Stabilisation of Parametric Active Contours using a Tangential Redistribution Term

V. Srikrishnan*, *Student Member, IEEE,* and Subhasis Chaudhuri, *Senior Member, IEEE,*

***Abstract*— Depending on implementation, active contours have been classified as geometric or parametric active contours. Parametric contours, irrespective of representation, are known to suffer from the problem of irregular bunching and spacing out of curve points during the curve evolution. In a spline based implementation of active contours, this leads to occasional formation of loops locally, and subsequently the curve blows up due to instabilities. In this work, we analyse the reason for this problem and propose a solution to alleviate the same. We propose an ordinary differential equation(ODE) for controlling the curve parametrisation during evolution by including a tangential force. We show that the solution of the proposed ODE is bounded. We demonstrate the effectiveness of the proposed method for segmentation and tracking tasks on closed as well as open contours.**

***Index Terms*— Parametric active contours, Stable contour evolution, Tangential redistribution of curve points, Image Segmentation.**

I. Introduction

ACTIVE contours or snakes, introduced in [1] by Kass, Witkin and Terzopoulos, are widely used in computer vision tasks like tracking [2] [3] and segmentation [4] [5]. Active contours are simply connected curves, generally assumed to be closed but not always so, which move so as to minimise the energy functional defined on them. The energy functional could be defined based on some property of the region enclosed within the curve or it could depend only on the curve points. The minimisation of the energy functional yields the corresponding Euler-Lagrange curve evolution equations.

Depending on the energy functional defined on the contour, the models have been classified into gradient and region based active contours. Although it is out of the scope of this paper to review the entire active contour literature in each of these subdivisions, we do mention some of the important works in each category. The seminal paper which started off the active contours research was the work by Kass *et al.* [1] which is a gradient based approach. The level set based method was introduced by Malladi *et al.* [6]. In [7] [8], the authors have recast the original formulation proposed in [1] into that of finding a geodesic defined on the image surface. A recent work on gradient based active contours is [9] which also contains a very extensive review of the major gradient based active contour models. Some of the important region based active contour models are [5] [10] and [11]. All these snake models mentioned above have assumed that the contours are closed curves. This restriction, however, need not always apply, at least not for gradient based strategies. In [12] and [13], the authors have proposed models for gradient based open active contours. A recent review paper by Cremers et al. [14] gives a very good overview of the region based strategies for active contours. Some other interesting works are [15] [16] [17] [18] [19] [20] [21].

Depending on the numerical implementation of the curve evolution equations, active contours have been classified as either parametric or geometric active contours. Parametric active contours are implemented using a representation like a spline [22] or a finite element method [23]. These were the initial choices of implementation. On the other hand, geometric active contours are implemented in an Eulerian framework using the level set methods [24] [25]. An interesting paper which links these two approaches is [21]. The relative merits and demerits of both these numerical methods are well documented by a number of researchers, for example [26]. We briefly state the properties for the sake of completion. Level set based methods handle any topology change during curve evolution in a very natural manner. However, level set methods are quite slow, though use of narrow band techniques [24] do improve the evolution speed. Recently, in [27] the authors have described an algorithm for real time implementation of level sets. Parametric contours, on the other hand, are much easier to implement. In fact, the simplest implementation is the multi-point implementation. Parametric contours are much faster than level set methods. The greatest drawback of parametric curve representation is that splitting and merging of curves is not possible, at least not without some additional and possibly heuristic techniques [28] [29]. However, for many practical applications, this curb on topology change does to restrict the scope of application. For example, while tracking a human being, we do not expect one person to split into two.

Parametric contours, irrespective of representation, are known to suffer from the problem of irregular bunching and spacing out of curve points during the curve evolution. In a spline based implementation of active contours, this leads to occasional formation of loops locally, and subsequently the curve blows up due to instabilities. In this work, we analyse mathematically, the reason for this problem and propose a definite solution to alleviate the same. We propose an ordinary differential equation(ODE) for controlling the curve parametrisation during evolution. This ODE provides a tangential force during curve evolution for lateral redistribution of curve points. We show that the solution of the proposed ODE is bounded. We demonstrate the effectiveness of the proposed technique for segmentation and tracking tasks on both closed and open contours.

A preliminary version of this work appeared at a conference

V. Srikrishnan and S. Chaudhuri are with the Department of Electrical Engineering, Indian Institute of Technology, Bombay, Mumbai, PIN 400076 INDIA e-mail: krishnan,sc@ee.iitb.ac.in

* A preliminary version of this paper was presented at CVPR 2007

[30]. In this work, we extend the framework to also deal with open curves, which is a non-trivial exercise. This is a non trivial extension for two reasons. The authors in [31] try to solve this problem by keeping the end points fixed, that is speed is set to zero. This is a highly restrictive condition. For our method, we impose no such restriction. Additionally, we show a theoretical proof of the boundedness of the solution of the proposed ODE, with respect to the solution of the ODE proposed in [31], using Gronwall's Inequality.

The outline of the paper is as follows. In section II, we discuss the problem associated with the implementation of curve evolution equation using parametric representation of curves. We also discuss the common practical solutions adopted during implementations and the more specialised methods proposed in literature to tackle the problem. In section III, we describe the proposed approach in detail. The boundedness of the proposed solution is proved in section IV. We show the experimental results to demonstrate the practical validation of the proposed ODE in section V. Section VI presents the conclusion.

II. Problems with Parametric Contour Evolution

A. Notation and Preliminaries

Before describing the problem with parametric curve evolution, we explain the notation used in this paper. A curve is denoted by $C(p,t)$, where p is the curve parameter and t is the artificial time parameter. Thus t parametrises a family of curves while p parametrises a single member of this family. The initial curve is $C(p,0)$ from which a family of curves is obtained. The local tangent and *inward* normal to the curve are denoted by $\mathbf{T}(p,t)$ and $\mathbf{N}(p,t)$, respectively. The curvature is denoted by $\kappa(p,t)$ and the arc length parameter by s. The most general form of the curve evolution equation is,

$$\frac{\partial C(p,t)}{\partial t} = \alpha(p,t)\mathbf{T}(p,t) + \beta(p,t)\mathbf{N}(p,t). \qquad (1)$$

In the above equation, $\alpha(p,t)$ and $\beta(p,t)$ are the tangential and normal components of the force moving the curve at each point.

An important quantity related to the curve is the speed parameter defined as $g(p,t) = |C_p|$. This quantity has the following interpretation. If we imagine a particle traversing along the curve perimeter with p as the artificial *time* variable, then from the above equation, clearly $g(p,t)$ measures the speed of this hypothetical particle. In other words, $g(p,t)$ measures the distance we have traversed along the curve for some change in parametrisation. This is crucial during discretisation since p is sampled usually in a uniform manner over its domain. Corresponding to the values of p, the curve points $C(p_i)$ are obtained. Hence, a constant $g(p,t)$ is essential for a uniform discretisation of the curve.

As mentioned earlier, an energy functional is defined on active contours for segmentation. Minimisation of the energy functional leads to the Euler-Lagrange equations. The general form of energy functional defined on the active contour is as

$$E(C(p)) = E_{im}(C(p)) + \mu E_{sm}(C(p)). \qquad (2)$$

The total energy on the contour $E(C(p))$ has two components $E_{im}(C(p))$, the energy depending on the underlying image, and a regularising component $E_{sm}(C(p))$, which ensures that the curve evolution proceeds smoothly. These are also known as the external and internal force terms, respectively. The parameter μ weighs the relative importance of these two terms. The relation between equations (1) and (2) is explained later. From, this point onward, we drop the dependence on the independent variables p and t for notational convenience, wherever not needed.

B. Problem with Evolution

Parametric contours exhibit a typical undesirable behaviour during their evolution. This has been reported in a number of places, for example [31] [32] [33]. During evolution, the curve points bunch together in some places while spreading out at other places along the curve. This uneven spread of points causes problems in computation of curve measures (e.g. curvature, tangent and normal vectors). Also, due to the spacing out of points the segmentation is not very accurate. For a spline based implementation, there is no problem in computing the curve measures since these are computed analytically, but it may lead to formation of loops because of the control points bunching together. This is highly undesirable as it reduces the curve smoothness and the normal $\mathbf{N}(p,t)$ becomes ill-defined and further curve evolution becomes meaningless. This problem which is very disturbing during image segmentation, becomes intolerable while tracking. As a motivational example, we show three frames from a tracking sequence of a hand in figure 1. The tracking algorithm is a very simple extension of the region segmentation approach. The final contour in the previous frame is the initialisation for the contour in the present frame. Although this is a very simple example, it suffices to illustrate the issue of degeneracy of curve evolution. Figure 1(a) shows the curve just after initialisation. The points on the curve are nearly equidistant. After four frames, as marked in figure 1(b), the points accumulated in two regions are marked by red circles. In the very next frame, in figure 1(c), we notice that small loops have formed in these regions. These loops subsequently blow up and the curve becomes unstable within the next few frames.

In the above example, there is a good amount of motion of the hand from the left to right but there is very little shape change. We show another example, where there is a rapid change in target shape but the overall global motion of the target is very small. Figure 2 shows three frames from an attempted tracking sequence on a target. In figure 2(a), the curve converges to the palm outline. Within the next two frames, there is a rapid shrinking of the palm. In figure 2(b), we can see the beginnings of the loop formation on the top left of the semi-closed palm. This immediately blows up into a loop as can be seen in figure 2(c). The above two examples illustrate very well that the curve evolution process could be quite unstable. For both these examples, the energy functional used was the region competition approach by Zhu and Yuille [5]. We give the exact form of the curve evolution equation in section V.

(a) Frame 26 (b) Frame 30 (c) Frame 31

Fig. 1. Illustration of curve degeneration:(a) Initial curve (in red). Convergence to target (in green). (b) Bunching of points (in red) starts due to target motion leading to (c) loop formation.

(a) Frame 66 (b) Frame 82 (c) Frame 83

Fig. 2. Illustration of curve degeneration due to rapid shrinking of target:(a) Initial curve in red and converged curve in first frame, (b) bunching of points starts due to target shape change leading to (c) loop formation.

C. Existing Approaches toward stabilisation

We now describe a few commonly used approaches to tackle this problem and discuss their limitations.

- Re-initialisation of the curve using a least mean squared criterion [22] can be done either after a fixed number of frames or when the distance between successive control points falls below a certain threshold. However, this is not a very good solution because the shape of the curve would change during the re-positioning of the control points. The computation is also increased in checking the distances in each frame after every iteration.
- Another ad-hoc solution is insertion and deletion of points from the curve when the distance between two consecutive curve points exceeds or falls below a certain threshold. For insertion of points, some kind of local interpolation is done to introduce new points. This again is not a very good solution; the thresholds have to be set manually and, in general, is a naive proposition.
- In a spline based implementation, we could also control the curve by deleting or inserting control points. This is because of the fact that looping is caused by nearness of control points. Although algorithms exist for such a procedure, for example [34]; this solution is not natural. Moreover, such a procedure is specific to splines. Also, if we were to use the control points to represent the shape space [32], these operations would change the dimensionality of the feature space.

The above methods are rather ad-hoc attempts to adjust distance among points. In [32], the authors have used third order B Splines for curve representation. To alleviate the problem discussed above, the authors attempt to maintain the distance between successive control points. This is done by defining an energy term which also acts as a regularising term for smooth curve evolution. The energy term defined by the authors is $E_{sm} = \int_0^L |C_p| dp$, as the regularising term rather than the L_2 definition, where subscript denotes differentiation. For the specific case of quadratic, uniform B splines, this has the effect of positioning each control point to be equidistant between its two adjacent ones. This method is however only applicable to the specific case of quadratic B-Splines.

In [26], the authors propose the use of a tangential evolution term to control the curve parametrisation. The tangential force evolving the curve is obtained by applying the diffusion equation,

$$\frac{\partial g}{\partial t} = \frac{\partial^2 g}{\partial p^2} + \beta \kappa g, \tag{3}$$

where as mentioned earlier, κ is the curvature and g is the curve speed term. If we imagine some particle to be traversing along the curve with p as the artificial time, then g gives the speed of this particle. The authors proposed the above form because when the normal force β becomes small enough, equation (3) becomes the diffusion equation for the curve parameter g. Hence, the final parametrisation g becomes some proportion to the arc length parameter s. This assumption, however, is not very valid in practice. Further, the redistribution of the point set in this case does not explicitly depend on the image data.

The term proposed in [33] is of the form $E_{sm} = \int_0^L (g^2 - M)^2 ds$, where M is proportional to the curve length. This

term may cause extra smoothing and shrinking of the curve. This is not a desirable effect because curve regularising terms are already defined.

A comprehensive effort on stable evolution was in [31], where the authors have proposed the following ODE for evolving α defined in equation (1),

$$\frac{\partial \alpha}{\partial s} = \kappa\beta - \frac{1}{L} < g\kappa\beta > +(\frac{L}{g} - 1)(k_1 + \frac{k_2}{L} < g\kappa\beta >) \quad (4)$$

where s is the arc length parameter, L is the curve length. $k_1, k_2 > 0$ are arbitrary constants and $< \phi >$ is the average of the quantity ϕ, i.e. $< \phi >= \int_0^1 \phi(p)\,\mathrm{d}p$. This is a fairly general equation governing α and the authors have proved that the evolution is stable. However, there is no well-defined method to select proper values of these constants. Thus for various values of k_1 and k_2, there is a family of temporal evolutions of the curve. However, it is difficult to predict which set of values would yield best results in actual implementation. The purpose of the current work is to avoid this kind of heuristic choice of parameters during the evolution without affecting the convergence issues.

We provide a new term for evolution of the tangential component that depends only on single parameter K, instead of two completely unknown parameters k_1 and k_2 in equation (4), but still gives an even redistribution of points. Further, assuming the natural periodic boundary conditions for closed contours, one can have an exact expression for K. This assumption is, however, not valid for evolution of an open curve where the boundary conditions are already given. In this case, we use a simple numerical discretisation scheme for calculation of K. This avoids any heuristic selection of the control parameter K. Further, we provide the necessary proof of boundedness of the curve evolution.

III. Stable Curve Evolution

The force at each point on the curve can be resolved into two components: along the local tangent and normal denoted by α and β, respectively. Therefore, for planar curve evolution, the most general form of curve evolution can be written as in equation (1). Given this form of the evolution equation, g varies as follows [35] [31]:

$$\frac{\partial g}{\partial t} = -g\kappa\beta + \frac{\partial \alpha}{\partial p}, \quad (5)$$

It is seen from the above equation that g depends on both the components. It has been shown by researchers [35] that only the normal component of the force β influences the shape of the curve. The tangential component α simply re-parametrises the curve. Based on this fact, most works have concentrated on defining the normal term to speed up the convergence, increase the capture range, etc. No specific efforts were made (except for few works cited previously) to give some form to the tangential term for the purpose of curve stabilisation. This did not pose any problems as these works used level set methods which do not suffer from the same problems. In this work we propose an exact tangential evolution term to stabilise the curve and derive precise bounds for important curve measures like the length and curvature.

We first qualitatively discuss the cause for the bunching of the points on the curve and the control points. It is seen from equation (5) that g depends on both components of the force. Therefore, while reconstructing the curve with a discrete set of points the spacing between the points may occasionally vary in an unpredictable manner. This leads to uneven spacing of points at certain locations of the curve which cannot be brought under control by normal smoothing term alone.

In our approach we ensure curve stability by using a very simple equation to control g. Though arc length parametrisation is the most desirable, it cannot always be achieved in practice due to the curve representation. Therefore, a proper choice of the control parameter $g = K$ is essential for maintaining uniform distribution of points. We further argue that this K should be independent of the parametrisation used and a function of time step t only, and denote this by K^t, for convenience we drop the superscript. The usefulness of this argument will be made clear in the next section.

It is then natural to use equation (5) to force the curve towards the parametrisation which would make $g = K$. The left hand side of this equation predicts how g changes given β and α. We know the normal component $\beta(p,t)$; these are given by equations 25, 27 or 28 depending on the model we have chosen. Equation (5) can be rewritten as:

$$\frac{\partial \alpha}{\partial p} = \frac{\partial g}{\partial t} + g\kappa\beta, \quad (6)$$

We propose to set,

$$\frac{\partial g}{\partial t} = K - g. \quad (7)$$

Qualitatively, at each point we try to find α by pushing g at that point to the constant K. We obtain α by substituting equation (7) in equation (6) and then numerically solving the resulting PDE

$$\frac{\partial \alpha}{\partial p} = K - g + g\kappa\beta. \quad (8)$$

After solving for $\alpha(p,t)$, we use the values in equation (1) to evolve the curve.

In the next section, we propose a choice of K and also prove that important curve properties like the length and curvature remains bounded for the given choice of K.

IV. Boundedness of Evolution

The goal of this section is to determine the term K entering the constitutive relation in equation (7) yielding asymptotically uniform redistribution of numerically computed grid points. Recall that $K = K^t$, i.e. K is independent of the spatial parameter p. Without loss of generality we assume that the parameter p belongs to the interval $[0,1]$. If we take into account periodic boundary conditions imposed on tangential velocity term α then the term $K = K^t$ has to satisfy:

$$\begin{aligned} 0 &= \alpha(1,t) - \alpha(0,t) = \int_0^1 \partial_p \alpha(p,t)\,\mathrm{d}p \\ &= \int_0^1 (K - g + \kappa\beta g)\,\mathrm{d}p \\ &= K - \int_0^1 g\,\mathrm{d}p + \int_0^1 \kappa\beta g\,\mathrm{d}p \end{aligned}$$

$$= \quad K - L^t + \int_{\Gamma^t} \kappa\beta \, \mathrm{d}s$$

and therefore $K = K^t$ is given by

$$K = L^t - \int_{\Gamma^t} \kappa\beta \, \mathrm{d}s \tag{9}$$

where L^t is the length of the curve Γ^t, i.e. $L^t = \int_{\Gamma^t} \mathrm{d}s = \int_0^1 g \, \mathrm{d}p$.

In what follows we shall assume that the normal velocity β has the general form: $\beta = \mu\kappa + f(C)$ where f is a bounded function depending on the position of a curve point C. The validity of this assumption is shown in section V, where the final evolution equations are shown. The parameter μ controls the weight of the two terms.

Concerning estimate of the length L^t of the curve Γ^t and the modulus of $|\kappa|$ and $|\kappa\beta|$ we have the following proposition:

Lemma 1: Assume the normal velocity $\beta = \mu\kappa + f(C)$ where $\mu > 0$ is a positive constant and $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ is a bounded function, i.e. $\|f\|_\infty = \sup_{C\in\Omega} |f(C)| < \infty$. Then following estimates are satisfied:

$$L^t \leq L^0 \exp\left(t\|f\|_\infty^2/(2\mu)\right)$$

$$\int_{\Gamma^t} |\kappa| \, \mathrm{d}s \leq \frac{\delta^t + \|f\|_\infty}{\mu} L^t$$

and

$$\int_{\Gamma^t} |\kappa\beta| \, \mathrm{d}s \leq \delta^t \frac{\delta^t + \|f\|_\infty}{\mu} L^t$$

where $\delta^t = \max_{\Gamma^t} |\beta|$.

Proof of the above lemma is presented in the appendix.

Now we are able to prove boundedness of the tangential velocity α proposed in section III. Recall that α can be computed from the equation

$$\partial_p \alpha = L - \int_\Gamma \kappa\beta \, \mathrm{d}s - g + \kappa\beta \tag{10}$$

by taking into account the boundary condition $\alpha(0, t) = 0$. Therefore, for any $p^* \in [0, 1]$ we have

$$\begin{aligned} |\alpha(p^*, .)| &= |\int_0^{p^*} \partial_p\alpha \, \mathrm{d}p| \leq \int_0^{p^*} |\partial_p\alpha| \, \mathrm{d}p \leq \int_0^1 |\partial_p\alpha| \, \mathrm{d}p \\ &\leq \int_0^1 |K^t - g + g\kappa\beta| \, \mathrm{d}p \\ &\leq |K^t| + \int_0^1 g \, \mathrm{d}p + \int_0^1 |g\kappa\beta| \, \mathrm{d}p \\ &= |K^t| + L^t + \int_{\Gamma^t} |\kappa\beta| \mathrm{d}s \end{aligned}$$

where $K^t = L^t - \int_{\Gamma^t} \kappa\beta \mathrm{d}s$. Taking into account the estimates from the previous lemma we can conclude:

Theorem 1: If the normal velocity satisfies $\beta = \mu\kappa + f(C)$ where $\mu > 0$ is a positive constant and $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ is a bounded function then the tangential velocity α given by (10) is globally bounded in spatial parameter p, and

$$\max_{p\in[0,1]} |\alpha(p, t)| \leq 2\left(L^t + \int_{\Gamma^t} |\kappa\beta| \, \mathrm{d}s\right) \leq CL^t(1 + |\delta^t|^2)$$

for any $t \in [0, T]$ where $\delta^t = \max_{\Gamma^t} |\beta|$ and $C > 0$ is a constant depending only on μ, T and $\|f\|_\infty$.

Remark 1. Clearly, $< g >= L$ and $< g\kappa\beta >= \int_\Gamma \kappa\beta \, \mathrm{d}s$. Comparing equation (4) for $\partial_s\alpha$ with our proposition in equation (7) where $K^t = L^t - \int_{\Gamma^t} \kappa\beta \, \mathrm{d}s$ we conclude that our choice of the tangential velocity sets $k_1 = 1$ and $k_2 = -1$, respectively. Note however that our approach cannot be derived from [31] as in their work they have assumed both $k_1, k_2 > 0$, failing which convergence cannot be guaranteed. Hence the proposed solution is different from that of [31].

A. Conditional boundedness of tangential evolution

In this section, we show that the ODE proposed in equation (8) is bounded with respect to that proposed in Mikula *et al.* [31] or in other words, the proposed curve evolution will be asymptotically at a finite distance from that of the evolved curve of Mikula *et al.* Mikula and Sevcovic [31] have shown in their work that using their proposed tangential component, the 4-tuple (C, κ, ν, L) remains bounded, under the condition that the normal force is bounded, where ν is the tangent angle. To prove this, we need the form of Gronwall's inequality [36] stated below. This inequality is well known in the area of adaptive control.

The Gronwall inequality as stated here is adapted closely from Howard [37]. We include the statement for the sake of completeness, the proof can be found in the same reference The inequality relates the solutions of two Odes. The theorem is stated below.

Gronwall's Inequality

Theorem 2: Let $\mathbf{X}$ be a Banach space. Let $U \subset \mathbf{X}$ and U be an open subset. Let $f, g : [a, b] \times U \to \mathbf{X}$. We assume that f, g are continuous functions. Let $y, z : [a, b] \to U$ satisfy the initial value problem

$$\dot{y}(t) = f(t, y(t)), \; y(a) = y_0 \tag{11}$$

$$\dot{z}(t) = g(t, y(t)), \; z(a) = z_0 \tag{12}$$

Let us assume that there is a constant H such that

$$\| \, g(t, x_2) - g(t, x_1) \, \| \leq \; H \, \| \, x_2 - x_1 \, \| \tag{13}$$

and a continuous function $\psi : [a, b] \to [0, \infty)$, such that

$$\|f(t, g(t)) - g(t, y(t))\| \leq \psi(t). \tag{14}$$

Then, for $t \in [a, b]$

$$\|y(t) - z(t)\| \leq e^{H|t-a|}\|y_0 - z_0\| + e^{H|t-a|} \int_a^t e^{-H|s-a|}\psi(s)ds. \tag{15}$$

The above inequality gives a bound on how different two temporally evolving functions are. In order to prove the conditional boundedness of the proposed solution, let us denote by α_1 and α_2 the proposed ODE in equation (8) and the ODE proposed in [31], respectively. Following Gronwall's inequality in the stated form, we can write equations (8) and (4) as:

$$\frac{\partial\alpha_1}{\partial s} = h_1(p, \alpha_1) = \frac{K}{g} - 1 + \kappa\beta \tag{16}$$

$$\frac{\partial \alpha_2}{\partial p} = h_2(p, \alpha_2) = \kappa\beta - \frac{1}{L} < g\kappa\beta > +(\frac{L}{g} - 1)\omega \quad (17)$$

in place of equations (11) and (12). Note that here the derivatives are with respect to the arc length parameter s. We note that in both the equations (16) and (17), the RHS is independent of variables α_1 and α_2, respectively.

First, we show that there exists such a constant H, so that $||h_2(p, x_2) - h_2(p, x_1)|| \leq H$. Since h_2 is independent of α_2, we set $H = 0$. Any finite non-negative choice can be used here.

Next, we show that there exists such a ψ such that $||h_2(p, x_2) - h_1(p, x_1)|| \leq \psi(p) \, \forall p \in [p_i, p_f]$, where p_i and p_f denote the limits of the curve parametrisation, for any x_1 and x_2. This expression after substitution of the right hand sides of equations (16) and (17) can be written as,

$$||\frac{L}{g}\omega - \omega - \frac{1}{L} < g\kappa\beta > -\frac{K}{g} + 1|| \leq ||\frac{L\omega - K}{g}|| + ||1 - < \kappa\beta > -\omega||, \quad (18)$$

where the inequality follows from the triangular inequality property of norms. The second term of the inequality is essentially constant, as $< \, . \, >$ denotes averaging over the curve. Let us denote this by C_1, i.e, $C_1 = 1 - < \kappa\beta > -\omega$. Therefore, we set,

$$\psi = ||\frac{L\omega - K}{g}|| + ||1 - < \kappa\beta > -\omega||. \quad (19)$$

Finally, we use the above results to prove that $||\alpha_1 - \alpha_2||$ is bounded over the curve. We use H and ψ as obtained earlier. Using Gronwall's inequality as given in equation (15) and keeping in mind that $H = 0$, we have:

$$||\alpha_2(p) - \alpha_1(p)|| \leq ||\alpha_2(0) - \alpha_1(0)|| + \int_{p_i}^{p} \left(||\frac{L\omega - K}{g}|| + ||C_1|| \right) ds. \quad (20)$$

The above expression can be easily shown to be bounded since g is finite and greater than zero, $||C_1||$ is finite and the first term in the RHS is bounded as curve initializations are usually done very close to each other for both the methods. If one uses the same initialization, then the first term also becomes zero. This proves that the tangential evolution is always bounded.

B. Computation of K: Open curve

Recall that the equation (9) gives the theoretical value of K per iteration of evolution. We assume existence of periodic boundary conditions along the curve for the calculation K. This value of K is valid only for the case of tangential term for evolution of closed curve. This is obvious because for open curves, we cannot assume that $\alpha(0, t) = \alpha(1, t)$. In fact, depending on the energy functional, each end point of the curve may have its motion defined for both the normal and tangential directions [12] [13]. Therefore, for the case of open curves, calculation of K follows a slightly different procedure, which is as follows.

We assume that the tangential motion at the end points of the curve points are known to be $\hat{\alpha}[1]$ and $\hat{\alpha}[N]$ respectively, assuming that the curve is discretised into N points. Note, that the tangential term for open curves is still defined as per equation (8). It follows that

$$\alpha(1, t) = \alpha(0, t) + \int_0^1 (K - g + g\kappa\beta) \, \mathrm{d}p. \quad (21)$$

Therefore, we get the value of K very simply to be as,

$$K = \hat{\alpha}(1, t) - \hat{\alpha}(0, t) + L^t - \int_0^1 \kappa\beta \, \mathrm{d}p. \quad (22)$$

Numerical implementation of equation (8) is done very simply using first order finite differences. It can be done using higher order differences also, but for our experiments first order difference methods are seen to give satisfactory results. It is possible to show that numerical implementation of equation (8) will lead to error in the value of $\alpha(1, t)$. Specifically, at the boundary point $C(1, t)$, the calculated value $\alpha[N]$ will not be equal to the given value $\hat{\alpha}(1, t)$, where $\alpha[N]$ is obtained by discretising equation (8) using any numerical technique. This residual error is significant enough to cause problems in evolution at the boundary $C(1, t)$. The problem which occurs during evolution is that at the end point $C(1, t)$, the residual error causes the curve to grow outward continuously. This phenomenon is particularly noticeable when the iterations are large. The reason for this residual is that equation (8) is a first order ODE. Therefore, this should be ideally be an initial value problem with one boundary condition specified. As explained above, because of the underlying snake model, both the end conditions have been specified. For such problems, different methods of discretisation has been discussed in [38]. We have adopted the strategy of determining the value of K *after* discretising the equation. We assume that the curve is discretised into N points and denote the value of g, κ, β at i-th point by g_i, κ_i and β_i respectively. Then the expression for K is as follows

$$K = \frac{\Sigma_{i=1}^{N-1}(g_i + g_i\kappa_i\beta_i)}{N - 1} + \frac{\hat{\alpha}[N] - \hat{\alpha}[0]}{(N - 1)\triangle p} \quad (23)$$

where $\triangle p$ is the step size of discretisation of the curve parameter p.

V. Results and Discussions

A. Implementation details

We have used closed, periodic, cubic B-Splines [39] to implement the curves. For all the examples in this work, we have used a B-Spline based implementation for curve evolution similar to that of [22]. The discretisation of equation (8) is as follows. Let us assume that the curve is discretised into N points. The numerical discretisation of equation (8) is as,

$$\alpha(p_{i+1}) = \alpha(p_i) + K - g(p_i) + g(p_i)\kappa(p_i)\beta(p_i), \; i = 2 \ldots N - 1 \quad (24)$$

The periodic boundary condition leads to $\alpha(p_0) = \alpha(p_N)$. The implementation is as follows. First the normal force is computed and β determined. It is to be noted that the normal

force consists of both the internal and external energy terms. This is entirely dependent on the active contour model used. Then, once we know β, we compute α using the discretisation scheme of equation (24) above. From equation (9) of the manuscript, we can very easily compute the value of K. If we neglect $< g(p,t)\kappa(p,t)\beta(p,t) >$ in comparison to the length L of the curve, one may use the approximation $K = L$. We compute K while computing the normal force β. Now, we have both β, the normal term and α, the tangential term. We simply plug these values back into equation (1) and evolve the curve.

The computation overhead for implementation of the tangential term is minimal. As we can see from the previous point, computation of K is done while computing the normal term. The only extra computation is the implementation of equation (24). Implementation of this has very less overhead because this is a simple summing operation over N points.

In a B-Spline representation, we need a higher number of control points to get a better delineation of the object boundary. An unfortunate side-effect of this necessity is that the tendency to form loops also increases [39]. Other representations for curves can also be used as the method proposed is independent of the choice of representation. The segmentation algorithms for closed curves used in this study are the gradient vector force(GVF) algorithm [4] and the region competition model [5]. For open curve evolution, we have implemented the model proposed in [13]. For the sake of completion, we write down the curve evolution equations here for the different models. For more details the reader is referred to the original papers. The normal term β_R for the region competition model is

$$\beta_R = log(p_B(I(C))) - log(p_T(I(C))) + \mu\kappa, \tag{25}$$

where $p_B(.)$ and $p_T(.)$ denote the probability that the pixel at image location $I(C(p,t))$ belongs to the background and target respectively. Keeping in mind that the local normal direction points inwards to the curve, the interpretation of the above equation is that the curve moves along the inward direction when the image point $I(C(p,t))$ on the curve $(C(p,t)$ has a higher probability of being a background pixel, otherwise the direction of motion is reversed. For the region competition model, we assume that there is one target and one background region. The probability distributions p_B and p_T are assumed to be known a priori. For our work, we have used histograms, rather than parametric Gaussian models as done in the original work, to model the target and the background. As, before the parameter μ weighs the importance of the image and smoothness terms.

In [4], the authors define a motion field $\mathbf{F} = (u, v)$ having two components in the image plane which moves the active contour toward the local edge, which is assumed to form the complete boundary of the target. The force field is obtained by minimising the following functional,

$$E_{im} = E_{GVF} = \int\int (\mathbf{F} - \nabla I)^2 + \gamma(u_x^2 + u_y^2 + v_x^2 + v_y^2)dxdy \tag{26}$$

where γ is a parameter which controls the spatial extent of the field. The normal force β_{GVF} driving the curve evolution in this case is

$$\beta_{GVF} = \mathbf{F} \cdot \mathbf{N} + \mu\kappa \tag{27}$$

Finally, for the open curve model, the normal term β_{KB} is as follows

$$\begin{aligned} \beta_{KB} &= sign(\nabla I \cdot \mathbf{N})\triangle I + \kappa \\ \beta_{KB}(0) &= -sign(\nabla I \cdot \mathbf{N})(\nabla I \cdot \mathbf{T}) \\ \beta_{KB}(1) &= sign(\nabla I \cdot \mathbf{N})(\nabla I \cdot \mathbf{T}) \end{aligned} \tag{28}$$

where without loss of generality we assume that the curve parametrisation $0 \leq p \leq 1$. The tangential force at the end points of the contour α_{KB} is as

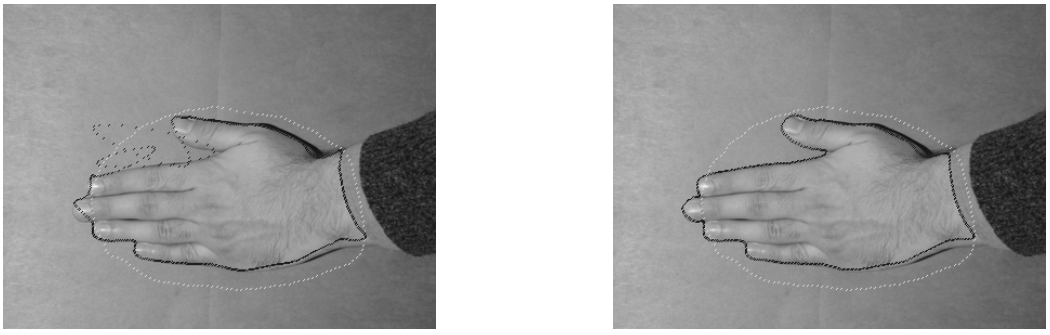
$$\begin{aligned} \alpha_{KB}(0) &= |\nabla I \cdot \mathbf{N}| - 1 \\ \alpha_{KB}(1) &= 1 - |\nabla I \cdot \mathbf{N}| \end{aligned} \tag{29}$$

The complete curve evolution equation is obtained by adding any of equations(25), (27) or (28) with the tangential term defined as per equation (8). This will then give the general form of the evolution equation (1). For the case of open curve evolution K is calculated as per equation (23), where the tangential force at the end points are defined as per equation (29).

B. Segmentation Results

1) Closed Curve: Figures 3 and 4 show the results of the stabilising term applied to static segmentation using region based approach and GVF model, respectively. Figures 3(a) and 4(a) show the results of segmentation without the use of the tangential term. The segmentation model for the hand images was the region competition method. For the segmentation of the aeroplane, we used the GVF model. The curves in yellow in figures 4(a) and 4(b) denote the initialisation. Without the use of the stabilising tangential term, we note that formation of loops has occurred, as expected. It is worth mentioning that for the hand segmentation, we found that the loop formation problem could be avoided by reducing drastically the number of control points. This however leads to a poor segmentation. The results of segmentation using the proposed stabilising tangential term is shown in figures 3(b) and 4(b). There is no formation of loop and the segmentation results are quite accurate.

In order to demonstrate the effect of the tangential term quantitatively, we take two measures defined on the curve. These are the *normalised* inter-point distance between the ith and $i+1$ point on the curve, $d_i = ||C(p_i,t) - C(p_{i+1},t||/L_t$, and its variance $\sigma = \frac{1}{N}\Sigma_i\big(d_i - \frac{1}{N}\big)^2$. We have normalised the inter-point distance with respect to the curve length to account for the fact that the curve length need not remain constant during evolution. This may increase or decrease depending on the initialisation and the target shape. In figure 5 we show the variation of these measures for the gradient based segmentation model of figures 4(a) and 4(b). Specifically, in figure 5(a), we show the normalised inter-point distance for all the points on the curve. For these experiments, the curve was discretised into 250 points. These distance were generated



(a) Region based segmentation

(b) Region based segmentation using tangential term

Fig. 3. Region segmentation results: (a) No stabilising term used. (b) Stabilisation with derived K.

(a) Edge based segmentation

(b) Edge based segmentation with tangential term

(c) Edge based segmentation with redistribution term of [31]

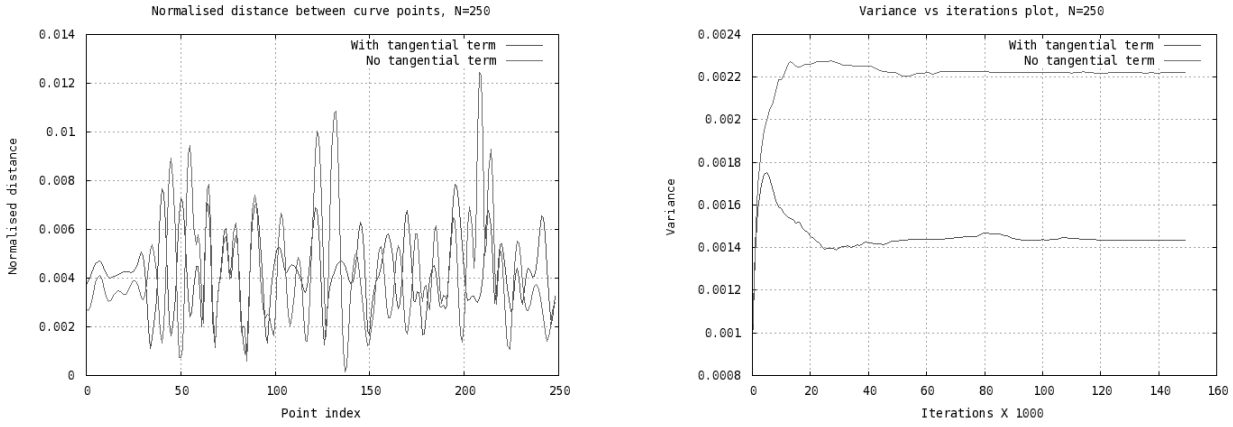
Fig. 4. Edge segmentation results: (a) No stabilising term used. (b) Stabilisation with derived K.

with and without incorporating the tangential term but after reducing the number of control points(as opposed to what is shown in figures 4(a) and 5(a)) such that the evolution remains stable even without the use of the tangential term. The reduction in the number of control points results in a poor segmentation of the contour. The curves in red and blue show the normalised distance with and without the tangential term, respectively. Clearly, the distribution is enclosed in a much tighter envelope for the red curve, signifying that the point spacing is more uniform. This is more clearly exemplified in figure 5(b) where we have plotted the variance σ as the iterations increase. The total number of iterations was 150000 and we plotted the variance every 1000 iterations. With the use of the tangential term, as shown in red, the variance is nearly half the value without the stabilising term. These plots clearly indicate the strong redistribution effect of the proposed term.

In figure 6, we show the similar plots for the case of region based evolution. The test cases were similar to the evolution shown in figures 3(a) and 3(b). Unlike in the previous case, in these sets of experiments, we did not try to prevent curve degeneration due to non usage of the tangential term. This could have been done by cutting down the number of control points but this yields a poor segmentation around the sharp corners. The curve degenerated in about 7000 iterations when the tangential term is not used. Using the tangential term, we evolved the curve for 27000 iterations to get the sharp corner near the fingertips and other places. The normalised distance without using the tangential term is shown in red in figure 6(a) at the end of 7000 iterations. It is to be noted that this distance fluctuated wildly along the curve, being very large near the end points and decreasing to a small value at at the middle. The curve in green shows the same measure for the tangential term case, where there is a much more narrow envelope delimiting the curve. In figure 6(b) we show the variance with respect to iteration number. The curve in red shows a drastic increase in variance at an early stage in the evolution. The curve in green maintains a tighter variance and hence, a better point distribution throughout the evolution.

For the purpose of comparison in figure 4(c), we show the segmentation result using the tangential redistribution term proposed in [31] for the image in figure 4(a). For implementation purposes, we used the values of $k_1 = 1.0$ and $k_2 = 0.0$ as mentioned by the authors. The initialisation is shown in red and the final converged curve is shown in yellow. This same initialisation is also used in figures 4(a) and 4(b). It is obvious from the image that the final curve does not have uniform point distribution as in figure 4(b). The variance of the distribution of the final points is shown in figure 7. The curve in red shows the plot of variance versus the iterations for the evolving curves in figure 4(c). Similarly, the figure in blue shows the variance plot for the evolving curves using the proposed redistribution term. It is again clear from the plots that the proposed method outperforms the redistribution term proposed in [31].

For the sake of brevity, we do not show any more results



(a) (b)

Fig. 5. Plots of (a) normalised inter-point distance and (b) the corresponding variance (as a function of number of iterations during evolution) for the edge based curve evolution of figures fig. 4(a) and 4(b). The curve was discretised into 250 points. Thus the average inter-point distance is 0.004.

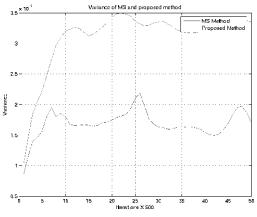


Fig. 7. Variance of point distance of the final converged curve in figure 4(c) . Plot in red shows the variance using the redistribution term of [31]. Plot in blue shows the variance using the proposed term. For implementation, values of k_1 and k_2 were set to 1.0 and 0.0 as chosen in [31].

of static segmentation. However, as mentioned earlier, since tracking of moving objects poses a bigger problem, we show some of these results in more details.

2) Tracking Results: The tracking algorithm is a naive extension of the one used for segmentation explained above. We take the final contour in previous frame as the initialisation for the contour in the current frame. For tracking, we observed that curve degeneration happens when there is a sudden change in shape or rapid motion. In figure 8, we show the same frames as in figure 1. We note that not only is the curve stabilised in figure 8(a) but also continues to remain so in figure 8(b), 9 frames later.

We show tracking results on three much more challenging sequences in figures 9, 10 and 11. Sequences shown in figures 10 and 11 have been taken from [40]. Figure 9 shows the tracking of a rapidly dropping balloon. The shape of the balloon also changes drastically at the end while it is still moving. Tracking for this sequence is done using the region competition model as mentioned above. Without the use of the tangential term, the curve degenerated within one or two frames. The whole sequence is of length 21 frames. Although we have successfully tracked sequences consisting of hundreds of frames, we chose this example to show the effectiveness of the proposed term even during rapid object motion and shape change.

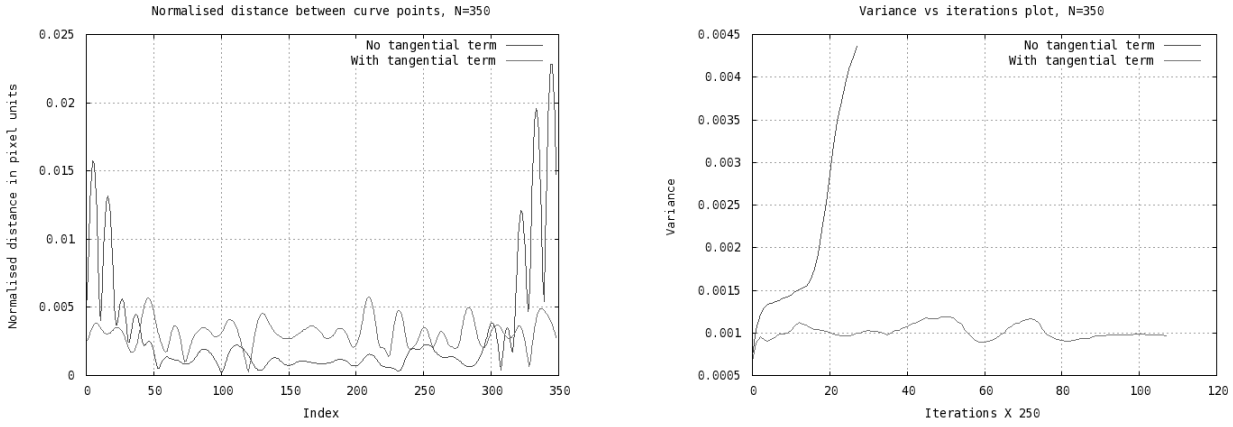
In figure 10, we show the tracking of a rapidly bending person. The length of this sequence is 93 frames. For segmentation, we initially learn the background and subtracted each incoming frame from the learned background. Since this is a static scene, the simple approach of frame differencing followed by thresholding was sufficient for the purpose. We then generated the force field using the gradient vector force [4] on the resulting difference map. This is the normal force moving the contour. Note here that we do not initialise the curve in each frame of the sequence. As before, we use the final converged contour of the previous frame as the initialising contour for the current frame. We see the efficacy of the proposed tangential term in stabilising the curve evolution. The tracked contour was able to follow the bend of the human target very well. As can be seen, the proposed tangential term makes the tracking possible, which was not possible without the redistribution term.

Finally, in figure 11, we show the tracking results on a sequence in which a person is moving rapidly by hopping on one foot while moving from the left to right. The motion field for curve evolution was generated as explained above for the case of the bending person. This sequence is of length 37 frames. Once again we note that the object has been tracked very successfully.

We have used the proposed term for processing sequences consisting of hundreds of frames. The results have been uniformly good for all the sequences.

C. Open curve

For the case of open curve evolution, we have used the model proposed by Kimmel *et al.* [13]. In this model, the curve



(a) (b)

Fig. 6. Plots of (a) normalised inter-point distance and (b) the corresponding variance as a function of the number of iterations for the region based curve evolution shown in figures fig. 3(a) and 3(b). The curve was discretised into 350 points. The average inter-point distance is thus 0.003. Note that the evolution without the tangential redistribution term degenerates after about 7000 iterations.

(a) (b)

Fig. 8. Tracking results for the same sequence with same initialisation as figure (1). The final curve remains stable despite significant motion and shape change. The image on the left (a) shows the result on the first frame and the image on the right(b) is 9 frames hence.

end points have a pre-defined tangential motion as per equation (29). This is because the curve shrinks in regions of small gradient and increases in length along an edge. Therefore, to calculate K for each iteration, we use the procedure described in section IV-B. In figure 12(a), the initial curves are indicated in green and the final curves are indicated in red. As expected, the curves initialised on the edges initially shrink to the edge and then expand to the length of the edge. The curve in the centre has no nearby edge to latch on to. Hence, it keeps on shrinking and the evolution was stopped after sometime. If the evolution is continued, then the curve shrinks to a point which is in complete accordance with the original model. The curves along the edge stopped at the edge of the square because of the sharp corners. This result is indicative of the effectiveness proposed tangential term. It performs the smooth re-parametrisation of points while keeping the edge or boundary conditions unchanged. Hence, the term in no way disturbs the original model.

In figure 12(b), we have used the cropped version of an image used in [13]. In this image, as in the previous image, the initialisation is given in green and the final curves are marked in red. We have used different initialising curve for each edge of the target. The curve latches onto the edge and expands along it till it reaches the corner of the target.

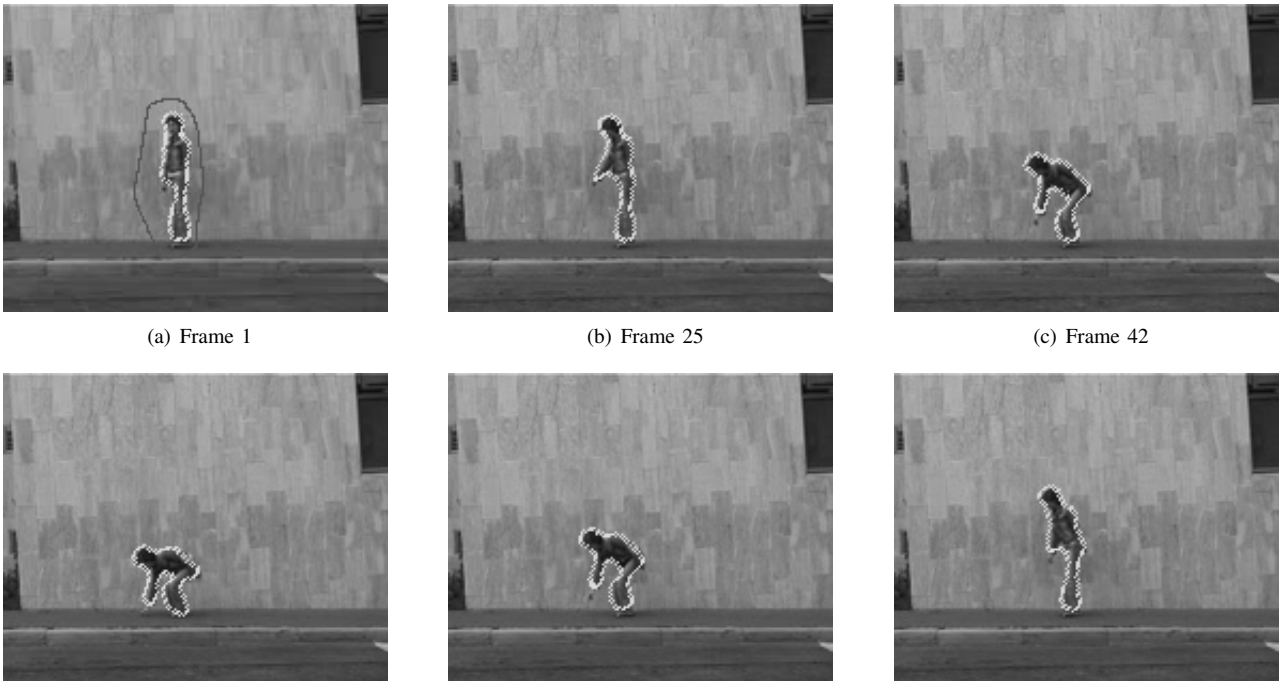
Figure 13 shows a sequence of figures which show the evolution the open curve. In these images, the curve in yellow is the initial curve. The curve in pink is the evolved curve. Initially, the curve gets attracted and lengthened by the weaker edges. As the evolution proceeds, the curve gets attracted to the stronger edge of the pool boundary. Once the curve latches onto this strong edge, evolution proceeds smoothly and the curve increases in length. The initial curve length is about 125 and the final length is about 725 pixel units. It may be noted that the points stretch smoothly as the curve expands.

It is interesting to note here that as mentioned in the original paper [13], the authors have implemented the evolution by using a multiple point representation of the curve. The distance between successive points in maintained by removing a point or adding point between them, when the distance becomes lesser or greater than a certain threshold. Hence, the number of points is not constant and varies during evolution. Our method is much more elegant for maintaining the distribution of curve points. We wish to remark here that for the case of open curves, the evolution is strongly dependent on the

(a) Frame 1 (b) Frame 4 (c) Frame 9

(d) Frame 12 (e) Frame 17 (f) Frame 21

Fig. 9. Tracking results for the balloon sequence: (a-g)Note the rapid motion at the beginning and shape shrinkage of the balloon at the end. The balloon is perfectly tracked even through the drastic change in position and shape change. There is no formation of loops or accumulation of points anywhere along the curve. Initial curve is marked in red in sub-figure (a).



(a) Frame 1 (b) Frame 25 (c) Frame 42

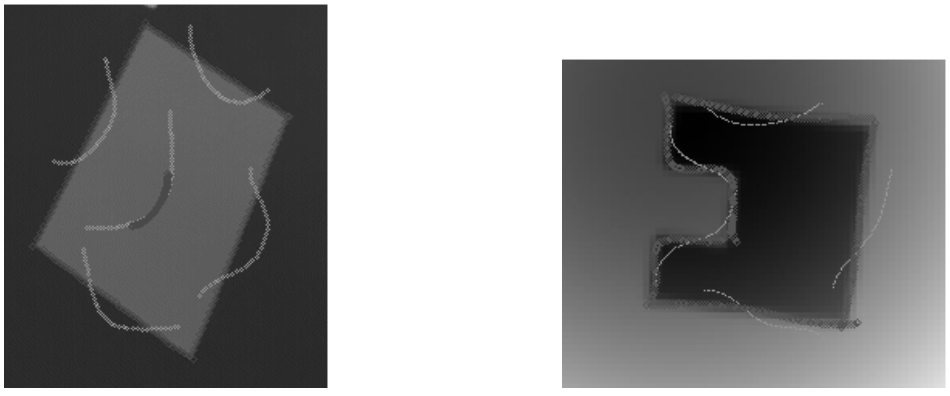
(d) Frame 57 (e) Frame 70 (f) Frame 85

Fig. 10. Tracking results: (a-f) The tracked person rapidly bends forward and regains the initial posture. Initial curve marked in red in sub-figure(a).

(a) Frame 1 (b) Frame 8 (c) Frame 15

(d) Frame 20 (e) Frame 28 (f) Frame 35

Fig. 11. Tracking results: (a-f) The tracked person moves from the left to right very rapidly. Initial curve is marked in red in sub-figure (a).



(a) (b)

Fig. 12. Evolution of open curve using the Kimmel-Bruckstein model [13]. The curve in green and red are the initial and final curves respectively. (a) For the curve in the centre of the box, the evolution was stopped after some time. This is because the curve shrinks to a point, since it is not located along any edge.(b) Result shown on image taken from [13].

relative weights of the tangential and normal evolution term. We have noted this phenomenon for all the cases of open curve evolution we have worked on. It is important to note that this is true for any implementation of the open curve and this is not particular to the proposed tangential term.

VI. Conclusions

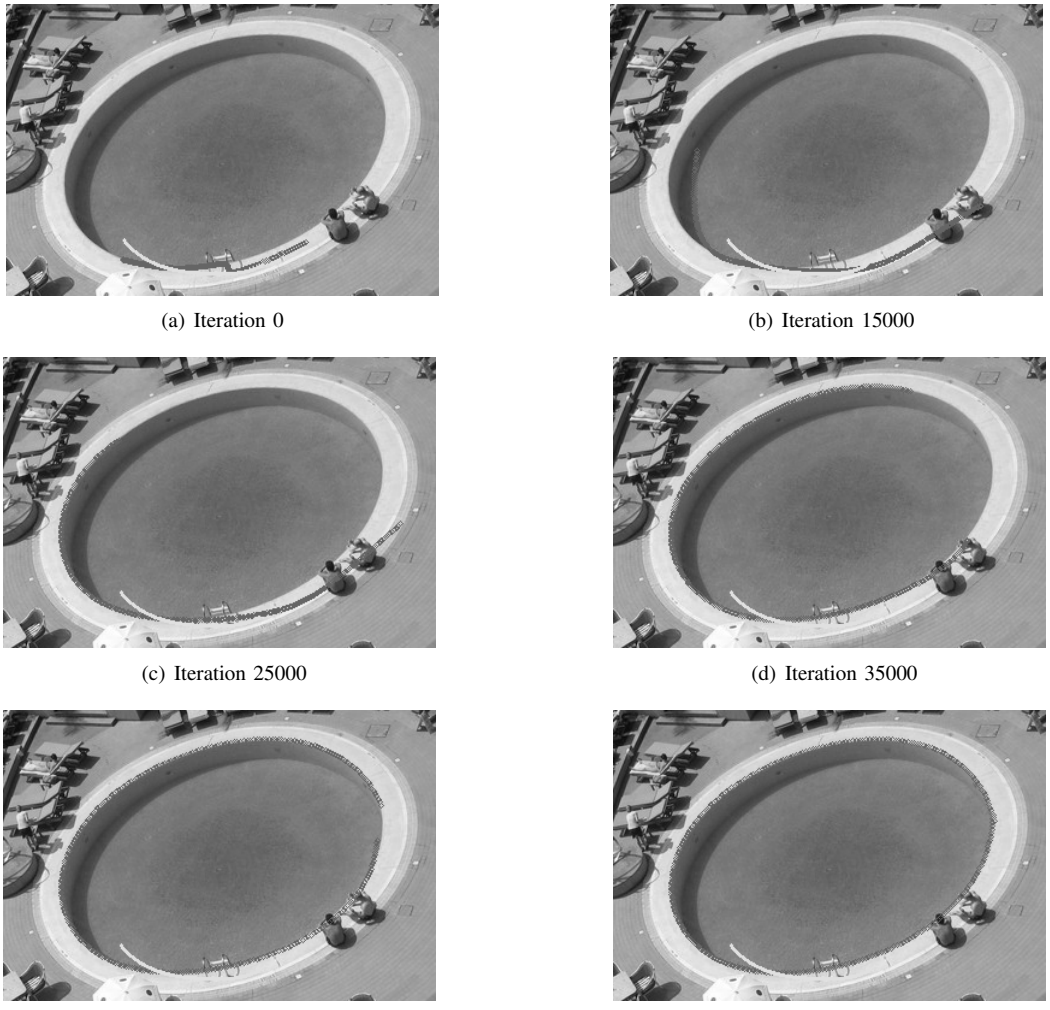
Parametric curves and B-Splines are simple methods to evolve an active contour. However, these suffer from the typical implementation problems of bunching and other associated instabilities. In this work, we have proposed a method for tangential redistribution of curve points and proved the boundedness of a tangential stabilising term. The proposed method is highly suitable for practical implementations of wide variety of curve evolution equations. We have also demonstrated how the proposed method can be used for open curve evolution.

Appendix
Proof of the boundedness of curvature

We present the proof of lemma 1 here.

Proof. Since $\frac{d}{dt}L^t = -\int_{\Gamma^t} \kappa\beta \,\mathrm{d}s = -\mu\int_{\Gamma^t} \kappa^2 \,\mathrm{d}s - \int_{\Gamma^t} f\kappa \,\mathrm{d}s$ Using Young's inequality $ab \leq (a^2+b^2)/2$ we obtain

$$|f\kappa| = \sqrt{\mu}\kappa f/\sqrt{\mu} \leq \frac{1}{2}\mu\kappa^2 + \frac{1}{2\mu}f^2.$$



(a) Iteration 0 (b) Iteration 15000

(c) Iteration 25000 (d) Iteration 35000

(e) Iteration 45000 (f) Iteration 55000

Fig. 13. Figure showing different stages of curve evolution. The curve in yellow is the initial curve and the curve in pink shows the evolved curve at different stages. Initially the curve gets attracted by the weaker edge, eventually it recovers and covers the complete circle.

Hence,

$$\frac{d}{dt}L^t \leq -\frac{1}{2}\mu\int_{\Gamma^t}\kappa^2\,\mathrm{d}s + \frac{\|f\|_\infty^2}{2\mu}\int_{\Gamma}\mathrm{d}s \leq \frac{\|f\|_\infty^2}{2\mu}L^t.$$

Integrating this inequality we end up with the desired bound $L^t \leq L^0\exp\left(t\|f\|_\infty^2/(2\mu)\right)$, as claimed.

Again using Young's inequality we obtain

$$|\kappa| = \sqrt{2\varepsilon\mu}|\kappa|\sqrt{1/2\varepsilon\mu} \leq \varepsilon\mu\kappa^2 + \frac{1}{4\varepsilon\mu} = \varepsilon\kappa(\beta - f) + \frac{1}{4\varepsilon\mu}.$$

Hence

$$|\kappa| \leq \varepsilon(\delta^t + \|f\|_\infty)|\kappa| + \frac{1}{4\varepsilon\mu}.$$

Taking

$$\varepsilon = \frac{1}{2(\delta^t + \|f\|_\infty)}$$

we obtain

$$|\kappa| \leq \frac{1}{2}|\kappa| + \frac{\delta^t + \|f\|_\infty}{2\mu}$$

and thus, by integrating $|\kappa|$ over the curve Γ^t we obtain

$$\int_{\Gamma^t}|\kappa|\,\mathrm{d}s \leq \frac{\delta^t + \|f\|_\infty}{\mu}\int_{\Gamma^t}\mathrm{d}s = \frac{\delta^t + \|f\|_\infty}{\mu}L^t$$

as claimed. The third inequality follows from $\int_{\Gamma^t}|\kappa\beta|\,\mathrm{d}s \leq \delta^t\int_{\Gamma^t}|\kappa|\,\mathrm{d}s$. $\diamondsuit$

ACKNOWLEDGMENT

The authors wish to thank Prof. D. Daniel Ševčovič, Comenius University, and Prof. Harish Pillai, I.I.T Bombay, for discussion and suggestions. The authors thank Anup Shetty, V.I.P Lab, for generating additional results. Additionally, the authors are grateful to the Bharti Centre for Communication, Naval Research Board and the JC Bose National Fellowship scheme for financial support. Suggestions by reviewers have been extremely useful in revising the manuscript.

REFERENCES

[1] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *IJCV*, pp. 321–331, 1988.

[2] D. Freedman and Zhang, "Active contours for tracking distributions," *IEEE Trans. Image Proc.*, vol. 13, no. 4, pp. 518–527, April 2004.

[3] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Trans. PAMI*, vol. 22, no. 3, pp. 266–280, March 2000.
[4] C.Xu and J.L.Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans.Image Proc.*, vol. 7, no. 3, pp. 359–369, 1998.
[5] S. C. Zhu and A. L. Yuille, "Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation." *IEEE Trans. PAMI.*, vol. 18, no. 9, pp. 884–900, 1996.
[6] R. Malladi, J. A.Sethian, and B. C.Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. on PAMI.*, vol. 17, no. 2, pp. 158–175, 1995.
[7] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *IJCV*, vol. 22, no. 1, pp. 61–79, 1997.
[8] S.Kichenessamy, A. Kumar, P. Olver, A.Tannenbaum, and A.Yezzi, "Gradient flows and geometric active contour models," in *ICCV*, 1995, pp. 810–815.
[9] X. Xie and M. Mirmehdi, "MAC: Magnetostatic active contour model," *IEEE Trans. PAMI*, vol. 30, no. 4, pp. 632–646, April 2008.
[10] R. Ronfard, "Region based strategies for active contour models." *IJCV*, vol. 13, no. 2, pp. 229–251, October 1994.
[11] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
[12] P. Fua and Y. Leclerc, "Model driven edge detection," *Machine Vision and Applications*, vol. 3, pp. 45–56, 1990.
[13] R. Kimmel and A. M. Bruckstein, "Regularized laplacian zero crossings as optimal edge integrators," *IJCV*, vol. 53, no. 3, pp. 225–243, 2003.
[14] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *IJCV*, vol. 72, no. 2, pp. 195–215, 2007.
[15] H. Delingette, "On smoothness measures of active contours and surfaces," in *IEEE Workshop on VLSM*, Vancouver, Canada, July 2001, pp. 43–50.
[16] C. Li, J. Liu, and M. Fox, "Segmentation of edge-preserving gradient vector flow:an approach towards automatic initialization and splitting of snakes," in *CVPR*, 2005.
[17] M. Rochery, I. H. Jermyn, and J. Zerubia, "Higher order active contours," *IJCV*, vol. 69, no. 1, pp. 27–42, 2006.
[18] G. Charpiat, P. Maurel, J.-P. Pons, R.Keriven, and O. Faugeras, "Generalized gradients: Priors on minimization flows," *IJCV*, vol. 73, no. 3, pp. 325–344, 2001.
[19] G. Aubert, M. Barlaud, O. Faugeras, and S. Jehan-Besson, "Image segmentation using active contours: Calculus of variations or shape gradients?" *SIAM Journal on Applied Mathematics*, vol. 63, no. 6, pp. 2128–2154, 2003.
[20] Y. Rathi, N. Vaswani, and A. Tannenbaum, "A generic framework for tracking using particle filter with dynamic shape prior," *IEEE Tans. Image Processing*, vol. 63, no. 6, pp. 2128–2154, 2007.
[21] C. Xu, J. L. Prince, and A. Y. Jr, "A summary of geometric level-set analogues for a general class of parametric active contour and surface models," in *IEEE VLSM*, 2001, p. 104.
[22] Menet, Saint-Marc, and Medioni, "Active contour models: Overview, implementation and application," in *Proc. IEEE Conf. on SMC*, 1990, pp. 194–199.
[23] L. Cohen and I. Cohen, "Finite-element methods for active contour models and balloons for 2-d and 3-d images," *IEEE PAMI*, vol. 15, no. 11, pp. 1131–1147, November 1993.
[24] J. A. Sethian, *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
[25] S.Osher and R.Fedkiw, *Level Set Method and Dynamic Implicit Surfaces*. Springer, 2003.
[26] H. Delingette and J. Montagnat, "Shape and topology constraints on parametric active contours," *CVIU*, vol. 83, no. 2, pp. 140–171, 2001.
[27] Y. Shi and W. C. Karl, "Real-time tracking using level sets," in *CVPR*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 34–41.
[28] T. McInerney and D. Terzopoulos, "T-snakes: Topology adaptive snakes," *Medical Image Analysis*, vol. 4, no. 2, pp. 73–91, June 2000.
[29] L. Ji and H. Yan, "Robust topology-adaptive snake for image segmentation," *Image and Vision Computing*, vol. 20, no. 2, pp. 147–164, February 2002.
[30] V. Srikrishnan, S. Chaudhuri, S. D. Roy, and D. Sevcovic, "On stabilisation of parametric active contours," in *CVPR*, 2007.
[31] K. Mikula and D.Ševčovič, "Computational and qualitative aspects of evolution of curves driven by curvature and external force," *Comput. and Vis. in Science*, vol. 6, no. 4, pp. 211–225, 2004.
[32] D. Cremers, Schnorr, and J.Weickert, "Diffusion snakes: Combining statistical shape knowledge and image information in a variational framework," in *IEEE VLSM*, Vancouver, 2001.
[33] M. Jacob, T. Blu, and M. Unser, "Efficient energies and algorithms for parametric snakes," *IEEE Trans. on Image Proc.*, vol. 13, no. 9, pp. 1231–1244, September 2004.
[34] T.-J. Cham and R. Cipolla, "Automated b-spline curve representation incorporating mdl and error-minimizing control point insertion strategies," *IEEE Trans.PAMI*, vol. 21, no. 1, pp. 49–53, 1999.
[35] B. B. Kimia, A. R. Tannenbaum, and S. W. Zucker, "On the evolution of curves via a function of curvature, I: The classical case," *JMAA*, vol. 163, no. 2, pp. 438–458, January 1992.
[36] T. Gronwall, "Note on the derivatives with respect to a parameter of the solutions of a system of differential equations," *Ann. of Math*, vol. 20, no. 4, pp. 292–296, 1919.
[37] R. Howard, "The Gronwall Inequality," http://www.math.sc.edu/~howard/Notes/gronwall.pdf.
[38] L. Fox, *The Numerical Solution of Two-Point Boundary Problems in Ordinary Differential Equations*. Oxford, 1957.
[39] D. F. Rogers and J. A. Adams, *Mathematical elements for computer graphics*. New Yorl, NY, USA: McGraw-Hill, Inc., 1990.
[40] "Weizman database:action as space-time shapes," Website, http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html#Database.

V. Srikrishnan is currently a Ph.D student in the Electrical Engineering department at the Indian Institute of Technology, Bombay. His current research areas are active contours and visual tracking in different scenarios.

Subhasis Chaudhuri (SM'02) was born in Bahutali, India. He received the B.Tech. degree in electronics and electrical communication engineering from the Institute of Technology (IIT), Kharagpur, in 1985. He received the M.S. and the Ph.D. degrees, both in electrical engineering, from the University of Calgary, Calgary, Canada, and the University of California, San Diego, respectively. He joined the IIT, Bombay, in 1990 as an Assistant Professor and is currently serving as a Professor and the Head of the department. He has also held visiting positions with the University of Erlangen-Nuremberg, Technical University of Munich, Germany, and the University of Paris XI, France. His research interests include image processing, computer vision, and multimedia. He is a coauthor of the books *Depth From Defocus: A Real Aperture Imaging Approach* and *Motion-Free Super-Resolution*(New York: Springer, 1999). He has also edited a book *Super-Resolution Imaging*(Boston,MA:Kluwer Academic, 2001).

Dr. Chaudhuri is a Fellow of the Alexander von Humboldt Foundation, Germany, the Indian National Academy of Engineering, and the National Academy of Sciences, India. He is the recipient of the Dr. Vikram Sarabhai Research Award for the year 2001, and the Swarnajayanti Fellowship in 2003. He received the S.S. Bhatnagar Prize in engineering sciences for the year 2004. He was a Program Co-Chair for ICCV 2005 Conference held in Beijing, China.