaries. Although the proposed method can segment objects with missing boundaries (as shown in Fig. 1), the fluid may flow through larger gaps. A solution to this problem would be to integrate region-based energy terms in our model, similar to the minimal-variance term proposed by Chan and Vese [22]; such extensions are the subject of ongoing research. Our method can segment objects with fuzzy boundaries, provided that fluid sources are placed both inside and

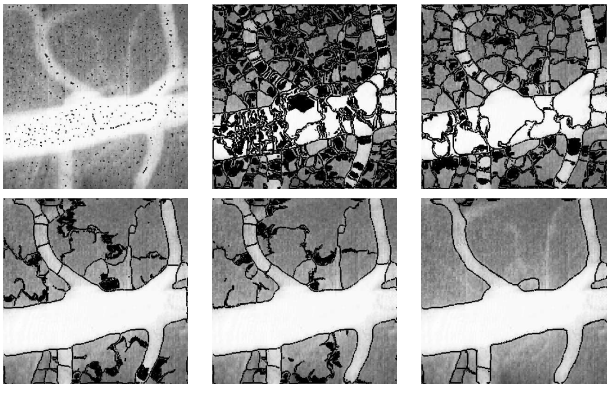

Fig. 6. *Left-to-right, top-to-bottom*: Successive snapshots

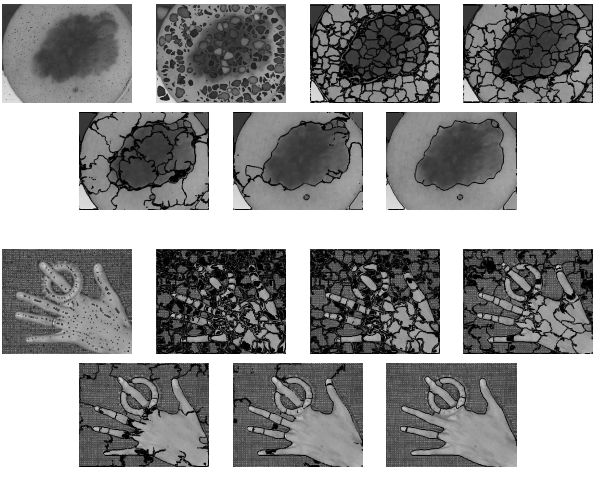

Fig. 7. *Left-to-right, top-to-bottom*: Successive snapshots

outside the object, see Fig. 8. However, when fluid sources are placed only outside (or inside) the object, the method may fail. We also performed experiments using the geodesic snake [6,7] augmented by the GVF field [16] to attract the contour towards the object boundaries, see Fig. 8, second row. Note that the geodesic snake fails to detect the boundary of the object even when the GVF field is used.

Parameter settings. In all our experiments we used the following parameter values: $\alpha = 0.2$, $\beta = 1.0$, $\gamma = 2.0$. Only the value of the viscosity parameter ν was adjusted when performing the boundary-leakage experiment. For the remaining experiments we also fixed the value of this parameter to $\nu = 0.1$. This indicates that the parameter setting of the proposed method is not critical.

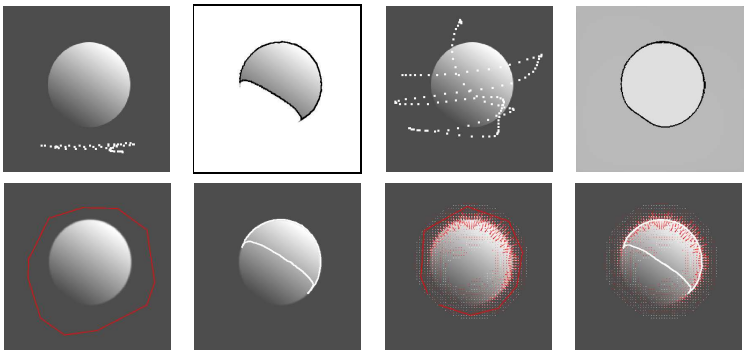

Fig. 8. Fuzzy boundaries. *First row*: Two different initializations and results by the proposed method; *second row*: results by the geodesic snake w/o attractive GVF field.

4 Conclusions

We have introduced a novel, physically-motivated deformable model for image segmentation, and demonstrated its flexibility and potential for shape recovery using manual initializations as well as automated segmentation. The proposed deformable model exhibits several important characteristics: (i) insensitivity to initialization, as opposed to snakes and watersheds, (ii) increased capture range of the attractive vector field, (iii) handling of topological changes, and (iv) good behaviour on noisy images. An important advantage is that the method can be used to perform automatic segmentation, with trivial initializations.

References

1. Terzopoulos, D.: Image analysis using multigrid relaxation methods. IEEE Trans. Pattern Anal. Machine Intell. **8** (1986) 129–139
2. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. J. Comput. Vis. **1** (1987) 321–331
3. Terzopoulos, D., Witkin, A., Kass, M.: Constraints on deformable models: Recovering 3D shape and nonrigid motion. Artificial Intelligence **36** (1988) 91–123
4. McInerney, T., Terzopoulos, D.: Deformable models in medical image analysis: a survey. Medical Image Analysis **1** (1996) 91–108
5. Suri, J., Liu, K., Singh, S., Laxminarayan, S., Zeng, X., Reden, L.: Shape recovery algorithms using level sets in 2-D/3-D medical imagery: A state of the art review. IEEE Trans. on Inf. Tech. in Biomed. **6** (2002) 8–28
6. Malladi, R., Sethian, J.A., Vemuri, B.C.: Shape modeling with front propagation: A level set approach. IEEE Trans. Pattern Anal. Machine Intell. **17** (1995) 158–175
7. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. 5th Int. Conf. Computer Vision. (1995) 694–699
8. Terzopoulos, D., Platt, J., Barr, A., Fleischer, K.: Elastically deformable models. ACM Comput. Graph. **21** (1987) 205–214

9. Christensen, G.E., Ribbitt, R.D., Miller, M.I.: Deformable templates using large deformation kinematics. IEEE Trans. Image Processing **5** (1996) 1435–1447
10. Bro-Nielsen, M., Gramkow, C.: Fast fluid registration of medical images. In: Proc. of the 4th International Conference VBC '96, London, UK, Springer-Verlag (1996) 267–276
11. Jain, A.K., Zhong, Y., Dubuisson-Jolly, M.P.: Deformable template models: A review. Signal Processing **71** (1998) 109–129
12. Meyer, F., Beucher, S.: Morphological segmentation. J. Visual Commun. and Image Repres. **1** (1990) 21–46
13. Roerdink, J.B.T.M., Meijster, A.: The watershed transform: Definitions, algorithms and parallelization strategies. Fundamenta Informaticae **41**(1-2) (2000) 187–228
14. Chorin, A.J., Marsden, J.E.: A mathematical introduction to fluid mechanics. Second edn. Texts in Applied Mathematics 4. Springer-Verlag (1990)
15. Stam, J.: Stable fluids. In: SIGGRAPH'99, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co. (1999) 121–128
16. Xu, C., Prince, J.L.: Snakes, shapes, and gradient vector flow. IEEE Trans. Pattern Anal. Machine Intell. **7**(3) (1998) 359–369
17. Jalba, A.C., Wilkinson, M.H.F., Roerdink, J.B.T.M.: CPM: A deformable model for shape recovery and segmentation based on charged particles. IEEE Trans. Pattern Anal. Machine Intell. **26** (2004) 1320–1335
18. Cohen, L.D.: On active contour methods and balloons. CVGIP: Image Understanding **53** (1991) 211–218
19. Cohen, L., Cohen, I.: Finite-element methods for active contour models and balloons for 2-D and 3-D images. IEEE Trans. Pattern Anal. Machine Intell. **15**(11) (1993) 1131–1147
20. Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: Numerical Recipes in C: The Art of Scientific Computing. Cambridge Univ. Press, Cambridge (1988)
21. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3D surface construction algorithm. In: SIGGRAPH'87. (1987) 1631–169
22. Chan, T., Vese, L.: Active contours without edges. IEEE Trans. Image Processing **10** (2001) 266–277