Multigrid Geometric Active Contour Models

George Papandreou, *Student Member, IEEE*, and Petros Maragos, *Fellow, IEEE*

***Abstract*—Geometric active contour models are very popular partial differential equation-based tools in image analysis and computer vision. We present a new multigrid algorithm for the fast evolution of level-set-based geometric active contours and compare it with other established numerical schemes. We overcome the main bottleneck associated with most numerical implementations of geometric active contours, namely the need for very small time steps to avoid instability, by employing a very stable fully 2-D implicit-explicit time integration numerical scheme. The proposed scheme is more accurate and has improved rotational invariance properties compared with alternative split schemes, particularly when big time steps are utilized. We then apply properly designed multigrid methods to efficiently solve the occurring sparse linear system. The combined algorithm allows for the rapid evolution of the contour and convergence to its final configuration after very few iterations. Image segmentation experiments demonstrate the efficiency and accuracy of the method.**

***Index Terms*—Geometric active contours, image segmentation, implicit-explicit schemes, level sets, multigrid, partial differential equations.**

I. Introduction

ACTIVE contours, also called snakes, are among the most important tools in computer vision. They were introduced in [1] and have been widely used for image and video analysis tasks such as object boundary detection and tracking [2].

Despite its success, the original parametric active contour model has some noticeable drawbacks. First, it depends not only on the intrinsic properties of the contour but also on its parameterization; thus, it is a nongeometric model. Second, it cannot naturally handle changes in the topology of the evolving contour; significant progress towards topologically adaptable parametric snakes has been done only recently [3]. These drawbacks of standard active contours were addressed by geometric active contours, introduced in [4] and [5]. An important development has been the introduction of geodesic active contours [6], [7]. In geodesic active contours, the energy functional to be minimized is the contour's geodesic length in a flat Riemannian manifold endowed with a metric induced by image features. The model does not entail a parameterization of the curve, rendering it purely geometric. Augmenting the edge-based geodesic active contours with other region-based visual cues has led to many powerful geometric active contour models. Many region-based geometric active contours, e.g., [8]–[10], have been inspired by the piecewise smooth Mumford–Shah image model [11]. Another class of region-based geometric active contour models builds on the statistical region competition model [12]; important developments include geodesic active regions [13] and models such as [14]–[17], which have demonstrated state-of-the-art performance in multiband segmentation tasks, like texture/motion segmentation. Prior information on object shapes can also be embodied into the formulation and constrain the front evolution, leading to segmentation schemes robust against significant image degradation and/or occlusion [18]–[21].

Most implementations of geometric active contour models build on the level-set method of Osher and Sethian [22], in which the active contour is given implicitly as the zero level set of a scalar embedding function defined on the whole image domain; this allows for changes in the curve's topology much more naturally than in parametric snakes. Graph cuts [23] and minimal paths [24] are efficient alternatives to level sets in implementing certain geometric active contour models, but they are not as generally applicable as level sets. Despite the advantages of the level-set method, its computational cost can be high, rendering its utilization for time-critical applications problematic. To restrict computations to the neighborhood of the evolving contour, narrowband methods in conjuction with reinitialization techniques from the level-set technology [25]–[27] have been used. Adopting a pyramidal approach [28] can lead to further improvement. Split schemes like the additive operator splitting (AOS) scheme [29], [30] have been adapted to the problem of geodesic active contours to relax the stability constraint on the size of the time step associated with explicit numerical schemes [31], [32]. Although the AOS scheme is very stable, splitting artifacts due to reduced rotational invariance can emerge, especially when big time steps are used. Thus, unless we sacrifice accuracy, the number of iterations needed for the contour to converge still remains quite large.

In this paper, we propose efficient multigrid algorithms for level-set implementations of geometric active contour models. Our algorithms retain their accuracy and demonstrate excellent stability and rotational invariance properties even with big time steps because fully 2-D nonsplit schemes are adopted. This allows for the rapid evolution and convergence of the contour after only very few iterations. In the proposed schemes, we make a distinction between the curvature-based active contour *internal forces*, designed to keep the curve smooth, and the remaining model-specific *external forces*, because the internal forces are particularly *stiff*, i.e., most difficult to handle stably in numerical implementations. Therefore, we treat the internal forces with implicit schemes and the external forces with explicit schemes. The resulting *implicit-explicit* [33] scheme combines the stability of implicit schemes and the flexibility of explicit schemes

Manuscript received January 24, 2006; revised May 17, 2006. This work was supported in part by the Greek Secretariat for Research and Technology under program PENED-2003, in part by the European Network of Excellence MUSCLE (IST-FP6-507752), in part by the European research project ASPI (IST-FP6-21324), and in part by a Greek State Scholarships Foundation (IKY) scholarship to GP. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Philippe Salembier.

The authors are with the School of Electrical and Computer Engineering, National Technical University of Athens, Athens 15773, Greece.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2006.884952

in a powerful blend. In the core of our algorithms is the efficient solution of a big sparse linear system which occurs at each time step. For this purpose, we apply *multigrid* methods [34], similarly to the use of multigrid techniques in the context of anisotropic diffusion [35]. The cost per iteration of the multigrid solver is comparable to the cost of tridiagonal solvers utilized in split schemes. However, the overall time of evolution is now typically smaller due to the reduced number of iterations needed for convergence. Adopting the low-cost separable distance transform of [36] further eases the computational burden. Standard pyramidal and/or narrowband techniques can naturally fit into the proposed framework and further accelerate the curve evolution.

Although multigrid techniques have been applied to geodesic active contours by Kenigsberg *et al.* in [37] and [38], our treatment is novel. In [37] and [38], a different discretization scheme is adopted, which leads to a nonlinear system of equations at each time step, whose solution by a nonlinear multigrid solver might be problematic. In our work, using implicit-explicit schemes and reinitializing the level-set embedding function before every new step leads to a linear system which is much easier to handle. Relative to [37] and [38], we also demonstrate our approach to a wide variety of geometric active contour models beyond geodesic active contours. Finally, we compare thoroughly the multigrid and the AOS techniques, illustrating their qualitative characteristics.

This paper extends our earlier conference work [39] on multigrid algorithms for geodesic active contours in several directions. First, we demonstrate the utility of multigrid curve evolution algorithms in efficiently solving a variety of additional computer vision models which encompass geometric active contours ideas, such as the Mumford–Shah and the region competition/geodesic active regions models; this is achieved in the framework of implicit-explicit schemes. Second, we elaborate on the optimal design of the multigrid solver components, depending on the numerical properties of the specific problem one solves, thus extending the scope of [39] which was confined to algebraic multigrid techniques. Third, we use the separable distance transform algorithm of [36] to rapidly reinitialize the front.

The rest of the paper is organized as follows. We begin in Section II with a review of some geometric active contour models which can be effectively treated by our algorithm. In Section III, we present discretization methods for geometric active contours from the unifying viewpoint of implicit-explicit schemes and discuss alternative algorithms for re-initializing the level-set embedding function to distance transform. Our multigrid algorithm for evolving geometric active contours is developed in Section IV. Finally, in Section V, we give experimental results demonstrating the performance of our method.

II. Geometric Active Contour Models

Let C be a planar curve with length $L(C)$ and $\vec{C}(s) \equiv (x(s), y(s)) : [0, L(C)] \to \mathbb{R}^2$ its arc-length parameterization; also, let I be an intensity image. In geodesic active contours, we minimize the curve's geodesic length [6], [7]

$$E(C) = \int_C g(|\nabla I(\vec{C}(s))|)ds \tag{1}$$

where the edge indication function $g : [0, +\infty) \to \mathbb{R}^+$ is a decreasing function of $|\nabla I|$ (other measures of edge strength can also be used), such that $g(0) = 1$, $g(r) \to 0$ as $r \to +\infty$. Minimization of the functional in (1) by means of variational techniques leads to a Euler–Lagrange partial differential equation (PDE); to reach (local) minima, we start with an initial contour and evolve it in the direction of steepest descent, introducing a pseudo-time variable t. Numerical implementation of the geodesic active contour model using level sets [22] naturally allows for changes in the contour's topology, permitting splits and merges. In the level-set framework, the moving contour $C(t)$ is defined implicitly as the zero level set of an embedding scalar function u with domain the whole image plane, i.e., $C(t) = \{(x,y) : u(x,y,t) = 0\}$, where $u(x,y,t) : \mathbb{R}^2 \times [0,+\infty) \to \mathbb{R}$. By convention, we assign negative values to the interior and positive values to the exterior of the curve. The signed distance function from the contour is often chosen for extending u away from the curve due to its good numerical properties. Laws describing the motion of the contour can be translated to compatible laws governing the motion of the embedding function u [22]. For geodesic active contours, the law is [6]

$$\frac{\partial u}{\partial t} = |\nabla u| \mathrm{div}\left(g(|\nabla I|)\frac{\nabla u}{|\nabla u|}\right) + cg(|\nabla I|)|\nabla u|. \tag{2}$$

In the last equation, we have added a so-called balloon force term, which acts like erosion/dilation, favoring the inward (if $c > 0$) or the outward (if $c < 0$) motion of the contour, respectively [40]. Improved contour alignment with image edges can be achieved by supplementing the model with the Laplacian edge detection term of [41].

The edge-based geodesic active contour model can be considerably enhanced by incorporating information from the whole image into the curve evolution process; advantages of such region-based models are less dependence on initialization and tolerance to noise. Many of these methods have been inspired by the model of Mumford–Shah [11], in which a curve C splits the domain Ω in foreground R and background R' and the image I is a degraded version of an ideal image f; f is assumed to be smooth inside the segments R and R', but not on the boundary C. The corresponding geometric active contour level-set evolution law is [9], [10]

$$\frac{\partial u}{\partial t} = |\nabla u| \left\{ \mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \frac{\alpha}{2}\left((f_R - I)^2 - (f_{R'} - I)^2\right) + \frac{\beta}{2}\left(|\nabla f_R|^2 - |\nabla f_{R'}|^2\right) \right\} \tag{3}$$

with α and β positive weights. Since f is also unknown, it needs to be re-estimated as the front moves [19]. A cartoon version of the Mumford–Shah model is the Chan–Vese model [8], where the image intensity is assumed piecewise constant ($c_R/c_{R'}$ inside/outside the contour) and the curve evolves by

$$\frac{\partial u}{\partial t} = |\nabla u| \left\{ \mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \frac{\alpha}{2}\left((c_R - I)^2 - (c_{R'} - I)^2\right) \right\}. \tag{4}$$

Beyond the piece-wise smooth intensity assumption, in a class of region-based geometric active contour models pioneered by Region Competition [12], one partitions the image into statistically homogeneous regions, where the statistics are computed on features that offer good discriminatory power for a particular application [13]–[17]. When a foreground model R competes with a background one R', we obtain the law

$$\frac{\partial u}{\partial t} = |\nabla u| \left\{ \operatorname{div}\left(g(|\nabla I|) \frac{\nabla u}{|\nabla u|} \right) + \alpha \log \frac{Pr(x|R)}{Pr(x|R')} \right\} \quad (5)$$

where $Pr(x|R)$ denotes the probability assigned to pixel x by model R. In purely region-based approaches, one typically sets $g \equiv 1$ in (5); boundary-based information can be integrated by utilizing an edge indicator function g, similar to that of (1) [13]. Applications of these models to image segmentation or object tracking tasks has given excellent results. Techniques that integrate object-specific shape prior statistical information into the curve evolution process can also be described by laws similar to (5) [18]–[21].

III. Numerical Schemes for Geometric Active Contour Models

A common characteristic of the geometric active contour models reviewed in Section II is that they can be cast as

$$\frac{\partial u}{\partial t} = |\nabla u| \left\{ \operatorname{div}\left(g(|\nabla I|) \frac{\nabla u}{|\nabla u|} \right) + F(u) \right\} \quad (6)$$

where the right-hand side term $F(u)$ groups the external forces of each specific model used [cf. (2)–(5)].

The last equation emphasizes that all these models share the same parabolic term $|\nabla u| \operatorname{div}(g(|\nabla I|)(\nabla u / |\nabla u|))$, which turns into the familiar nonhomogeneous heat diffusion term when the embedding function u is constrained to be a distance transform [42] (in which case $|\nabla u| = 1$). This internal geodesic curvature force term acts on the evolving front as a geometric regularizer, penalizing nonsmooth boundaries. It plays a similar role to the membrane and thin-plate force terms of snakes [1] and can be interpreted probabilistically as enforcing a generic prior favoring short region borders. Although this regularizing term is indispensable in geometric active contour models, it makes the resulting PDE particularly *stiff* [43] numerically: If one uses simple explicit methods for numerically evolving the contour, then instability incurs unless very small time steps are applied. To overcome this shortcoming, implicit integration in time is needed for this term. On the other hand, the stability constraint associated with the remaining external forces term $F(u)$ is typically much looser; therefore, simple explicit in time schemes are adequate for $F(u)$. Before explaining these further, we will first discuss appropriate discretization of the different terms in (6).

A. Spatial Discretization

Finite difference discretization schemes for the various terms involved in (6) have been extensively studied in the level-set literature [25], [44].

The first term on the right-hand side of (6) describes motion under geodesic curvature and is of parabolic nature. Therefore, central difference schemes are suitable for descritizing it [22], [32]. As noted in [31], a particularly attractive simplification occurs if we reinitialize the embedding function u to be a signed distance transform before each iteration, which implies $|\nabla u| = 1$. Then, the short-time curve evolution driven by this term is described by the linear nonhomogeneous diffusion equation

$$\frac{\partial u(x,y,t)}{\partial t} = \operatorname{div}(g(x,y)\nabla u) \quad (7)$$

where the conduction coefficient $g(x,y) = g(|\nabla I|)$ depends on the image I and not on the evolving function u. We adopt this reinitialization technique in our scheme because it yields the time-independent operator $\operatorname{div}(g(x,y)\nabla \cdot)$; this is particularly desirable in the multigrid context of our approach, since the system matrices at all levels need to be computed only once in a setup phase. Moreover, reinitialization of the embedding function to a distance transform improves the robustness of level-set methods [25]. Fast distance transform algorithms for this task are discussed in Section III-C.

Following the extensive literature on numerical methods for anisotropic diffusion, we discretize $\operatorname{div}(g\nabla u) = (gu_x)_x + (gu_y)_y$ of (7) by the standard five-point stencil [30], [32]

$$\begin{aligned} \operatorname{div}(g\nabla u)|_{ij} \approx \frac{1}{h_x^2} \Big\{ & g_{i+\frac{1}{2},j}(u_{i+1,j} - u_{ij}) \\ & - g_{i-\frac{1}{2},j}(u_{ij} - u_{i-1,j}) \Big\} \\ + \frac{1}{h_y^2} \Big\{ & g_{i,j+\frac{1}{2}}(u_{i,j+1} - u_{ij}) \\ & - g_{i,j-\frac{1}{2}}(u_{ij} - u_{i,j-1}) \Big\} \end{aligned} \quad (8)$$

where h_x, h_y are the spatial finite difference discretization mesh grid lengths. In the following we assume for convenience that $h_x = h_y = h$. We also make a linear approximation for g, obtaining, for example, $g_{i+(1/2),j} \approx (g_{i+1,j} + g_{ij})/2$.

If we raster-scan the pixels lexicographically (in a row-major order) into a column vector u, we can compactly write (8) in matrix-vector notation as

$$\operatorname{div}(g\nabla u) \approx Au \quad (9)$$

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

$$a_{ij} = \begin{cases} \frac{g_i + g_j}{2h^2}, & j \in \mathcal{N}(i) \\ -\sum\limits_{k \in N(i)} \frac{g_k + g_i}{2h^2}, & j = i \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

where $\mathcal{N}(i)$ denotes the 4-neighborhood of pixel P_i. This matrix is extremely sparse: It has only five nonzero elements per row, apart from the rows corresponding to pixels on the sides or corners of the image where it has four or three nonzero elements, respectively, and properly incorporates the reflecting boundary conditions.

Finally, the remaining external force terms $F(u)$ of (6) should be discretized by appropriate numerical schemes, depending on their nature. For example, upwind schemes might be needed for the discretization of the balloon force term of (2) and other hyperbolic or convective terms that might exist in a particular model [25], [44]. Treating the external forces of the models presented in Section II is straightforward and needs not be discussed further here.

B. Timestepping With Implicit-Explicit Schemes

Combining the spatially discrete expressions for the internal (9) and external force terms while keeping a continuous-in-time formulation [45], we end up with the following system of N (as many as the number of pixels) ordinary differential equations in time

$$\frac{\partial u}{\partial t} = Au + F(u) \tag{11}$$

where u is the raster-scanned vectorized level-set function. A time-integration scheme needs to be adopted next for discretizing the time variable t and numerically advancing the solution of (11).

The *explicit* scheme $(u^{n+1} - u^n)/\tau = Au^n + F(u^n)$ has been widely used for level-set calculations since it is the most straightforward and easy to implement: both right hand side terms are evaluated using only current function values. Unfortunately, due to the stiff internal parabolic term Au, the explicit scheme is only stable for very small time steps $\tau \leq h^2/4$ [30]. This stability constraint is practically very restrictive [43]: If the active contour moves by this internal force by a distance D to reach its steady state, we need to run the algorithm for time $T \propto D^2$. Therefore, the number of steps needed is $(T/\tau) \propto (D^2/h^2) = \mathcal{O}(N)$. Thousands of iterations are required to process a moderately sized picture. Since the cost per iteration is also $\mathcal{O}(N)$, the overall cost of the explicit algorithm is $\mathcal{O}(N^2)$. For example, the processing cost for a 200×200 image is roughly 16 times bigger than that for the same image at half the resolution 100×100.

To improve performance, a key observation is that stability and not accuracy is the performance bottleneck of the explicit scheme [30]. One can sacrifice some accuracy by employing schemes which permit bigger time steps, since for most image processing applications the accuracy lost is hardly visible. This is particularly true for active contours, since for this application (unlike anisotropic diffusion) one is interested in the steady-state of the system and not in accurately describing transient phenomena [43]. Therefore, utilizing big time steps in conjuction with implicit schemes, which are excellent for stably integrating stiff sets of differential equations [43], [46], is well suited to geometric active contours.

A *fully implicit* scheme for (11) is the backward Euler method $(u^{n+1} - u^n)/\tau = Au^{n+1} + F(u^{n+1})$. This scheme can stably treat the stiff parabolic term Au no matter how large the time step is [30], but, depending on the particular form of F, solving the resulting (usually dense and nonlinear) algebraic system for u^{n+1} might be complicated. This is generally true for most of the geometric active contour models of Section II, making the fully implicit scheme impractical.

However, implicit integration of the external force term $F(u)$ may not be required. This term is usually amenable to stable integration by explicit schemes with much bigger time steps than that allowed for the internal force term treated explicitly. Combining the benefits of stable implicit integration for the stiff term and simple explicit integration for the nonlinear term yields the hybrid class of so-called *implicit-explicit* (IMEX) schemes [33], the simplest of them being

$$\frac{u^{n+1} - u^n}{\tau} = Au^{n+1} + F(u^n). \tag{12}$$

Obviously, in the absence of an external force term F, as in the pure geodesic active contour case we studied in [39], the fully implicit and the IMEX schemes coincide. Various IMEX schemes have been successfully used in practice for years. For example, algorithms for integrating parametric snakes with IMEX schemes have been used early on [1].

Adopting the IMEX scheme of (12) means that one needs to solve the following linear system at each iteration:

$$\left[\frac{1}{\tau}I - A\right] u^{n+1} = \frac{1}{\tau}u^n + F(u^n). \tag{13}$$

While solving an analogous system in 1-D (which is essentially the case of parametric spline snakes [1]) is quite easy, the situation in our 2-D level-set case is not so straightforward. The system matrix $L \equiv (1/\tau)I - A$ is very large and inherits the sparse structure of A described in Section III-A. This specific sparse structure does not lend itself to application of efficient elimination techniques (the matrix does not have small bandwidth). Although L is strictly diagonally dominant, simple iterative methods such as Jacobi or Gauss–Seidel converge slowly for such big systems. Convergence gets even slower with bigger time steps τ, since then the regularizing diagonal term $1/\tau$ diminishes, and, hence, the diagonal dominance of L weakens. With these simple iterative methods, the number of iterations required to reduce the error by a predefined factor is proportional to the number of pixels N[43]. Since their cost per iteration is also $\mathcal{O}(N)$, this leads to an $\mathcal{O}(N^2)$ cost for each step of (13)—no improvement over the explicit algorithm.

To avoid solving the full-blown system (13), the authors in [31] and [32] adopt the AOS scheme, whose IMEX variant is

$$u^{n+1} = \frac{1}{2}\sum_{l\in\{x,y\}} \left[\frac{1}{2\tau}I - A_l\right]^{-1} \left[\frac{1}{2\tau}u^n + \frac{1}{2}F(u^n)\right] \tag{14}$$

where the matrices $A_l, l \in \{x, y\}$ are the 1-D counterparts of A [cf. (10)]. The AOS scheme is also unconditionally stable with respect to the stiff internal force term and was first introduced in the context of nonlinear diffusion [30]. It belongs to the class of alternating direction implicit (ADI) methods [46]; a multiplicative locally one-dimensional (LOD) variant of it has been used in [47]. The AOS scheme is a simplified version of the fully 2-D IMEX scheme (13) in which a 2-D diffusion process of duration τ is approximated as the average of two independent 1-D ones of duration 2τ each. The advantage of this approach is that the 2-D problem is approximated with two simpler 1-D ones. The two linear systems involved in (14) are tridiagonal and can be solved efficiently with $\mathcal{O}(N)$ cost per step [30]. The disadvantages are, first, reduced accuracy, since the effective 1-D diffusion time step is double in the case of AOS (2τ instead of τ); this implies that for a given desired accuracy one should take roughly twice as many steps with the AOS scheme. More importantly, unless the time step τ is very small, splitting artifacts can emerge due to loss of rotational invariance. The LOD scheme, treating nonsymmetrically the x and y axes, has even worse rotational invariance properties [30]. 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 adopt the fully 2-D IMEX scheme in its complete form (12). In order to efficiently solve the corresponding full-blown system (13) in optimal $\mathcal{O}(N)$ time, we resort to multigrid techniques, as described in Section IV. By carefully selecting the constituting multigrid components, the fully 2-D IMEX scheme has essentially the same computational cost per time step as the split AOS scheme of (14), while at the same time preserving the benefits of the fully 2-D approach, namely increased accuracy and rotational invariance. These advantageous properties of the proposed 2-D IMEX/multigrid algorithm allow using even bigger values of the time step τ than in the AOS case, resulting in the rapid evolution of the contour to its final configuration after very few iterations.

It is worth mentioning that the proposed discretization of (12) is not the only possible IMEX scheme [33]. For example, one can evaluate the derivative of the stiff term half-way between the current and the next step, to obtain the Crank–Nicolson (CN) scheme $(u^{n+1} - u^n)/\tau = A(u^n + u^{n+1})/2 + F(u^n)$. This scheme, in comparison to all previously mentioned schemes which are only $\mathcal{O}(\tau)$ accurate, has superior $\mathcal{O}(\tau^2)$ accuracy. However, the CN scheme is not as stable as (12): Although u^{n+1} computed by CN remains bounded even for very large τ, undesirable oscillations of high spatial frequency modes can emerge (for $\tau/h^2 \gtrsim 100$, in practice), as can be verified by a Fourier-type analysis of the scheme [43], [46]. This renders the CN scheme less competitive in our application, since we are mainly interested in the steady-state position of the active contour and stability is of utmost importance. In contrast, for applications like anisotropic diffusion, where one needs to accurately describe the evolution of the contour and smaller time steps are usually utilized, the second-order accuracy of the CN scheme (or split variants of it) could lead to visible quality improvements [47].

Finally, we comment on appropriate schemes to treat the balloon force term of (2), described by the hyperbolic equation $\partial u/\partial t = cg(|\nabla I|)|\nabla u|$ which corresponds to adaptive dilation/erosion [48]. The first option, following [31] and [32], is to incorporate the balloon force into the external force terms F and treat it explicitly into the framework of IMEX schemes just described. This approach leads to a CFL stability constraint to the allowable time step [25]: $|c|\tau/h \leq 0.25$. The second option we used in [39] is to adopt a fractional steps scheme [43], in which the front first moves under the balloon force and then by the remaining (internal and external) force terms. Motion under the balloon force term is treated separately by an unconditionally stable numerical scheme and the remaining forces are integrated by IMEX schemes as usual. The idea behind the numerical scheme for the balloon force term is that evolution of the zero level-set $C(t)$ under this law can be cast in stationary form as $C(t) = \{(x,y) : T(x,y) = -ct\}$, where the function $T(x,y)$ satisfies the eikonal PDE $|\nabla T|g(x,y) = 1$. The fast marching (FM) algorithm can then be applied to compute T [25], [42]. Although the resulting scheme is unconditionally stable, its drawback is the increased computational cost of applying the FM algorithm every time step. However, no matter which approach one follows in discretizing it, the precence of the balloon force term can lead to problems when very big time steps are utilized, because the contour may skip over and miss weak object boundaries, something also noticed in [31]. A heuristic modification of the edge indicator function g to ameliorate this effect has been proposed in [38]. A better solution is to altogether replace the balloon force with the gradient vector flow (GVF) external force [49], which does not exhibit this problem, while offering most of the advantages of the balloon force, such as convergence to boundary concavities and reduced sensitivity to initial conditions.

C. Distance Transform Algorithms for Reinitialization

As explained in Section III-B, it is advantageous before every iteration to reinitialize the embedding function u to be a signed distance transform, i.e., signed distance function from the contour, so that $|\nabla u| = 1$ and $u < 0$ $(u > 0)$ for the pixels in the interior (exterior) of the front. Moreover, using the distance transform as level-set embedding function facilitates the computation of the Hausdorff distance [50] between shapes, which is useful in checking the convergence of the contour to its final configuration.[1]

We have compared various distance transform algorithms for level-set reinitialization. The most common approaches in the level-set community are the FM algorithm [25] and the PDE-based method of [26] and [27]. The FM algorithm utilizes an efficient insert-sort procedure based on heaps and has $\mathcal{O}(N \log N)$ complexity. A nice property of the FM algorithm is that it first computes the distance function near the zero level set and gradually expands further. Therefore, the procedure costs less if the distance function is only needed in a narrow band around the contour. The alternative method of [26], [27] drives the PDE $\partial u/\partial t + \operatorname{sign}(u)(|\nabla u| - 1) = 0$ to steady state, thus yielding $|\nabla u| = 1$ for $t \to \infty$. While this PDE method is easy to implement and effective if the embedding function u is close to a distance function, it converges slowly in our case since we typically utilize big time steps which cause to u big deviations from the signed distance function.

We have also investigated an algorithm by Felzenszwalb and Huttenlocher to reinitialize u to a distance function [36]. This algorithm is very simple to implement and, to the best of our knowledge, has not been used in level-set applications before. It first computes in linear cost the 1-D distance transform of each row as the lower envelope of parabolas. It then exploits the min-plus separability of the Euclidean distance transform (distance transforms can be seen as morphological min-plus convolutions [42]) and applies the 1-D procedure columnwise on the intermediate result to obtain the 2-D distance function.

[1] If C, C' are two active contours and u, u' their distance transform embedding functions, then their Hausdorff distance is $H = \max(h(C,C'), h(C',C))$, where the directed Hausdorff metric $h(C,C') = \max_{c \in C} \min_{c' \in C'} \|c - c'\|_2$ can be computed efficiently as $h(C,C') = \max_{c \in C} |u'(c)|$ and similarly for $h(C',C)$.

Extension to more dimensions is straightforward. Being separable, this algorithm has optimal cost—only $\mathcal{O}(N)$ comparisons. Indeed, we have found it to be the fastest in practice in our experiments.

IV. Multigrid Evolution of the Contour

Multigrid numerical methods employ a hierarchy of grids of different mesh sizes to efficiently solve a wide range of problems, most notably those arising from PDE models [51]. Some excellent books on multigrid are [34], [52], and [53]. Multigrid algorithms were first employed systematically in computer vision problems in [54]. For multigrid optical flow algorithms, see [55] and the references therein; other relevant multigrid computer vision applications include [19], [35], [38], [39], [56], and [57].

A. *The Multigrid Principle*

Multigrid techniques overcome the typically slow convergence properties of conventional uni-grid iterative methods [43] (e.g., Jacobi, Gauss–Seidel, and conjugate gradients.) While uni-grid relaxation procedures are particularly effective at eliminating the high spatial frequency part of the error, they suppress very slowly its low-frequency part. This behavior stems from the local nature of computations in uni-grid methods, which allows attenuation of error components that vary on scales comparable to the discretization mesh grid size h, but prevents quick suppression of larger scale errors. As discretization grids get finer, information propagates even more slowly and the shortsightedness of local uni-grid methods becomes more pronounced. Multigrid cycles overcome these difficulties by employing conventional relaxation procedures, called *smoothers* in multigrid terminology, in a hierarchy of grids. Since relaxation performed at each resolution level smooths the error components at scales comparable to the grid size of this level, one multigrid cycle can effectively eliminate error components at the whole range of frequencies. Moreover, applying a relaxation procedure at coarse scales costs significantly less than applying it at fine scales, since less variables are involved. Such multigrid techniques are optimal for a wide range of interesting problems, both linear and nonlinear, in the sense that discretization accuracy can be achieved after a number of cycles which is independent of the problem size with total cost $\mathcal{O}(N)$. The multigrid method is also applicable to 3-D problems, such as active surfaces implemented with level sets; however, this is not pursued further here.

Algorithm 1 Generic two-level multigrid

```
Relax $n_1$-times on $L_h u_h = f_h$.
Compute the residual $r_h = f_h - L_h u_h$ in the fine grid.
Restrict the residual $r_{2h} = R r_h$ to the coarse grid.
Solve directly the coarse-grid system $L_{2h} e_{2h} = r_{2h}$.
Prolong the error to the fine grid $e_h = P e_{2h}$.
Correct the fine grid solution $u_h \leftarrow u_h + e_h$.
Relax $n_2$-times on $L_h u_h = f_h$.
```

We introduce the multigrid idea by giving the generic two-level Algorithm 1 for solving a system of equations $L_h u_h = f_h$, like (13) [52]. Subscripts denote the discretization mesh grid size. Concrete procedures follow from Algorithm 1 by specifying the following *multigrid components*.

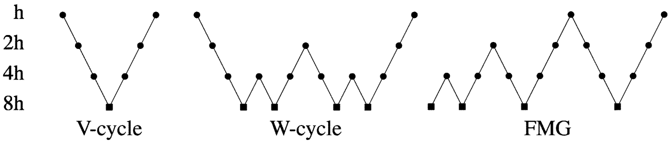


Fig. 1. Usual multigrid schedules. Grid size in the left column. Illustration for four levels.

1) The *smoother* and the *number of pre/postsmoothing steps* n_1/n_2; usually $n_1 + n_2 \leq 3$ Gauss–Seidel relaxations suffice.
2) The intergrid transfer matrices: the *prolongation* (interpolation) $N_h \times N_{2h}$ matrix P and the *restriction* (decimation) $N_{2h} \times N_h$ matrix R, where $N_{(\cdot)}$ is the number of unknowns at a certain level, typically $N_{2h} \approx (1/4) N_h$ for 2-D problems; simple choices are bi-linear interpolation for P and injection (subsampling) for R.
3) The *coarse grid system matrix* L_{2h}; in simple cases rediscretizing the continuous problem at the coarse grid works well.

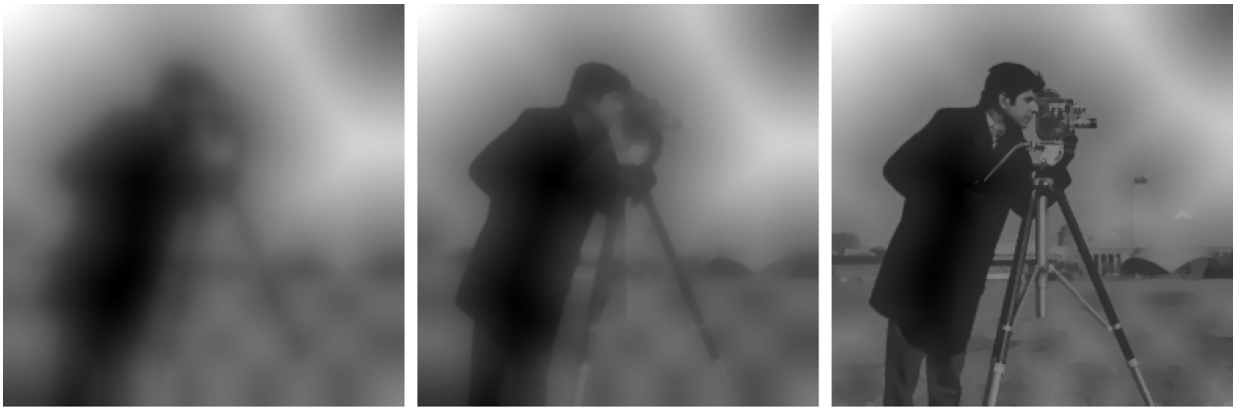
However, the aforementioned "standard" choices for multigrid components are not always effective. We discuss more proper alternatives for our application in Section IV-B.

In extending the two-level Algorithm 1 to multiple grids we have many options, depending on the order we visit the various levels. Fig. 1 depicts examples of the most popular multigrid schedules. These schedules are usually defined recursively [34]; using enough levels ensures that, in the coarsest grid, very few unknowns are left and applying a direct solver is computationally cheap. There are two types of multigrid algorithms (cf. Fig. 1): *Correction* algorithms (e.g., the V and W cycles) start at the finest level and use coarser levels to smooth the low-frequency error of the residual equation; *nested-iteration* (e.g., the FMG cycle) starts at coarser grids and use them to generate initial guesses for finer-grid problems. A good rule of thumb is to use correction algorithms if a good initial estimate is available, as is the case in our application.

The previous discussion highlights an important distinction between multigrid techniques and other coarse-to-fine multiresolution algorithms, such as [28], [58], and [59]. Multigrid algorithms utilize both coarse-to-fine and fine-to-coarse transitions to achieve optimal performance, while traditional pyramidal image analysis algorithms use coarse grids just to obtain estimates of the fine-grid solution; after that, coarse grids are not revisited. These issues are further discussed in [55] and [60].

B. *Selection of Multigrid Components*

Designing effective multigrid components for the linear system of (13), which arises at every front update step when we employ the IMEX scheme, requires that we first examine some qualitative properties of its system matrix $L_h = (1/\tau) I_h - A_h$. The positive diagonal constant $1/\tau$ has a regularizing effect, improving the diagonal dominance of L_h; this results in better convergence rates, particularly for small time steps when $1/\tau$ is substantial. As $\tau \rightarrow +\infty$, we obtain the system



$\sigma = \infty$ $(g \equiv 1)$ $\sigma = 0.1$ $\sigma = 0.01$

Fig. 2. Level-set embedding function u^{n+1}, as given by (13) (i.e., before reinitialization), depicted as gray-value image for various values of the edge threshold σ in a geometric active contour with $g(x, y) = \exp\left(-|\nabla I_s|/\sigma\right)$. As σ decreases, image edges cause sharper changes to u.

$-Au = f$, corresponding to the inhomogeneous Poisson equation $-\mathrm{div}(g\nabla u) = f$. Thus, multigrid solvers designed for inhomogeneous Poisson boundary-value problems perform at least as well in our case [34].

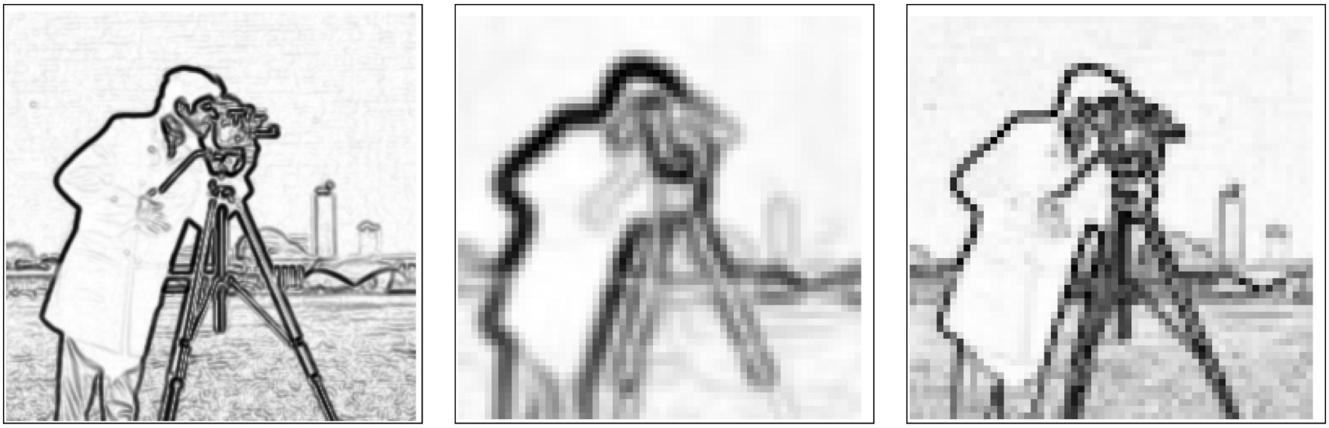
Among the various geometric active contour models reviewed in Section II, the easiest to handle with multigrid techniques are the purely region-based ones, such as the Mumford–Shah and the Chan–Vese. For these models, $g \equiv 1$, and the corresponding Poisson equation, $-\nabla^2 u = f$ is homogeneous. For the homogeneous Poisson equation, the standard multigrid components mentioned in Section IV-A perform excellently [34].

On the other hand, a serious complication with *edge-based* (i.e., $g \neq 1$) geometric active contour models is that the diffusivity $g(x,y)$ in (7) can vary several orders of magnitude throughout the image; we typically have $g \approx 1$ in homogeneous regions and $g \approx 10^{-3}$ at edges of the image. Such strong discontinuities in g impair the convergence of multigrid when the simple intergrid transfer operators and coarse grid matrices described in Section IV-A are utilized [19], [53], [61], [62]. To see why, we recall from Section III-A that the short-time evolution of the level-set embedding function u corresponds to heat diffusion in a nonhomogeneous medium with conductivity g [cf. (7)]. Therefore, although u^n has been reinitialized to be a distance function, and is, thus, smooth, the embedding function at the next time step u^{n+1} before reinitialization, given as solution to the linear system (13), is not smooth at image edges, since there the conductivity is almost zero; the physical analogy is that image edges act as insulators and obstruct heat transfer, causing sharp changes to the "temperature" u across them. Since u is nonsmooth for strongly varying g, using standard linear interpolation as prolongation operator P in transfering corrections from coarse to fine grids will blur the solution estimate and deteriorate multigrid convergence rates. The situation is illustrated in Fig. 2. Similarly, if the coarse-grid matrix L_{2h} is calculated by computing image gradients on a Gaussian pyramid of the original image and using (10) at every level, then solutions to coarse-grid problems might be too smooth to give useful corrections to fine-grid problems. This discussion highlights the need for *edge-preserving* multigrid components for edge-based geometric active contours.

We have explored two alternative approaches from the multigrid literature, both well suited to the case of strongly discontinuous edge indicator $g(x, y)$: *matrix-dependent restriction/ prolongation multigrid* (MDMG) [53], [62] and *algebraic multigrid* (AMG) [34], [52], [56], [57], [63]. MDMG is applicable to problems discretized on rectangular grids and coarse grid meshes are conventionally formed by doubling the discretization mesh size. Good multigrid performance in the presense of discontinuous g is achieved by using edge-preserving adaptive interpolation operators P which avoid smoothing the coarse-grid corrections across image edges; see [62] for details and concrete interpolation formulae. Using specialized smoothers, such as incomplete LU factorization, makes MDMG even more robust [53], [62].

While applicability of MDMG is typically confined to problems discretized on regular meshes, AMG methods are designed to also handle problems $L_h u_h = f_h$ defined on more general unstructured graphs $G_h = (\mathcal{V}_h, \mathcal{E}_h)$, with vertex set $\mathcal{V}_h$ and edge set $\mathcal{E}_h$. In AMG, one partitions the fine-grid vertices $\mathcal{V}_h$ into two disjoint sets $\mathcal{V}_h = \mathcal{C}_h \cup \mathcal{F}_h$, $\mathcal{C}_h \cap \mathcal{F}_h = \emptyset$; the vertices in $\mathcal{C}_h$ are retained in the coarse grid, i.e., $\mathcal{C}_h = \mathcal{V}_{2h}$, and are chosen in such a way that each vertex $i \in \mathcal{F}_h$ not included in the coarse grid is *strongly coupled* to vertices in $\mathcal{C}_h$, e.g., $\sum_{j \in \mathcal{C}_h} |l_{ij}| \geq \beta \sum_{j \in \mathcal{V}_h} |l_{ij}|$, for some fixed $\beta \in (0, 1)$, where l_{ij} are the elements of the system matrix L_h. This ensures that errors and corrections are transferred accurately across grids and makes AMG very robust in the presence of discontinuous coefficients. Moreover AMG can easily treat the irregular grids associated with narrowband calculations and extends more readily to 3-D problems. On the other hand, the flexibility of AMG costs in multigrid setup time and in memory; thus, MDMG is preferable in the non-narrowband case. Hybrid approaches in the spirit of [56], [57], and [64], having elements of AMG while still benefitting from the image grid structure, could be faster than fully general AMG and similarly support narrowband calculations.

System matrices at coarse grids must also be carefully designed in the case of discontinuous g, so that the problem there closely resembles the fine-grid one. The simplest method to obtain coarse-grid system matrices (cf. Section IV-A) is to re-discretize the continuous problem at all coarse grids: a Gaussian



(a) Fine-grid (256×256) (b) Rediscretization (64×64) (c) Galerkin (64×64)

Fig. 3. Continuous problem rediscretization versus Galerkin generation of coarse-grid matrices. Pictures show the modulus of the matrix L diagonal elements $\mathrm{diag}(L)$, which correspond to image edge strength, cf. (10); (a) fine grid L_h; (b) rediscretization level-2 L_{4h}; (c) Galerkin level-2 L_{4h}. The Galerkin approximation preserves sharp edge transitions.

TABLE I
COST OF MAIN OPERATIONS FOR 256 × 256 IMAGES

Operation		Algorithm	Time (milli-sec)	
			Setup	Per time-step
Scheme	2D IMEX	Mat.-Dep. MG (MDMG)	35±1	23±1
		Algebraic MG (AMG)	300±20	40±5
	AOS	Tridiag. Solver (TRI)	13±1	5±1
Distance Transf.		Separable DT (SDT)	n/a	27±1
		Fast Marching (FMDT)	n/a	38±2

pyramid I^k, $k \geq 0$, of the original image I is created and all coarse-grid matrices $L_{2^k h}, k \geq 1$, are computed similarly to the fine-grid matrix L_h, i.e., by calculating $g(|\nabla I^k|)$ and then using (10). However, this approach leads to overly smooth coarse-grid corrections, since image edges get excessively dispersed, as illustrated in Fig. 3(b).

In our context, a much better choice for determining the restriction operator R and coarse-grid system matrices is through the so-called *variational* approach, which is a constituent component of both MDMG and AMG algorithms [34], [52], [53], [62]. In the variational approach, one only needs to feed the algorithm with the fine grid matrix L_h and right-hand side f_h; their coarse-grid counterparts are computed recursively by

$$L_{2h} = RL_hP \text{ (Galerkin Condition)} \tag{15a}$$

$$R = P^T. \tag{15b}$$

This choice naturally arises from a variational interpretation of the solution to $L_h u_h = f_h$ common to finite element formulations: for L_h symmetric positive-definite (SPD) (as in our case) the solution u_h minimizes the functional $J_h(u_h) = (1/2)(L_h u_h, u_h) - (f_h, u_h)$, where $(\cdot,\cdot)$ denotes the Euclidean vector inner-product; seeking the coarse-grid correction Pe_{2h} which minimizes $J_h(u_h + Pe_{2h})$ yields the variational conditions (15); see [52, Ch. 10] for details. Another interesting property of the variational condition is that if L_h is SPD, then L_{2h} is also SPD, for arbitrary full-rank prolongation matrix P [34]. Note that computation of coarse-grid matrices by the Galerkin condition (15a) can be performed efficiently by exploiting the sparse structure of R, L_h, and P [53]. Additional efficiency can be attained in the non-narrowband case: since we reinitialize u to be a signed distance function before every new front update and the problem domain is unaltered, the system matrix L_h does not change over time, thus building up the hierarchy of grids and coarse-grid matrices needs only be done once. The effectiveness of adaptive interpolation P in conjuction with decimation R and coarse grid matrices selected according to the variational principles (15) in treating edge-based geometric active contours is illustrated in Fig. 3(c): the edge map is well preserved in coarse grids, ensuring sufficiently sharp coarse-grid corrections.

Finally, two additional qualitative characteristics of the active contour problem are that, first, the equation at each front update does not need to be solved at discretization accuracy since the steady state solution matters mostly, and, second, the solution to each step provides a good initial guess for the next step. The first property means that typically one multigrid cycle at each front update suffices; further cycles have usually no perceptible effect on the final solution. The second implies that multigrid correction schemes such as the V- and W-cycles of Fig. 1 better fit our application since they exploit the good initial guess to the solution. In practice, we utilize the $V(1,1)$ or even the so-called sawtooth $V(0,1)$ multigrid schedule, where $V(n_1, n_2)$ denotes a V-cycle with n_1 pre- and n_2 postsmoothing steps per level; both are computationally cheap and sufficiently accurate for our application.

V. EXPERIMENTS AND COMPARISONS

We have tested the performance of the proposed algorithm in conjuction with various of the geometric active contour models of Section II in segmentation experiments. We have experimented with two popular publicly available multigrid (MG) codes: 1) the matrix-dependent multigrid code `mgd9v` [62] (MDMG); 2) the algebraic multigrid code `amg1r5` [63] (AMG). For distance transform (DT) calculations, we have implemented the separable DT transform algorithm of [36] (SDT) and the FM method [25] (FMDT). Our implementation of the AOS scheme [30] utilizes LAPACK tri-diagonal solvers (TRI)

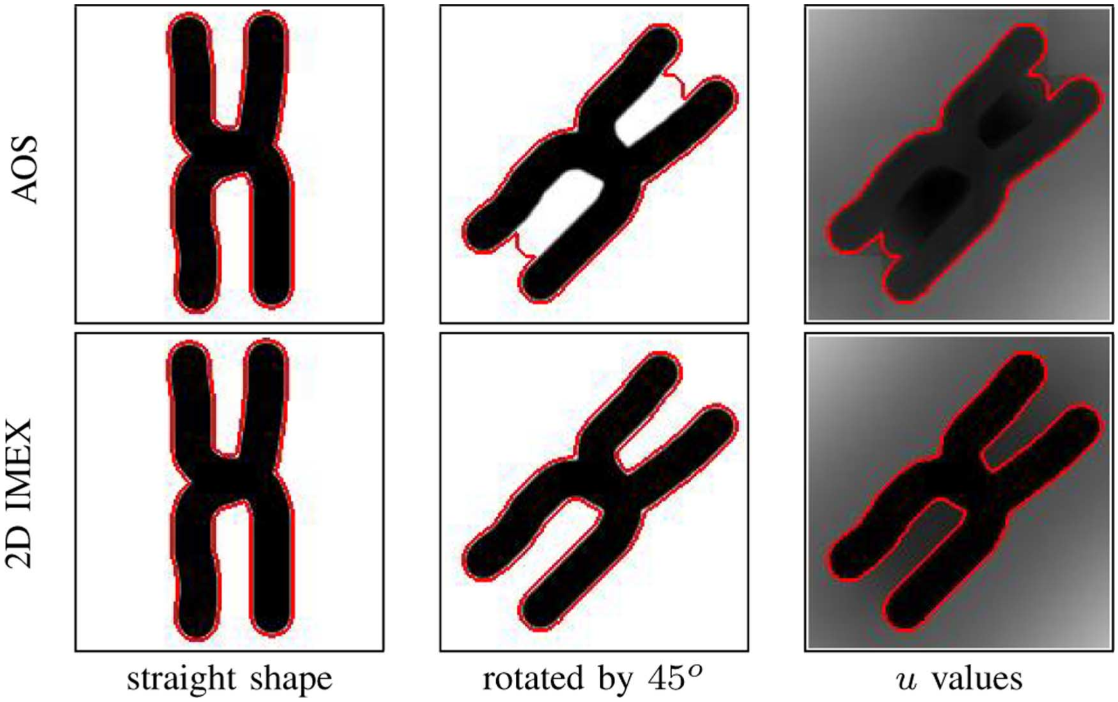


Fig. 4. Anisotropicity artifact of the AOS scheme. Top: The AOS scheme gives different results after a 45° rotation of the image. This is due to the shading effect of the AOS scheme, better visualized in the graph of the embedding function u (third column). Bottom: The rotation has no effect on the result of the fully 2-D IMEX scheme. (geodesic active contour, $c = 0$, $\tau = 1000$).

without row pivoting, since the system matrices are strictly diagonally dominant. Contour convergence is established as soon as the Hausdorff distance between two consecutive snapshots of the front, computed as discussed in Section III-C, is less than H_T; typically $H_T = 1 - 5$ pixels. Execution times refer to our C++ implementation[2] and a laptop computer with a Pentium M710 1.4-GHz processor.

In Table I, we give execution times on typical 256×256 images (without narrowband) of the main algorithmic components, namely: 1) *front update*, using either our fully 2-D IMEX scheme and multigrid solution of the linear system (13) (both MDMG and AMG have been tested), or the AOS scheme in conjuction with tridiagonal system solvers; 2) *front reinitialization* with distance transform algorithms. For both multigrid and tridiagonal algorithms, we give separately the setup overhead and the cost per time step: by re-initializing the front, unless we employ narrowband, the system matrices do not change; hence, in either case, solver setup (multigrid: computation of prolongation/restriction operators and coarse-grid matrices; tridiagonal solver: LU-decomposition) needs to be performed only once.

As can be seen from Table I, the setup overhead is substantial for AMG, but much lighter for MDMG; this clearly makes MDMG more efficient than AMG, even if we combine AMG with narrowband as we did in [39]: although fewer variables are involved in each narrowband iteration, most pixels are typically included in the narrow-band at some iteration of the algorithm and the costly AMG setup needs to be repeated every time the domain changes. The cost of both MDMG and AMG algorithms per front update is low, since one V(0,1) multigrid cycle suffices to suppress the error sufficiently. Especially for MDMG, the cost per front update is of the same order with the cost of split schemes like LOD/AOS, while retaining all the advantages of the fully 2-D IMEX scheme. For the purely region-based $(g \equiv 1)$ active contour models, for which the cheaper standard multigrid components of Section IV-A are applicable, one could expect a further efficiency improvement. Regarding distance transform (DT) computations, the separable algorithm SDT is faster than the alternative FMDT method. We see that the combination of matrix-dependent multigrid and separable distance transform components $(\mathrm{MDMG+SDT})$ costs ≈ 50 msec for each front update, thus allowing 20 non-narrowband updates/sec; since very few front updates with the big time steps allowed by the fully 2-D IMEX scheme (12) suffice for the contour to converge, we typically segment 256×256 images in less than a second. The computational cost also scales *linearly* with the number of pixels N for $\mathrm{MDMG+SDT}$; the experiments reported next have been conducted with this advantageous choice for multigrid/distance transform algorithms.

We demonstrate next the improved accuracy and rotational invariance properties of the fully 2-D IMEX scheme over the AOS split scheme. In Fig. 4, we illustrate on a synthetic image an anisotropicity artifact of the AOS scheme which we call *shading effect*. Since diffusion is carried out independently in the x and y directions, cavities with interior not completely visible by an observer who scans the image only in these directions do not get "lit" properly and the contour cannot get into them. The fully 2-D IMEX scheme faces no such problem because it permits omni-directional diffusion. The better qualitative properties of the fully 2-D IMEX discretization are further illustrated in the segmentation detail of Fig. 5; we notice that, even for time steps as small as $\tau = 2$, the AOS scheme can lead to visible artifacts.

[2] To facilitate experimentation with the proposed algorithms, we make our source code publicly available at http://cvsp.cs.ntua.gr/gpapan

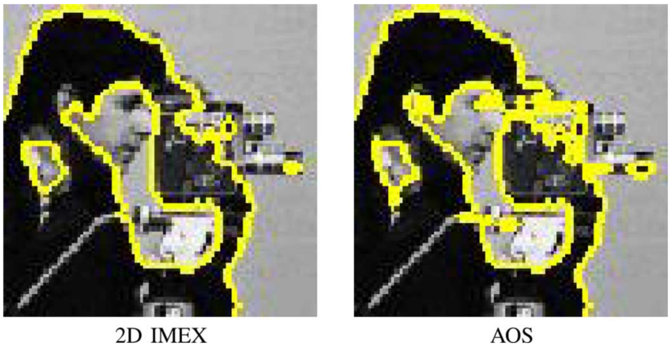


Fig. 5. Fully 2-D IMEX versus AOS discretizations. The AOS scheme does not regularize properly the curve and favors the formation of lines parallel to the axes. Chan–Vese segmentation of Cameraman image (detailed view), $\tau = 2$.

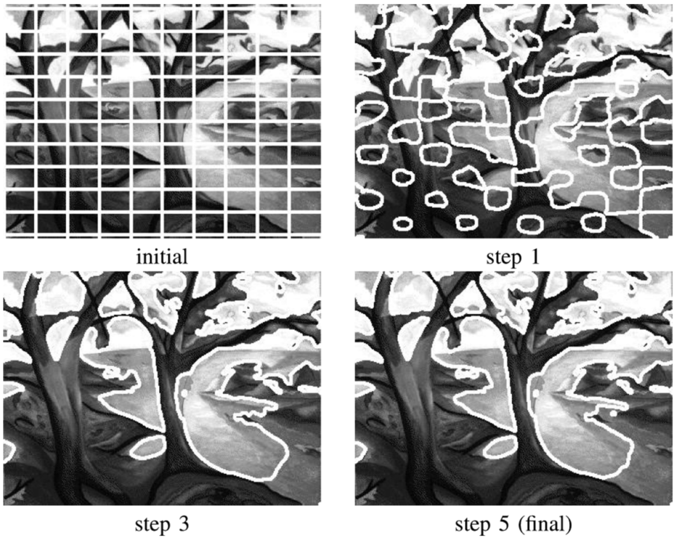


Fig. 6. Chan–Vese segmentation of a 350 × 258 image. Convergence after five iterations ($\tau = 50$, execution time 0.55 s).

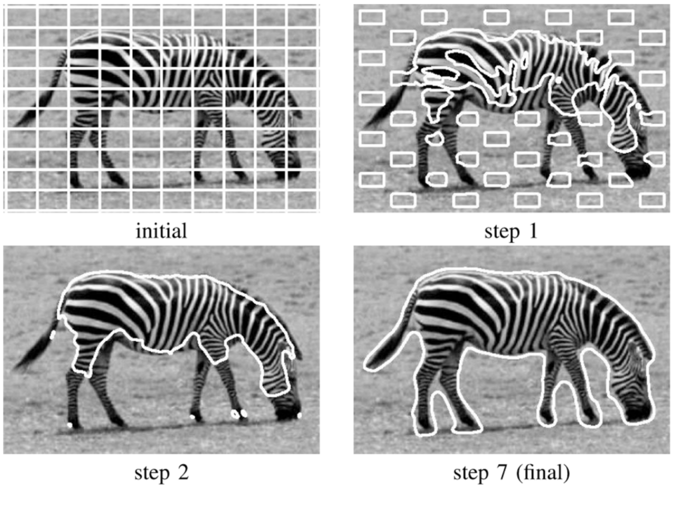


Fig. 7. Structure tensor-based texture segmentation [15] of a 436 × 286 image. Convergence after seven iterations ($\tau = 25$, execution time 1.8 s).

Two more region-based segmentation examples employing our algorithm are presented next; the one in Fig. 6 uses the Chan–Vese model and the other in Fig. 7 uses the unsupervised structure tensor-based texture segmentation approach of [15]. In both cases, the front converges rapidly, successfully capturing the trees and the zebra, respectively.

VI. Conclusion

In this paper, we have presented a novel algorithm for the rapid evolution of geometric active contours. It utilizes a very stable fully 2-D implicit-explicit numerical scheme which exhibits excellent rotational invariance properties and relies on multigrid methods for the efficient solution of the occuring sparse linear system at each front update. We have discussed both its applicability in conjuction with a wide range of geometric active contour models and its accuracy in comparison to the AOS scheme. The experimental results we have presented show that the new multigrid algorithm can be a promising tool in the further adoption of geometric active contour methods in time-critical applications.

Acknowledgment

The authors would like to thank I. Kokkinos for his extensive feedback on an early manuscript and the reviewers for their constructive comments. The algorithm implementation has been facilitated by publicly available multigrid code written by P. M. de Zeeuw (`mgd9v`) and J. Ruge, K. Stüben, and R. Hempel (`amg1r5`).

References

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1988.

[2] A. Blake and M. Isard, *Active Contours*. New York: Springer, 1997.

[3] T. McInerney and D. Terzopoulos, "T-snakes: Topology adaptive snakes," *Med. Image Anal.*, vol. 4, pp. 73–91, 2000.

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

[5] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modelling with front propagation: a level set approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 2, pp. 158–175, Feb. 1995.

[6] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, 1997.

[7] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum, "A geometric snake model for segmentation of medical imagery," *IEEE Trans. Med. Imag.*, vol. 16, no. 2, pp. 199–209, Apr. 1997.

[8] T. Chan and L. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.

[9] A. Tsai, A. Yezzi, and A. Willsky, "Curve evolution implementation of the Mumford–Shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1169–1186, Aug. 2001.

[10] L. A. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *Int. J. Comput. Vis.*, vol. 50, no. 3, pp. 271–293, 2002.

[11] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, no. 5, pp. 577–685, 1989.

[12] S.-C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 18, no. 9, pp. 884–900, Sep. 1996.

[13] N. Paragios and R. Deriche, "Geodesic active regions: A new framework to deal with frame partition problems in computer vision," *J. Vis. Commun. Image Represent.*, vol. 13, no. 1-2, pp. 249–268, 2002.

[14] A. Yezzi, A. Tsai, and A. Willsky, "A statistical approach to snakes for bimodal and trimodal imagery," in *Proc. Int. Conf. Computer Vision*, 1999, vol. 2, pp. 898–903.

[15] M. Rousson, T. Brox, and R. Deriche, "Active unsupervised texture segmentation on a diffusion based feature space," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003, vol. II, pp. 699–704.

[16] S. Jehan-Besson, M. Barlaud, and G. Aubert, “DREAM^{2}S: Deformable regions driven by an Eulerian accurate minimization method for image and video segmentation,” *Int. J. Comput. Vis.*, vol. 53, no. 1, pp. 45–70, 2003.

[17] I. Kokkinos, G. Evangelopoulos, and P. Maragos, “Modulation-feature based textured image segmentation using curve evolution,” in *Proc. Int. Conf. Image Processing*, 2004, vol. 2, pp. 1201–1204.

[18] M. Leventon, E. Grimson, and O. Faugeras, “Statistical shape influence in geodesic active contours,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, vol. 1, pp. 316–323.

[19] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr, “Diffusion snakes: Introducing statistical shape knowledge into the Mumford–Shah functional,” *Int. J. Comput. Vis.*, vol. 50, no. 3, pp. 295–313, 2002.

[20] M. Rousson and N. Paragios, “Shape priors for level set representations,” in *Proc. Eur. Conf. Computer Vision*, 2002, vol. 2, pp. 78–92.

[21] I. Kokkinos and P. Maragos, “An expectation maximization approach to the synergy between image segmentation and object categorization,” in *Proc. Int. Conf. Computer Vision*, 2005, vol. I, pp. 617–624.

[22] S. Osher and J. A. Sethian, “Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations,” *J. Comput. Phys*, vol. 79, pp. 12–49, 1988.

[23] V. Kolmogorov and Y. Boykov, “What metrics can be approximated by geo-cuts, or global optimization of length/area and flux,” in *Proc. Int. Conf. Computer Vision*, 2005, pp. 564–571.

[24] L. D. Cohen and R. Kimmel, “Global minimum for active contour models: A minimal path approach,” *Int. J. Comput. Vis.*, vol. 24, no. 1, pp. 57–78, 1997.

[25] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[26] M. Sussman, P. Smereka, and S. Osher, “A level set approach for computing solutions to incompressible two-phase flow,” *J. Comput. Phys*, vol. 114, pp. 146–159, 1994.

[27] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang, “A PDE-based fast local level set method,” *J. Comput. Phys*, vol. 155, pp. 410–438, 1999.

[28] D. Geiger, A. Gupta, L. A. Costa, and J. Vlontzos, “Dynamic programming for detecting, tracking, and matching deformable contours,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 3, pp. 294–302, Mar. 1995.

[29] T. Lu, P. Neittaanmäki, and X.-C. Tai, “A parallel splitting up method and its application to Navier-Stokes equations,” *Appl. Math. Lett.*, vol. 4, no. 2, pp. 25–29, 1991.

[30] J. Weickert, B. t. H. Romeny, and M. Viergever, “Efficient and reliable schemes for nonlinear diffusion filtering,” *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 398–410, Mar. 1998.

[31] R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky, “Fast geodesic active contours,” *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1467–1475, Oct. 2001.

[32] J. Weickert and G. Kühne, “Fast methods for implicit active contour models,” in *Geometric Level Set Methods in Imaging, Vision and Graphics*, S. Osher and N. Paragios, Eds. New York: Springer, 2003.

[33] U. M. Ascher, S. J. Ruuth, and B. T. R. Wetton, “Implicit-explicit methods for time-dependent partial differential equations,” *SIAM J. Numer. Anal.*, vol. 32, no. 3, pp. 797–823, 1995.

[34] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*. New York: Academic, 2001.

[35] S. Acton, “Multigrid anisotropic diffusion,” *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 280–291, Mar. 1998.

[36] P. Felzenszwalb and D. Huttenlocher, “Distance Transforms of Sampled Functions,” Tech. Rep. TR2004-1963, Cornell Comput. Inf. Sci., Cornell Univ., Ithaca, NY, 2004.

[37] A. Kenigsberg, “A multigrid approach for fast geodesic active contours,” presented at the Copper Mountain Conf. Multigrid Methods, 2001.

[38] A. Kenigsberg, R. Kimmel, and I. Yavneh, “A Multigrid Approach for Fast Geodesic Active Contours,” Tech. Rep. CIS-2004-06, Technion–Israel Inst. Technol., Haifa, 2004.

[39] G. Papandreou and P. Maragos, “A fast multigrid implicit algorithm for the evolution of geodesic active contours,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognit.*, 2004, vol. II, pp. 689–694.

[40] L. D. Cohen, “On active contour models and balloons,” *Comput. Vis. Graph. Image Process.: Image Understand.*, vol. 53, no. 2, pp. 211–218, 1991.

[41] R. Kimmel and A. M. Bruckstein, “Regularized laplacian zero crossings as optimal edge integrators,” *Int. J. Comput. Vis.*, vol. 53, no. 3, pp. 225–243, 2003.

[42] P. Maragos and M. A. Butt, “Curve evolution, differential morphology, and distance transforms applied to multiscale and eikonal problems,” *Fundam. Inf.*, vol. 41, pp. 91–129, 2000.

[43] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[44] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. New York: Springer-Verlag, 2003.

[45] J. Weickert, *Anisotropic Diffusion in Image Processing*. Stuttgart, Germany: Teubner-Verlag, 1998.

[46] W. Ames, *Numerical Methods for Partial Differential Equations*, 3rd ed. New York: Academic, 1992.

[47] D. Barash, T. Schlick, M. Israeli, and R. Kimmel, “Multiplicative operator splittings in nonlinear diffusion: From spatial splitting to multiple timesteps,” *J. Math. Imag. Vis.*, vol. 19, no. 1, pp. 33–48, 2003.

[48] P. Maragos, “PDEs for morphological scale-spaces and eikonal applications,” in *The Image and Video Processing Handbook*, A. Bovik, Ed., 2nd ed. New York: Academic, 2005, ch. 4.16.

[49] C. Xu and J. Prince, “Snakes, shapes, and gradient vector flow,” *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 359–369, Mar. 1998.

[50] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, “Comparing images using the Hausdorff distance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.

[51] A. Brandt, “Multi-level adaptive solutions to boundary-value problems,” *Math. Comput.*, vol. 31, no. 138, pp. 333–390, 1977.

[52] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd ed. Philadelphia, PA: SIAM, 2000.

[53] P. Wesseling, *An Introduction to Multigrid Methods*. New York: Wiley, 1992.

[54] D. Terzopoulos, “Image analysis using multigrid relaxation methods,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 2, pp. 129–139, Mar. 1986.

[55] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schjörr, “Variational optical flow computation in real-time,” *IEEE Trans. Image Process.*, vol. 14, no. 5, pp. 608–615, May 2005.

[56] E. Sharon, A. Brandt, and R. Basri, “Fast multiscale image segmentation,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, vol. I, pp. 70–77.

[57] R. Kimmel and I. Yavneh, “An algebraic multigrid approach for image analysis,” *SIAM J. Sci. Comput.*, vol. 24, no. 4, pp. 1218–1231, 2003.

[58] F. Heitz, P. Pérez, and P. Bouthemy, “Multiscale minimization of global energy functions in some visual recovery problems,” *Comput. Vis. Graph. Image Process.: Image Understand.*, vol. 59, no. 1, pp. 125–134, 1994.

[59] B. Leroy, I. L. Herlin, and L. Cohen, “Multiresolution algorithms for active contour models,” in *Proc. Int. Conf Anal. Optimiz. Syst.*, 1996, pp. 58–65.

[60] A. Willsky, “Multiresolution markov models for signal and image processing,” *Proc. IEEE*, vol. 90, no. 8, pp. 1396–1458, Aug. 2002.

[61] R. E. Alcouffe, A. Brandt, J. E. Dendy, and J. W. Painter, “The multigrid method for the diffusion equation with strongly discontinuous coefficients,” *SIAM J. Sci. Statist. Comput.*, vol. 2, pp. 430–454, 1981.

[62] P. M. d. Zeeuw, “Matrix-dependent prolongations and restrictions in a blackbox multigrid solver,” *J. Comput. Appl. Math.*, vol. 33, pp. 1–27, 1990.

[63] J. W. Ruge and K. Stüben, “Algebraic multigrid,” in *Multigrid Methods*, S. Mc Cormick, Ed. Philadelphia, PA: SIAM, 1987.

[64] Y. Shapira, *Matrix-Based Multigrid*. Norwell, MA: Kluwer, 2003.

George Papandreou (S’03) received the Diploma in electrical and computer engineering (with highest honors) in 2003 from the National Technical University of Athens, Athens, Greece, where he is currently pursuing the Ph.D. degree.

As an undergraduate, he was a Research Associate with the Institute of Informatics and Telecommunications of the Greek National Center for Scientific Research “Demokritos” (2001 to 2003). His current research interests include computer vision, image restoration, and probabilistic approaches to multimodal fusion.

Petros Maragos (S'81–M'85–SM'91–F'95) received the Diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1980, and the M.Sc.E.E. and Ph.D. degrees from the Georgia Institute of Technology (Georgia Tech), Atlanta, GA, in 1982 and 1985.

In 1985, he joined the faculty of the Division of Applied Sciences at Harvard University, Cambridge, MA, where he was a Professor of electrical engineering for eight years affiliated with the interdisciplinary Harvard Robotics Lab. He has also been a Consultant to several industry research groups, including Xerox's research on image analysis. In 1993, he joined the faculty of the School of Electrical and Computer Engineering, Georgia Tech. During parts of 1996 to 1998, he was on sabbatical and academic leave working as a Research Director with the Institute for Language and Speech Processing, Athens. In 1998, he joined the faculty of the National Technical University of Athens, where he is currently a Professor of electrical and computer engineering. His current research and teaching interests include the general areas of signal processing, systems theory, communications, pattern recognition, and their applications to image processing and computer vision, and computer speech processing and recognition.

He has served as Associate Editor for the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (currently), Guest Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, and an editorial board member for *Signal Processing* (currently) and *Visual Communications and Image Representation*; General Chairman and Co-Chairman of image and signal processing conferences; and a member of IEEE DSP committees. His research work has received several awards, including a 1987 U.S. National Science Foundation Presidential Young Investigator Award; the 1988 IEEE Signal Processing Society's Young Author Paper Award for the paper "Morphological Filters"; the 1994 IEEE Signal Processing Society's Senior Award and the 1995 IEEE Baker Award for the paper "Energy Separation in Signal Modulations with Application to Speech Analysis"; the 1996 Pattern Recognition Society's Honorable Mention Award for the paper "Min-Max Classifiers"; and election to IEEE Fellow for his contributions to the theory and applications of nonlinear signal processing systems.