Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 28 (2007) 884–893
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Stability and convergence of the level set method in computer vision

Kunal N. Chaudhury *, K.R. Ramakrishnan

Department of Electrical Engineering, Indian Institute of Science, Bangalore, India

Received 14 June 2005; received in revised form 5 October 2006
Available online 1 February 2007

Communicated by H.H.S. Ip

Abstract

Several computer vision problems, like segmentation, tracking and shape modeling, are increasingly being solved using level set methodologies. But the critical issues of stability and convergence have always been neglected in most of the level set implementations. This often leads to either complete breakdown or premature/delayed termination of the curve evolution process, resulting in unsatisfactory results. We present a generic convergence criterion and also a means of determining the optimal time-step involved in the numerical solution of the level set equation. The significant improvement in the performance of level set algorithms, as a result of the proposed changes, is demonstrated using object tracking and shape-contour extraction results.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Level-set method; Tracking; Curve evolution; Difference equation; Numerical stability; Courant–Friedrichs–Levy condition

1. Introduction

The level set approach has been extensively used in computer vision for motion segmentation (Mansouri and Konrad, 2003), object tracking (Chang et al., 2004; Paragios and Deriche, 2000) and shape modeling (Malladi et al., 1995). All these problems are formulated using curve (contour) evolution concepts. The central idea of the level set technique, as developed by (Adalsteinsson and Sethian, 1995; Osher and Sethian, 1988), is to evolve a higher dimensional function whose zero-level set always corresponds to the position of the propagating contour. Compared to the *active contours* methods (Kass and Witkin, 1988), often used in the curve evolution context, the advantages of the level set methodology are its ability to handle merging/splitting of the evolving contour, numerical stability of the solution and the elegance of the implicit representation of the evolving contour using the zero-level set concept. Also the final extracted contour is independent of the curve initialization, unlike other active contour models such as the snakes (Kass and Witkin, 1988), where the final object contour is very much determined by the contour initialization. The only major drawback of this level set method is that by embedding the evolving contour as the zero-level set of a higher dimensional function, a one-dimensional curve evolution problem is transformed into a two-dimensional problem. This adds to the computational complexity and renders the standard level set method incapable of real time implementation. To reduce the computational labor involved, two well-known schemes exist, namely the *Narrow Band* and the *Fast Marching* approach (Adalsteinsson and Sethian, 1995). Paragios and Deriche (2000) have proposed a new approach that combines the Narrow Band and Fast Marching methods by employing the idea of selective propagation over a small window.

Level set methodologies ultimately involve the solution of a continuous spatio-temporal partial differential equation. While a closed form solution has not been found as

* Corresponding author. Present address. Biomedical Imaging Group, Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland.
E-mail addresses: kunal.chaudhury@epfl.ch (K.N. Chaudhury), krr@ee.iisc.ernet.in (K.R. Ramakrishnan).

0167-8655/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2006.12.003

yet, an approximate numerical solution of the differential equation (using a finite difference scheme) has been used in different level set based applications (Mansouri, 2002; Mansouri and Konrad, 2003; Chang et al., 2004; Osher and Sethian, 1988). But in all these level set based applications, the critical issues of numerical stability and convergence have been ignored. The time-step, involved in the numerical solution of the differential equations, has been assigned an arbitrary value independent of the speed of the evolution process. Also, arbitrary convergence (stopping) criteria have been employed for the iterative solution of the level set equation (Mansouri, 2002; Malladi et al., 1995). Either these issues have not been attended to Chang et al. (2004), or the time-step value and the convergence criteria employed do not guarantee satisfactory results (Mansouri, 2002; Malladi et al., 1995; Mansouri and Konrad, 2003). We discuss the stability issues related to the solution of the difference equations, in general, and the level set equation in particular. Based on these, we present a scheme for determining the optimal time-step. We also propose a generic convergence criteria based on the length of the contour at different stages of the evolution process.

2. Level set method

Given a closed $(N-1)$ dimensional hyper-surface $\gamma(t=0)$, we briefly discuss the *Eulerian* formulation for the motion of the hyper-surface $\gamma(t)$, propagating along its normal direction with speed $F(x)$, where $x \in \Re^N$. The main idea of the level set methodology is to embed the propagating interface as the zero-level set of a higher dimensional function $\Phi(x,t)$. Let $\Phi(x,t=0)$ be defined by

$$\Phi(x,t=0) = \pm d \tag{1}$$

where d is the distance from x to $\gamma(t=0)$, and the plus (minus) sign in Eq. (1) is chosen if the point x is outside (inside) the initial hyper-surface $\gamma(t=0)$. Thus, we have an initial function $\Phi(x,t=0) : \Re^N \to \Re$ with the property that

$$\gamma(t=0) = \{x | \Phi(x,t=0) = 0\} \tag{2}$$

The goal is to produce an equation for the evolving function $\Phi(x,t)$ that contains the embedded motion of $\gamma(t)$ as the zero level set $\Phi = 0$. The spatio-temporal partial differential equation describing the dynamics of the level set function, introduced by Osher and Sethian (1988), is given by

$$\frac{\partial \Phi(x(t),t)}{\partial t} + F(x(t))|\nabla \Phi(x(t),t)| = 0, \quad \text{given } \Phi(x,t=0) \tag{3}$$

We refer to this as the *Eulerian formulation* of the curve evolution problem because the underlying coordinate system remains fixed. The approximate discrete solution of the differential equation (3) obtained by using a finite grid in the spatial domain and by substitution of finite difference approximations for the spatial and temporal derivatives, also proposed by Osher and Sethian, is as given below:

$$\Phi_{ij}^{n+1} = \Phi_{ij}^{n} - \Delta t(\max(F_{ij},0)\Delta^{+} - \min(F_{ij},0)\Delta^{-}) \tag{4}$$

where Δt is the finite time-step of the above iterative solution, F_{ij} is the speed function at the grid location (i,j), and

$$\begin{aligned}\Delta^{+} &= [\max(D_{ij}^{-x})^2 + \min(D_{ij}^{+x})^2 + \max(D_{ij}^{-y})^2 + \min(D_{ij}^{+y})^2]^{1/2} \\ \Delta^{-} &= [\max(D_{ij}^{+x})^2 + \min(D_{ij}^{-x})^2 + \max(D_{ij}^{+y})^2 + \min(D_{ij}^{-y})^2]^{1/2}\end{aligned} \tag{5}$$

where

$$\begin{aligned}D_{ij}^{-x} &= (\Phi_{i,j}^{n} - \Phi_{i-1,j}^{n})/h_x \\ D_{ij}^{+x} &= (\Phi_{i+1,j}^{n} - \Phi_{i,j}^{n})/h_x \\ D_{ij}^{-y} &= (\Phi_{i,j}^{n} - \Phi_{i,j-1}^{n})/h_y \\ D_{ij}^{+y} &= (\Phi_{i,j+1}^{n} - \Phi_{i,j}^{n})/h_y\end{aligned} \tag{6}$$

are the forward and backward differences of the level set function $\Phi_{i,j}^{n}$ at some particular time iteration n and grid location (i,j). h_x and h_y are the grid spacings in the x and y direction, respectively.

3. Stability and CFL condition

In most level set implementations, the value of the time-step Δt has been arbitrarily assigned. A large time-step speeds up the inherently slow curve evolution process, but the question is: how large value of Δt can we use? In (Mansouri, 2002; Malladi et al., 1995) an arbitrarily assigned $\Delta t = 0.0001$ is used, while in (Mansouri and Konrad, 2003), $\Delta t = 7 \times 10^{-5}$ is used. While approximating the solution to differential equation in the continuous domain (3) by a finite difference scheme (4), the question of numerical stability arises. In fact, we implemented the difference equation (4) with different time-steps Δt. We observed that, for small values of the time-step, though the convergence was slow, curve evolution results were satisfactory. When large time-steps were used to speed up the evolution process, we often got garbage outputs. Clearly there is a stability issue that depends on the value of the time-step Δt employed.

We clarify what stability we are talking about. While solving a differential equation numerically using a difference scheme, we encounter approximation errors at each iteration. If these approximation errors keep accumulating over time, the process might become unstable and can cause the solution to go haywire. However, the interesting thing is that, even if we had used a very accurate scheme, the same problem could arise. One might wonder how this can happen since we know that the scheme is accurate!

The answer is that the accuracy considers only local error, whereas the stability problem that is occurring in this case is a global issue. This is exactly where the Courant–Friedrichs–Levy (CFL) condition comes into the picture. This states that the numerical range of dependence must always contain the theoretical range of dependence of the

solution or instability will occur. Otherwise, the scheme, no matter how accurate, will always magnify errors in each step and therefore be unstable. The application of the above principle to the level set theory yields the following stability condition:

$$F_{\max} \cdot \Delta t \leqslant \min(h_x, h_y) \tag{7}$$

where h_x and h_y are the grid spacing in the x and y direction, respectively and $F_{\max}$ is the maximum absolute speed of all the points on the grid. This means that the evolving contour can cross no more than one grid at each time step. Thus, the CFL condition places an upper bound on the time-step Δt and hence on the rate of curve evolution. Hence, the need for an optimal time-step, which not only avoids numerical instability but also keeps the rate of curve evolution as high as possible.

4. Convergence criteria

In almost all level set based curve evolution problems (Mansouri, 2002; Malladi et al., 1995; Mansouri and Konrad, 2003), the evolution process is stopped when the iteration count n exceeds a particularly large threshold N_0. In (Mansouri, 2002) an N_0 of 1000 and in (Mansouri and Konrad, 2003) an N_0 of 20,000 are employed. Thus, they evolve the curve for a large number of iterations assuming that, at the end of the process, the desired contour will be obtained. But, there are few inherent problems associated with this form of stopping criteria, specially when a large value of N_0 is used.

- The desired contour might have been obtained long before the iteration process ends. Thus, a lot of computational resource is wasted, and considering the time complexity of the level set implementations, this unnecessarily increases the execution time.
- Ideally, the speed function which guides the dynamics of the higher dimensional function, must force the embedded curve to slow down and finally attain steady-state at a certain stage of the curve evolution process. Very often, the speed function is "leaky" in the sense that it cannot force the evolution process to reach steady-state and hence the curve never stops propagating. The "leaky" nature is mostly due to aberrant nature of the data based on which the speed function is computed. Thus, if the iteration process is continued for a long time, the results would be unsatisfactory.

These two issues make it obvious that we cannot run the iteration process for an arbitrarily long time. Also, we must be careful so as not to stop the process prematurely. Hence, the need for a convergence criteria. We propose a general convergence criteria which stops the curve evolution process at the correct instant giving satisfactory results. This requires us to determine the length of the evolving contour at various stages of the iteration process. We store the current length of the contour in a variable $L_c(n)$. When the curve evolution process reaches steady-state, it is obvious that the variable $L_c(n)$ remains almost constant. Based on these observations, we state the convergence criteria:

If the iteration count n exceeds a particular threshold N_0 and the rate of change of contour length $\left|\frac{dL_c(t)}{dt}\right| < \epsilon$ over a fixed number of iterations Δn, STOP the iteration process. The zero crossings at that instant gives the desired contour.

N_0 is chosen sufficiently large to ensure that the convergence criteria is never met in the initial stages of the curve evolution process, when the evolution process is slow. Thus, only when n exceeds N_0, we check for convergence. If the condition $\left|\frac{dL_c(t)}{dt}\right| < \epsilon$ is satisfied at a later stage, we do not stop the process immediately. Instead, we check whether the condition is maintained over a fixed number of iterations Δn before halting the process. This check needs to be performed because the evolution process often slows down temporarily even before steady-state is reached. Thus, only if the rate of evolution is low for a long time we terminate the process. As far as the final results are concerned, the choice of the thresholds is very flexible. Typical values are $N_0 = 500$, $\epsilon = 5$ pixels/iteration and $\Delta n = 50$.

5. Results: level set based algorithms

5.1. Object tracking

To demonstrate the criticality of the stability and convergence issues in level set based algorithms, we use a contour-based object tracking algorithm similar to the one described in (Mansouri, 2002). We modify the algorithms suitably so as to introduce the concept of an optimal time-step Δt_{opt} and also replace their stopping criteria with our proposed convergence criteria. Finally, we compare the results obtained using our algorithm with that using the algorithms in (Mansouri, 2002). We have used images (frames) from two real video sequences to demonstrate the superiority of our modified level set algorithm.[1]

Assume that I_0 and I_1 are the consecutive color images, indexed by coordinates (i,j). Let Φ_{ij}^n denote the level set function value at coordinate (i,j) on the grid at iteration n. Starting from a region R_0 in I_0, we wish to estimate a region $\widehat{R}_1$ in image I_1. Let h_x and h_y be the grid spacing in the x and y directions respectively. The level set algorithm, using the optimal time-step Δt and the proposed convergence criteria is given below:

(1) $n := 0$, $k := 0$, flag $:= 0$.
(2) For all grid points (i,j):
- if $(i,j) \in R_0$, $\Phi_{ij}^0 = + \min_{(h,k)\in\delta R_0} \max(|i-h|h_x, |j-k|h_y)$,
- if $(i,j) \in R_0^c$, $\Phi_{ij}^0 = - \min_{(h,k)\in\delta R_0} \max(|i-h|h_x, |j-k|h_y)$,

where δR_0 is the boundary of the region R_0.

[1] One of then being the popular "hall-monitor" video sequence.

(3) For all grid points (i,j):

$$F(i,j) = \min_{\{l_1,l_2:\|l_1\|\leqslant\delta,\|l_2\|\leqslant\delta,(i+l_1,j+l_2)\in R_0\}} \|I^1(i,j) - I^0(i+l_1,j+l_2)\|^2 + \min_{\{l_1,l_2:\|l_1\|\leqslant\delta,\|l_2\|\leqslant\delta,(i+l_1,j+l_2)\in R_0^c\}} \|I^1(i,j) - I^0(i+l_1,j+l_2))\|^2$$

(4) The optimal time-step Δt is given by

$$\Delta t_{\text{opt}} = c \cdot \min(h_x, h_y)/F_{\max}$$

where $F_{\max}$ is the maximum absolute speed and c (=0.75–0.90) is the factor of safety.

(5) For all grid points (i,j)

$$\Phi_{ij}^{n+1} = \Phi_{ij}^{n} - \Delta t_{\text{opt}}(\max(F_{ij},0)\Delta^{+} - \min(F_{ij},0)\Delta^{-})$$

where the symbols have the same meaning as in Eqs. (6) and (5).

(6) $n := n+1$.

(7) If $n > N_0$, based on the zero level set of $\Phi^n(i,j)$, determine the current contour $C(n)$ and its length $L_c(n)$. Else go to step (5).

(8) Perform the convergence test.

- case (a): if $|L_c(n) - L_c(n-1)| < \epsilon$ AND flag $= 0$ flag $:= 1$, $k := k+1$, go to step (5),
- case (b): if $|L_c(n) - L_c(n-1)| < \epsilon$ AND flag $= 1$ if $k > \Delta N$, go to step (9) else $k := k+1$, go to step (5),
- case (c): if $|L_c(n) - L_c(n-1)| > \epsilon$ AND flag $= 1$ flag $:= 0$, $k := 0$, go to step (5).

(9) STOP. The estimated region is $\widehat{R}_1 = \{(i,j) : \Phi(i,j) > 0\}$.

In Fig. 1, we show the tracking results obtained in three successive frames using our tracking algorithm, involving the optimal time-step and the convergence criteria. Fig. 1a shows the person to be tracked and his contour in white. The tracking problem is to track the person (right side) by tracking his contour in the successive frames. Fig. 1b–d shows the tracked person and his contour in three successive frames obtained using our algorithm. To appreciate the results shown in Fig. 1, we need to simulate situations which demonstrate the criticality of the stability and the convergence issues.

To demonstrate the superiority of our algorithm, we simulate two situations. In the first situation, the time-step used deliberately violates the CFL condition (7), for some given speed function F. This situation can arise when the time-step is arbitrarily set with complete disregard to the speed function information that dictates the rate of curve evolution (Mansouri, 2002; Mansouri and Konrad, 2003; Malladi et al., 1995). In Fig. 2, we compare the results of the tracking algorithm used in (Mansouri, 2002) with that of our algorithm. Using an arbitrary time-step $\Delta t = 0.001$, that violates the CFL condition, obscure results are obtained after 500 time-iterations due to the numerical instability (see Fig. 2b). However, as shown in Fig. 2c, our algorithm avoids this instability problem by ensuring (using the speed function) that the optimal time-step value satisfies the CFL condition and hence gives satisfactory tracking results.

To further demonstrate the instability problem, we use a different example-the "hall-monitor" video sequence. As shown in Fig. 3, we try to use the same tracking algorithm

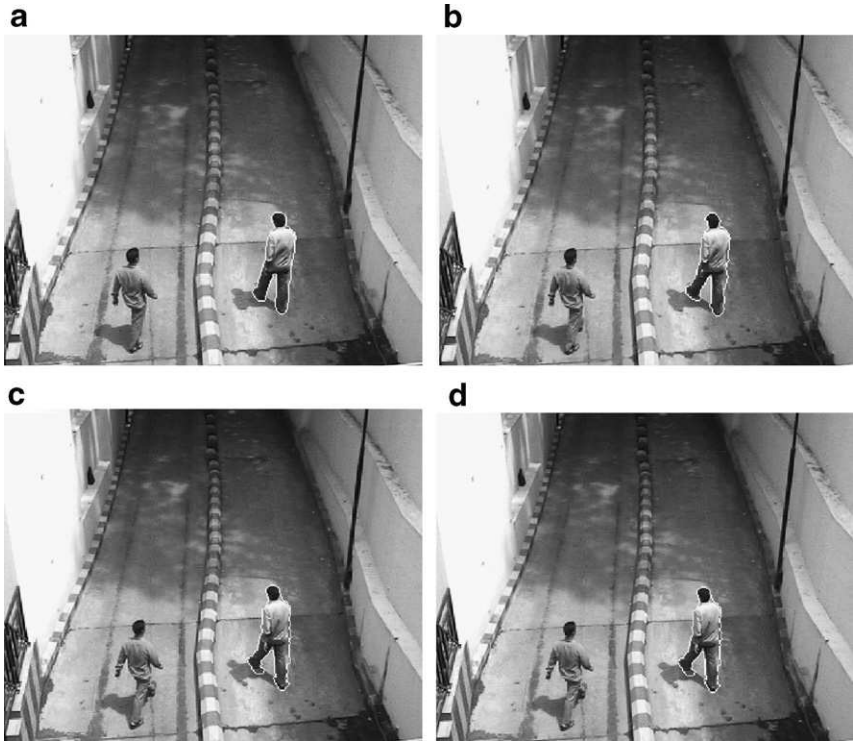

Fig. 1. Level set based tracking using our algorithm: (a) The person on the right as the tracked object and his contour is shown in white. (b)–(d) The tracked person and his contour in three successive frames obtained using the optimal time-step and the convergence test.

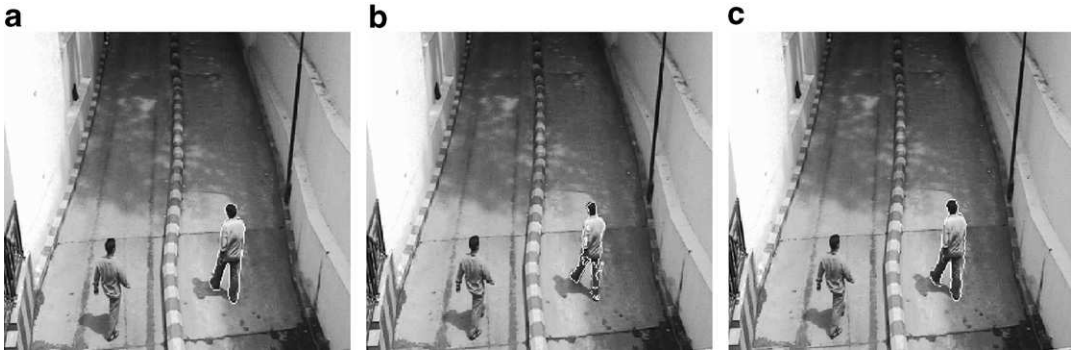

Fig. 2. Demonstration of the numerical instability issue: (a) The person on the right as the tracked object and his contour is shown in white. (b) and (c) The tracked person and his contour in two successive frames obtained using time-steps $\Delta t = 0.001$ (that violates the CFL condition) and $\Delta t_{\text{opt}} = 0.00035$ (that preserves the CFL condition), respectively.

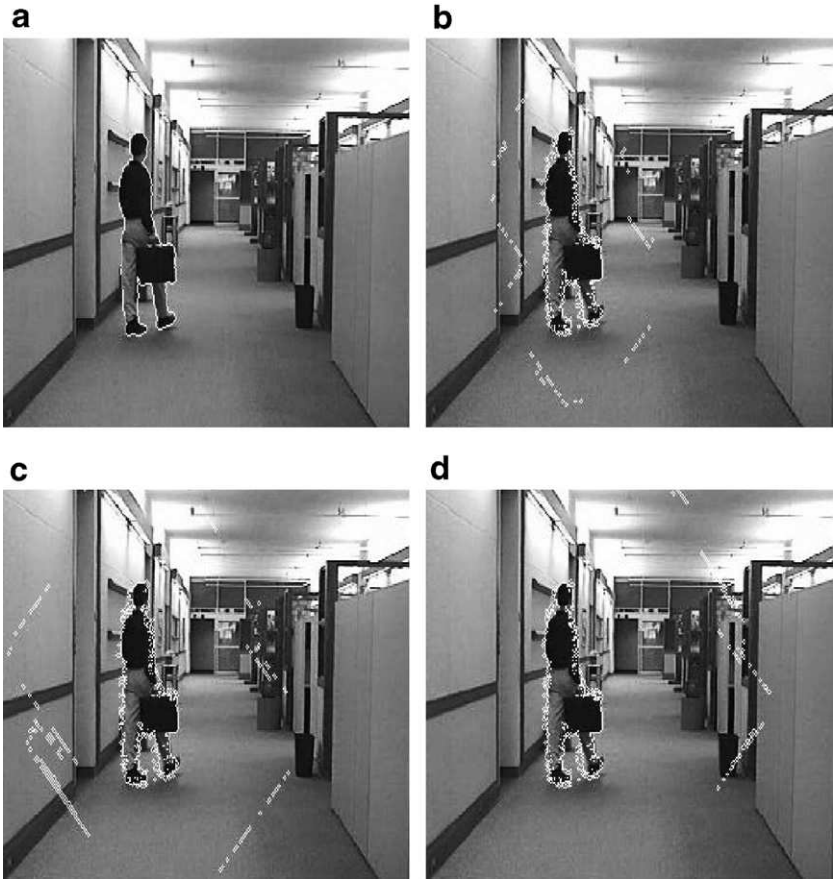

Fig. 3. Illustration of the instability problem: (a) The person as the tracked object in the first frame and his contour is shown in white. (b)–(d) The person and his evolving contour in the successive frame at intermediate steps of the evolution process. The time-step Δt was set to violate the CFL condition and inject numerical stability in tracking process.

to track the person from one frame to the next frame. Fig. 3a shows the person as the tracked object in the first frame and his contour is shown in white. Our problem is to track the person in the next (successive) video frame using the given contour in the first frame. We use the tracking algorithm described earlier to solve this problem. To demonstrate the adverse effects of numerical instability, the time-step Δt is set to deliberately violate the CFL condition and inject numerical stability in tracking process. This situation can arise when the time-step is arbitrarily set with complete disregard to the speed function information. Fig. 3b–d shows the person and the evolving contour at intermediate steps of the evolution process. It is obvious that the curve evolution process has broken down in the face of numerical instability and given rise to obscure and unsatisfactory tracking output.

Next, the above instability problem is deliberately suppressed by using a time-step value that satisfies the CFL condition, so as to highlight the importance of the convergence issue. In most level set algorithms, either the convergence criteria is not been explicitly mentioned (Chang et al., 2004) or the basis for the termination of the curve evolution process is not obvious (Mansouri, 2002; Mansouri and Konrad, 2003; Malladi et al., 1995). In the latter case, the process is arbitrarily stopped after a few thousand time iterations assuming that the propagating curve has finally locked on to the desired object boundary. In the absence of a good stopping criteria, the level set algorithm cannot detect the slowing down of the curve propagation in the vicinity of the desired object boundary. If the iteration process is not terminated at the right instant and the speed function is "leaky", the curve often permeates through the object boundary resulting in unsatisfactory outputs. Fig. 4b and c depicts the tracking results obtained in two successive frame as a result of the evolution process being blindly continued up to 1000 iterations, similar to Mansouri (2002). In spite of the assured numerical stability, the absence of an effective convergence criteria degrades the tracking output in each frame.

Using the above examples, we clearly demonstrated the relevance of the CFL condition in maintaining numerical stability and the role of a good convergence criterion that

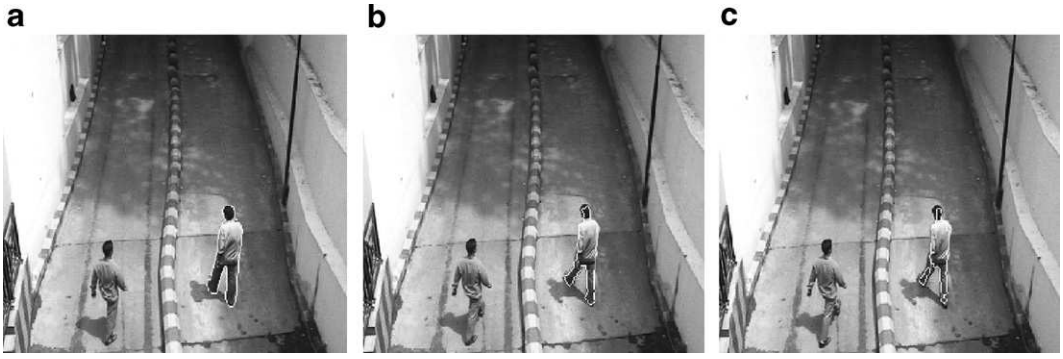

Fig. 4. Importance of the convergence criteria: (a) The person on the right as the tracked object and his contour is shown in white. (b) and (c) The tracked person and his contour in two successive frames using some poor convergence criteria. The time-step Δt was set to violate the CFL condition and inject numerical stability in either case.

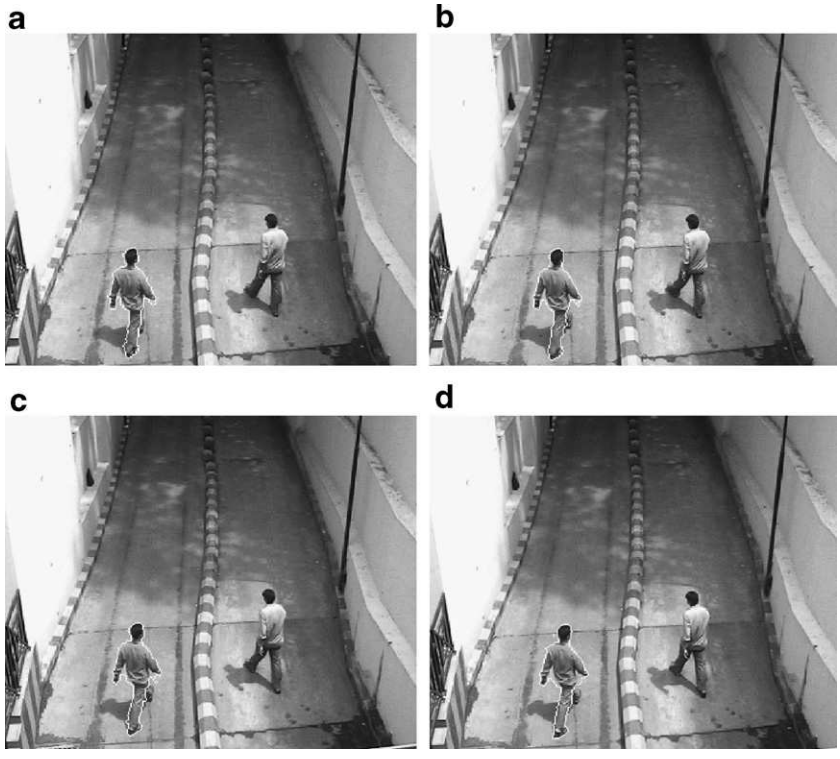

Fig. 5. Level set based tracking using our algorithm: (a) The person on the right as the tracked object and his contour is shown in white. (b)–(d) The tracked person and his contour in three successive frames obtained using the optimal time-step and the convergence test.

terminates the curve evolution process at the correct instant. We have incorporated both the stability and the convergence criteria in a tracking algorithm (Mansouri, 2002) and showed that the performance improves with these criteria.

In Fig. 5, we show the tracking results obtained in three successive frames of the earlier video sequence using our modified tracking algorithm. The algorithm involves the optimal time-step and the convergence criteria as described earlier. Fig. 5a shows the person (left side) to be tracked and his contour in white. The tracking problem is to track the person by tracking his contour in the successive frames. Fig. 5b–d shows the tracked person and his contour in three successive frames obtained using our algorithm.

In most of the above examples, the same video sequence is used. As a final demonstration of our tracking algorithm, we use images from the popular "hall-monitor" video sequence and try tracking the person in this particular sequence. Fig. 6a shows the person to be tracked and his contour is shown in white in a particular frame. The tracking problem is to track the person in the next frame by tracking his contour. Fig. 6b–i depicts the curve evolution process using two successive frames. Fig. 6b shows the person in the next frame along with the initial contour. The contour in the previous frame is used to initialize the level set function in next frame. This contour gradually evolves, as depicted in the Fig. 6c–i, to lock on to the true shape-contour of the person. After convergence of the contour (curve) evolution process, the final contour of the person in this next frame is show in Fig. 6i. The above tracking process continues for the succeeding frame and so on.

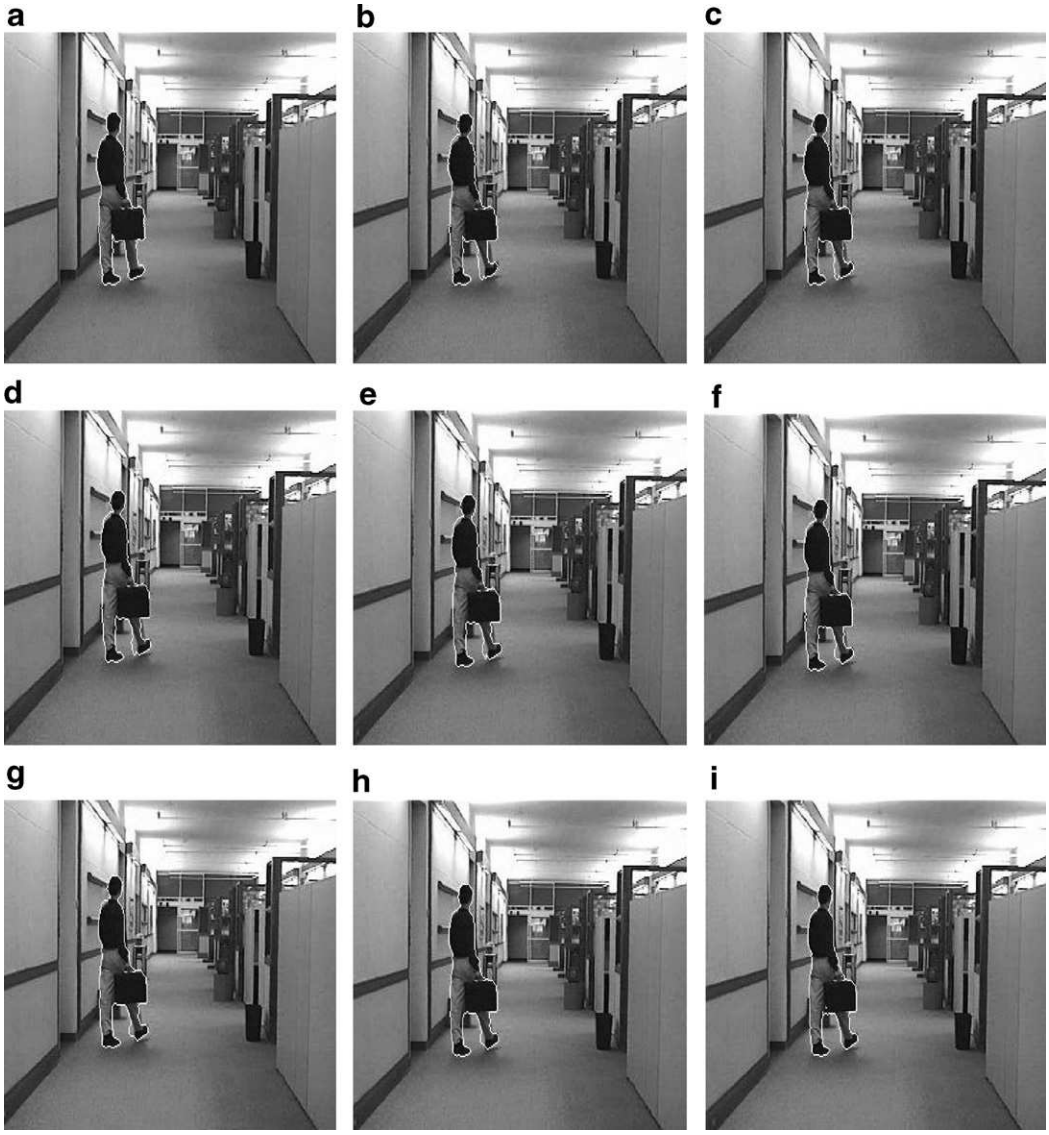

Fig. 6. Level set based tracking for the "hall-monitor" sequence: (a) The person as the tracked object in the first frame and his contour is shown in white. (b) and (i) The tracked person and his contour in the successive frame during intermediate stages of the curve evolution process. An optimal time-step Δt was used to ensure numerical stability along with our proposed convergence criteria.

5.2. Shape-contour extraction using curve evolution

In this section, we demonstrate the stability and convergence issues using the level set based shape-contour extraction algorithm similar to those described in (Malladi et al., 1995; Osher and Sethian, 1988). Discussion of underlying shape-contour extraction problem that leads to the level set implementation is beyond the scope of this paper, but the proposed algorithmic modifications would be similar to those described in the tracking algorithm in Section 5. The problem of shape-contour extraction is one in which we keep evolving a contour (curve), starting from some initial estimate, until it locks on to the true object boundary. Our prime motive is to demonstrate how the shape-contour extraction algorithms described in (Malladi et al., 1995; Osher and Sethian, 1988) overlooks the stability and convergence issues and how our proposed modifications improve the performance of these algorithms. The images from the popular "hall-monitor" video sequence are used for demonstrating the curve evolution process for shape-contour extraction.

To illustrate the consequences of using a time-step that violates the CFL condition we use $\Delta t = 0.01$. The upper bound of the time-step, as determined by (7) for some given speed function F, is $\Delta t = 0.0025$. Fig. 7 shows the results of the curve evolution process starting from the initial contour. Here the initial contour is the bounding box shown in white in Fig. 7a. Numerical instability injected into the evolution process is obvious from the results shown in the Fig. 7. After a few iterations, the system becomes unstable and entire curve evolution process breaks down. These figures demonstrate the importance of numerical stability and the CFL condition.

In Fig. 8, we have once more shown the results of the curve evolution process except for the fact that we have used a safe $\Delta t_{\text{opt}} = 0.001$ (<0.0025) to ensure the numerical stability of the evolution process. Using Fig. 8a–f, we demonstrate the importance of the convergence criteria once numerical stability is guaranteed. Clearly, as far as the shape-contour extraction is concerned, the evolving contours are more meaningful in the Fig. 8a–f than in Fig. 7a–f. As shown in Fig. 8e–f, in the absence of a convergence criteria we are unable to terminate the evolution

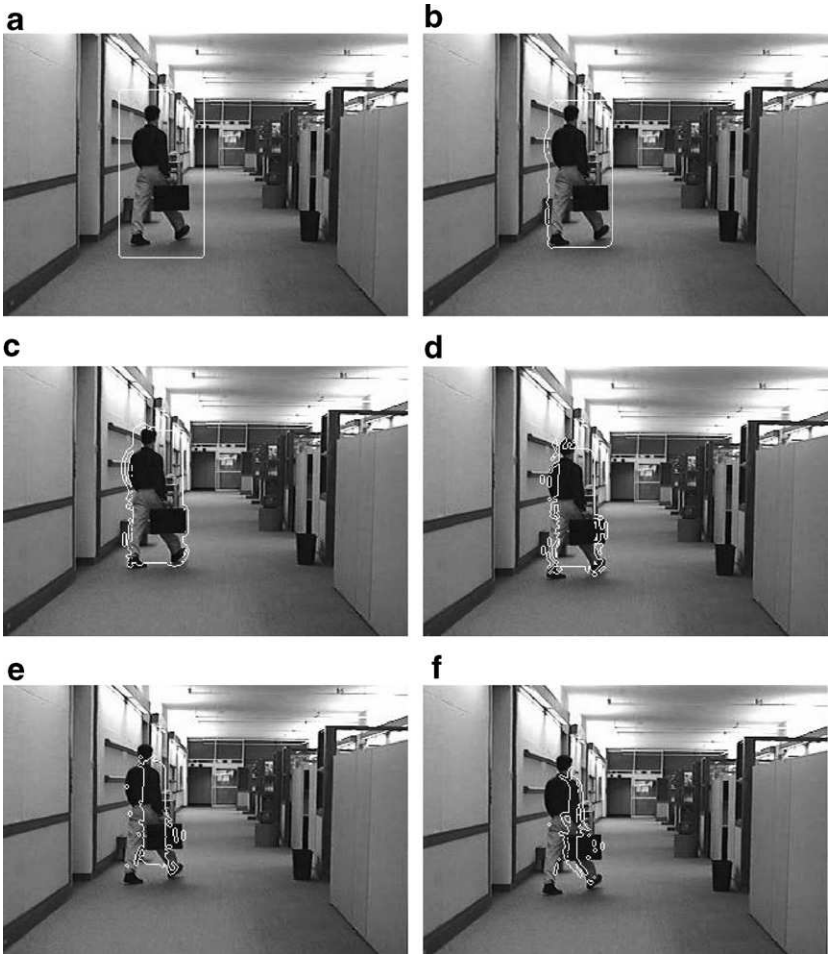

Fig. 7. Illustration of the instability problem: (a)–(f) Certain intermediate results of the curve evolution process using a level set algorithm. The curve in white is the evolving contour and our goal is to extract the contour of the person. The time-step Δt for the level set algorithm was deliberately set to violate the CFL condition and inject numerical stability into the evolution process.

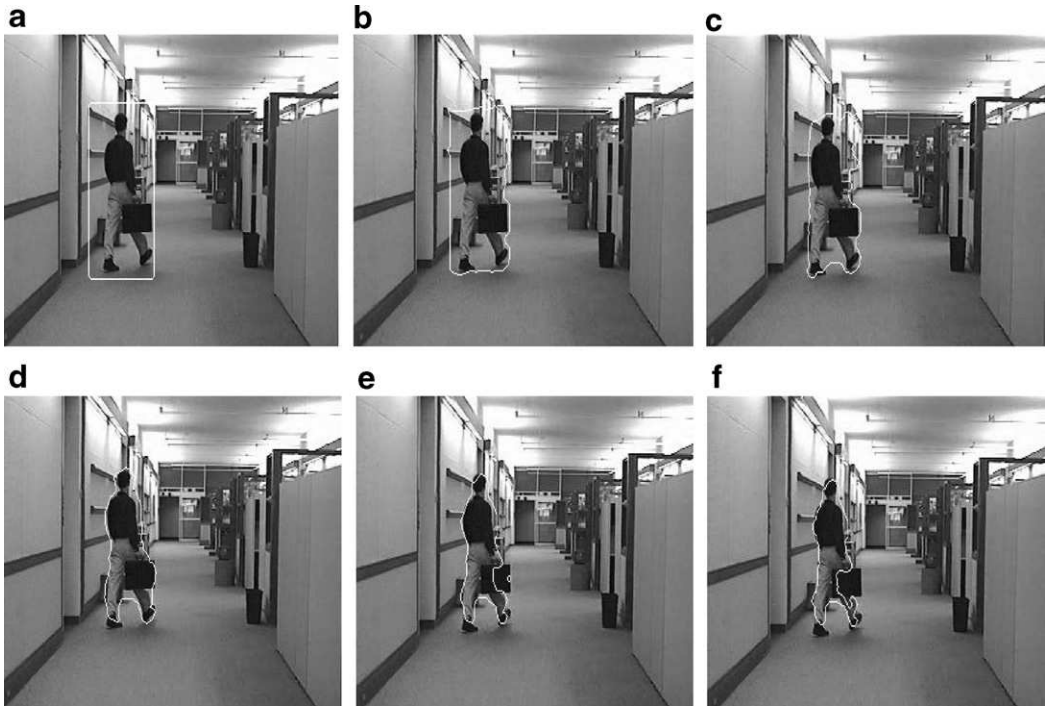

Fig. 8. Importance of the convergence criteria: (a)–(f) The level set based curve evolution process at certain time iterations. The curve in white is the evolving contour and our goal is to extract the contour of the person. The time-step Δt was set to ensure numerical stability but no convergence criteria was employed to stop the process at the correct instant.

process at the correct instant. This not only leads to a wastage of computational resource but also leads to unsatisfactory shape-contours as shown in Fig. 8f. Thus, employing our convergence criteria, we are able to stop the evolution process at the correct instant and extract a shape-contour similar to the one shown in Fig. 8d.

6. Comparison of performances

To summarize the improvement in the performance of existing level set based algorithms, as a result of the introduction of our proposed optimal time-step and convergence criteria, we compare the performance of our modified algorithm along with the primitive level set algorithms in the Table 1. It is not feasible to demonstrate the improvement in stability of our algorithm quantitatively. But, in the earlier sections, we have demonstrated pictorially how the instability problems are avoided using results from the curve evolution process. Below we depict the improvement in performance only in terms of the execution time (convergence issues).

Table 1
Comparison of performances

Reference to the level set paper/algorithm	No/unsatisfactory convergence criteria used		Using our proposed convergence criteria	
	Iteration count	Execution time (s)	Iteration count	Execution time (s)
Malladi et al. (1995)	2000	40	850	20
Mansouri (2002)	1000	35	640	25
Mansouri and Konrad (2003)	1000	35	800	20
Chang et al. (2004)	–	–	750	15

While generating the results used in the Table 1, we have assumed that the stability criteria was adhered to by both our algorithm and the corresponding primitive algorithms. In our case, the iteration count and hence the execution time is purely determined by the convergence criteria. In the reference algorithms, the iteration count (stopping criteria) was chosen empirically based on the image size and the length of the initial contour (Mansouri, 2002; Malladi et al., 1995; Mansouri and Konrad, 2003; Chang et al., 2004).

It is clear from the table that, as the number of iterations in the level set algorithm is set empirically, a lot of computational resource goes waste, and considering the time complexity of the level set implementations, this unnecessarily increases the execution time. Our convergence test stops the iteration process once the desired contour has been attained and hence does away with the undue wastage of computational resources.

7. Conclusion and future work

Our goal was to demonstrate that the existing level set based algorithms are not only vulnerable to numerical instabilities but can also lead to unsatisfactory results due to the deployment of poor stopping criteria. As evident from the results shown in this paper, both numerical

stability and a good convergence criteria are essential for satisfactory performance of level set algorithms.

We have proposed a robust convergence criteria and a method of determining the optimal time-step for the numerical solution of the level set equation. The time-step is optimal in the sense that, for a given speed function, it ensures maximum rate of curve evolution without threatening the numerical stability of the solution. The convergence criteria takes a lot of factors into account while deciding when to halt the propagating curve. This makes the convergence criteria robust to situations, like the temporary slowdown of the evolution process prior to steady state, which could prematurely halt the curve propagation yielding unsatisfactory results.

A detailed discussion of the computer vision problems used to validate our findings is beyond the scope of this paper. Though we use object tracking and shape-contour extraction algorithms for validation purpose, the stability and convergence issues are generic in nature and can be extended to other level set based applications. Fast level set algorithms, like "Narrow Band Level Set", along with our proposed stability and convergence solutions can drastically improve the reliability and performance of level-set based applications in computer vision.

References

Adalsteinsson, D., Sethian, J., 1995. A fast level set method for propagating interfaces. J. Comput. Phys., 269–277.

Chang, C.J., Hseih, J.W., Chein, Y., 2004. Tracking multiple moving objects using the level set method. Int. J. Pattern Recognition Artif. Intell. 18 (2), 101–125.

Kass, M., Witkin, A., 1988. Snakes: Active contour models. Int. J. Comput. Vision 118, 321–331.

Malladi, R., Sethian, J., Vemuri, B., 1995. Shape modeling with front propagation: A level set approach. IEEE Trans. Pattern Recognition Artif. Intell. 17 (2).

Mansouri, A., 2002. Region tracking via level set PDEs without motion computation. IEEE Trans. PAMI 24 (7), 947–961.

Mansouri, A., Konrad, J., 2003. Multiple motion segmentation with level sets. IEEE Trans. PAMI 12, 201–219.

Osher, Sethian, J., 1988. Front propagation with curvature-dependent speed: Algorithms based on Jacobi–Hamilton formulation. J. Comput. Phys. 79, 12–49.

Paragios, N., Deriche, R., 2000. Geodesic active contours and level sets for the detection and tracking of moving objects. IEEE Trans. Pattern Recognition Artif. Intell. 22 (3).