Joint Registration and Active Contour Segmentation for Object Tracking

Jifeng Ning[a,b], Lei Zhang[b,1], *Member, IEEE,* David Zhang[b], *Fellow, IEEE* and Wei Yu[a]

[a] College of Information Engineering, Northwest A&F University, Yangling, Shaanxi, China

[b] Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong, China

Abstract: This paper presents a novel object tracking framework by joint registration and active contour segmentation (JRACS), which can robustly deal with the non-rigid shape changes of the target. The target region, which includes both foreground and background pixels, is implicitly represented by a level set. A Bhattacharyya similarity based metric is proposed to locate the region whose foreground and background distributions can best match those of the tracked target. Based on this metric, a tracking framework which consists of a registration stage and a segmentation stage is then established. The registration step roughly locates the target object by modeling its motion as an affine transformation, and the segmentation step refines the registration result and computes the true contour of the target. The robust tracking performance of the proposed JRACS method is demonstrated by real video sequences where the objects have clear non-rigid shape changes.

Index Terms: Object tracking, active contour model, level set, segmentation, registration

[1] Corresponding author: cslzhang@comp.polyu.edu.hk.

This work is supported by the National Science Foundation Council of China under Grants 61003151 and the Fundamental Research Funds for the Central Universities under Grant No.QN2009091.

1. Introduction

Visual tracking is an important yet challenging task in various computer vision applications. In the past decades many algorithms have been proposed for robust object tracking, aiming to overcome the difficulties arising from noises, occlusions, clutters, and changes in the foreground object and/or in the background environment [1-5]. In particular, how to design the trackers that can handle the target object shape (or contour) changes is one of the hottest topics of object tracking [6-15].

Because level set can deal with object topological changes seamlessly, many trackers aim to describe the motion change information of the object using the active contour method. Freedman and Zhang [11] located the best candidate region by matching object distributions using the Bhattacharyya similarity and Kullback-Leibler distance, respectively. Afterwards, they improved it by combining foreground matching flow and background mismatching flow, proposing the so-called the combination flow method [12]. However, both the methods in [11] and [12] are level set based image segmentation method with prior distribution. They often need many times of iterations to converge when the initial contour is far from the target true contour.

On the other hand, some template (e.g., using a simple rectangle or an ellipse) based trackers [16-17] often perform well in real time under complex scenes, but they are difficult to track the target with complex contour. Methods in [18] and [19] track the target as a changing ellipse obtained by estimating the covariance matrix in scale and orientation. Yilmaz [8] made an attempt to deal with the scale and orientation changes of the target. He extended the traditional mean shift tracker [17] by using an asymmetric kernel (level set) to represent the target. Riklin-Raviv *et al.* [20] used projective transformation to segment an image by using a

single prior shape but without using any point correspondences. Recently, Chiverton *et al.* [21] proposed an online active contour based learning model, which consists of two components: a motion based bootstrapper to extract the target shape information from previous tracking results and a region based tracker to optimize the extracted shape by active contour. In summary, it is of high interest to develop a tracking method which can possess the advantages of both the template based trackers (e.g., having less iteration numbers for converge) and the segmentation based trackers (e.g., tracking the true contour of the target).

To achieve the above mentioned goal, in this paper we propose a novel tracking framework to track the true contour change of the target. In the proposed method, the target region, including the foreground and background components, is represented by a level set. By using the Bhattacharyya similarity [22-24] as a measure for target matching, we locate the candidate target region such that its foreground distribution and background distribution are most similar to the user defined target model. A novel tracking algorithm via joint registration and active contour segmentation (JRACS) is then developed. The proposed JRACS consists of two stages. First, the registration procedure estimates the affine deformation of the target. This stage can be considered as a template based tracker, while it uses arbitrary shape (level set), instead of the simple rectangle or ellipse template, to represent the target. This makes the proposed method powerful to estimate non-rigid motion of the target. Second, the segmentation procedure refines the affine transformation estimated in the registration stage, and computes the target's true shape accurately. Finally, on-line target appearance updating is used to remove tracking drift. Extensive experiments on typical videos validate the effectiveness of our methods.

The proposed method is partially inspired by the works in [11, 12, 16, 17]. However, different from mean-shift [16] and foreground flow [11] methods which perform only foreground matching, the proposed JRACS model matches both foreground and background. In addition, the classical mean-shift tracker considers the target motion as a translation transformation; the lately developed EM-shift [18] and SOAMST [19] methods track the target with a changing ellipse; methods in [11] and [12] are actually two image segmentation methods; in contrast, the proposed JRACS method considers the non-rigid arbitrary shape changes of the target object.

The rest of this paper is organized as follows. Section 2 describes the target representation. Section 3 describes in detail the proposed JRACS method. Section 4 presents the implementation of the JRACS. Section 5 presents the experimental results and Section 6 concludes the paper.

2. Target Representation

To track robustly the target object in the video sequence, we need to represent the target as robust as possible. The region where the target locates can be represented as an ellipse, a rectangle or an arbitrary contour. Once the region is fixed, there are various features that can be used to describe the target defined in it, such as color, edge and texture features, or the combination of them. In this paper, we select color histogram to model the target object because of its merits such as independence of scaling and rotation, robustness to partial occlusions, low computational cost, etc. Of course, when object and background differs much from each other on texture features, the texture histogram can also be used to represent the target. Follow the notation in [16], we denote by u the feature space of the object indicated by the user, denote by m the number of features, and define the foreground distribution $\mathbf{q}$ and

background distribution **o** as follows:

$$\mathbf{q} = \{q_u\}_{u=1,\cdots,m} \quad \sum_{u=1}^{m} q_u = 1 \tag{1}$$

$$\mathbf{o} = \{o_u\}_{u=1,\cdots,m} \quad \sum_{u=1}^{m} o_u = 1 \tag{2}$$

To track the target object whose region is initialized in the first frame, in the subsequent frames we attempt to find the best candidate region under a certain metric. We use level set to represent the target region because of its flexibility in representing an arbitrary target contour. Let level set function Φ denote a candidate target. We use $p(\Phi)$ to stand for the distribution of foreground region $\Phi \geq 0$, and use $v(\Phi)$ to stand for the distribution of background region $-d < \Phi < 0$, where threshold d is used to restrict the interested region into a small area. The two distributions $p(\Phi)$ and $v(\Phi)$ will match **q** and **o**, respectively, under certain metrics.

Fig. 1 shows an example by using level set to represent the contour of a car. The region enclosed by the internal curve (in blue color) is the foreground, while the region enclosed between the internal and external (in red color) curves is the background. $p(\Phi)$ and $v(\Phi)$ are respectively defined as follows:

$$\mathrm{p}(\Phi) = \{p_u(\Phi)\}_{u=1,\cdots,m} \quad \sum_{u=1}^{m} p_u = 1 \tag{3}$$

$$\mathrm{v}(\Phi) = \{v_u(\Phi)\}_{u=1,\cdots,m} \quad \sum_{u=1}^{m} v_u = 1 \tag{4}$$

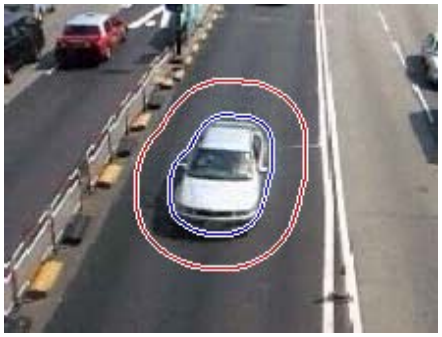


Figure 1: Target representation implicitly by using level set.

Let $\left\{\mathrm{x}_{f,i}^{*}\right\}_{i=1\cdots n_f}$ and $\left\{\mathrm{x}_{b,i}^{*}\right\}_{i=1\cdots n_b}$ be the pixels falling into the foreground part and background part, respectively. Next we define the foreground model and background model. The function $b:R^2\to\{1\cdots m\}$ maps the pixel at location x_i into the bin $b\left(\mathrm{x}_i^{*}\right)$ in the quantified feature space. Then, the probability of the feature *u*=1, 2, …, *m* in the foreground model and background model are respectively calculated as

$$q_u=\frac{1}{n_f}\sum_{i=1}^{n_f}\delta\left[b\left(\mathrm{x}_{i,f}^{*}\right)-u\right] \tag{5}$$

$$o_u=\frac{1}{n_b}\sum_{i=1}^{n_b}\delta\left[b\left(\mathrm{x}_{b,i}^{*}\right)-u\right] \tag{6}$$

where δ is the Kronecker delta function. Normalized constants n_f and n_b are the numbers of foreground pixels and background pixels, which make $\sum_{u=1}^{m}q_u=1$ and $\sum_{u=1}^{m}o_u=1$.

Similarly, we compute foreground distribution and background distribution in the current candidate region $\left(-d<\Phi<0\right)$ as follows:

$$p_u\left(\Phi\right)=\frac{1}{A_f}\sum_{i=1}^{n}H\left(\Phi\left(\mathrm{x}_i\right)\right)\delta\left[b\left(\mathrm{x}_{\mathrm{i}}\right)-u\right] \tag{7}$$

$$v_u\left(\Phi\right)=\frac{1}{A_b}\sum_{i=1}^{n}\left(1-H\left(\Phi\left(\mathrm{x}_i\right)\right)\right)\delta\left[b\left(\mathrm{x}_i\right)-u\right] \tag{8}$$

where *n* is the number of pixels in the candidate region, the Heaviside function $H(\cdot)$ is used to select foreground region, and thus $1-H(\cdot)$ is employed to select background region. The normalization factors A_f and A_b, making $\sum_{u=1}^{m}p_u\left(\Phi\right)=1$ and $\sum_{u=1}^{m}v_u\left(\Phi\right)=1$, are derived as follows:

$$A_f=\sum_{i=1}^{n}H\left(\Phi\left(\mathrm{x}_i\right)\right) \tag{9}$$

$$A_b=\sum_{i=1}^{n}\left(1-H\left(\Phi\left(\mathrm{x}_i\right)\right)\right) \tag{10}$$

We select $H(x)=\frac{1}{2}\left(1+\frac{2}{\pi}\arctan\left(\frac{x}{\varepsilon}\right)\right)$ as the Heaviside function in the level set because of its many advantages [25]. As shown in Fig. 2, the Heaviside function can also be regarded as a kernel function, like the Epanechnikov kernel or Gaussian kernel in the mean shift tracker [16].

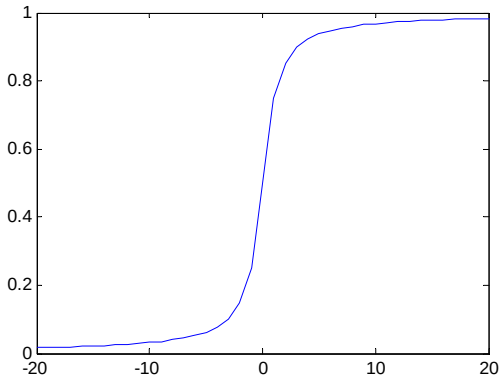


Figure 2: Heaviside function with ε=1.0.

3. The joint registration and segmentation

In the proposed method, the Bhattacharyya similarity is adopted for target matching. Based on this metric, we derive a registration formula and a segmentation formula. The registration formula is used to estimate the affine deformation of the target, and the segmentation formula is used to refine the registration results so that the contour of the target can be obtained.

3.1 Matching metric

Some recently proposed tracking methods achieve good results by constructing an online discriminative classifier using both the foreground and background features [26-28]. The combination flow method [12] also demonstrates that the background information is important to obtain good result for level set based tracking methods. Nonetheless, because both the foreground and background information of the target changes smoothly in most videos, the background information is also useful to estimate accurately the target contour change.

Our goal is to find a region in the current frame such that its foreground distribution p(Φ) and background distribution v(Φ) can best match the model foreground distribution **q** and the model background distribution **o**, respectively. There are many kinds of criteria [29] that can be used to compare the similarity of these distributions. In this paper, we adopt the Bhattacharyya similarity [22-24] because of its successful applications in object tracking. The Bhattacharyya coefficient is a divergence-type measure which has a straightforward geometric interpretation. It is the cosine of the angle between **q** and p(Φ) or between **o** and v(Φ). The higher the Bhattacharyya coefficient between target model and candidate target model, the higher the similarity between them. In our model, since both the foreground and background are considered, we define the similarity distance measure as

$$E(\Phi)=\sum_{u=1}^{m}\left(\sqrt{p_u(\Phi)q_u}+\lambda\sqrt{v_u(\Phi)o_u}\right) \tag{11}$$

where the weight λ balances the contributions of foreground and background in the matching.

3.2 Deformation estimation

Deformation estimation is used to handle the shape change of the target. The template based trackers [16-19] usually consider the motion of the target as a translation transform or simple zooming change, but they encounter difficulties when the contour of the target presents large non-rigid change. On the other hand, although the segmentation based tracking methods can handle non-rigid motion, they usually require more time to converge and are prone to local minima. Therefore, our goal is to present a novel framework which can combine the advantages of the above two kinds of trackers.

Let Φ_0 be the initial position of the target in the current frame. The contour of the target can be obtained by letting Φ_0=0. Thus, the two probabilities $\mathrm{p}(\Phi_0)=\{p_u(\Phi_0)\}_{u=1,\cdots,m}$ and $\mathrm{v}(\Phi_0)=\{v_u(\Phi_0)\}_{u=1,\cdots,m}$ can be computed first. Suppose that Φ (Φ>-*d*) is the next position of

the target, which is adjacent to Φ_0. Similar to the derivation of classical mean shift tracker [16], by applying Taylor expansion around $p_u(\Phi_0)$ and $v_u(\Phi_0)$, we have

$$\begin{aligned} E(\Phi) = & \frac{1}{2}\left(\sum_{u=1}^{m} \sqrt{p_u(\Phi_0) q_u} + \sum_{u=1}^{m} p_u(\Phi) \sqrt{\frac{q_u}{p_u(\Phi_0)}} \right) \\ & + \frac{1}{2}\lambda \left(\sum_{u=1}^{m} \sqrt{v_u(\Phi_0) o_u} + \sum_{u=1}^{m} v_u(\Phi) \sqrt{\frac{o_u}{v_u(\Phi_0)}} \right) \end{aligned} \tag{12}$$

Furthermore, by substituting Eq. (7) and Eq. (8) into Eq. (12), we have

$$\begin{aligned} E(\Phi) = & \frac{1}{2}\left(\sum_{u=1}^{m} \sqrt{p_u(\Phi_0) q_u} + \frac{1}{A_f} \sum_{i=1}^{n} w_{f,i} H(\Phi(\mathrm{x}_i)) \right) \\ & + \frac{1}{2}\lambda \left(\sum_{u=1}^{m} \sqrt{v_u(\Phi_0) o_u} + \frac{1}{A_b} \sum_{i=1}^{n} w_{b,i} (1 - H(\Phi(\mathrm{x}_i))) \right) \end{aligned} \tag{13}$$

where

$$w_{f,i} = \sum_{u=1}^{m} \sqrt{\frac{q_u}{p_u(\Phi_0)}} \delta[b(\mathrm{x}_i) - u] \tag{14}$$

$$w_{b,i} = \sum_{u=1}^{m} \sqrt{\frac{o_u}{v_u(\Phi_0)}} \delta[b(\mathrm{x}_i) - u] \tag{15}$$

It is worth noting that the original mean-shift tracking method [16] considers the motion of the target as purely translation, and the weight $w_{f,i}$ plays a key role in finding the new centroid of the target. It indicates the possibility that pixel x_i belongs to the foreground. In our method, $w_{f,i}$ plays the similar role. Similarly, $w_{b,i}$ indicates the possibility that pixel x_i belongs to the background. As we will see later, $w_{f,i}$ and $w_{b,i}$ will guide the registration and segmentation of level set Φ.

Maximizing Eq. (13) is equivalent to maximizing the Bhattacharyya coefficient defined in Eq. (11), which is a function about location x and level set function Φ. Next, we derive the formulas for optimizing location and contour, respectively.

3.2.1 Registration. We model the motion of the target as an affine transformation. To this end, we introduce a warp $\mathrm{x}=\mathrm{w}(\mathrm{x},\Delta T)$ [30] into Eq. (13) to model and estimate the affine transformation of the target:

$$\mathrm{x}=\mathrm{w}(\mathrm{x},\Delta T)=\begin{pmatrix}1+p_1 & p_3 & p_5\\ p_2 & 1+p_4 & p_6\end{pmatrix}\begin{pmatrix}x\\ y\\ 1\end{pmatrix} \tag{16}$$

where the column vector $\Delta T=(p_1,p_2,p_3,p_4,p_5,p_6)'$ has 6 parameters to characterize the pose change of the object, and (x,y) is the column and row coordinate at pixel x.

Substituting Eq. (16) into Eq. (13) and omitting the terms that are not a function of ΔT, we have

$$E(\Phi)=\frac{1}{2A_f}\sum_{i=1}^{n}H\left(\Phi\left(\mathrm{w}(\mathrm{x}_i,\Delta T)\right)\right)w_{f,i}+\frac{1}{2A_b}\lambda\sum_{i=1}^{n}\left(1-H\left(\Phi\left(\mathrm{w}(\mathrm{x}_i,\Delta T)\right)\right)\right)w_{b.i} \tag{17}$$

ΔT is the incremental warp of the shape kernel represented implicitly by level set function Φ. When ΔT tends to **0**, the affine deformation estimation will converge.

In order for the convenience to derive ΔT, we rewrite $H\left(\Phi\left(\mathrm{w}(\mathrm{x},\Delta T)\right)\right)$ and $1-H\left(\Phi\left(\mathrm{w}(\mathrm{x},\Delta T)\right)\right)$ as $\left(\sqrt{H\left(\Phi\left(\mathrm{w}(\mathrm{x},\Delta T)\right)\right)}\right)^2$ and $\left(\sqrt{1-H\left(\Phi\left(\mathrm{w}(\mathrm{x},\Delta T)\right)\right)}\right)^2$, respectively [9]. The Taylor expression around ΔT leads to

$$\left(\sqrt{H\left(\Phi\left(\mathrm{w}(\mathrm{x},\Delta T)\right)\right)}\right)^2\approx\left[\sqrt{H(\Phi)}+\frac{1}{2\sqrt{H\left(\Phi(\mathrm{x})\right)}}\mathrm{J}\Delta T\right]^2 \tag{18}$$

$$\left(\sqrt{1-H\left(\Phi\left(\mathrm{w}(\mathrm{x},\Delta T)\right)\right)}\right)^2\approx\left[\sqrt{1-H(\Phi)}+\frac{1}{2\sqrt{H\left(\Phi(\mathrm{x})\right)}}(\text{-J})\Delta T\right]^2 \tag{19}$$

where

$$\mathrm{J}=\frac{\partial H}{\partial\Phi}\frac{\partial\Phi}{\partial\mathrm{x}}\frac{\partial W}{\partial\Delta T}=\delta\left(\Phi(\mathrm{x})\right)\nabla\left(\Phi(\mathrm{x})\right)\frac{\partial W}{\partial\Delta T} \tag{20}$$

$$\nabla\left(\Phi\left(\mathrm{x}\right)\right)=\left[\Phi_x\left(\mathrm{x}\right),\Phi_y\left(\mathrm{x}\right)\right] \tag{21}$$

$$\frac{\partial W\left(\mathrm{x},\Delta T\right)}{\partial \Delta T}=\begin{bmatrix} x & 0 & y & 0 & 1 & 0 \\ 0 & x & 0 & y & 0 & 1 \end{bmatrix} \tag{22}$$

By differentiating Eq. (17) with respect to ΔT, we have

$$\Delta T=-\left[\sum_{i=1}^{n}\left(\frac{1}{A_f}\frac{w_{f,i}}{2H\left(\Phi\left(\mathrm{x}_i\right)\right)}+\frac{1}{A_b}\frac{\lambda w_{b,i}}{2\left(1-H\left(\Phi\left(\mathrm{x}_i\right)\right)\right)}\right)\mathrm{J}_i^{\,T}\mathrm{J}_i\right]^{-1}\times\sum_{i=1}^{n}\left(\frac{1}{A_f}w_{f,i}-\frac{1}{A_b}\lambda w_{b,i}\right)\mathrm{J}_i^{\,T} \tag{23}$$

The level set function Φ can then be updated by using Eq. (16). Note that the denominator $H\left(\Phi\left(\mathrm{x}_i\right)\right)$ and $1-H\left(\Phi\left(\mathrm{x}_i\right)\right)$ in Eq. (23) will never be zero with the used Heaviside function $H\left(x\right)=\frac{1}{2}\left(1+\frac{2}{\pi}\arctan\left(\frac{x}{\varepsilon}\right)\right)$ and each x_i corresponds to a different J_i according to Eq. (20).

This registration step by solving an affine transformation problem can be viewed as a template based tracking process. It iteratively estimates the shape change until convergence. For the majority of target shape change types, the affine transformation can usually describe them well.

3.2.2 Segmentation. The affine transformation tracker proposed in Section 3.2.1 can estimate the non-rigid motion better than traditional template based trackers, such as mean shift tracker [16] and the EM-shift tracker [18], but it cannot extract the contour of the target accurately. Therefore we propose to refine the registration result by using a novel segmentation based tracking procedure.

By viewing Eq. (13) as the function of $\Phi\left(\mathrm{x}_i\right)$, we optimize it with respect to $\Phi\left(\mathrm{x}_i\right)$ by calculus of variations [31]. The first variation of the functional is

$$\frac{\partial E\left(\Phi\left(\mathrm{x}_i\right)\right)}{\partial \Phi\left(\mathrm{x}_i\right)}=\frac{1}{2}\delta\left(\Phi\left(\mathrm{x}_i\right)\right)\left(\frac{1}{A_f}w_{f,i}-\lambda\frac{1}{A_b}w_{b,i}\right) \tag{24}$$

where $\delta\left(\Phi\left(\mathrm{x}_i\right)\right)=\varepsilon\Big/\left(\pi\left(\Phi\left(\mathrm{x}_i\right)^2+\varepsilon^2\right)\right)$ is the derivative of the smoothed Heaviside step function, i.e., a smoothed Dirac delta function. $\delta\left(\Phi\left(\mathrm{x}_i\right)\right)$ acts on all level curves, and it tends to compute a global minimization [25].

We seek $\frac{\partial E\left(\Phi\left(\mathrm{x}_i\right)\right)}{\partial\Phi\left(\mathrm{x}_i\right)}=0$ by carrying out the steepest-ascent method using the following gradient flow:

$$\frac{\partial E\left(\Phi\left(\mathrm{x}_i\right),t\right)}{\partial t}=\frac{\partial E\left(\Phi\left(\mathrm{x}_i\right)\right)}{\partial\Phi\left(\mathrm{x}_i\right)} \tag{25}$$

Eq. (24) is actually a segmentation based active contour tracker, where the sign of $\frac{1}{A_f}w_{f,i}-\lambda\frac{1}{A_b}w_{b,i}$ determines if the pixel $\mathrm{x_i}$ belongs to foreground or background. Because the contour obtained by the registration step is very close to the true edge of the target, we evolve Eq. (24) by using the result from the registration step as the initial curve.

3.2.3 Target update. In dynamic scenes, the illumination and viewpoint might change and the object might be occluded, and thus the foreground distribution and background distribution of the target often change gradually. Since our method can estimate the shape deformation accurately, the target model updating becomes easy and this can prevent effectively the target from drifting. Our updating method is very simple:

$$\begin{cases}\mathrm{q\_update}=\alpha\cdot\mathrm{q}+\left(1\text{-}\alpha\right)\cdot\mathrm{p_t}\\ \mathrm{o\_update}=\beta\cdot\mathrm{o}+\left(1\text{-}\beta\right)\cdot\mathrm{v_t}\end{cases} \tag{26}$$

where $\mathrm{p_t}$ and $\mathrm{v_t}$ are respectively the foreground and background model at time t. In our experiments, we set $\alpha,\beta\in[0.7,0.95]$ by experience.

4. Implementation

In this section, we present the numerical implementation of the proposed algorithm in detail. Let h be the step length and obviously h=1 in image grid. Let (x_i, y_i) be the spatial coordinate corresponding to pixel x_i. For convenience, we represent $\Phi(\mathrm{x}_i)$ as $\Phi(x_i, y_i)$ in another form. $\nabla(\Phi(\mathrm{x}_i))$ in Eq. (20), i.e., $\nabla\Phi(x_i, y_i)$, is approximated by the center difference scheme as follows:

$$\nabla\Phi(x_i, y_i) = \left[\frac{1}{2}\left(\Phi(x_i+1, y_i) - \Phi(x_i-1, y_i)\right);\ \ \frac{1}{2}\left(\Phi(x_i, y_i+1) - \Phi(x_i, y_i-1)\right)\right] \quad (27)$$

For the registration step, we first compute J in Eq. (20) to estimate the affine transformation vector ΔT in Eq. (23), where $\Delta T = (p_1, p_2, p_3, p_4, p_5, p_6)'$ is a 6×1 vector. In Eq. (20), $\delta(\Phi(\mathrm{x}))$ is a scalar, $\nabla(\Phi(\mathrm{x}_i))$ is 1×2 vector, and $\frac{\partial W}{\partial \Delta T}$ is a 2×6 matrix. So J_i which corresponds to each x_i is a 1×6 vector and $\mathrm{J}_i^T\mathrm{J}_i$ is a 6×6 matrix. Thus we can get ΔT by Eq. (23) after computing $\mathrm{J}_i^T\mathrm{J}_i$ and J_i^T. In practice, we can only use those pixels around zero level set to estimate ΔT, and this can speed up the estimation of ΔT.

For segmentation, Eq. (25) can be implemented by using a numerical scheme on a discrete grid. Let Δt be the time step. Then we compute Φ^{l+1} by the following discretization and linearization of Eq. (25),

$$\frac{\Phi^{l+1}(x_i, y_i) - \Phi^{l}(x_i, y_i)}{\Delta t} = \frac{1}{2}\delta\left(\Phi^{l}(x_i, y_i)\right)\left(\frac{1}{A_f^l}w_{f,i}^l - \lambda\frac{1}{A_b^l}w_{b,i}^l\right) \quad (28)$$

Eq. (28) can be rewritten as

$$\Phi^{l+1}(x_i, y_i) = \Phi^{l}(x_i, y_i) + \frac{1}{2}\Delta t\delta\left(\Phi^{l}(x_i, y_i)\right)\left(\frac{1}{A_f^l}w_{f,i}^l - \lambda\frac{1}{A_b^l}w_{b,i}^l\right) \quad (29)$$

The convergence of Eq. (29) is guaranteed by

$\Delta t \le 1 \Big/ \max_{i=1,\cdots,n} \left| \frac{1}{2}\delta\left(\Phi^{1}\left(x_i, y_i\right)\right)\left(\frac{1}{A_f^l} w_{f,i}^l - \lambda \frac{1}{A_b^l} w_{b,i}^l \right) \right|$ according to the Courant-Friedrichs-Lewy step-size restriction [32]. The level set is represented by a signed distance function and its re-initialization can be solved efficiently by using the method proposed in [33].

In general, the procedures of JRACS can be summarized as follows.

i) Calculate the foreground model **q** and background model **o**.

ii) Initialize the position Φ_0 of the candidate region in the current frame.

iii) Calculate the foreground candidate $p(\Phi_0)$ and background candidate $v(\Phi_0)$.

iv) Initialize the iteration number $k = 0$ for registration.

v) Calculate ΔT by using Eq. (23).

vi) Let $e \leftarrow \|\Delta T\|$, $k \leftarrow k+1$. Set the error threshold ε and the maximum iteration number *N*. If $e < \varepsilon$ and $k < N$, then update Φ_0 by using (16) and go to step iii. Otherwise, registration converges and then executes segmentation.

vii) Refine the registration result by using Eq. (25).

viii) Update the foreground model and background model by using Eq. (26).

5. Experimental results and discussions

Since the proposed JRACS is inspired by both template based tracking methods and segmentation based tracking methods, we evaluate it in comparison with the combination flow algorithm [12], which is a representative and state-of-the-art segmentation-based tracking method, and the EM-shift algorithm[2] [18], which improves the template based mean shift tracker by estimating iteratively the scale and orientation changes of the target. In addition, the recently developed SOAMST algorithm [19], which is robust to scale and orientation changes of the target, and the tracking method developed by Chiverton *et al.* [21],

[2] We thank Dr. Zivkovic for sharing the code in [35].

which consists of a bootstrap stage and an adaptive shape memory based active contour stage, are also employed in the experiment. Note that the results of Chiveton *et al.*'s method were obtained by using only the active tracking part without the bootstrap stage[3] for a fair comparison with the other methods which use a non-automatic (manual) initialization.

In the JRACS method, there are several parameters to set. First, d (Φ>-d) is used to select the size of background region. In our experiments, the size of background region is set approximately that of the foreground region. Second, λ is used to balance the foreground matching score and background matching score. In the experiments, we set $\lambda = A_b / A_f$, and then $\frac{1}{A_f} w_{f,i} - \frac{1}{A_b} \lambda w_{b,i}$ can be rewritten as $\frac{1}{A_f}\left(w_{f,i} - w_{b,i}\right)$, in which $w_{f,i} - w_{b,i}$ will guide the registration and segmentation and $1/A_f$ is a normalization constant. $\lambda > A_b / A_f$ implies that some foreground pixels will be considered as the background. Third, a big ε in the Heaviside function can speed up the convergence of registration and segmentation process. We select $\varepsilon \in [1, 5]$ and set the foreground and background distribution update parameters α and β in Eq. (25) as 0.9 in our experiments.

We first use an example to illustrate the registration and the segmentation stages of the proposed JRACS method. Then we present the experimental results on five real video sequences. The algorithms are implemented in the environment of MATLAB 7.10 and run on a PC with Intel Core i7 CPU (2.93 GHZ) and 8GB RAM. The RGB color model is selected as the feature space to represent the target. The videos of all the tracking results in the following experiments can be downloaded at http://www4.comp.polyu.edu.hk/~cslzhang/JRACS.htm.

5.1 An example of the proposed JRACS method

In this section, we demonstrate the proposed JRACS method by an example of hand, which

[3] We thank Dr. Chiverton for providing the experimental results of their tracking algorithm.

shows obvious non-rigid motion. We first demonstrate the registration performance of JRACS, i.e., estimating the affine transformation of the hand, in Figure 3. Due to the high flexibility of human hand, it is not accurate to represent the hand shape by using an ellipse template [18, 19], while the level set can be used to accurately represent the hand shape. Figure 3 (b) shows the initial position of level set in a certain frame and it can be clearly seen that the target has large non-rigid deformation. Figures 3 (c)-(f) illustrate the registration process of JRACS. It can be seen that although there is a significant change of scale and orientation, the JRACS can still estimate the hand shape well. The combination flow method [12] uses 180 iterations (about 1.95s) to convergence, while JRACS uses only 8 iterations to converge (about 0.13s) since it matches the template quickly by estimating the affine transformation of the hand.

It can also be seen from Figure 3 that although the registration process in JRACS can estimate the general deformation of hand, the boundary is not very accurate. Therefore we further perform segmentation to refine the registration result for a more accurate hand shape contour. The segmentation process is illustrated in Figure 4.

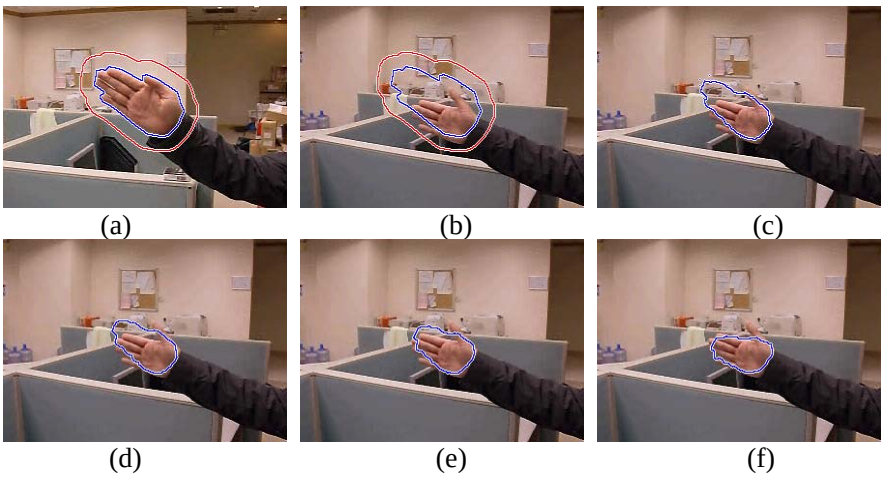


Figure 3: The registration process by the proposed JRACS method. (a) The tracked target (including foreground and background). (b) Initial level set location in the current frame. (c)-(f) The registration process and the final results.

(a) (b) (c)

Figure 4: The segmentation process in the proposed JRACS method.

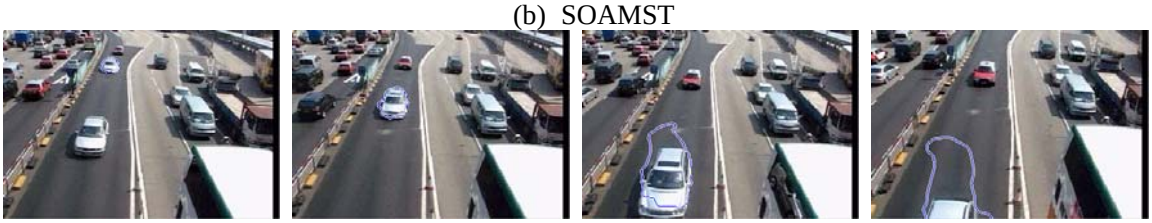
5.2 Experimental results on real video sequences

We then use several real video sequences to validate the proposed JRACS method. Because Chiverton *et al.*'s method [21] fails to track the sequences of car, head and outdoor face, we only show its tracking results on the hand and fish sequences in the following figures.

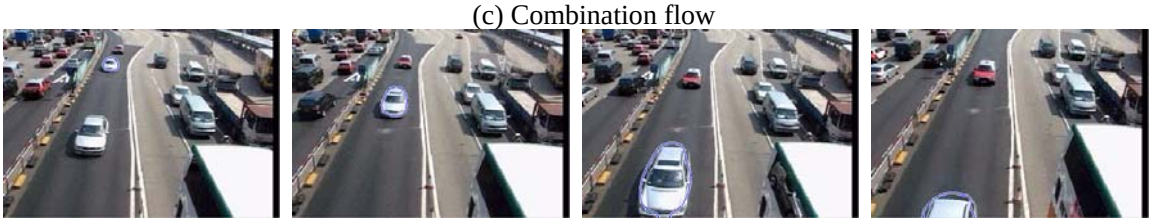
The first one is a car sequence, whose scale grows gradually. In this sequence, some features of the car present in the background might disturb the estimation of scale and orientation. As can be seen in Figure 5(d), the JRACS tracks the target over the whole sequence with good scale estimation, while the combination flow algorithm (refer to Figure 5(c)) does not perform well because the main features of background are similar to some features of the car, which disturbs the evolution of the level set. The EM-shift (Figure 5(a)) and SOAMST (Figure 5(b)) methods, which are template based trackers, cannot capture the true contour of the car.

(a) EM-shift

(b) SOAMST



(c) Combination flow



(d) JRACS

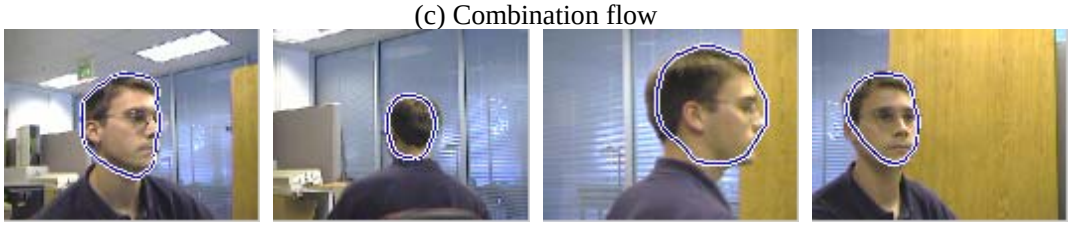
Figure 5: Tracking results on the car sequence with large scale changes by the competing methods.

The second sequence was the one used in [36]. The target is an indoor face that moves quickly. In this sequence, the face shows some complex changes of viewpoint, background and pose. Figure 6 illustrates the tracking results. The skin color of neck (which is part of the background) is close to that of the face. Because the combination flow method considers the dissimilarity between the background and target, the neck of the man is wrongly regarded as a part of the target, leading to inaccurate tracking results. On the other hand, the proposed JRACS method simultaneously matches foreground and background of the target, and it performs much better in tracking the target face. Because of the big pose change of the face, the EM-shift method and SOAMST method do not perform well.

(a) EM-shift

(b) SOAMST

(c) Combination flow



(d) JRACS

Figure 6: Tracking results on the indoor face sequence with obvious viewpoint and background changes by the competing methods.

In the third sequence, our goal is to track a moving hand with large non-rigid deformation (which was used in Section 5.1). In this hand sequence, the fast stretching and clenching actions and the disturbance of some background features raise many difficulties to estimate the contour changes of hand. In the proposed JRACS method, the registration step estimates rigid deformation of the target, and then the segmentation step makes the contour of the target complete. The tracking results in Figure 7 show that JRACS performs much better than the combination flow method on this sequence. We can see that EM-shift and SOAMST are hard to handle the complex shape change of the target, while the active contour based tracking method in [21] does not perform well.

(a) EM-shift

(b) SOAMST

(c) Combination flow

(d) Chiverton et al.'s method

(e) JRACS

Figure 7: Tracking results on the hand sequence with obvious non-rigid changes. (Note that the result of Chiveton *et al.*'s method was obtained by the active tracking part only.)

The fourth video is an outdoor face sequence which has obvious viewpoint and illumination changes and occlusion. Figure 8 shows the experimental results by the four tracking methods. Because of the illumination and viewpoint changes, segmentation based combination flow algorithm fails to track after the 60^{th} frame. The template based trackers such as EM-Shift and SOAMST perform better than combination flow. For the proposed JRACS method, the registration step of it estimates the non-rigid deformation of the object by affine transformation and overcomes effectively the effect of the illumination and viewpoint change, and then the segmentation step further optimizes the object area by active contour.

Note that in Figure 8 we only show the selected results from the first 60 frames for the combination flow method because it fails to track after the 60th frame. For EM-Shift, SOAMST and JRACS method, we show the representative experimental results selected from all the 380 frames.

The last experiment is on a fish sequence, where the motion and shape of the fish are very irregular, and there are similar objects appearing around the target object. The experimental results in Figure 9 show that our method performs well even when a similar fish is presented near the target fish. However, the combination flow method cannot handle it well because it considers another fish as the desired target as well. Meanwhile, the active contour stage of the tracking method in [21] does not estimate the contour change of the target well.

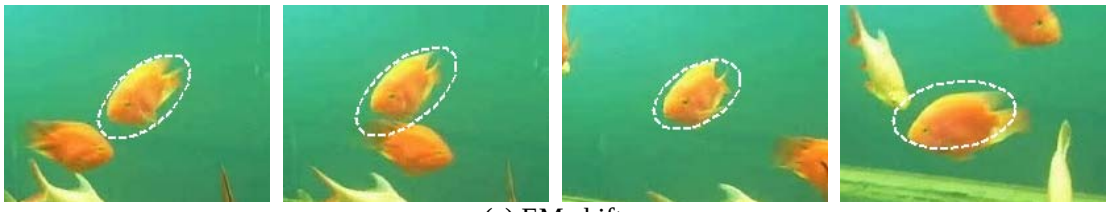
(a) EM-shift

(b) SOAMST

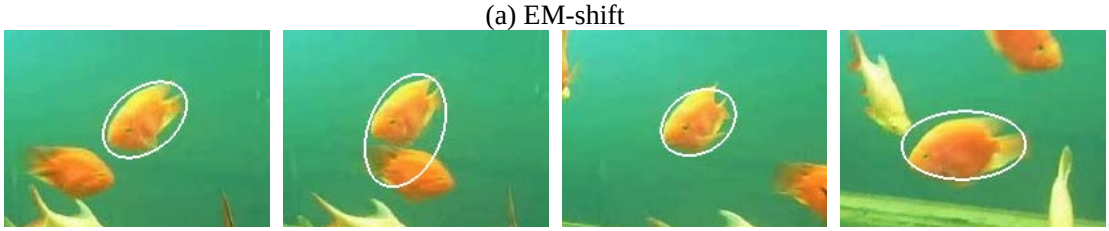
(c) Combination flow

(d) JRACS

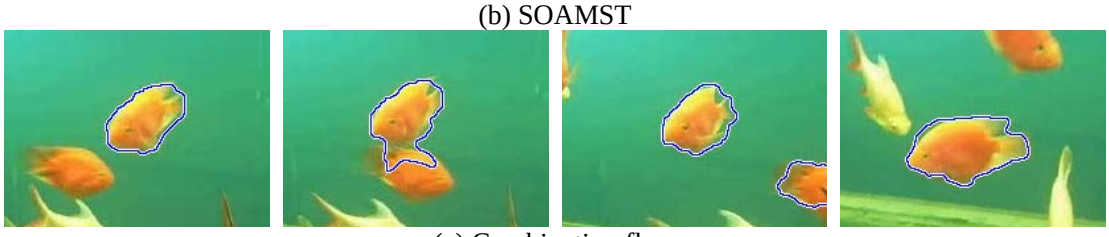
Figure 8: Tracking results on the outdoor face sequence with illumination and viewpoint changes and occlusion.



(a) EM-shift



(b) SOAMST



(c) Combination flow

(d) Chiverton et al.'s method

(e) JRACS

Figure 9: Tracking results on fish sequence with obvious non-rigid changes and similar objects. (Note that the result of Chiveton *et al*.'s method was obtained by the active tracking part only.)

In order to compare quantitatively the proposed JRACS method with the other four methods, we manually labeled the ground truth contours of the target objects in the five videos, and evaluate the tracking performance by applying the overlap index (OI) [34]:

$$\mathrm{OI} = \frac{\left|\mathrm{A_G} \cap \mathrm{A_M}\right|}{\left|\mathrm{A_G} \cup \mathrm{A_M}\right|} \tag{30}$$

where $\mathrm{A_G}$ represents the ground truth area of the interesting object and $\mathrm{A_M}$ represents the area of tracking outputs. A big OI generally implies that the tracking method has good localization

accuracies. Table 1 lists the OI values of the five tracking methods on the five video sequences. Because the tracking method in [21] fails to track the car and two face sequences, we only show its target localization accuracies on the hand and fish sequences. A similar case arises for the combination flow method to handle the outdoor face sequence. The proposed JRACS method achieves the highest OI among the five tracking methods.

Table 1: The target localization accuracies for the five tracking methods according to OI. (Note that the result of Chiveton *et al.*'s method was obtained by the active tracking part only.)

Method	Car	Face (indoor)	Hand	Fish	Face (outdoor)
EM-shift [18]	57%	45%	55%	63%	25%
SOAMST [19]	60%	40%	66%	70%	48%
Combination flow [12]	64%	66%	81%	71%	-
Chiverton et al.'s method [21]	-	-	19%	50%	-
JRACS	75%	67%	82%	82%	69%

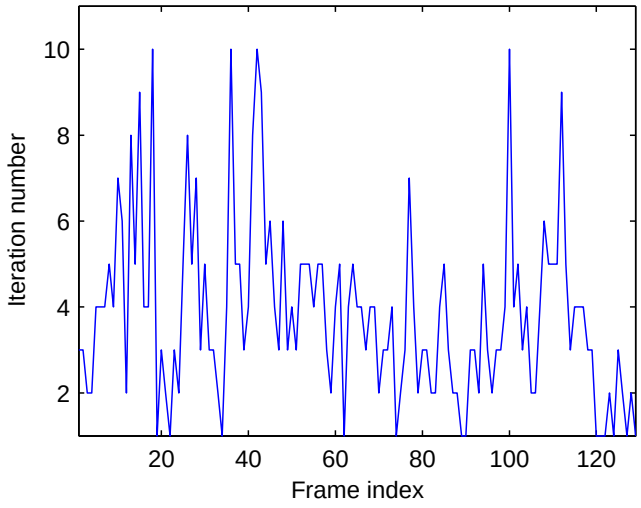


Figure 10: The iteration number by the proposed JRACS on the indoor face sequence in registration stage.

The proposed JRACS algorithm has a registration stage and a segmentation stage. By many experiments, we found that the registration stage needs 3~5 iterations in average to converge for small deformation, but more iterations are necessary for severe deformation. Figure 10 plots the number of iterations of the proposed algorithm on the face sequence for each frame. The average number of iteration is 3.8. After estimating the affine transformation

of the target in the registration step, the segmentation step furthermore refines the result of registration in order to better approximate the true shape of the target. Theoretically, the more iterations used in the segmentation stage, the more accurate result can be obtained, but it may consume more computational time. According to our experiments, 5~15 iterations are appropriate. For the combination flow method, it will require about 35 iterations in average because segmentation-based tracking methods often require more iterations than template matching based tracking methods.

Table 2: Average speed for the four tracking methods (frames/per second)

Method	Car	Face (indoor)	Hand	Fish	Face (outdoor)
Combination flow [12]	7	4	2	3	-
EM-shift [18]	49	31	21	25	22
SOAMST [19]	57	38	30	29	31
JRACS	32	22	15	19	17

In MATLAB environment, the tracking speed of JRACS is faster than foreground flowing and the combining flowing methods reported in [11, 12]. Certainly, JRACS is slower than EM-shift and SOAMST because the time complexity of estimating affine transformation (in JRACS) is higher than that of estimating covariance matrix (in EM-Shift and SOAMST). Table 2 compares the average tracking speed for the combination flow, EM-shift, SOAMST and JRASC methods[4].

The proposed JRACS method combines the advantages of the segmentation based tracker and the template based tracker. The registration step estimates adaptively the target shape change by using affine transformation based on level set method. Because the registration result is close to the true contour of the object, the segmentation step can easily optimize it and get accurate object contour. Actually, the work by Chiverton *et al* [21] shares this merit with our work.

[4] Note that the speed of Chiverton et al.'s method [21] is not listed here because the results of this algorithm were run on a different PC and software system.

6. Conclusions

This paper presented a novel tracking framework with joint registration and active contour segmentation (JRACS). The tracked target was implicitly represented by using a level set, which can handle seamlessly the topological changes of the target. The goal is to find a candidate region, whose foreground distribution and background distribution can best match the template foreground and background based on the Bhattacharyya similarity. A joint registration and segmentation scheme was developed, which first estimates the rigid deformation of the object and then refines the registration result.

The advantages of JRACS come from the two key weights, which indicate the possibility of a pixel in the candidate region belonging to foreground or background, and guide the evolution process of registration and segmentation. The good performance of JRACS was demonstrated by representative testing videos where the targets show large scale non-rigid shape changes. Experimental results validated that JRACS overcomes some limitations of previous works, including EM-shift tracker, SOAMST tracker and the combination flow tracker, and JRACS can be considered as an extension of them. In the future, we will consider how to extend JRACS by integrating the spatial information into the target representation.

References

[1] A. Yilmaz, O. Javed and M. Shah, “Object Tracking: a Survey,” ACM Computing Surveys, vol. 38, no. 4, Article 13, 2016.

[2] M. Isard, and A. Blake, “CONDENSATION - conditional density propagation for visual tracking,” Int. J. Computer Vision, vol. 29, no.1, pp. 5-28, 1998.

[3] P. Chockalingam, N. Pradeep, and S. Birchfield, “Adaptive fragments-based tracking of non-rigid objects using level sets,” In Proceedings of IEEE International Conference on Computer Vision, 2009.

[4] Zulfiqar Hasan Khan, Irene Yu-Hua Gu, and Andrew G. Backhouse, “Robust Visual Object Tracking Using Multi-Mode Anisotropic Mean Shift and Particle Filters”, IEEE Trans. Circuits and Systems for Video Technology, vol. 21, no. 1, 74-87, 2011.

[5] N. M. Artner, A. Ion, and W. G. Kropatsch, “Multi-scale 2D tracking of articulated objects using hierarchical spring systems,” Pattern Recognition, vol. 44, no. 4, pp. 800-810, 2011.

[6] N. Paragios, R. Deriche, “Geodesic active contours and level sets for the detection and tracking of moving objects,” IEEE Trans. Pattern Anal. Machine Intell., vol. 22, no.3, pp. 266–280, 2000.

[7] Qiang Chen, Quan-Sen Sun, Pheng Ann Heng, and De-Shen Xia. “Two-Stage Object Tracking Method Based on Kernel and Active Contour.” IEEE Trans. Circuits and Systems for Video Technology, vol. 20, no. 4, 605-609, 2010.

[8] A. Yilmaz, “Object Tracking by Asymmetric Kernel Mean Shift with Automatic Scale and Orientation Selection,” In Proc. IEEE Conf. on Computer Vision and pattern Recognition, Minnesota, USA, Vol. I, pp.1-6, 2017.

[9] C. Bibby and I. Reid. “Robust Real-Time Visual Tracking using Pixel-Wise Posteriors,” In Proc. European Conf. on Computer Vision, part II, pp. 831-844, 2008.

[10] X. Sun, H. Yao and S. Zhang. “A novel supervised level set method for non-rigid object tracking,” In Proceedings of IEEE International Conference on Computer Vision, pp. 3393 – 3400, 2011.

[11] D. Freedman and T. Zhang, “Active Contours for Tracking Distributions,” IEEE Trans. Image Processing, vol. 13, no. 4, pp. 518-526, 2014.

[12] T. Zhang, D. Freedman, “Improving performance of distribution tracking through background matching,” IEEE Trans. Pattern Anal. Machine Intell., vol. 27, no. 2, pp. 282–287, 2015.

[13] D. Cremers, “Dynamical statistical shape priors for level set based tracking,” IEEE Trans. Pattern Anal. Machine Intell., vol. 28, no. 8, pp. 1262–1273, 2006.

[14] D. Cremers. M. Rousson, R. Deriche, “A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape,” Int’l Journal of Computer Vision, vol. vol. 72, no. 2, pp. 195–215, 2017.

[15] M. Roh, T. Kim, J. Park, and S. Lee, “Accurate object contour tracking based on boundary edge selection,” Pattern Recognition, vol. 40, no. 3, pp. 931-941, 2007.

[16] D. Comaniciu, V. Ramesh and P. Meer, “Kernel-Based Object Tracking,” IEEE Trans. Pattern Anal. Machine Intell., vol.25, no. 5, pp. 564-577, May, 2003.

[17] C. Yang C, D. Ramani, and L. Davis, “Efficient Mean-Shift Tracking via a New Similarity Measure,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, San Diego, CA, 2005, vol. 1, pp.176-183.

[18] Z. Zivkovic and B. Kröse, “An EM-like Algorithm for Color-Histogram-Based Object Tracking,” In Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Washington, D.C., USA, vol. I, pp. 798-803, 2004.

[19] J. Ning, L. Zhang, D. Zhang, and C. Wu, “Scale and Orientation Adaptive Mean Shift Tracking,” to appear in IET Computer Vision, 2011. http://www4.comp.polyu.edu.hk/~cslzhang/papers.htm

[20] T. Riklin-Raviv, N. Kiryati and N. Sochen. “Unlevel-Set: Geometry and Prior-based Segmentation,” In Proc. European Conf. on Computer Vision. pp.50--61, 2004.

[21] J. Chiverton, X. Xie, and M. Mirmehdi. “Automatic Bootstrapping and Tracking of Object Contours. IEEE Trans. on Image Processing,” vol. 21, no. 3, pp, 1231-1245, 2012.

[22] T. Kailath, “The divergence and bhattacharyya distance measures in signal selection,” IEEE Trans. Communication Technology, vol. 15, no. 1, pp. 52-60, 1967.

[23] O. Michailovich, Y. Rathi, and A. Tannenbaum, “Image Segmentation Using Active Contours Driven by the Bhattacharyya Gradient Flow,” IEEE Trans. Image Processing. vol. 16, no. 11, pp. 2787-2801, 2007.

[24] F. Goudail, P. Refregier, and G. Delyon, “Bhattacharyya distance as a contrast parameter for statistical processing of noisy optical images,” J. Opt. Soc. Am. A, vol. 21, no. 7, pp. 1231–1240, July 2004.

[25] T. F. Chan and L. A. Vese, “Active Contours without Edges,” IEEE Trans. Image Processing, vol. 10, no. 2, pp. 266-277, 2001.

[26] Grigorios Tsagkatakis and Andreas Savakis, “Online Distance Metric Learning for Object Tracking” IEEE Trans. Circuits and Systems for Video Technology, vol. 21, no. 12, 1810-1821, 2011.

[27] B. Babenko, B., M. Yang, and S. Belongie, “Tracking with Online Multiple Instance Learning,” IEEE Trans. Pattern Anal. Machine Intell., vol.33, no. 8, pp. 1619 - 1632, 2011.
[28] Ying-Jia Yeh and Chiou-Ting Hsu, “Online Selection of Tracking Features Using AdaBoost”, IEEE Trans. Circuits and Systems for Video Technology, vol. 19, no. 3, 442-446, 2009.
[29] T. M. Cover and J. A. Thomas, “Elements of information theory,” New York: Wiley, 1991.
[30] S. Baker, and I. Matthews, “Lukas-kanade 20 years on: A unifying framework,” Int’l Journal of Computer Vision, vol. 69, no. 3, pp. 221–255, 2004.
[31] G. Aubert, K. Pierre, Mathematical problems in image processing: partial differential equations and the calculus of variations. Springer, 2002.
[32] W. F. Ames, Numerical Methods for Partial Differential Equations, 3rded. New York: Academic, 1992.
[33] K. Zhang, L. Zhang, H. Song and W. Zhou., “Active contours with selective local or global segmentation: a new formulation and level set method., Image and Vision Computing,” vol.28, issue 4, pp.668-676, April 2010
[34] G. H. Rosenfield and K. Fitzpatrick Lins, “A coefficient of agreement as a measure of thematic classification accuracy,” Photogramm. Eng. Remote Sens., vol. 52, no. 2, pp. 223-227, 1986.
[35] Z. Zivkovic, EM-shift code, http://staff.science.uva.nl/~zivkovic/PUBLICATIONS.html
[36] www.ces.clemson.edu/~stb/research/headtracker/seq/