A Fast Multigrid Implicit Algorithm for the Evolution of Geodesic Active Contours

George Papandreou and Petros Maragos

School of Electrical and Computer Engineering
National Technical University of Athens
Zografou 15773, Athens, Greece
E-mail: {gpapan,maragos}@cs.ntua.gr

Abstract

Active contour models are among the most popular PDE-based tools in computer vision. In this paper we present a new algorithm for the fast evolution of geodesic active contours and compare it with other established numerical schemes. The new algorithm employs a full time-implicit and unconditionally stable numerical scheme and applies multigrid methods for the efficient solution of the occurring sparse linear system. When we utilize very big time-steps for the numerical evolution of the front, the proposed scheme has increased accuracy and better rotational invariance properties compared with the alternative AOS scheme. This allows for the rapid evolution and convergence of the contour to its final configuration after only very few iterations. Standard pyramidal and/or narrowband techniques can be easily integrated into our algorithm and further accelerate the curve evolution. Experimental results in object boundary detection demonstrate the power of the method.

1. Introduction

Active contour models (also called snakes), are one of the most important tools in computer vision. They were introduced by Kass, Witkin and Terzopoulos [10] in 1987 and have been subsequently used heavily for machine vision tasks such as object boundary detection and moving object tracking [3]. In an active contour model an initial contour evolves on the image plane towards the object boundaries. This is achieved by minimizing an image-dependent energy functional of the contour, designed specifically so that the contour corresponding to a (local) minimum of the functional coincides with the desired object boundaries. Regularization of the contour is achieved by additional energy terms which favour the smoothness of the curve and limit the bending effect. The whole model is formulated in a variational framework and the minimization of the functional is usually accomplished by steepest descent techniques, which lead to curve evolution governed by a partial differential equation (PDE).

Despite its success, the original active contour model has some noticeable drawbacks. First of all, it depends not only on the intrinsic properties of the contour but also on its parameterization, which means that it is a non-geometric model. Secondly it cannot naturally handle changes in the topology of the evolving contour, which are necessary in situations where no a priori knowledge of the number of objects to be detected is available.

These drawbacks of standard active contours were addressed by geometric active contour models, which were introduced in [5] and their further development led to the geodesic active contours model [6], [23]. According to the geodesic active contours model, an initial curve evolves towards the minimum of an energy functional, which in this case is a geodesic in a Riemannian manifold endowed with a metric induced by image features. The geodesic active contours model is thus parameterization-independent. In order for it also to be topologically adaptable, level set ideas [13] are utilized. Apart from object boundary detection, this model has been successfully employed in other problems in vision, such as moving object tracking [14].

Although the geodesic active contours model implemented in the level set framework has many improvements over classic snakes, these improvements come at higher computational cost, which renders its utilization for time-critical applications problematic. In order to limit computations in the neighbourhood of the evolving contour, narrowband techniques in conjuction with reinitialization techniques borrowed from the level set technology [2], [19], [15] have been used [7]. Adopting a pyramidal approach can lead to further improvement. For the purpose of removing the stability constraint on the size of the time-step associated with the simple explicit numerical scheme, the addi-

tive operator splitting (AOS) scheme [22] has been adapted to the problem of geodesic active contours [7]. Although no stability constraint on the time-step is present when the AOS scheme is utilized, the size of the time-step cannot be very large because, as we will see, splitting artifacts associated with loss of rotational invariance will emerge. The practical implication of this is that the number of iterations needed for the contour to converge remains quite large.

The main contribution of our paper is that the algorithm we propose remains sufficiently accurate and maintains its rotational invariance even when very big time-steps are utilized. This allows for the rapid evolution and convergence of the contour to its final configuration after only very few iterations. The new algorithm is based on the full-matrix version of the implicit scheme and is unconditionally stable as far as the size of the time-step is concerned. In the core of our algorithm is the efficient solution of a big sparse linear system which occurs at each time-step. For this purpose we apply multigrid methods [20], which were also used recently in a similar context in [1] and [11]. Although the cost per iteration of the multigrid solver is now bigger than in the case of the AOS scheme, the overall time of evolution is now typically smaller due to the drastically reduced number of iterations needed for the convergence of the evolving front. Standard pyramidal and/or narrowband techniques can naturally fit into our algorithm and further accelerate the curve evolution. Experimental results in object boundary detection demonstrate the applicability and performance of the method.

Although multigrid techniques have been applied for the solution of the geodesic active contours problem in [11], our treatment is novel. In [11] a completely different discretization scheme is utilized, which leads to a nonlinear system of equations per timestep, whose solution by a nonlinear multigrid solver might be complicated, requiring the evolution of the contour on a denoised version of the image (cf. Subsection 3.3). In our work the reinitialization of the embedding function before every new step leads to a linear system which is easy to solve efficiently with multigrid techniques. We additionally make a thorough comparison between the multigrid approach and the AOS technique, illustrating each one's shortcomings and applicability.

2. Geodesic active contours model

Let $C(q) \equiv (x(q), y(q)) : [0,1] \rightarrow R^2$ be a planar curve with parameter q and $I : [0,a] \times [0,b] \rightarrow R^+$ be the intensity image on which we would like to detect the object boundaries. The image-dependent non-parametric curve functional to be minimized in the geodesic active contours model is (see [6]):

$$L_R = E(C) = \int_0^{L(C)} g(\|\nabla I(C(s))\|)ds, \qquad (1)$$

where s is the arc-length parameter, the edge indication function $g : [0,+\infty) \rightarrow R^+$ is a decreasing function such that $g(0) = 1$, $g(r) \rightarrow 0$ while $r \rightarrow +\infty$, $L(C)$ is the Euclidean length of the contour C and L_R is the geodesic length of the contour induced by the new metric $g(\|\nabla I\|)$, which depends on the image and the form of the function g.

The minimization of the functional in eq. (1) by means of variational techniques leads to an Euler-Lagrange PDE. One way to reach (local) minima is to start with an initial contour and evolve it in the direction of steepest descent, introducing an artificial time variable. In order to allow the curve to split and merge freely, the numerical implementation of the geodesic active contours model needs to easily deal with changes in the topology of the contour. This can be naturally achieved utilizing level set techniques [13]. In the level set framework the contour is defined implicitly by means of an embedding scalar function $u(x,y,t) : R^2 \times [0,+\infty) \rightarrow R$, which has the whole image as domain. More specifically, the contour is defined as the zero level set of the embedding function, i.e. $C(t) = \{(x,y) : u(x,y,t) = 0\}$. The extension of u away from the contour is rather arbitrary, although the signed distance transform from the contour (by convention, we assign negative values in the interior and positive values in the exterior of the curve) is often chosen for its good numerical properties. The evolution of the contour is done implicitly as we evolve the embedding function under a suitable law. In the case of geodesic active contours this law can be shown to be [6]:

$$\frac{\partial u(x,y,t)}{\partial t} = \|\nabla u\| \mathrm{div}\Big(g\frac{\nabla u}{\|\nabla u\|}\Big) + cg\|\nabla u\|, \qquad (2)$$

where t is the pseudo-time variable. In the last equation we have added a "balloon" force term which acts like erosion/dilation, favouring the inward (if $c > 0$) or the outward (if $c < 0$) motion of the contour, respectively.

3. Numerical solution of the model

While the geodesic active contours model of eq. (2) has the advantage that it can easily deal with changes in the topology of the evolving contour, its straightforward implementation is typically less efficient than that of classic snakes. Improvements can be sought in two directions: a) Reduction of the workload per iteration by limiting the number of pixels on which we update the values of the embedding function u and b) Convergence of the contour in fewer iterations by adopting numerical schemes which stably and accurately allow the evolution of the contour with big strides. Our main focus is on the second direction by proposing an implicit numerical scheme and an efficient method to solve the occurring linear system. We then discuss how narrowband/pyramidal techniques can be com-

bined with the proposed algorithm in order to reduce the workload per iteration.

3.1. Discretization

We will first describe the discretization of eq. (2) without the balloon force term. We will separately discuss the inclusion of the balloon force term in Subsection 3.4.

In [7] Goldenberg et al. noticed the connection between the equation of geodesic active contours evolution and the diffusion equation. The first term in eq. (2) is of parabolic nature and describes motion under geodesic curvature. This can be seen clearly if we reinitialize the embedding function u to be a signed distance transform before each iteration, which implies $\|\nabla u\| = 1$. Then from eq. (2) for $c = 0$ we see that every step of the geodesic active contours evolution corresponds to linear non-homogenous diffusion according to the equation:

$$\frac{\partial u(x,y,t)}{\partial t} = \operatorname{div}\big(g(x,y)\nabla u\big), \tag{3}$$

where the conduction coefficient $g(x,y) = g(\|\nabla I\|)$ depends on the image I and not on the evolving function u. Reflecting boundary conditions are applied.

For the time derivative in eq. (3) a forward time difference can be used:

$$\frac{\partial u}{\partial t} \approx \frac{u^{n+1} - u^n}{\tau}, \tag{4}$$

where τ is the size of the time-step.

The right hand side parabolic term of eq. (3) can be approximated by central differences (see [22], [21]). For $h_x = h_y = 1$ this gives:

$$\begin{aligned}\operatorname{div}(g\nabla u)|_{ij} \approx\ & g_{i+\frac{1}{2},j}(u_{i+1,j} - u_{ij}) - g_{i-\frac{1}{2},j}(u_{ij} - u_{i-1,j}) \\ & + g_{i,j+\frac{1}{2}}(u_{i,j+1} - u_{ij}) - g_{i,j-\frac{1}{2}}(u_{ij} - u_{i,j-1})\end{aligned} \tag{5}$$

We can set $g_{i+\frac{1}{2},j} \approx (g_{i+1,j} + g_{ij})/2$ and use analogous expressions for the other similar terms.

If we number the pixels in a row-major order, we can write eq. (5) compactly in matrix-vector notation as:

$$\operatorname{div}(g\nabla u) \approx Au \tag{6}$$

where $A = [a_{ij}]$ is the $N \times N$ ($N = N_x N_y$ is the number of pixels of the image) time-independent matrix with elements:

$$a_{ij} = \begin{cases} \frac{g_i + g_j}{2} & j \in N(i) \\ -\sum_{k \in N(i)} \frac{g_k + g_i}{2} & j = i \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where $N(i)$ denotes the 4-neighborhood of pixel P_i. This matrix is extremely sparse, having only 5 non-zero elements per row (apart from the rows corresponding to pixels on the sides or corners of the image where it has 4 or 3 non-zero elements, respectively).

3.2. The proposed scheme

Combining the expressions in eqs. (4) and (6) we end up with the following general discretization formula:

$$(u^{n+1} - u^n)/\tau = Au \tag{8}$$

In eq. (8) we have not yet determined when the right hand side should be evaluated. The explicit scheme $(u^{n+1} - u^n)/\tau = Au^n$ is the most straightforward and easy to implement choice, but it is stable only for $\tau \leq 0.25$ [22]. This constraint is practically very restrictive, since it typically leads to the need for thousands of iterations before the contour converges to a geodesic.

On the other hand, one can prove that the semi-implicit scheme $(u^{n+1} - u^n)/\tau = Au^{n+1}$ is stable no matter how large the time-step is (see [22]). However at each iteration the following linear system needs to be solved:

$$[I - \tau A]u^{n+1} = u^n \tag{9}$$

The system matrix $I - \tau A$ is very large and it inherits the sparse structure of A described in Subsection 3.1. This specific sparse structure is not susceptible to the application of efficient elimination techniques (the matrix doesn't have small bandwidth). Simple iterative methods such as Jacobi or Gauss-Seidel converge slowly for such big systems. The number of iterations of these iterative methods required to reduce the error by a predefined factor is proportional to the number of pixels N [16]. Since the per iteration cost of these iterative methods is also $O(N)$, this leads to a $O(N^2)$ cost for each step of (9), which is quite prohibitive.

In order to avoid solving the full-blown system (9), the authors in [7] and [21] adopt the AOS scheme:

$$u^{n+1} = \frac{1}{2} \sum_{l \in \{x,y\}} [I - 2\tau A_l]^{-1} u^n \tag{10}$$

The AOS scheme is unconditionally stable and was first introduced in the context of non-linear diffusion [22]. It is a simplified version of the semi-implicit scheme (9) in which a 2-dimensional diffusion process is approximated as the average of two independent 1-dimensional ones. The advantage of this approach is that the two linear systems involved in eq. (10) are tridiagonal and can be solved efficiently (with cost $O(N)$ per step) by means of the Thomas algorithm. The disadvantage is that when the time-step τ gets very big, splitting artifacts emerge due to significant loss in rotational invariance. This constrains the size of the time-step and keeps the number of iterations needed for the contour to converge still large.

In the algorithm we propose, we choose to attack the semi-implicit scheme in its complete form (9). In order to efficiently solve the full-blown system (9) we resort to multigrid techniques, which are described briefly in Subsection 3.3. While the cost of each step is now larger, this is compensated by the fact that really big values of the time-step τ give results of good accuracy and rotational invariance. Using such large values for τ leads to the rapid evolution of the contour to its final configuration after very few iterations (typically less than 5, as we will see in Section 4).

We should briefly mention here that we also tested the behaviour of the Crank-Nicholson scheme (see e.g. [17]) $(u^{n+1}-u^n)/\tau = 0.5A(u^n+u^{n+1})$ in our problem . While the solution remains bounded no matter how large τ gets, undesirable oscillations emerge with big time-steps. Since a linear system analogous to (9) is involved in this case, too, the Crank-Nicholson scheme is not competitive as a candidate for the fast evolution of geodesic active contours and will not be examined any more in this paper.

3.3. Multigrid evolution of the contour

Multigrid methods (see e.g. [4], [8], [20]) are one of the most general tools for the efficient solution of large sparse linear systems that arise during the numerical solution of PDEs. They have already been used in a similar context in [1] and [11]. In many cases their complexity can be shown to be $O(N)$. We have experimentally verified this for the case of our problem, too.

The main drawback of standard iterative methods for the solution of linear systems (e.g. Jacobi, Gauss-Seidel) is that, while they eliminate the high-frequency (spatially speaking) part of the error quickly, they are particularly inefficient in the suppression of its low-frequency part. For these reasons these methods are called smoothers. The idea behind multigrid is to attack both high- and low-frequency components of the error in a multiresolution framework. Few iterations of a smoother on the finest grid will smooth the error. The residual is then restricted to a coarser representation of the problem where it appears more oscillatory because of the coarser sampling. This means that a new application of the smoother at this coarser scale (which also has reduced cost) effectively eliminates a lower-frequency part of the error. The process goes on recursively until we reach a scale coarse enough for Gaussian elimination to be applied. Then the error estimates computed at the coarser levels are prolonged and correct the solution at the finer levels, until the solution at the original resolution is reached.

For two levels this translates to the following multigrid cycle (the superscript $()^h$ denotes fine grid quantities while the $()^{2h}$ denotes coarse grid quantities):

1. Relax on $A^h u^h = f^h$ and then compute the residual $r^h = f^h - A^h u^h$ in the fine grid.
2. Restrict the residual $r^{2h} = Rr^h$ to the coarse grid.
3. Solve directly the system $A^{2h}e^{2h} = r^{2h}$ in the coarse grid.
4. Prolong the error to the fine grid $e^h = Pe^{2h}$.
5. Correct the solution at the fine grid $u^h \leftarrow u^h + e^h$.

In geometric multigrid one needs to define both the problem in all resolutions, specifying the corresponding system matrices, and the way the intergrid transfers are carried out, specifying the restriction/prolongation operators R and P. This is cumbersome when dealing with non-rectangular geometries, such as the narrowband case in our problem. Moreover, since the "diffusion coefficient" $g(\|\nabla I\|)$ is an abruptly varying function, the convergence rate for geometric multigrid with simple, matrix-independent restriction/prolongation operators can be poor, a phenomenon well studied in the multigrid community (see e.g. [20]). This might be the reason why [11] had to use a denoised version of the image, effectively smoothing $g(\|\nabla I\|)$.

Algebraic multigrid techniques [18], [20] are particularly convenient in this case because they require as input only the system matrix corresponding to the finest grid. The matrices for the coarser grids and the intergrid transfer operators are then computed automatically. Additionally, algebraic multigrid is very robust in the presence of discontinuous coefficients, which is our case. Therefore we have used algebraic multigrid in our implementation.

The proposed reinitialization of u to be a signed distance transform before every new step, apart from leading to the linearized eq. (3), has the additional advantage that the coefficients (7) of the system matrix A do not change over time. This means that the process of building up the hierarchy of grids in our multigrid solver needs only be done once, which is another important advantage of our approach in comparison to [11].

3.4. Details and extensions

One important extension of the model is to integrate narrowband techniques into it. This is straightforward since we utilize algebraic multigrid methods in our algorithm. Now the matrix A defined in (7) contains only rows corresponding to pixels in the narrowband and a pixel needs to be in the narrowband in order to belong to the neighborhood $N(i)$ of pixel P_i. The integration of pyramidal techniques is also straightforward and need not be further discussed here. Extension to the 3-dimensional case is also possible.

We examine next how the balloon force can be incorporated into the proposed algorithm. A simple idea is to perform the evolutions due to the two different terms of eq. (2) sequentially. Isolating the balloon force term from eq. (2), we get the hyperbolic equation $u_t = cg\|\nabla u\|$, corresponding to adaptive dilation/erosion [12]. In order to derive an

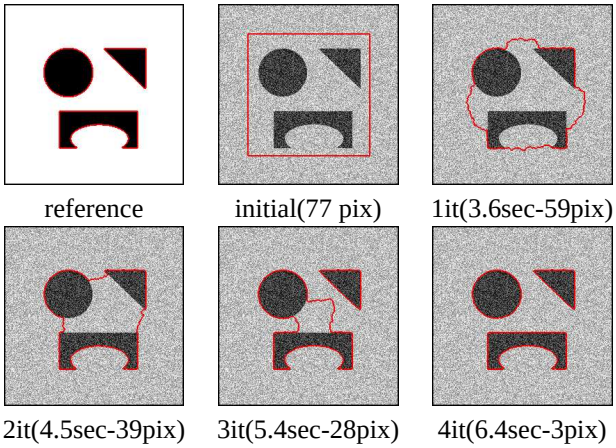

Figure 1. Example on a 256x256 noisy synthetic image with $\tau = 5 \times 10^4$ and $\sigma = 0.01$. In parentheses the execution time and the Hausdorff distance to the reference contour. Convergence after 4 iterations.

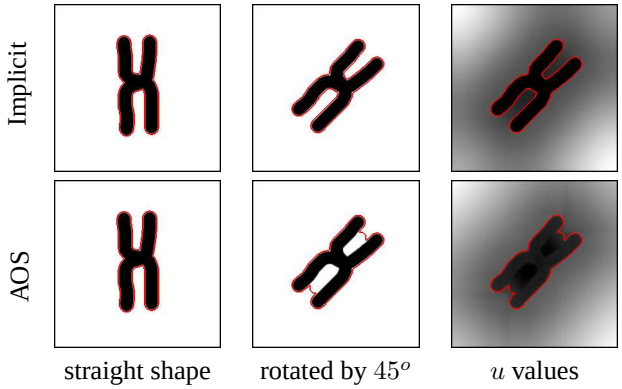

Figure 2. Rotational invariance test. The rotation has no effect on the result of the semi-implicit scheme. The AOS scheme with large τ gives different results after a 45^o rotation of the image. This anisotropicity is caused by the "shading effect" of the AOS scheme, better visualized in the graph of the embedding function (third column). The curve evolutions have converged with $\tau = 1000$ in both cases.

unconditionally stable numerical scheme for the evolution of the zero level set $C(t)$ under this law, we cast it in stationary form as $C(t) = \{(x, y) : T(x, y) = -ct\}$, where the function $T(x, y)$ satisfies the eikonal PDE $\|\nabla T\| g(x, y) = 1$. We then apply the fast marching algorithm [19] to efficiently compute T. The combined algorithm utilizes the fast marching algorithm twice per iteration: once for the evolution of the front under the balloon force and once for the reinitialization of u to be a signed distance transform. We should however note here that utilization of the balloon force with the big time-steps typically used in the context of our algorithm can be problematic, because the contour evolving under the balloon force may skip over and miss the object boundaries, something also noticed in [7].

As far as the reinitialization of u to be a signed distance transform is concerned, apart from the fast marching algorithm, we have also considered another method [15] which is based on the PDE $d_t + sign(d)(\|\nabla d\| - 1) = 0$. This method gives worse results in our case, because the utilization of big time-steps causes to u big deviations from the signed distance transform, which results to slow convergence of the previous PDE.

4. Experiments and comparisons

The performance of the proposed algorithm has been tested in object boundary detection experiments performed on both synthetic and natural images. As edge indicator function we used $g = g(\|\nabla I\|) = \exp\big(-\frac{\|\nabla I_s\|}{\sigma}\big)$, where I_s denotes a smoothed version of the image produced after convolution with a Gaussian of variance s. In all the experiments that follow $s = 2$. The Hausdorff distance [9] is utilized whenever quantitative measures regarding the quality of the results are supplied. In our implementation we use the algebraic multigrid code `amg1r5` [18]. The timing measurements refer to overall execution time with I/O for our implementation running on a Pentium 2 at 700 MHz under Linux. No balloon force term has been used.

In Fig. 1 we demonstrate the rapid convergence and quality of result in the case of a noisy synthetic image with additive Gaussian noise. We use as reference the final position of the contour in the case of non-noisy image.

In Fig. 2 we show a demonstration of anisotropicity of the AOS scheme which we call "shading effect". Since diffusion is carried out only parallel to the x- and y- directions, cavities with interior not completely visible by an observer who scans the image only in these directions don't get "lit" properly and the contour cannot get into them. The semi-implicit scheme faces no such problem because it permits omni-directional diffusion.

What is notable in the previous experiments, where our algorithm with very large time-steps is utilized, is that the contour overcomes the noisy spurious details in Fig. 1 and manages to get into cavities in Fig. 2, reaching in both cases the real object boundaries, despite the absence of a balloon force term. This is due to the fact that the contour propagates in big strides and thus does not get stuck easily in shallow local minima. Although this behavior is not consistent with that of the continuous model of eq. (2) with $c = 0$, it is usually highly desirable. It is also remarkable that this phenomenon does not exhibit any a priori inclination towards inward or outward motion, in contrast to curve evolution under the balloon force. The only case where the balloon force seems to be indispensable is when we want to

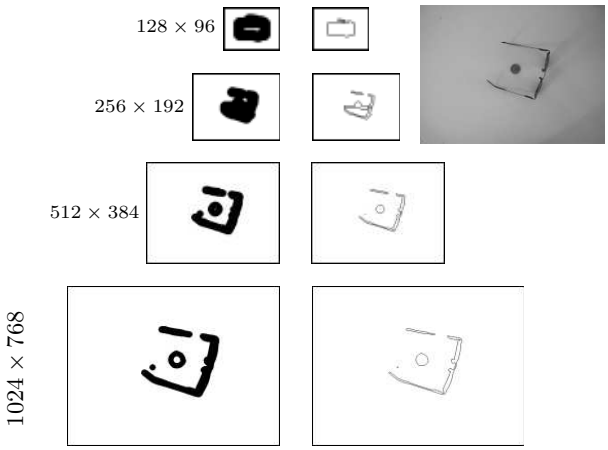

Figure 3. Multigrid combined with narrowband and pyramidal techniques. *Up right*: Original 1024×768 image. *Left collumn*: The narrow band. *Right collumn*: The contour. (Exec. time 10.2 sec.)

initialize the contour from a point in the interior of an object. Since in that case we usually cannot utilize very big time-steps due to the problems reported in Subsection 3.4, the AOS-based methods seem to be preferable in that particular scenario because of their smaller cost per iteration.

Finally, in Fig. 3 we apply our multigrid algorithm along with narrowband and pyramidal techniques and accelerate the contour evolution on a big image. We performed one step per pyramid level, starting from the coarsest one.

5. Conclusions

In this paper we presented a novel algorithm for the rapid evolution of geodesic active contours. It utilizes a semi-implicit and unconditionally stable numerical scheme which exhibits increased rotational invariance and relies on multigrid methods for the efficient solution of a sparse linear system per step. We discussed both its applicability in various scenarios and its performance in comparison with the AOS scheme. The experimental results we have presented show that the new algorithm can be a promising enabling tool in the adoption of geodesic active contours methods in time-critical applications.

6. Acknowledgements

We wish to thank the anonymous reviewers for their constructive comments and for drawing our attention to [11]. This work was supported by the research program 'Pythagoras' of the Greek Ministry of Education and by the European Union's Network of Excellence MUSCLE.

References

[1] S. Acton. Multigrid anisotropic diffusion. *IEEE IP*, 7(3):280–291, Mar. 1998.

[2] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *J. C. Ph.*, 118(2):269–277, 1995.

[3] A. Blake and M. Isard. *Active Contours*. Springer, 1997.

[4] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM Books, Philadelphia, 2 edition, 2000.

[5] V. Caselles, F. Catte, T. Coll, and F. Dibos. A geometric model for active contours. *Numer. Math.*, 66:1–31, 1993.

[6] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *IJCV*, 22(1):61–79, Feb. 1997.

[7] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *IEEE IP*, 10:1467–1475, 2001.

[8] W. Hackbusch. *Multigrid methods and applications*. Springer-Verlag, Berlin, 1985.

[9] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the Hausdorff distance. *IEEE PAMI*, 15(9):850–863, Sept. 1993.

[10] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, Jan. 1988.

[11] A. Kenigsberg. A multigrid approach for fast geodesic active contours. In *Cop. Mnt. Conf. on Multigrid Methods*, 2001.

[12] P. Maragos and M. A. Butt. Curve evolution, differential morphology, and distance transforms applied to multiscale and eikonal problems. *Fund. Informaticae*, 41:91–129, 2000.

[13] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. C. Ph.*, 79:12–49, 1988.

[14] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE PAMI*, 22(3):266–280, Mar. 2000.

[15] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *J. C. Ph.*, 155:410–438, 1999.

[16] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes*. Cambridge University Press, 1992.

[17] R. Richtmyer and K. Morton. *Difference Methods for Initial Value Problems*. Wiley, New York, 2 edition, 1967.

[18] J. W. Ruge and K. Stüben. Algebraic multigrid. In S. Mc Cormick, editor, *Multigrid Methods*. SIAM, 1987.

[19] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *PNAS*, 93:1591–1595, Feb. 1996.

[20] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.

[21] J. Weickert and G. Kühne. Fast methods for implicit active contour models. In S. Osher and N. Paragios, editors, *Geometric Level Set Methods in Imaging, Vision and Graphics*. Springer, 2002.

[22] J. Weickert, B. ter Haar Romeny, and M. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE IP*, 7(3):398–410, Mar. 1998.

[23] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum. A geometric snake model for segmentation of medical imagery. *IEEE Med. Im.*, 16(2):199–209, 1997.