Object Contour Tracking Using Level Sets

Alper Yilmaz
School of Computer Science
Univ. of Central Florida
yilmaz@cs.ucf.edu

Xin Li
Mathematics Dept.
Univ. of Central Florida
xli@math.ucf.edu

Mubarak Shah
School of Computer Science
Univ. of Central Florida
shah@cs.ucf.edu

Abstract— **High level vision tasks (recognition, understanding, etc.) for video processing require tracking of the complete contour of the objects. In general, objects undergo non-rigid deformations, which limit the applicability of using motion models (e.g. affine, projective) that impose rigidity constraints on the objects. In this paper, we propose a contour tracking algorithm for video captured using mobile cameras of different modalities. The proposed tracking algorithm uses Bayesian inference based on the probability density functions (PDFs) of texture and color features. These feature PDFs are fused in an independent opinion polling strategy, where the contribution of each feature is defined by its discrimination power. We formulate the evolution of the object contour as a variational calculus problem and solve the system using level sets. The associated energy functional combines region-based and boundary-based object segmentation approaches into one framework for object tracking in video, evaluated in the vicinity of the object contour. In this regard, it can be viewed as generalization of formerly proposed methods where the shortcomings of other methods (color, shape, gradient constraints, etc.) are overcome. The robustness of the proposed algorithm is demonstrated on real sequences.**

I. Introduction

In the recent years, a great deal of effort has been expanded on object tracking. There are three broad classes of methods for tracking multiple objects:

- *Correspondence-based object tracking:* requires object detection in every frame. Tracking is performed by establishing correspondence of the objects in consecutive frames. Objects are represented by their centroids or silhouettes.
- *Transformation-based object tracking:* requires object detection only once. Tracking is performed by estimating the motion of objects in consecutive frames. Object are represented by planar surfaces, such as rectangle and ellipse.
- *Contour-based object tracking:* requires object detection only once. Tracking is performed by finding the object contour given an initial contour from the previous frame. Objects are represented by contour.

Object detection, which is common in all the classes can be performed by feature detection (corner detectors), background subtraction or segmentation. Due to space limitations, we will not discuss all possible object detectors, however it is worth to mention several.

Harris corner detector is the most favorite feature detector [1]. It finds the variation of image intensities by generating a 2x2 matrix, $\mathbf{M}$, from first order image derivatives and computes the eigenvalues, λ, of $\mathbf{M}$. Pixels with high λ represent existence of a corner.

Background subtraction is the most popular detection method used in object trackers [2], [3], [4], where color observations of individual pixels in a reference frame are statistically modeled. Detection is performed by labeling the pixels that deviate from the background model. In [2], Wren *et al.* used a single Gaussian kernel to model YUV color space of each pixel. Stauffer and Grimson [3] generalized this scheme by modeling the RGB values of pixels using a Gaussian mixture model.

Segmentation is another object detection method. Although there are various ways to perform segmentation, due to the relevance to our approach, we will only review active contours (snakes) in the discussion on related work.

In the sequel of an detection method, correspondence based tracking can be achieved using object's state, where state variables can be composed of linear motion models (constant velocity and acceleration) coupled with regional information (color, texture, area and shape). These variables can be modeled using dynamic linear models [4], Gaussian kernels (Kalman filtering) [3] or Monte-Carlo techniques (particle filtering) [5]. Based on the model, object correspondence is performed by predicting the objects new position using past observations and verifying the existence of the object at the predicted position.

Transformation based tracking is performed directly by computing the transformation of the planar object region from one frame to the next. Object transformations used are translation, translation+scaling and affine motion models. One of the most common transformation based tracker is "template matching", where translation of an object template (rectangle) is computed by searching the image for a similar template. Template matching is computationally expensive and sensitive to illumination variation. Limitations of template matching is solved by the mean-shift tracker, which uses translation based motion model for rectangular or elliptical objects [6]. In mean-shift tracking, for each object, color priors are computed using weighted kernel density estimation, where the weights are obtained based on their distance from the object centroid. Mean-shift vector is then computed iteratively by maximizing likelihood ratio between the object color prior and the model generated from hypothesized object position. In [7], Shi and Tomasi proposed the KLT tracker, where object transformation is modeled by affine. Objects, detected using Harris detector, are represented by 25×25 windows. Similarly, Jepson et al. [8] proposed an object tracker which computed affine motion of an ellipsoidal object in consecutive frames using a variation of EM algorithm based on maximizing the likelihood of observing stable features (phase response of steerable pyramids) and object structure (ellipse), while dealing with the outliers (occlusions).

Tracking non-rigid objects can not be performed by using rigid-body motion models. Due to the fact that all rigid and non-rigid transformations reduce to translation of pixels inside the region, tracking can be defined as pixel translations. This leads to tracking both rigid and non-rigid objects using contours. Contour tracking can be performed by evolving the contour based on minimizing an energy. Compared to the silhouettes obtained from background subtraction, contour tracking produces tighter object boundaries. There are two variations of contour trackers in the literature, with two types of energy functionals:

- *Motion-based energy functional:* is defined based on optical flow to evolve an initial object contour.
- *Segmentation-based energy functional:* clusters the image into object and background regions using past observations.

In this paper, we propose a novel segmentation based energy functional for contour tracking, which is minimized by variational approach. Proposed functional is motivated by a Bayesian framework and is derived without imposing any constraints on object's shape, color or gradient, and it does not use parametric motion models. It fuses color and texture features computed from the object using independent opinion polling. Proposed functional builds a bridge between the boundary-based and region-based variational tracking methods which will be detailed in the following section. The functional is evaluated in the vicinity of the contour which increases stability of the solution due to locality. Thanks to texture features proposed method is less prone to lighting changes. The model priors are updated online to learn changes in the background model, thus it is suitable for tracking objects using mobile camera.

The paper is organized as follows: In Section II, we discuss related work on active contours, and state differences and similarities between our work and others. Section III details the proposed method, outlines the features used (§III-A) and presents the proposed contour energy functional (§III-B). Contour representation and related evolution equations based on the proposed energy functional is given in Section IV. Finally, the experimental results and conclusions are sketched in Sections V and VI respectively.

II. Related Work

Active contour (snake) was introduced to the vision community by Kass et al. [9]. In an active contour approach, the objective is to get tight contour enclosing the object. The contour, Γ, is usually represented explicitly by control points, $\mathbf{v}$, and is initialized by placing the control points outside the object region. Segmentation is obtained by evolving the contour toward the object region. Contour evolution is associated by minimizing an energy functional. Energy functionals in active contour framework include regularity and smoothness (internal) terms along with either boundary or region based image energies. Some approaches use external energies to increase stability of the evolution. A typical functional has the following form:

$$E(\Gamma) = \int_0^1 E_{internal}(\mathbf{v}) + \mathbf{E_{image}}(\mathbf{v}) + \mathbf{E_{external}}(\mathbf{v})\mathbf{ds} \quad (1)$$

where s is the arc-length of Γ. Kass et al. [9] used first order ($\nabla \mathbf{v}$) and second order ($\nabla^2 \mathbf{v}$) continuity terms for $E_{internal}$, which guaranteed elimination of gaps and rapid bending of the contour. The evolution was terminated by gradient magnitude, $|\nabla I|$, based boundary energy. In [10], authors used a greedy algorithm to minimize the contour energy, which was composed of second order regularity and curvature terms. The energy was computed using image gradient $|\nabla I|$ at each control point, i, which was normalized by $\max(|\nabla I_{i-1}|, |\nabla I_{i+1}|) - \min(|\nabla I_{i-1}|, |\nabla I_{i+1}|)$.

Caselles et al. [11] replaced the contour energy of (1) with $E(\Gamma) = \int_0^1 E_{internal}(\mathbf{v})\mathbf{E_{image}}(\mathbf{v})\mathbf{ds}$. E_{image} was set to $g(|\nabla I|)$, where g was a sigmoid function. In contrast to using control points, they implicitly represented the contour by a *level set* function (see §IV for details).

Image gradient based contour energy is not suitable to segment objects in textured background. To overcome this limitation, Ronfard [12] proposed a region-based energy, where statistical models were used for object and background regions. All contour points with a neighborhood that fits the object model are pushed outside, while, all contour points with a neighborhood that fits the background model are pulled inside. The force on the contour was the difference of the statistical fits to the object and background, which was formulated as Ward distance. Since Ward distance depends on the image data in a very intricate way, it is impossible to use conventional techniques to minimize the contour energy. Thus Ronfard followed a heuristic evolution approach. In another region based approach, Zhu and Yuille [13] defined a region to be homogeneous and its brightness (color) could be modeled using simple statistical models, such as a single Gaussian. In order to avoid over-segmenting the image, Zhu and Yuille used circular windows of m pixels around each point. At each evolution iteration, they computed the probability of pixels coming from the region prior, and evolved the boundary by minimizing $E(\Gamma)$. During iterations, if the adjacent regions had similar models, they merged them offline and continued the iterations until the image was segmented. Paragios and Deriche [14] extended the region model to the mixture of Gaussians for magnitude of Gabor filter responses. Their functional had both boundary energy (similar to [11]) and region energy (similar to [13]) in E_{image}, which were combined using convex combination, $E(\Gamma) = \int_0^1 \lambda E_{boundary}(\mathbf{v}) + (\mathbf{1} - \lambda)\mathbf{E_{region}}(\mathbf{v})\mathbf{ds}$. The boundary term was used to increase numerical stability.

If the contour is initialized with its previous position, contour segmentation approaches become object trackers (*segmentation-based* functional). On the other hand, contour tracking using *motion-based* functional is motivated by the availability of comprehensive study on optical flow. In its simplest form, optical flow constraint is defined by $I^{t+1}(x, y) - I^t(x - u, y - v) = 0$, where t is time t, and

(u, v) is flow vector.In [15], Bertalmio et al. used flow constraint to evolve object contour in consecutive frames. Their objective was to compute u and v iteratively for each contour position using level set representation. At each iteration, the contour speed in normal direction, $\overrightarrow{n}$, was computed by projecting the temporal gradient $|\nabla I_t|$ onto $\overrightarrow{n}$. The authors used two energy functions: contour tracking, $E_t(\Gamma) = \int_0^1 E_{external}(\mathbf{v})\mathbf{ds}$, and intensity morphing, $E_m(\Gamma) = \int_0^1 E_{image}(\mathbf{v})\mathbf{ds}$, where $E_{external}$ energy of E_t is based on E_m. $E_m(\Gamma)$, which minimized intensity changes in the current and previous frames, $\nabla I_t = I_t - I_{t-1}$, on the hypothesized object contour was coupled with $E_m(\Gamma)$. For instance, if $\nabla I_t(x, y) \gg 0$, then the contour moves with maximum speed in its normal direction and $I^{t-1}(x, y)$ is morphed into $I^t(x, y)$. Similarly, Mansouri [16] used the optical flow constraint for contour evolution. In contrast to [15], his approach was motivated by computing the flow vector for each regional pixel by brute-force-search in a circular neighborhood with radius r. Note that, motion-based methods do not require background modeling.

Contour tracking approach proposed in this paper is motivated by [12] (both him and us use subregion around the contour) and [13] (in terms of statistical ideas: conditional probabilities are used in an essential way in defining the functionals). Our work is different from [12] in terms of defining these subregions. We followed a rigorous approach motivated by Bayesian framework, which can be minimized by conventional techniques such as gradient descent. However, in [12], the energy was minimized heuristically, ie. solution is not stable, since derivatives of the energy functional could not be taken. Our functional is very similar in appearance to [13], but has very different interpretations. First, the probability in [13] is assumed to belong to a preassigned family of distributions. Second, within each region, all points are assumed to have the same distribution ([13, Equation (18)]). Third, the averaging operation is not directly related with the minimization since it only serves as a noise reduction operator similar to a low-pass filter, whereas our minimization functional is directly derived from the contour neighborhood. Finally, since [13] uses parametric form of the contour for segmentation, it makes the numerical implementation difficult and inefficient. In comparison, by using the level set, we only calculate energy in a small neighborhood of the object contour.

Our approach naturally is a generalization of previously proposed boundary based [9], [10], [11] and region based [12], [13], [14], [16] active contour methods by definition of a band around the contour, such that if the band size is set to 1, it becomes a boundary based method, or if the band is set such that it covers the complete object region, then it becomes a region based method. Details on the band definition will be given in Section III-B.

III. Methodology

Tracking an object in an image sequence $I^n : 0 < n < \infty$ can be treated as discriminant analysis of pixels in two non-overlapping classes, where the classes correspond to the object region, R_{obj} , and the background region, R_{bck}. The performance of discriminant analysis depends on selection of features. In this section, we detail the features used for this purpose and derive the tracking functional associated with discriminant analysis in the spatial domain.

A. Object Features And Modeling

During the last two decades, two classes of features have been widely considered for object tracking: *color feature* (is obtained from raw color values in an image) and *texture feature* (codes the repetitive details in an image).

We believe an ideal tracking approach should use both color and texture features. Therefore, in our approach, we used both of these features. In particular, raw color values are modeled by kernel density estimat*ion* using Epanechnikov kernel:

$$K(\mathbf{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1- \parallel \mathbf{x} \parallel^{\mathbf{2}}) & \text{if } \parallel \mathbf{x} \parallel < \mathbf{1} \\ 0 & \text{otherwise} \end{cases},$$

where c_d is the volume of unit d-dimensional sphere. For texture, we select a multi-scale and multi-oriented linear basis, specifically steerable pyramids (SP). In order to generate a disjoint feature space for creating independent PDFs based on texture analysis, we used Gabor wavelets, which create an orthonormal sub-band basis in SP:

$$G_i(x, y) = e^{-\pi[x^2/\alpha^2 + y^2/\beta^2]} \cdot e^{-2\pi i[u_0 x + v_0 y]},$$

where α and β specify width and height, while u_0 and v_0 specify modulation of the filter. We modeled both the magnitude and the phase responses of SP using Gaussian mixture model $N(\mu_k, \sigma_k)$. The probability of observing a value x is computed by $p(x|\Theta) = \sum_{k=1}^{C_N} P_k p_k(x|\mu_k, \sigma_k)$, where P_k is a priori component probability and $\Theta = \{P_k, \mu_k, \sigma_k : k = 1 \ldots C_N\}$. Fixing C_N to 3, unknowns are computed using Expectation Maximization (EM) algorithm.

Color and texture models together construct a semi-parametric feature model. Clustering pixels using this model should take discrimination power of the features into account. For this purpose, we generated the semi-parametric model using *independent opinion polling strategy* [17], in which integration of features are evaluated prior to object/background membership: $p(\mathbf{x}|\mathbf{M}_\alpha) = \prod_\beta \mathbf{p}_\beta(\mathbf{x}|\mathbf{M}_{\alpha,\beta})$, where $\mathbf{x}$ is the spatial variable, $\alpha \in \{\text{object}, \text{background}\}$ and $\beta \in \{\text{color}, \{\text{steerable sub-bands}\}\}$. Using Bayes' rule, a posteriori estimate of membership can be computed by:

$$p(\alpha|\mathbf{x}) = \frac{\prod_\beta \mathbf{p}_\beta(\mathbf{x}|\mathbf{M}_{\alpha,\beta})\mathbf{p}(\alpha)}{\sum_\gamma \prod_\beta \mathbf{p}_\beta(\mathbf{x}|\mathbf{M}_{\gamma,\beta})\mathbf{p}(\gamma)}, \tag{2}$$

where $\gamma \in \{\text{object}, \text{background}\}$. It can be easily observed from (2) that discriminant features will be emphasized.

B. Tracking Energy Functional

In the spatial domain, the object and background regions define $R = R_{obj} \cup R_{bck}$. Object contour, Γ, is defined by the intersection of directional curves corresponding to each region, such that $\Gamma = \Gamma_{obj} \cap \Gamma_{bck}$, where Γ_{obj} and

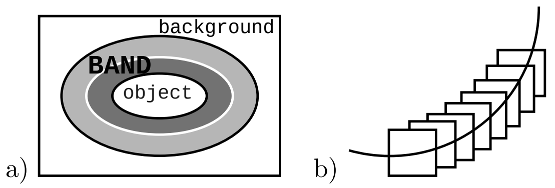

Fig. 1. (a) Object band (light gray), background band (dark gray) around the object contour (white ellipse), (b) subregions around the contour Γ defined by rectangles.

Γ_{bck} are respectively borders of the object and background regions in the counterclockwise direction. The likelihood of observing the boundary (contour) between two regions is equal to the likelihood of partitioning the regions:

$$P(\Gamma) = P\left(\varphi(R) = \{R_{obj}, R_{bck}\}\right), \tag{3}$$

where φ is the partitioning operator [13]. *Posteriori probability* for the boundary (left side of (3)) can be used interchangeably with *posteriori probability* of partitioning the space. Thus, for frame I^n, we formulate the object tracking problem in terms of the boundary probability, P_Γ, given I^n, and the previous object boundaries, $\Gamma^{n-1}, \Gamma^{n-2}, \ldots, \Gamma^1$: $P_\Gamma = P\left(\varphi(R^n)|I^n, \Gamma^{n-1}, \Gamma^{n-2}, \ldots, \Gamma^1\right)$. Using Markovian assumption and Bayes' rule, P_Γ is simplified to:

$$P_\Gamma \approx P\left(I^n|R^n_{obj}, \Gamma^{n-1}\right) P\left(I^n|R^n_{bck}, \Gamma^{n-1}\right) P\left(\varphi(R^n)|\Gamma^{n-1}\right) \tag{4}$$

The last term in (4) is based on only the previous contour observation, which corresponds to the shape of the object. For the sake of simplicity, we will assume $P\left(\varphi(R^n)|\Gamma^{n-1}\right) = 1$ and drop the term. Thus, the contour probability becomes $P_\Gamma = P\left(I^n|R^n_{obj}, \Gamma^{n-1}\right) P\left(I^n|R^n_{bck}, \Gamma^{n-1}\right)$. Let there be two subregions R^Γ_{obj} and R^Γ_{bck} defined in the neighborhood of the curve, such that $R^\Gamma_{obj} \subset R^n_{obj}$ and $R^\Gamma_{bck} \subset R^n_{bck}$. Due to the artifacts and noise, $P_\Gamma$can be defined in terms of subregions R^Γ_{obj} and R^Γ_{bck} as shown in Figure 1a using:

$$P_\Gamma \leqslant P'_\Gamma = P\left(I^n|R^\Gamma_{obj}, \Gamma^{n-1}\right) P\left(I^n|R^\Gamma_{bck}, \Gamma^{n-1}\right). \tag{5}$$

In the remainder of the paper, we will replace P_Γ with P'_Γ and assume that observation at each pixel is independent. Let $R_{obj}(\mathbf{x})$ and $R_{bck}(\mathbf{x})$ denote a neighborhood of $\mathbf{x}$ in R^Γ_{obj} and R^Γ_{bck} respectively. Then, we can estimate P'_{obj} as:

$$P'_\Gamma = \prod_{\mathbf{x_1}} \Bigg(\underbrace{\prod_{\mathbf{x_2}} P_{R^\Gamma_{obj}}(I^n(\mathbf{x_2}))}_{\text{object likelihood}} \underbrace{\prod_{\mathbf{x_3}} P_{R^\Gamma_{bck}}(I^n(\mathbf{x_3}))}_{\text{background likelihood}} \Bigg), \tag{6}$$

where $\mathbf{x_1} \in \mathbf{\Gamma}$, $\mathbf{x_2} \in \mathbf{R_{obj}}(\mathbf{x_1})$ and $\mathbf{x_3} \in \mathbf{R_{bck}}(\mathbf{x_1})$.

The maximum a posteriori (MAP) estimate of the contour being tracked, $\widehat{\Gamma^n}$, is found by maximizing the probability P'_Γ over the subsets $\Gamma \subset \Omega$, where Ω is the space of all object contours. Based on (6), $\widehat{\Gamma^n}$ can be written in terms of the subregions defined by Γ:

$$\widehat{\Gamma^n} = \arg\max_{\Gamma \subset \Omega} \prod_{\mathbf{x_1}} \Bigg(\prod_{\mathbf{x_2}} P_{R^\Gamma_{obj}}(I^n(\mathbf{x_2})) \prod_{\mathbf{x_3}} \mathbf{P_{R^\Gamma_{bck}}(I^n(x_3))} \Bigg). \tag{7}$$

In (7), the band around the hypothesized object contour serves both as *boundary constraint* [9], [10], [11] and as a *region constraint* [13], [14] generalizing previously proposed active contour methods into one framework. The advantages of using the band around the boundary compared to using the complete region are:

- Reduced contour search space;
- Noise and artifacts (holes in the object) are not considered in contour estimation;
- Boundary and region region based energy functionals are generalized into one framework.
- It allows object tracking using mobile cameras by adapting to the local changes around the object contour.

A convenient way of converting a MAP estimation to a minimization problem is by computing the negative log-likelihood of probabilities. Thus, the tracking scheme proposed in (7) becomes:

$$\begin{aligned} E(\Gamma) &= \int_{x_1 \in \Gamma} \Bigg(\underbrace{\int\!\!\int_{\mathbf{x_2} \in \mathbf{R_{obj}(x_1)}} \Psi_{obj}(\mathbf{x_2})\mathbf{dx_2}}_{E_A} + \\ &\quad \underbrace{\int\!\!\int_{\mathbf{x_3} \in \mathbf{R_{bck}(x_1)}} \Psi_{bck}(\mathbf{x_3})\mathbf{dx_3}}_{E_B} \Bigg) d\mathbf{x_1}, \end{aligned} \tag{8}$$

where $\Psi_{obj}(\mathbf{x}) = -\log \mathbf{P_{R_{obj}}(I^n(x))}$ and $\Psi_{bck}(\mathbf{x}) = -\log \mathbf{P_{R_{bck}}(I^n(x))}$. The planar integrals given in (8) are not defined. To define these integrals, we select subregion $R_{obj} \cup R_{bck}$ to be a square region $[-m, m]$x$[-m, m]$ centered around $\mathbf{x_i}$ (Figure 1b). The pixels inside the $R_{obj} \cup R_{bck}$ are defined for each contour position $(f(s), g(s))$ by $x = \tilde{x} + f(s)$ and $y = \tilde{y} + g(s)$, where f and g are parametric curve functions with parameter s, $\tilde{x}, \tilde{y} \in [-m, m]$. To compute object and background probabilities inside subregion, we define an indicator function of the form:

$$1^\Gamma_\alpha\{\mathbf{x} \in \mathbf{R}_\alpha\} \quad = \quad \begin{cases} 1 & \mathbf{x} \in \mathbf{R}_\alpha \\ 0 & \textit{otherwise} \end{cases},$$

Using the above definitions, the functional E_A becomes:

$$E_A(s) = \int_{-m}^{m} \int_{-m}^{m} \Psi\left(x, y\right) J 1^\Gamma_{obj}(x, y) d\tilde{x} d\tilde{y},$$

where J is the *Jacobian* introduced due to the change of variables and x, y are defined above. Due to the translation of the square window around Γ the $J = 1$ and is dropped. Once E_B is rewritten, (8) results in the tracking functional:

$$\begin{aligned} E \quad &= \quad \int_0^1 \Bigg[- \overbrace{\iint_{-m}^{m} \log P_{R_{obj}}(I(\mathbf{x}))\mathbf{1^\Gamma_{obj}}\{\mathbf{x}\}\mathbf{d\tilde{x}d\tilde{y}}}^{\Phi_{obj} \Rightarrow \text{posteriori object log likelihood}} - \\ &\quad \underbrace{\iint_{-m}^{m} \log P_{R_{bck}}(I(\mathbf{x}))\mathbf{1^\Gamma_{bck}}\{\mathbf{x}\}\mathbf{d\tilde{x}d\tilde{y}}}_{\Phi_{bck} \Rightarrow \text{posteriori background log likelihood}} \Bigg] ds \end{aligned} \tag{9}$$

where l is contour length and $1^{\partial\Gamma}_{\text{bck}} = 1 - 1^{\partial\Gamma}_{\text{obj}}$.

The functional given in (9) is very general and functionals proposed in [11], [13], [14], [16] are special cases of (9):

- Limiting the discriminant analysis to the pixels inside the region and setting the probabilities of (9) to

$$P_\alpha(\mathbf{x}) = \max_{\mathbf{z}:\|\mathbf{z}\|\leq\mathbf{m}} \mathbf{e}^{-\frac{\left(\mathbf{I^{n-1}(x)-I^{n}(x+z)}\right)^{2}}{\mathbf{2\sigma^2}}}$$

and dropping the plane integrals (due to max operation) results in the tracking scheme proposed in [16].

- Replacing the probabilities in (9) with the $e^{-|\Delta I|}$ reduces to energy proposed in [11].
- Functional given in (9) can be simplified to [13] by increasing m such that the band covers both object and background regions. In this case, the front dependent subregions R_α^Γ of (8) are changed to region terms R_α.
- Similarly, convex combination of boundary and regional forces given in [14] is unified through regional probabilities in (9) attracted by the contour which defines the regions.

C. Minimizing the Tracking Functional

Tracking using (9) requires that evolved object contour has minimum energy. The first order necessary condition in this regard is to compute the derivative of the energy functional. Associated Euler-Lagrange equations of (9) are:

$$\tfrac{\delta E}{\delta x} = -(\Phi_{\mathrm{obj}} + \Phi_{\mathrm{bck}})\dot{y}, \quad \tfrac{\delta E}{\delta y} = (\Phi_{\mathrm{obj}} + \Phi_{\mathrm{bck}})\dot{x}, \qquad (10)$$

where Φ_{obj} and Φ_{bck} are defined in (9). An interesting observation about the speed function in (10) is that it is a differential equation with integrals, and are called nonlinear first order partial integro-differential equations, which are not well-studied. This is different than the integral terms in [13], which was used for averaging purposes. Although convergence issue needs analysis, a practical approach to show its convergence is to assume that the rectangle centered on the contour has equal object and background regions on all the contour positions. Thus, the planar integral can be dropped and the system of equations can be transformed into a second order differential equation.

IV. Level Sets and Contour Evolution

Contours can be implicitly represented using level sets. Level set function, $\phi : R^2 \to R^1$, is a grid, where each grid position has a value representing a level. The contour, Γ, in level set function is defined by the 0^{th} level, $\phi(\Gamma(s,t),t) = 0$, and other grid positions in ϕ carry negative (inside Γ) and positive (outside Γ) values. Evolution of the contour in level set function is obtained by modifying ϕ: $\phi_t(\Gamma(s,t),t) = F(\mathbf{x},\mathbf{t})|\nabla\phi(\mathbf{\Gamma}(\mathbf{s},\mathbf{t}),\mathbf{t})|$ with speed F in the normal direction, $\vec{n}$, such that changing the values creates new zero crossings. The Euler-Lagrange equations given in (10) can be related to contour evolution on level sets by defining F. To do so, we rewrite (10):

$$\frac{\delta E}{\delta \vec{\mathbf{v}}} = -(\Phi_{obj} + \Phi_{bck}) \begin{bmatrix} -\dot{y} & \dot{x} \end{bmatrix}^T \qquad (11)$$

where $\vec{\mathbf{v}} = (x, y)$. Note that, object and the background normals are $\vec{\mathbf{n}}_{obj} = (\dot{y}, -\dot{x})$ and $\vec{\mathbf{n}}_{bck} = (-\dot{y}, \dot{x})$. Thus, (11) relates to F:

$$\begin{aligned} F_{x,y} \quad = \quad & -\sum_{i=-m}^{m}\sum_{j=-m}^{m} \log P_{R_{obj}}(I_{\mathbf{x}'}) 1_{obj}^{\Gamma}\{(\mathbf{x}')\} + \\ & \sum_{i=-m}^{m}\sum_{j=-m}^{m} \log P_{R_{bck}}(I_{\mathbf{x}'}) 1_{bck}^{\Gamma}\{(\mathbf{x}')\}, \qquad (12) \end{aligned}$$

where $\mathbf{x}' = (\mathbf{x} + \mathbf{i}, \mathbf{y} + \mathbf{j})$. Negative and positive terms correspond to the shrinking and expanding forces and are due to the opposite normal directions. Equation 12 can be interpreted as follows:

- When the contour is correctly positioned, $F \approx 0$.
- If the contour is not correct, background (object) probability will be higher and F becomes negative (positive).

V. Experiments

To demonstrate the performance of the proposed contour tracker, we have experimented with various sequences captured with infrared and electro-optical cameras and obtained excellent results. Model priors are computed online by reevaluating the change in the object and background features. The level set contour evolution is implemented using the narrow band method, where (12) is used as the speed function. The algorithm is initialized with boundaries of objects in the first frame. Selection of m in (12) is not sequence dependent and is fixed to 6 for all sequences. In contrast to [16], magnitude of motion is not constrained.

In Figure 2i, the results are shown for the standard tennis player sequence. The player generates non-rigid motion while the camera pans. The tracking performance is the best among other tracking methods including [18] and [19].

Shown in Figure 2ii, the method is tested on a sequence captured using mobile camera. Both texture and color features played roles during the course of tracking. For instance, in regions, where the texture is similar with the background, the method chose color to perform tracking, while in regions, where the color of the pants is similar with the background, the texture features are emphasized.

We also tested our method on a low quality sequence captured using surveillance camera (Figure 2iii). The tracked person in the sequence undergoes high nonrigid motion. Due to the similarities in the object and background texture color feature contributes more to the tracking. The object contour is correctly tracked throughout the sequence. Results given in Figure 2iv show where both features simultaneously contribute to the tracking. The contour of the object is correctly tracked while the camera pans.

We also tested the method on IR sequences. In addition to the problems due to clutter, objects sometimes are not visible or distinguishable from the background. Due to these limitations [16] (use intensity differences) [14] (use only texture) will perform poor on these sequences. In Figure 3, we show two different closing sequences, where the target size increases. Since the imagery is of low quality and the atmospheric effects cause changes in the features, online models can adapt to the changes in the scene.

To conclude, experiments demonstrated here show the superiority of the proposed method over the methods cited

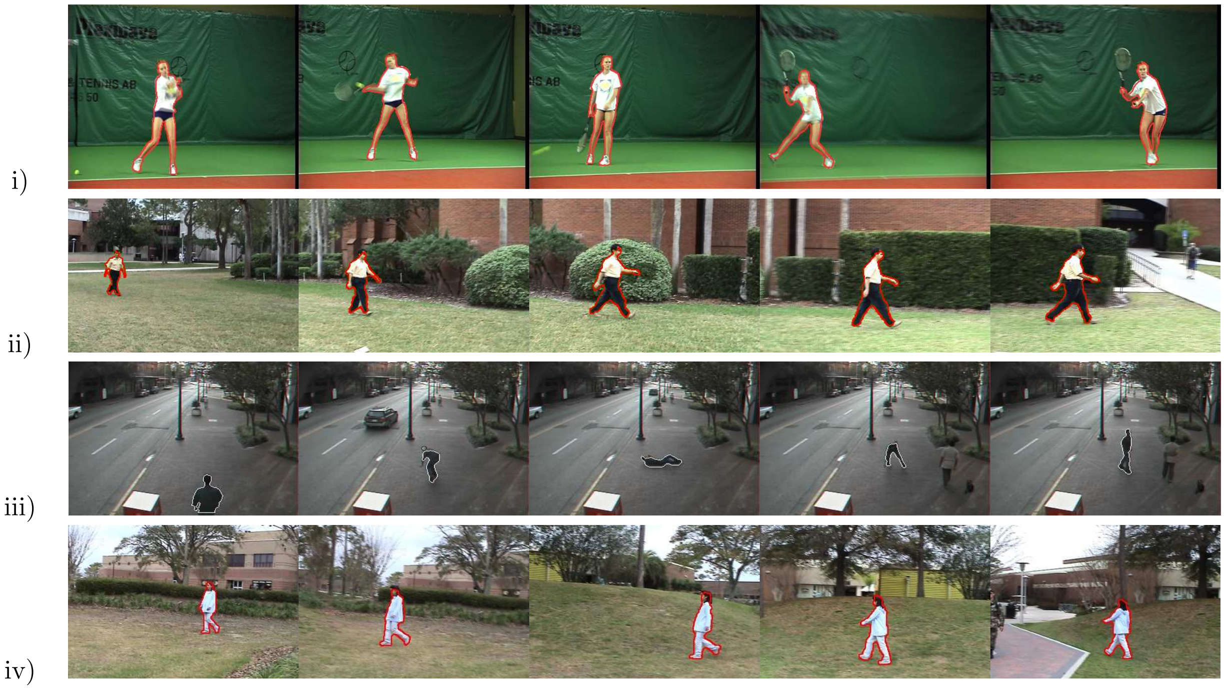

Fig. 2. Contour tracking results for: (i) non-rigid tennis player sequence; (ii) object that has similar feature set with the changing background (iii) sequence captured by a stationary surveillance camera; (iv) changing background. We suggest colored printout to view the results.

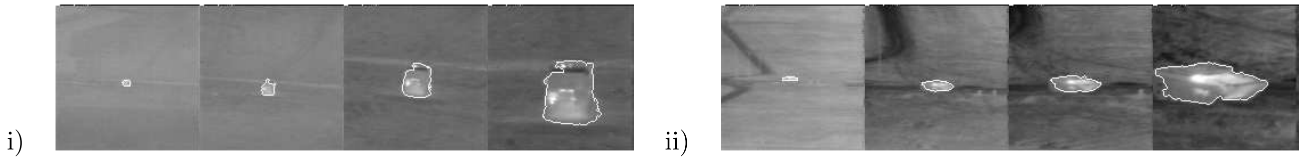

Fig. 3. Contours tracking of targets in closing infrared sequences taken from an airborne vehicle. (i) Sequence RNG14_15 every 30^{th} frame; (ii) sequence RNG16_18 every 30^{th} frame.

here. In general, all background subtraction based algorithms will fail for sequences given in Figure 2 except for 2iii where the camera is stationary. The methods given in [9], [10], [11] will not work for all these sequences due to the dependence on image or temporal gradients. Texture based methods given in [8], [14] will fail for sequences in Figures 2i, iii and 3 due to the dependence on purely repetitive texture information which is not present.

VI. Conclusions

We proposed an contour tracking method for tracking non-rigid objects in video captured from mobile cameras. The method generates online color and texture models for both object and background regions. Based on the feature priors, object tracking is formulated as maximization of a posteriori contour probability evaluated in the vicinity of the contour, given the previous contour observations. The locality introduced by a band around the contour suppresses the noise and artifacts that generally occur during the course of tracking and increase stability of the solution. The minimization of the proposed tracking method is performed by evolving the contour in the steepest descent direction, using the implicit level set representation. The results presented show robustness of tracking algorithm for EO and IR imagery.

References

[1] C.G. Harris and M. Stephens, "A combined corner and edge detector," in *4th Alvey Vision Conference*, 1988, pp. 147–151.

[2] C.R. Wren, A. Azarbayejani, and A. Pentland, "Pfinder: Real-time tracking of the human body," *PAMI*, vol. 19, no. 7, pp. 780–785, 1997.

[3] C. Stauffer and W.E.L. Grimson, "Learning patterns of activity using real time tracking," *PAMI*, vol. 22, no. 8, pp. 747–767, 2000.

[4] O. Javed and M. Shah, "Tracking and object classification for automated surveillance," in *ECCV*, 2002, vol. 4, pp. 343–357.

[5] J. Rittscher, J. Kato, S. Joga, and A. Blake, "A probabilistic background model for tracking," in *ECCV*, 2000, vol. 2, pp. 336–350.

[6] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *CVPR*, 2000, vol. 2, pp. 142–149.

[7] J. Shi and C. Tomasi, "Good features to track," in *CVPR*, 1994, pp. 593–600.

[8] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi, "Robust online appereance models for visual tracking," in *CVPR*, 2001, vol. 1, pp. 415–422.

[9] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *IJCV*, vol. 1, pp. 321–332, 1988.

[10] D. Williams and M. Shah, "A fast algorithm for active contours and curvature estimation," *CVGIP*, vol. 55, no. 1, pp. 14–26, 1992.

[11] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," in *ICCV*, 1995, pp. 694–699.

[12] R. Ronfard, "Region based strategies for active contour models," *IJCV*, vol. 13, no. 2, pp. 229–251, 1994.

[13] S.C. Zhu and A. Yuille, "Region competition: unifying snakes,

region growing, and bayes/mdl for multiband image segmentation," *PAMI*, vol. 18, no. 9, pp. 884–900, 1996.

[14] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *IJCV*, vol. 46, no. 3, pp. 223–247, 2002.

[15] M. Bertalmio, G. Sapiro, and G. Randall, "Morphing active contours," *PAMI*, vol. 22, no. 7, pp. 733–737, 2000.

[16] A.R. Mansouri, "Region tracking via level set pdes without motion computation," *PAMI*, vol. 24, no. 7, pp. 947–961, 2002.

[17] P. Torr, R. Szeliski, and P. Anandan, "An integrated bayesian approach to layer extraction from image sequences," *PAMI*, vol. 23, no. 3, pp. 297–303, 2001.

[18] E. Hayman and J. Eklundh, "Probabilistic and voting approaches to cue integration for figure ground segmentation," in *ECCV*, 2002, pp. 469–486.

[19] S. Khan and M. Shah, "Object based segmentation of video using color motion and spatial information," in *CVPR*, 2001, vol. 2, pp. 746–751.