Scale and Orientation Adaptive Mean Shift Tracking

Jifeng Ning, Lei Zhang[1], David Zhang and Chengke Wu

Abstract – A scale and orientation adaptive mean shift tracking (SOAMST) algorithm is proposed in this paper to address the problem of how to estimate the scale and orientation changes of the target under the mean shift tracking framework. In the original mean shift tracking algorithm, the position of the target can be well estimated, while the scale and orientation changes can not be adaptively estimated. Considering that the weight image derived from the target model and the candidate model can represent the possibility that a pixel belongs to the target, we show that the original mean shift tracking algorithm can be derived using the zeroth and the first order moments of the weight image. With the zeroth order moment and the Bhattacharyya coefficient between the target model and candidate model, a simple and effective method is proposed to estimate the scale of target. Then an approach, which utilizes the estimated area and the second order center moment, is proposed to adaptively estimate the width, height and orientation changes of the target. Extensive experiments are performed to testify the proposed method and validate its robustness to the scale and orientation changes of the target.

Keywords: object tracking, mean shift, moment, scale and orientation estimation

[1] Corresponding author. Lei Zhang is with the Biometrics Research Center, Dept. of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China. Email: cslzhang@comp.polyu.edu.hk. This work is supported by the National Science Foundation Council of China under Grants 60532060, 60775020 and 61003151 and the Chinese University Scientific Fund under Grant No.QN2009091.
Jifeng Ning is with the College of Information Engineering, Northwest A&F University, Yangling, China, and the Biometrics Research Center, Dept. of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China and the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, China. Email: jf_ning@sina.com.
David Zhang is with the Biometrics Research Center, Dept. of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China. Email: csdzhang@comp.polyu.edu.hk.
Chengke Wu is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, China. Email: ckwu@xidian.edu.cn.

1. Introduction

Real-time object tracking is a critical task in computer vision, and many algorithms have been proposed to overcome the difficulties arising from noise, occlusions, clutters, and changes in the foreground object and/or background environment [14]. Among various tracking methods, the mean shift tracking algorithm is a popular one due to its simplicity and efficiency. The mean shift algorithm was originally developed by Fukunaga and Hostetler [2] for data analysis, and later Cheng [3] introduced it to the field of computer vision. Bradski [6] modified it and developed the Continuously Adaptive Mean Shift (CAMSHIFT) algorithm for face tracking. Comaniciu and Meer successfully applied mean shift algorithm to image segmentation [8] and object tracking [7, 9]. Some optimal properties of mean shift were discussed in [13, 15].

In the classical mean shift tracking algorithm [9], the estimation of scale and orientation changes of the target is not solved. Although it is not robust, the CAMSHIFT algorithm [6], as the earliest mean shift based tracking scheme, could actually deal with various types of movements of the object. In CAMSHIFT, the moment of the weight image determined by the target model was used to estimate the scale (also called area) and orientation of the object being tracked. Based on Comaniciu *et al*'s work in [9], many tracking schemes [10, 11, 17, 18, 23] were proposed to solve the problem of target scale and/or orientation estimation. Collins [10] adopted Lindeberg *et al*'s scale space theory [19, 20] for kernel scale selection in mean-shift based blob tracking. However, it cannot handle the rotation changes of the target. An EM-shift algorithm was proposed by Zivkovic and Kröse in [11], which simultaneously estimates the position of the local mode and the covariance matrix that can approximately describe the shape of the local mode. In [23], a distance transform based asymmetric kernel is used to fit the object shape through a scale adaptation followed by a segmentation process. Hu

et al [17] developed a scheme to estimate the scale and orientation changes of the object by using spatial-color features and a novel similarity measure function [12, 16].

In this paper, a scale and orientation adaptive mean shift tracking (SOAMST) algorithm is presented under the mean shift framework. Unlike CAMSHIFT, which uses the weight image determined by the target model, the proposed SOAMST algorithm employs the weight image derived from the target model and the target candidate model in the target candidate region to estimate the target scale and orientation. Such a weight image can be regarded as the density distribution function of the object in the target candidate region, and the weight value of each pixel represents the possibility that it belongs to the target. Using this density distribution function, we can compute the moment features and then estimate effectively the width, height and orientation of the object based on the zeroth order moment, the second order center moment and the Bhattacharyya coefficient between target model and target candidate model. The experimental results demonstrate that SOAMST can deal with various movements of the tracked object flexibly and robustly.

The rest of the paper is organized as follows. Section 2 introduces the classical mean shift algorithm. Section 3 analyzes the moment features of the target candidate region and then describes in detail the proposed SOAMST approach. Section 4 performs extensive experiments to test the proposed SOAMST algorithm in comparison with state-of-the-art schemes. Section 5 concludes the paper.

2. Mean Shift Tracking Algorithm

2.1 Target Representation

In object tracking, a target is usually defined as a rectangle or an ellipsoidal region in the image. Currently, a widely used target representation is the color histogram because of its independence of scaling and rotation and its robustness to partial occlusions [9, 21]. Denote

by $\left\{\mathrm{x}_i^*\right\}_{i=1\cdots n}$ the normalized pixels in the target region, which is supposed to be centered at the origin point and have n pixels. The probability of the feature u (u=1, 2,…, m) in the target model is computed as [9]

$$\begin{cases}\hat{\mathrm{q}}=\left\{\hat{q}_u\right\}_{u=1\cdots m} \\ \hat{q}_u = C\sum_{i=1}^{n} k\left(\left\|\mathrm{x}_i^*\right\|^2\right)\delta\left[b\left(\mathrm{x}_i^*\right)-u\right]\end{cases} \tag{1}$$

where $\hat{\mathrm{q}}$ is the target model, $\hat{q}_u$ is the probability of the u^{th} element of $\hat{\mathrm{q}}$, δ is the Kronecker delta function, $b\left(\mathrm{x}_{\mathrm{i}}^*\right)$ associates the pixel x_i^* to the histogram bin, and $k(x)$ is an isotropic kernel profile. Constant C is a normalization function defined by

$$C = 1\Big/\sum\nolimits_{i=1}^{n} k\left(\left\|\mathrm{x}_i^*\right\|^2\right) \tag{2}$$

Similarly, the probability of the feature u in the target candidate model from the candidate region centered at position y is given by

$$\begin{cases}\hat{\mathrm{p}}(\mathrm{y})=\left\{\hat{p}_u(\mathrm{y})\right\}_{u=1\cdots m} \\ \hat{p}_{\mathrm{u}}(\mathrm{y}) = C_h\sum_{i=1}^{n_h} k\left(\left\|\frac{\mathrm{y}-\mathrm{x}_i}{h}\right\|^2\right)\delta\left[b\left(\mathrm{x}_i\right)-u\right]\end{cases} \tag{3}$$

$$C_h = 1\Big/\sum_{i=1}^{n_h} k\left(\left\|\frac{\mathrm{y}-\mathrm{x}_i}{h}\right\|^2\right) \tag{4}$$

where $\hat{\mathrm{p}}(\mathrm{y})$ is the target candidate model, $\hat{p}_u(\mathrm{y})$ is the probability of the u^{th} element of $\hat{\mathrm{p}}(\mathrm{y})$, $\left\{\mathrm{x}_i\right\}_{i=1\cdots n_h}$ are pixels in the target candidate region centered at y, h is the bandwidth and C_h is the normalization function which is independent of y [9].

In order to calculate the likelihood of the target model and the candidate model, a metric based on the Bhattacharyya coefficient [1] is defined by using the two normalized histograms $\hat{\mathrm{p}}(\mathrm{y})$ and $\hat{\mathrm{q}}$ as follows

$$\rho[\hat{\mathrm{p}}(\mathrm{y}),\hat{\mathrm{q}}]=\sum_{u=1}^{m}\sqrt{\hat{p}_u(\mathrm{y})\hat{q}_u} \tag{5}$$

The distance between $\hat{\mathrm{p}}(\mathrm{y})$ and $\hat{\mathrm{q}}$ is then defined as

$$d[\hat{\mathrm{p}}(\mathrm{y}),\hat{\mathrm{q}}]=\sqrt{1-\rho[\hat{\mathrm{p}}(\mathrm{y}),\hat{\mathrm{q}}]} \tag{6}$$

2.2 Mean Shift

Minimizing the distance $d\left[\hat{\mathrm{p}}(\mathrm{y}),\hat{\mathrm{q}}\right]$ in Eq. (6) is equivalent to maximizing the Bhattacharyya coefficient $\rho\left[\hat{\mathrm{p}}(\mathrm{y}),\hat{\mathrm{q}}\right]$ in Eq. (5). The optimization process is an iterative process and is initialized with the target position, denoted by y_0, in the previous frame. By using the Taylor expansion around $\hat{p}_u(\mathrm{y}_0)$, the linear approximation of the Bhattacharyya coefficient $\rho\left[\hat{\mathrm{p}}(\mathrm{y}),\hat{\mathrm{q}}\right]$ in Eq. (5) can be obtained as:

$$\rho[\hat{\mathrm{p}}(\mathrm{y}),\hat{\mathrm{q}}]\approx\frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(\mathrm{y}_0)\hat{q}_u}+\frac{C_h}{2}\sum_{u=1}^{n_h}w_i k\left(\left\|\frac{\mathrm{y}-\mathrm{x}_i}{h}\right\|^2\right) \tag{7}$$

where

$$w_i=\sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathrm{y}_0)}}\delta[b(\mathrm{x}_i)-u] \tag{8}$$

Since the first term in Eq. (7) is independent of y, to minimize the distance in Eq. (6) is to maximize the second term in Eq. (7). In the mean shift iteration, the estimated target moves from y to a new position y_1, which is defined as

$$\mathrm{y}_1=\frac{\sum_{i=1}^{n_h}\mathrm{x}_i w_i g\left(\left\|\frac{\mathrm{y}-\mathrm{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h}w_i g\left(\left\|\frac{\mathrm{y}-\mathrm{x}_i}{h}\right\|^2\right)} \tag{9}$$

When we choose the kernel $k(x)$ with the Epanechnikov profile, there is $g(x)=-k(x)=1$, and Eq. (9) can be reduced to [9]

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i} \tag{10}$$

By using Eq. (10), the mean shift tracking algorithm finds in the new frame the most similar region to the object.

From Eq. (10) it can be observed that the key parameters in the mean shift tracking algorithm are the weights w_i. In this paper we will focus on the analysis of w_i, with which the scale and orientation of the tracked target can be well estimated, and then a scale and orientation adaptive mean shift tracking algorithm can be developed.

3. Scale and Orientation Adaptive Mean Shift Tracking

In this section, we first analyze how to calculate adaptively the scale and orientation of the target in sub-sections 3.1 ~ 3.5, then in sub-section 3.6, a scale and orientation adaptive mean shift tracking (SOAMST) algorithm is presented.

The enlarging or shrinking of the target is usually a gradual process in consecutive frames. Thus we can assume that the scale change of the target is smooth and this assumption holds reasonably well in most video sequences. If the scale of the target changes abruptly in adjacent frames, no general tracking algorithm can track it effectively. With this assumption, we can make a small modification of the original mean shift tracking algorithm. Suppose that we have estimated the area of the target (the area estimation will be discussed in sub-section 3.2) in the previous frame, in the current frame we let the window size or the area of the target candidate region be a little bigger than the estimated area of the target. Therefore, no matter how the scale and orientation of the target change, it should be still in this bigger target candidate region in the current frame. Now the problem turns to how to estimate the real area and orientation from the target candidate region.

3.1 The Weight Images in Target Scale Changing

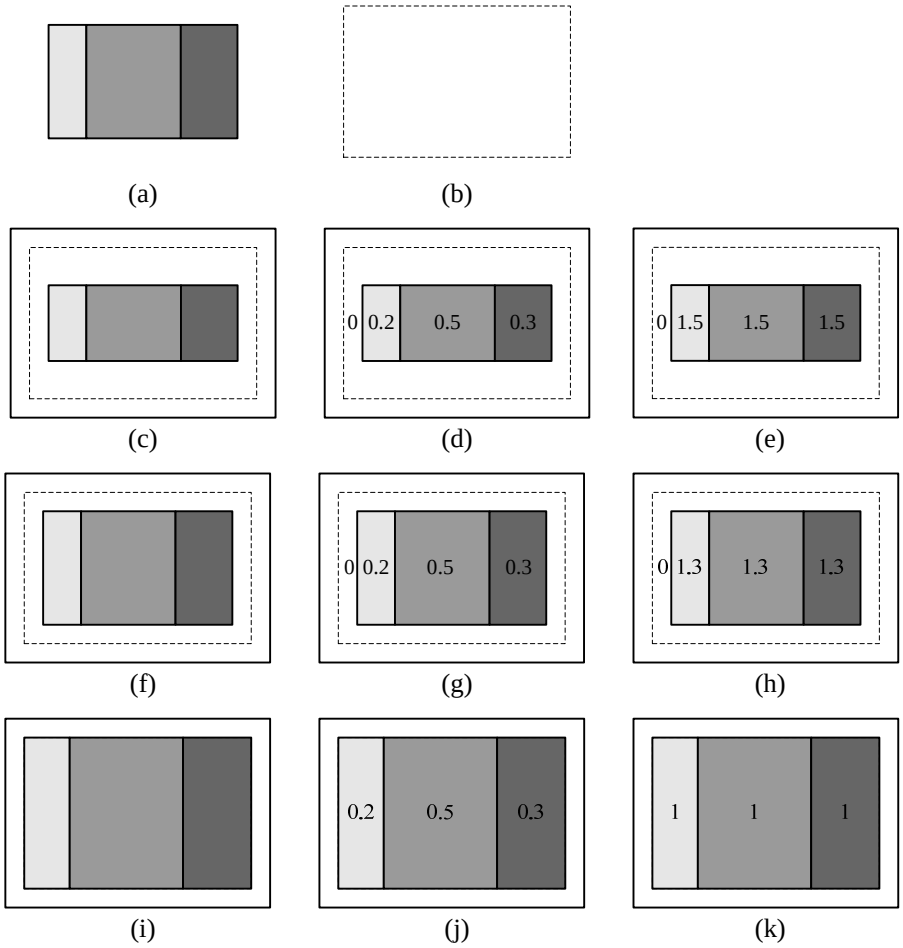


Fig. 1: Weight images in CAMSHIF [6] and mean shift tracking [9] algorithms when the object scale changes. (a) A synthesized target with three gray levels. (b) A target candidate window that is bigger than the target. (c), (f) and (i) are the target candidate regions enclosed by the target candidate window (dashed box) when the scale of the target decreases, keeps invariant and increases, respectively. (d), (g) and (j) are respectively the weight images of the target candidate regions in (c), (f) and (i) calculated by CAMSHIFT. (e), (h) and (k) are respectively the weight images of the target candidate regions in (c), (f) and (i) calculated by mean shift tracking.

In the CAMSHIFT and the mean shift tracking algorithms, the estimation of the target location is actually obtained by using a weight image [10, 24]. In CAMSHIFT, the weight image is determined using a hue-based object histogram where the weight of a pixel is the probability of its hue in the object model. While in the mean shift tracking algorithm, the weight image is defined by Eq. (8) where the weight of a pixel is the square root of the ratio of its color probability in the target model to its color probability in the target candidate model. Moreover, it is not accurate to use the weight image by CAMSHIFT to estimate the location

of the target, and the mean shift tracking algorithm can have better estimation results. That is to say, the weight image in the mean shift tracking algorithm is more reliable than that in the CAMSHIFT algorithm.

As in the CAMSHIFT algorithm, in the SOAMST scheme to be developed, the scale and orientation of the target will be estimated by using the moment features [4-6] of the weight image. Since those moment features depend only on the weight image, a properly calculated weight image could lead to accurate moment features and consequently good estimates of the target changes. Therefore, let's analyze the weight images in the CAMSHIFT and mean shift tracking methods in order for the development of the SOAMST algorithm.

As mentioned at the beginning of Section 3, we will track the target in a larger candidate region than its size to ensure that the target will be within this candidate region when the tracking process ends. With this strategy, let's compare the weight images in CAMSHIFT and mean shift tracking under different scale changes by using the following experiments. Figure 1-(a) shows a synthesized target that has three gray levels. Figure 1-(b) shows the candidate region that is a little bigger than the target. Figures 1-(c), (f) and (i) are the tracking results when the scale of the synthesized target decreases, keeps invariant and increases, respectively. Figures 1-(d), (g) and (j) illustrate the weight images calculated by the CAMSHIFT algorithm in the three cases, while Figures 1-(e), (h) and (k) illustrate the weight images calculated by the mean shift tracking algorithm in the three cases.

From Figure 1, we can see clearly the difference of the weight images between CAMSHIFT and mean shift tracking. First, the weight image in the CAMSHIFT algorithm is constant and it only depends on the target model, while the weight image in the mean shift tracking algorithms will change dynamically with the scale changes of the target. Second, the weight image is closely related to the target scale change in mean shift tracking. The closer the real scale of the target is to the candidate region, the better the weight image approaches to

1. That is to say, the weight image in mean shift tracking can be a good indicator of the scale change of the target. However, the weight image in CAMSHIFT does not reflect this.

Based on the above observation and analysis, we could consider the weight image in the mean shift tracking algorithm as a density distribution function of the target, where the weight value of a pixel reflects the possibility that it belongs to the target. In the following sections, we can see that the scale and orientation of the target can be well estimated by using this density distribution function together with the moment features of the weight image.

3.2 Estimating the Target Area

Since the weight value of a pixel in the target candidate region represents the probability that it belongs to the target, the sum of the weights of all pixels, i.e., the zeroth order moment, can be considered as the weighted area of the target in the target candidate region:

$$M_{00} = \sum_{i=1}^{n} w(\mathrm{x}_i) \tag{11}$$

In mean shift tracking, the target is usually in the big target candidate region. Due to the existence of the background features in the target candidate region, the probability of the target features is less than that in the target model. So Eq. (8) will enlarge the weights of target pixels and suppress the weight of background pixels. Thus, the pixels from the target will contribute more to target area estimation, while the pixels from the background will contribute less. This can be clearly seen in Figures 1-(e), 1-(h) and 1-(k).

On the other hand, the Bhattacharyya coefficient[2] (referring to Eq. (5)) is an indicator of the similarity between the target model $\hat{\mathrm{q}}$ and the target candidate model $\hat{\mathrm{p}}(\mathrm{y})$. A smaller Bhattacharyya coefficient means that there are more features from the background and fewer features from the target in the target candidate region, vice versa. If we take M_{00} as the

[2] In the remaining of the paper, for the convenience of expression we will only use "Bhattacharyya coefficient" to represent the "Bhattacharyya coefficient between the target model and the target candidate model".

estimation of the target area, then according to Eq. (11), when the weights from the target become bigger, the estimation error by taking M_{00} as the area of the target will be bigger, vice versa. Therefore, the Bhattacharyya coefficient is a good indicator of how reliable it is by taking M_{00} as the target area. Table 1 lists the real area of the target in Figure 1 and the estimation error by taking M_{00} as the target area. We can see that with the increase of the Bhattacharyya coefficient, the estimation accuracy by taking M_{00} as the target area will also increase (e.g., the estimation error will decrease).

Based on the above analysis, we see that the Bhattacharyya coefficient can be used to adjust M_{00} in estimating the target area, denoted by A. We propose the following equation to estimate it:

$$A = c(\rho)M_{00} \tag{12}$$

where $c(\rho)$ is a monotonically increasing function with respect to the Bhattacharyya coefficient ρ ($0 \le \rho \le 1$). As can be seen in Figures 1-(e), 1-(h) and 1-(k) and Table 1, M_{00} is always greater than the real target area and it will monotonically approach to the real target area with ρ increasing. Thus we require that $c(\rho)$ should be monotonically increase and reach maximum 1 when ρ is 1. Such a correction function $c(\rho)$ is possible to shrink M_{00} back to the real target scale. There can be alternative candidate functions of $c(\rho)$, such as linear function $c(\rho)=\rho$, Gaussian function, etc. Here we choose the exponential function as $c(\rho)$ based on our experimental experience[3]:

$$c(\rho) = \exp\left(\frac{\rho - 1}{\sigma}\right) \tag{13}$$

From Eqs. (12) and (13) we can see that when ρ approaches to the upper bound 1, i.e., when the target candidate model approaches to the target model, $c(\rho)$ approaches to 1 and in

[3] By our experimental experience, both exponential and Gaussian functions can achieve satisfying results, and we choose the former here for simplicity.

this case it is more reliable to use M_{00} as the estimation of target area. When ρ decreases, i.e., the candidate model is not identical to the target model, M_{00} will be much bigger than the target area but $c(\rho)$ is less than 1 so that A can avoid being biased too much from the real target area. When ρ approaches to 0, i.e., the tracked target gets lost, $c(\rho)$ will be very small so that A is close to zero.

Table 1. The area estimation (pixels) of the target under different scale changes by the proposed method.

Tracking result		Fig. 1 (e)		Fig. 1 (h)		Fig. 1 (k)	
Real area of target		100		150		240	
Background area		140		90		0	
Bhattacharyya coefficient		0.6454		0.7906		1	
Estimated area A under different σ and the relative estimation error (%) in comparison with M_{00}.	M_{00}	150	+50%	195	+30%	240	0%
	σ=1.5	118.42	+18.42%	169.59	+13.06%	240	0%
	σ=1	105.22	+5.22%	158.16	+5.44%	240	0%
	σ=0.8	96.29	-3.71%	150.09	+0.06%	240	0%
	σ=0.5	73.81	-26.19%	128.28	-14.48%	240	0%

Table 1 lists the area estimation results of the target by using Eq. (12) under different scale changes in Figures 1-(e), 1-(h) and 1-(k). Though an optimal value of σ should be adaptive to the video content, by our experimental experiences it was found that when the target model is appropriately defined (containing not too many background features), setting σ between 1 and 2 can achieve very robust tracking results for most of the testing video sequences.

3.3 The Moment Features in Mean Shift Tracking

In this sub-section, we analyze the moment features in mean shift tracking and then combine them with the estimated target area to further estimate the width, height and orientation of the target in the next sub-section. Like in CAMSHIFT, we can easily calculate the moments of the weight image as follows:

$$M_{10} = \sum_{i}^{n_h} w_i \mathrm{x}_{i,1} \qquad M_{01} = \sum_{i=1}^{n_h} w_i \mathrm{x}_{i,2} \tag{14}$$

$$M_{20} = \sum_{i=1}^{n_h} w_i \mathrm{x}_{i,1}^2, M_{02} = \sum_{i=1}^{n_h} w_i \mathrm{x}_{i,2}^2, M_{11} = \sum_{i=1}^{n_h} w_i \mathrm{x}_{i,1} \mathrm{x}_{i,2} \tag{15}$$

where pair ($\mathrm{x}_{i,1}$, $\mathrm{x}_{i,2}$) is the coordinate of pixel *i* in the candidate region.

Comparing Eq. (10) with Eqs. (11) and (14), we can find that y_1 is actually the ratio of the first order moment to the zeroth order moment:

$$\mathrm{y}_1 = \left(\overline{\mathrm{x}}_1, \overline{\mathrm{x}}_2\right) = (M_{10} / M_{00}, M_{01} / M_{00}) \tag{16}$$

where $(\overline{\mathrm{x}}_1, \overline{\mathrm{x}}_2)$ represents the centroid of the target candidate region. The second order center moment could describe the shape and orientation of an object. By using Eqs. (10), (11), (15) and (16), we can convert Eq. (9) to the second order center moment as follows

$$\mu_{20} = M_{20} / M_{00} - \overline{x}_1^2 \qquad \mu_{11} = M_{11} / M_{00} - \overline{x}_1 \overline{x}_2 \qquad \mu_{02} = M_{02} / M_{00} - \overline{x}_2^2 \tag{17}$$

Eq. (17) can be rewritten as the following covariance matrix in order to estimate the width, height and orientation of the target:

$$Cov = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} \tag{18}$$

3.4 Estimating the Width, Height and Orientation of the Target

By using the estimated area (sub-section 3.2) and the moment features (sub-section 3.3), the width, height and orientation of the target can be well estimated. The covariance matrix in Eq. (18) can be decomposed by using the singular value decomposition (SVD) [22] as follows

$$Cov = U \times S \times U^T = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \times \begin{bmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}^{\mathrm{T}} \tag{19}$$

where $U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$ and $S = \begin{bmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{bmatrix}$. λ_1^2 and λ_2^2 are the eigenvalues of *Cov*. The vectors $(u_{11}, u_{21})^{\mathrm{T}}$ and $(u_{12}, u_{22})^{\mathrm{T}}$ represent, respectively, the orientation of the two main axes of the real target in the target candidate region.

Because the weight image is a reliable density distribution function, the orientation estimation of the target provided by matrix U is more reliable than that by CAMSHIFT. Moreover, in the CAMSHIFT algorithm, λ_1 and λ_2 were directly used as the width and height of the target, which is actually improper [4, pp. 12-14]. Next, we present a new scheme to more accurately estimate the width and height of the target.

Suppose that the target is represented by an ellipse, for which the lengths of the semi-major axis and semi-minor axis are denoted by *a* and *b*, respectively. Instead of using λ_1 and λ_2 directly as the width *a* and height *b*, it has been shown [4, pp. 12-14] that the ratio of λ_1 to λ_2 can well approximate the ratio of *a* to *b*, i.e., $\lambda_1/\lambda_2 \approx a/b$. Thus we can set $a = k\lambda_1$ and $b = k\lambda_2$, where *k* is a scale factor. Since we have estimated the target area *A*, there is $\pi ab = \pi(k\lambda_1)(k\lambda_2) = A$. Then it can be easily derived that

$$k = \sqrt{A/(\pi\lambda_1\lambda_2)} \tag{20}$$

$$a = \sqrt{\lambda_1 A/(\pi\lambda_2)} \qquad b = \sqrt{\lambda_2 A/(\pi\lambda_1)} \tag{21}$$

Now the covariance matrix becomes

$$Cov = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \times \begin{bmatrix} a^2 & 0 \\ 0 & b^2 \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}^{\mathrm{T}} \tag{22}$$

The adjustment of covariance matrix *Cov* in Eq. (22) is a key step of the proposed algorithm. It should be noted that the EM-like algorithm by Zivkovic and Kröse [11] estimates iteratively the covariance matrix for each frame based on the mean shift tracking algorithm. Unlike the EM-like algorithm, our algorithm combines the area of target, i.e., *A*, with the covariance matrix to estimate the width, height and orientation of the target.

In Section 4.1, we listed the estimated width, height and orientation of the synthetic ellipse sequence in Figure 1 together with the relative estimation error by using the proposed SOAMST algorithm. It can be seen that the estimation accuracy is very satisfying.

3.5 Determining the Candidate Region in Next Frame

Once the location, scale and orientation of the target are estimated in the current frame, we need to determine the location of the target candidate region in the next frame. With Eq. (22), we define the following covariance matrix to represent the size of the target candidate region in the next frame

$$Cov_2 = U \times \begin{bmatrix} (a+\Delta d)^2 & 0 \\ 0 & (b+\Delta d)^2 \end{bmatrix} \times U^{\mathrm{T}} \tag{23}$$

where Δd is the increment of the target candidate region in the next frame. The position of the initial target candidate region is defined by the following ellipse region

$$(\mathrm{x}-\mathrm{y}_1) \times Cov_2^{-1} \times (\mathrm{x}-\mathrm{y}_1)^{\mathrm{T}} \le 1 \tag{24}$$

3.6 Implementation of the SOAMST Algorithm

Based on the above analyses in sub-sections 3.1 ~ 3.5, the scale and orientation of the target can be estimated and then a scale and orientation adaptive mean shift tracking algorithm, i.e. the SOAMST algorithm, can be developed. The implementation of the whole algorithm is summarized as follows.

Algorithm of Scale and Orientation Adaptive Mean Shift Tracking (SOAMST)

1) Initialization: calculate the target model $\hat{\mathrm{q}}$ and initialize the position y_0 of the target candidate model in the previous frame.
2) Initialize the iteration number $k \leftarrow 0$.
3) Calculate the target candidate model $\hat{\mathrm{p}}(\mathrm{y}_0)$ in the current frame.
4) Calculate the weight vector $\{w_i\}_{i=1\cdots n}$ using Eq. (8).
5) Calculate the new position y_1 of the target candidate model using Eq. (10).
6) Let $\mathrm{d} \leftarrow \|\mathrm{y}_1 - \mathrm{y}_0\|$, $\mathrm{y}_0 \leftarrow \mathrm{y}_1$. Set the error threshold ε (default 0.1) and the maximum Iteration number N (default 15).

If ($d<\varepsilon$ or $k \geq N$)	Stop and go to step 7;
Otherwise	$k \leftarrow k+1$ and go to step 3.

7) Estimate the width, height and orientation from the target candidate model using Eq. (22).

8) Estimate the initial target candidate model for next frame using Eq. (24).

4. Experimental Results

This section evaluates the proposed SOAMST algorithm in comparison with the original mean shift algorithm, i.e., mean shift tracking with a fixed scale, the adaptive scale algorithm [9] and the EM-shift algorithm[4] [11, 25]. The adaptive scale algorithm and the EM-shift algorithm are two representative schemes to address the scale and orientation changes of the targets under the mean shift framework. Because the weight image estimated by CAMSHIFT is not reliable, it is prone to errors in estimating the scale and orientation of the object. So CAMSHIFT is not used in the experiments.

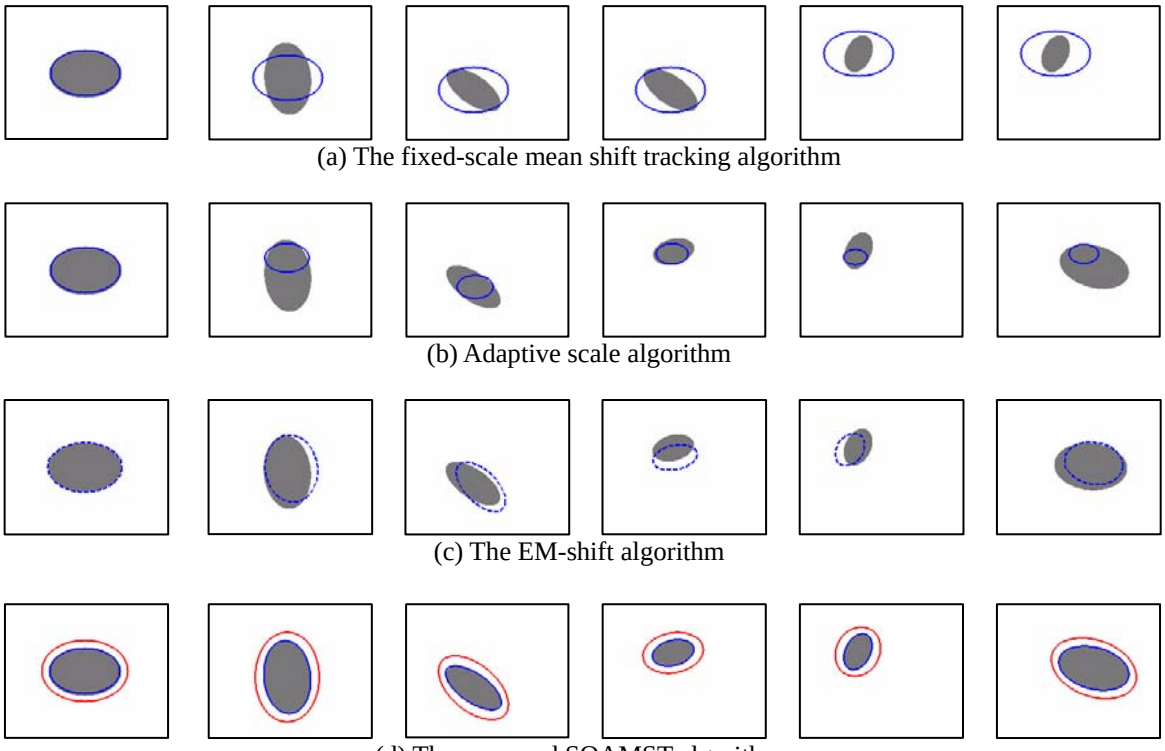
We selected RGB color space as the feature space and it was quantized into 16×16×16 bins for a fair comparison between different algorithms. It should be noted that other color space such as the HSV color space can also be used in SOAMST. One synthetic video sequence and three real video sequences are used in the experiments. The MATLAB source codes and all the experimental results of this paper can be downloaded in the website http://www.comp.polyu.edu.hk/~cslzhang/SOAMST.htm.

4.1 Experiments on a Synthetic Sequence

We first use a synthetic ellipse sequence to verify the efficiency of the proposed SOAMST algorithm. As shown in Figure 2-(d), the window size of the initial target (blue ellipse) is

[4] We thank Dr. Zivkovic for sharing the code in [25].

59×89. We select Δk =10 in the proposed SOAMST algorithm so that the window size of the initial target candidate region (red ellipse in Figure 2-(b)) is 79×109 in frame 1. For other frames in the SOAMST results, the external ellipses represent the target candidate regions, which are used to estimate the real targets, i.e., the inner ellipses. The experimental results show that the proposed SOAMST algorithm could reliably track the ellipse with scale and orientation changes. Meanwhile, the experimental results by the fixed-scale mean shift is not good because of significant scale and orientation changes of the object. The adaptive scale algorithm does not estimate the target orientation change and has bad tracking results. The EM-shift algorithm fails to correctly estimate the scale and orientation of the synthetic ellipse, although the target in this sequence is very simple.



(a) The fixed-scale mean shift tracking algorithm

(b) Adaptive scale algorithm

(c) The EM-shift algorithm

(d) The proposed SOAMST algorithm

Fig. 2: Tracking results of the synthetic ellipse sequence by different tracking algorithms. The red ellipses represent the target candidate region while the blue ellipse represents the estimated target region. The frames 1, 20, 30, 40, 50, 70 are displayed.

Table 2 lists the estimated width, height and orientation of the ellipse in this sequence by

using the SOAMST scheme. The orientation is calculated as the angle between the major axis and x-axis. The first frame of the sequence was used to define the target model and the rest frames were used for testing. It can be seen that the proposed SOAMST method achieves good estimation accuracy of the scale and orientation of the target.

Table 2. The estimation result and accuracy of the width, height and orientation of the ellipse by the proposed SOAMST method.

Frame no.	Semi-major length *a*			Semi-minor length *b*			Orientation		
	Real length	Estimated length	Error (%)	Real length	Estimated length	Error (%)	Real angle	Estimated angle	Error (%)
20	45	46.13	2.51	29	29.81	2.79	95	95.26	0.27
30	39	41.25	5.77	18	18.62	3.44	145	145.03	0.02
40	26	27.03	3.97	16	16.58	3.63	15	14.68	2.13
50	24	24.72	3	16	16.41	2.56	65	63.38	2.49
60	36	37.93	5.36	16	16.57	3.56	115	114.7	0.26
70	44	45.12	2.55	26	26.58	2.23	165	165.01	0.01
Average error over 71 frames			3.50			2.81			1.47

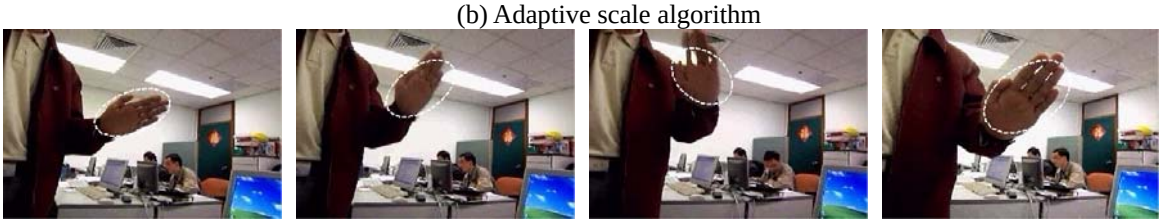
4.2 Experiments on Real Video Sequences

The proposed SOAMST algorithm is then tested by using three real video sequences. The first video is a palm sequence (Figure 3) where the object has clearly scale and orientation changes. Neither the fixed-scale mean shift algorithm nor the adaptive scale algorithm achieves good tracking results. On the other hand, we see that both EM-shift and SOAMST track the palm well in the sequence. However, when the palm is moving fast, such as in frames 27 and 94, the estimated target scale and orientation by EM-shift are not as accurate as those by the SOAMST algorithm.

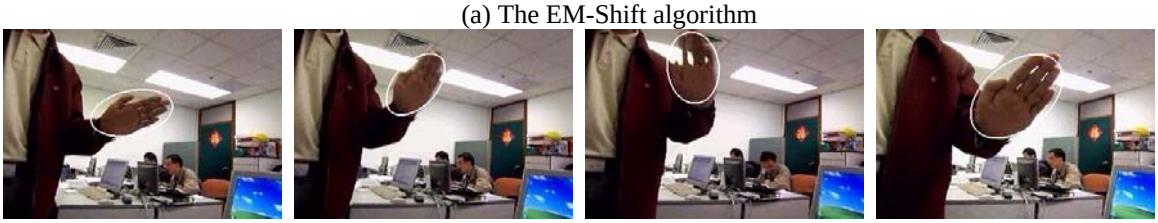
The second video is a car sequence where the scale of the object (a white car) increases gradually as shown in Figure 4. The experimental results show that the proposed SOAMST algorithm estimates more accurately the scale changes than the adaptive scale and the EM-shift algorithms.

(a) The fixed-scale mean shift tracking algorithm

(b) Adaptive scale algorithm



(a) The EM-Shift algorithm



(b) The proposed SOAMST algorithm

Fig. 3: Tracking results of the palm sequence by different tracking algorithms. The frames 10, 27, 94, and 140 are displayed.

(a) Adaptive scale algorithm

(b) The EM-Shift algorithm

(c) The proposed SOAMST algorithm

Fig. 4: Tracking results of the car sequence by different tracking algorithms. The frames 15, 40, 60 and 75 are displayed.

The last experiment is on a more complex sequence of walking man. The object exhibits large scale changes with partial occlusion. To save space we only show the results by EM-shift and SOAMST here. As can be seen in Figure 5, both EM-shift and SOAMST algorithm can track the target over the whole sequence. However, the SOAMST scheme works much better in estimating the scale and orientation of the target, especially when occlusion occurs.

(a) The EM-shift algorithm

(b) The proposed SOAMST algorithm

Fig. 5: Tracking results of the walking man sequence with occlusion by the EM-shift and SOAMST algorithms. The frames 10, 60, 110 and 150 are displayed.

Table 3. The average number of iterations by different methods on the four sequences.

Methods	Fixed-scale mean shift	Adaptive scale	EM-shift	SOAMST
Synthetic ellipse	2.34	13.62	6.27	2.59
Palm sequence	3.92	14.43	6.52	4.28
Car sequence	3.82	11.25	6.34	3.34
Walking-man sequence	3.97	12.84	6.35	3.69

Table 3 lists the average numbers of iterations by different schemes on the four video sequences. The average number of iterations of the proposed SOAMST is approximately equal to that of the original mean shift algorithm with fixed scale. The iteration number of the adaptive scale algorithm is the highest because it runs mean shift algorithm three times. The main factors which affect the convergence speed of the EM-shift and the SOAMST algorithms are the computation of the covariance matrix. EM-shift estimates it in each iteration while

SOAMST only estimates it once for each frame. So SOAMST is faster than EM-shift.

To better evaluate the competing methods, in Table 4 we list the mean localization errors (MLE) and the true area ratios (TAR) by the three trackers on the three real video sequences, palm, car and walking man. The TAR is defined as the ratio of the overlapped area between the tracking result and ground truth to the area of ground truth. The MLE and TAR are closely related to scale and orientation estimation of the target being tracked. Table 4 shows that the proposed SOAMST method achieves the best performance among the three tracking methods.

Table 4. The MLE and TAR values by the competing tracking methods.

Method	Adaptive scale		EM-shift		SOAMST	
	MLE	TAR	MLE	TAR	MLE	TAR
Palm	7.72	64.41%	10.58	86.80%	3.36	96.05%
Car	8.64	72.07%	7.09	95.36%	4.41	92.14%
Walking man	13.40	46.10%	11.63	65.43%	8.57	87.11%

In general, the proposed SOAMST algorithm, which is motivated by the CAMSHIFT algorithm [6], extends the mean shift algorithm when the target has large scale and orientation variations. It inherits the simplicity and effectiveness of the original mean shift algorithm while being adaptive to the scale and orientation changes of the target.

5. Conclusions

By analyzing the moment features of the weight image of the target candidate region and the Bhattacharyya coefficients, we developed a scale and orientation adaptive mean shift tracking (SOAMST) algorithm. It can well solve the problem of how to estimate robustly the scale and orientation changes of the target under the mean shift tracking framework. The weight of a pixel in the candidate region represents its probability of belonging to the target, while the zeroth order moment of the weights image can represent the weighted area of the candidate

region. By using the zeroth order moment and the Bhattacharyya coefficient between the target model and the candidate model, a simple and effective method to estimate the target area was proposed. Then a new approach, which is based on the area of the target and the corrected second order center moments, was proposed to adaptively estimate the width, height and orientation changes of the target. The proposed SOAMST method inherits the merits of mean shift tracking, such as simplicity, efficiency and robustness. Extensive experiments were performed and the results showed that SOAMST can reliably track the objects with scale and orientation changes, which is difficult to achieve by other state-of-the-art schemes. In the future research, we will focus on how to detect and use the true shape of the target, instead of an ellipse or a rectangle model, for a more robust tracking.

References

[1] Kailath T.: 'The Divergence and Bhattacharyya Distance Measures in Signal Selection', IEEE Trans. Communication Technology, 1967, 15, (1), pp. 52-60.

[2] Fukunaga F., Hostetler L. D.: 'The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition', IEEE Trans. on Information Theory, 1975, 21, (1), pp. 32-40.

[3] Cheng Y.: 'Mean Shift, Mode Seeking, and Clustering', IEEE Trans on Pattern Anal. Machine Intell., 1995, 17, (8), pp. 790-799.

[4] Mukundan R., Ramakrishnan K. R.: 'Moment Functions in Image Analysis: Theory and Applications', World Scientific, Singapore, 1996.

[5] Wren C., Azarbayejani A., Darrell T., Pentland A.: 'Pfinder: Real-Time Tracking of the Human Body', IEEE Trans. Pattern Anal. Machine Intell., 1997, 19, (7), pp. 780-785.

[6] Bradski G.: 'Computer Vision Face Tracking for Use in a Perceptual User Interface', Intel Technology Journal, 1998, 2(Q2), pp. 1-15.

[7] Comaniciu D., Ramesh V., Meer P.: 'Real-Time Tracking of Non-Rigid Objects Using Mean

Shift'. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head, SC, June, 2000, vol. 2, pp. 142-149.

[8] Comaniciu D., Meer P.: 'Mean Shift: a Robust Approach toward Feature Space Analysis', IEEE Trans Pattern Anal. Machine Intell., 2002, 24, (5), pp. 603-619.

[9] Comaniciu D., Ramesh V., Meer P.: 'Kernel-Based Object Tracking', IEEE Trans. Pattern Anal. Machine Intell., 2003, 25, (2), pp. 564-577.

[10] Collins R.: 'Mean-Shift Blob Tracking through Scale Space', Proc. IEEE Conf. Computer Vision and Pattern Recognition, Wisconsin, USA, 2003, pp. 234-240.

[11] Zivkovic Z., Kröse B.: 'An EM-like Algorithm for Color-Histogram-Based Object Tracking', Proc. IEEE Conf. Computer Vision and Pattern Recognition, Washington, DC, USA, 2004, vol.1 , pp. 798-803.

[12] Yang C., Ramani D., Davis L.: '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.

[13] Fashing M., Tomasi C.: 'Mean Shift is a Bound Optimization', IEEE Trans. Pattern Anal. Machine Intell., 2005, 27, (3), pp. 471-474.

[14] Yilmaz A., Javed O., Shah M.: 'Object Tracking: a Survey', ACM Computing Surveys, 2006, 38, (4), Article 13.

[15] Carreira-Perpinan M. A. 'Gaussian Mean-Shift is an EM Algorithm', IEEE Trans. Pattern Anal. Machine Intell., 2007, 29, (5), pp. 767-776.

[16] Birchfield S., Rangarajan S.: 'Spatiograms versus histograms for region-based tracking', Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2005, vol. 2, pp. 1158–1163, 2005.

[17] Hu J., Juan C., Wang J.: 'A spatial-color mean-shift object tracking algorithm with scale and orientation estimation', Pattern Recognition Letters, 2008, 29, (16), pp. 2165-2173.

[18] Srikrishnan V., Nagaraj T., Chaudhuri S.: 'Fragment Based Tracking for Scale and Orientation Adaption', Proc. Indian Conf. on In Computer Vision, Graphics & Image Processing, 2008, pp. 328-335.

[19] Linderberg T.: 'Feature Detection with Automatic Scale Selection', International Journal of

Computer Vision. 1998, 30, (2), pp. 79-116.

[20] Bretzner L., Lindeberg T.: ‘Qualitative Multi-Scale Feature Hierarchies for Object Tracking’, Journal of Visual Communication and Image Representation, 2000, 11, (2), pp.115-129.

[21] Nummiaro K., Koller-Meier E., Gool L. V.: ‘An Adaptive Color-Based Particle Filter’, Image and Vision Computing, 2003, 21, (1), pp. 99-110.

[22] Horn R. A., Johnson C. R., Topics in Matrix Analysis, Cambridge University Press, U.K., 1991.

[23] Quast K., Kaup A.: ‘Scale and Shape adaptive Mean Shift Object Tracking in Video Sequences’, Proc. European Signal Processing Conference, Glasgow, Scotland, 2009, pp. 1513-1517.

[24] Collins R.: “Lecture on Mean Shift,” http://www.cse.psu.edu/~rcollins/CSE598G/.

[25] Zivkovic Z.: EM-shift code, http://staff.science.uva.nl/~zivkovic/PUBLICATIONS.html.