International Journal of Image and Graphics
Vol. 13, No. 3 (2013) 1350012 (29 pages)

DOI: 10.1142/S0219467813500125

RECENT ADVANCE ON MEAN SHIFT TRACKING: A SURVEY

LIWEN HE

Shenzhen Graduate School, Harbin Institute of Technology
Shenzhen 518055, P. R. China

YONG XU

Key Laboratory of Network Oriented Intelligent Computation
Bio-Computing Research Center
Shenzhen Graduate School
Harbin Institute of Technology
Shenzhen 518055, P. R. China
laterfall2@yahoo.com.cn

YAN CHEN and JIAJUN WEN

Shenzhen Graduate School, Harbin Institute of Technology
Shenzhen 518055, P. R. China

Received 12 April 2012
Accepted 8 March 2013
Published

Though there have been many applications of object tracking, ranging from surveillance and monitoring to smart rooms, object tracking is always a challenging problem in computer vision over the past decades. Mean Shift-based object tracking has received much attention because it has a great number of advantages over other object tracking algorithms, e.g. real time, robust and easy to implement. In this survey, we first introduce the basic principle of the Mean Shift algorithm and the working procedure using the Mean Shift algorithm to track the object. This paper then describes the defects and potential issues of the traditional Mean Shift algorithm. Finally, we summarize the improvements to the Mean Shift algorithm and some hybrid tracking algorithms that researchers have proposed. The main improvements include scale adaptation, kernel selection, on-line model updating, feature selection and mode optimization, etc.

Keywords: Object tracking; mean shift; challenges in tracking; feature selection; scale adaptation; on-line model updating; hybrid tracking.

1. Introduction

Object tracking aims at detecting interesting moving objects and tracking such objects from frame to frame.[1] Object tracking is a challenging computer vision problem. Object tracking has been widely used in intelligent human–computer interaction (HCI), medical diagnosis, intelligent robot, video surveillance, video encoding, military and other fields.[1] In general, the object tracking methods are divided into three categories, i.e. Filter Theory-based target tracking method, Mean Shift-based target tracking method and Partial Differential Equations-based target tracking method.[2]

In 1975, Fukunaga and Hostetler[3] first used the Mean Shift algorithm to estimate the probability density gradient function. This algorithm was a simple iterative statistical method and it had not been paid enough attention until Cheng emphasized the importance of Mean Shift in 1995.[4] He improved the basic Mean Shift algorithm in two ways. The first way is to define a cluster of kernel functions, which is helpful to resolve the issue that data points cannot converge to a single position in a blurring process. The second way is to introduce a weight coefficient and to extend the Mean Shift algorithm. The extension of Mean Shift algorithm proposed by Cheng[4] can be applied to address more issues. For example, some conventional clustering algorithms can be viewed as special cases of an extension of the Mean Shift algorithm.[4]

Comaniciu and Meer[5] successfully integrated a feature space of a target with the Mean Shift algorithm, and they applied it to image smoothing and image segmentation. Bradski[6] first introduced the Mean Shift algorithm into object tracking field. He used this algorithm to track face in video frame sequences and named the modified algorithm Continuously Adaptive Mean Shift (CAMSHIFT) algorithm. In Ref. 7, Comaniciu *et al.* showed that Mean Shift-based object tracking algorithm had many advantages, such as real time, robustness and easy implementation. They used the nonparametric probability density to model a color distribution and employed a metric derived from the Bhattacharyya coefficient as the cost function to measure the similarity between the target model and the target candidate. The actual location of the target can be fixed by Mean Shift iteration. The Mean Shift converges to an extreme point of the cost function which is usually the actual location of the target. In addition, Comaniciu proved that the Mean Shift algorithm enabled the center point of a target to converge to a stable point.

The conventional Mean Shift algorithm has many desirable properties. It is an efficient and simple adaptive tracking algorithm and its implementation is straightforward. Moreover, the Mean Shift algorithm can work in real time, due to its fast convergence speed. However, the Mean Shift tracking algorithm also suffers from some problems. First, the background influence is a common problem to all tracking algorithms because the target cannot be represented accurately without any background information known. The Mean Shift algorithm has certain adaptability to background changes. That is to say, when the difference between the object and background is not significant, we might also get a good tracking effect at first. However, in a changing background the tracking effect may get worse gradually and the target may "lose". In order to get rid of background influence, a number of improved algorithms have been proposed.[8–11] Second, varying illumination leads to unstable appearances of the target and background, and the strong illumination usually weakens the difference between the tracking object and background. Consequently, the color-based Mean Shift algorithm which uses the histogram of color to represent the target cannot overcome the influence of the illumination change, although the histogram of the color is somewhat robust against the shift, partial occlusion, rotation, changes in size and posture. To update the parameters of the method is a mean to reduce the influence of the illumination changes over time.[12] Moreover, other features such as the spatial distribution are also used to represent the target.[9] Third, it is difficult for the appearance-based tracking algorithm to track an object with occlusion. The Mean Shift algorithm is able to obtain robust tracking result when the occlusion area is small. However, if an object is seriously or completely occluded by another object, its visual appearance would dramatically deviate from its appearance template set.[13] To address this problem, several methods to detect and handle occlusion have been proposed.[13–17] Fourth, object rotation and shape transformation also change the appearance of the target. The Mean Shift algorithm can track the target with a little rotation and shape change, but it fails to track the target that rotates fast and is quite different from the initial target model within a small number of frames. A number of methods such as the method that updates the target model have been designed to resolve this problem.[9,18]

The above problems are mainly caused by complex environment. Besides these problems, the traditional Mean Shift tracking algorithm also has its own drawbacks. First, the initial location of the tracking object should be determined manually or semi-automatically, therefore, the application of the traditional Mean Shift tracking algorithm is limited. Second, the actual tracking object often changes in size and shape, but the bandwidth of the tracking object is fixed since we have selected the tracking window manually in the initial frame. On one hand, the noise of the tracking window will increase when the tracking object diminishes over time. On the other hand, when the size of the tracking object becomes large over time, some features of the tracking object cannot be captured by using the fixed window. In order to overcome the defects of the traditional Mean Shift algorithm mentioned above, many researches proposed a number of improvements to the traditional Mean Shift tracking algorithm.

The purpose of this paper is to summarize and compare mainstream Mean Shift tracking algorithms, and we hope readers can get inspiration from this paper. The remainder of this survey is organized as follows: the Mean Shift object tracking algorithm is introduced in Sec. 2. Then the difficulties of object tracking and possible solutions are showed in Sec. 3. In Sec. 4, we summarize the improvements to the Mean Shift object tracking algorithm. The improvements have one or more goals such as scale adaptation, the use of background information, adaptive model, automatic selection of the object and robust features. Hybrid tracking methods are described in Sec. 5. In the end of this article, we offer a brief conclusion.

2. Mean Shift Tracking

2.1. *Theory of mean shift*

The Mean Shift algorithm is a nonparametric method to find local maxima in a density function represented by a set of samples.[3] Let X be the n-dimensional Euclidean space R^n and S_h be a finite set with sphere radius h. Data points are denoted by $x_i \in X$. The sample mean at $x \in X$ is

$$m(x) = \frac{1}{k} \sum_{x_i \in S_h} x_i, \tag{1}$$

where k stands for the number of data points in the area of S_h.

The difference $M(x) = m(x) - x$ is called Mean Shift in 3. The Mean Shift algorithm aims at iteratively computing $x \leftarrow m(x)$.[3] From the definition of the Mean Shift algorithm, we know that the Mean Shift iterative procedure shifts every data point to the average of data points in its neighborhood.

Cheng extended the basic Mean Shift algorithm and introduced the kernel function and weight coefficient.[4] The extended Mean Shift is defined as follows.

$$M_h(x) = \frac{\sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right) w(x_i) x_i}{\sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right) w(x_i)} - x, \tag{2}$$

where K is the kernel function and w is the weight function. The profile of a kernel K can be defined as a function $k : [0, \infty) \to R$ such that $K(x) = k(\|x\|^2)$.[7] Then the Mean Shift can be written as

$$M_h(x) = \frac{\sum_{i=1}^{n} k(\left\|\frac{x_i - x}{h}\right\|^2) w(x_i) x_i}{\sum_{i=1}^{n} k(\left\|\frac{x_i - x}{h}\right\|^2) w(x_i)} - x. \tag{3}$$

The Mean Shift algorithm iteratively moves the data points toward the Mean Shift until it eventually converges, and the center point is the local maximal of probability density, i.e. extreme point.[4] The Mean Shift procedure can obtain a quadratic bound maximization for all kernels.[19]

2.2. *Basic mean shift tracking*

The Mean Shift algorithm has been successfully applied in many fields and the following briefly introduces its applications in object tracking.

The Mean Shift tracking algorithm consists of two steps, namely, the target representation step and the target location step. In order to describe the characteristics of a target and reduce the computational cost, m-bin histogram feature space is usually used as the estimation of the target probability density.

At first, a region is selected as the tracking target. The Mean Shift algorithm searches the surrounding areas to find a candidate area that is most similar to the target. If the tracking object is truly located in the obtained candidate region, we can track the target successfully.

After a target region that contains the tracking object is selected, we use m to stand for the dimension of its feature space, and denote the feature vector of the target by

$$\mathbf{q} = \{q_u\}_{u=1,\dots,m}, \quad \sum_{u=1}^{m} q_u = 1. \tag{4}$$

In the same way, a candidate region whose center is located in point y is selected. The corresponding feature vector is

$$\mathbf{p}(y) = \{p_u(y)\}_{u=1,\dots,m}, \quad \sum_{u=1}^{m} p_u = 1. \tag{5}$$

Here features of the target are indicated by a color histogram. Let b be a function that maps a pixel to its histogram bin, i.e. $b : R^2 \rightarrow \{1, \dots, m\}$, then the eigenvector q_u in Eq. (4) can be expressed as

$$q_u = C \sum_{i=1}^{n} k \left(\left\| \frac{x_i}{h} \right\|^2 \right) \delta[b(x_i) - u], \tag{6}$$

where n is the number of data points in target region, and C is a normalization constant that makes sum of q_u equal to 1. And $p_u(y)$ in Eq. (5) can be written as

$$p_u(y) = C_h \sum_{i=1}^{n} k \left(\left\| \frac{x_i - y}{h} \right\| \right) \delta[b(x_i) - u], \tag{7}$$

where C_h is the corresponding normalization constant.

The kernel function k ensures that pixels closer to the center are given a higher weight than pixels on the periphery of the object window. h is bandwidth of the object window, which determines the number of pixels that are used in the estimation of the kernel and reflects the size of the target. δ is the Kronecker delta function.

In order to find the most similar candidate, the similarity function between $\mathbf{q}$ and $\mathbf{p}(y)$ is defined as follows.

$$\hat{\rho}(y) \equiv \rho[\mathbf{p}(y), \mathbf{q}]. \tag{8}$$

Several similarity functions have been proposed in literatures, and here we choose Bhattacharyya coefficient as the similarity function. Bhattacharyya coefficient is the cosine of the angle between two vectors. It has many good properties such as it gives better results than the divergence, its relation to Fisher's information, and explicit formulas for a large class of distributions.[20] The most important advantage of Bhattacharyya coefficient is that it allows us to use the Mean Shift algorithm to find the maximum value of the similarity function skillfully. Bhattacharyya coefficient is defined as follows.

$$\hat{\rho}(y) \equiv \rho[\mathbf{p}(y), \mathbf{q}] = \cos\langle \mathbf{p}(y), \mathbf{q} \rangle = \sum_{u=1}^{m} \sqrt{p_u(y) q_u}. \tag{9}$$

The larger the $\hat{\rho}(y)$ is, the more similar the two vectors are.[8]

In visual object tracking, in order to track the target correctly, we should choose the candidate that makes the similarity function $\hat{\rho}(y)$ maximal for every frame. Suppose that the center position of the target region in the previous frame is y_0 and the center point of the candidate region in the current frame is y. Using Taylor expansion to expand Eq. (9) around $p_u(y_0)$, we can obtain the following linear approximation of Bhattacharyya coefficient defined in Eq. (9):

$$\begin{aligned}\hat{\rho}(y) &\approx \hat{\rho}(y_0) + \hat{\rho}'(y_0)(p_u(y) - p_u(y_0)) \\ &= \sum_{u=1}^{m} \sqrt{p_u(y_0)q_u} + \frac{1}{2}\sum_{u=1}^{m}\sqrt{\frac{q_u}{p_u(y_0)}}(p_u(y) - p_u(y_0)) \\ &= \frac{1}{2}\sum_{u=1}^{m} \sqrt{p_u(y_0)q_u} + \frac{1}{2}\sum_{u=1}^{m} p_u(y)\sqrt{\frac{q_u}{p_u(y_0)}}. \end{aligned} \tag{10}$$

Substituting Eq. (7) into Eq. (10), we obtain

$$\hat{\rho}(y) = \frac{1}{2}\sum_{u=1}^{m}\sqrt{p_u(y_0)q_u} + \frac{C_h}{2}\sum_{u=1}^{m} w_i k\left(\left\|\frac{x_i - y}{h}\right\|^2\right), \tag{11}$$

where

$$w_i = \sum_{u=1}^{m} \delta[b(x_i) - u]\sqrt{\frac{q_u}{p_u(y_0)}}. \tag{12}$$

To maximize the similarity function in Eq. (11), we only need to maximize the second term since the first term in Eq. (11) is independent of y. The second term represents the density estimation computed with kernel profile $k(x)$ at y in the current frame and with the data being weighted by w_i. Therefore we can use the Mean Shift algorithm to find position y where similarity function $\hat{\rho}(y)$ reaches the maximum value.

The following context describes the procedure to maximize Bhattacharyya coefficient $\hat{\rho}[p(y_0), q]$. Given the distribution $\{q_u\}_{u=1,\ldots,m}$ of a target model and its initial location y_0 in the previous frame, the procedure of the Mean Shift algorithm is summarized as follows:

Step 1. Initialize the location of the target in the current frame with y_0, then compute the distribution $\{p_u(y_0)\}_{u=1,\ldots,m}$ using Eq. (7), and evaluate $\hat{\rho}[p(y_0), q] = \sum_{u=1}^{m}\sqrt{p_u(y_0)q_u}$.

Step 2. Use Eq. (12) to derive corresponding weights $\{w_i\}_{i=1,\ldots,n}$.

Step 3. Apply the following formula to obtain the new location of the target

$$y_1 = \frac{\sum_{i=1}^{n} k(\left\|\frac{y_0 - x_i}{h}\right\|^2)w_i x_i}{\sum_{i=1}^{n} k(\left\|\frac{y_0 - x_i}{h}\right\|^2)w_i}. \tag{13}$$

Step 4. Exploit Eq. (7) to update $\{p_u(y_1)\}_{u=1,\ldots,m}$ and calculate $\hat{\rho}[p(y_1), q] = \sum_{u=1}^{m} \sqrt{p_u(y_1) q_u}$.

Step 5. Repeatedly calculate $y_1 = \frac{1}{2}(y_0 + y_1)$ till $\hat{\rho}[p(y_1), q] > \hat{\rho}[p(y_0), q]$.

Step 6. If $\|y_1 - y_0\| < \varepsilon$ Stop.
Otherwise set $y_0 \leftarrow y_1$ and go to Step 1.

3. Problems and Possible Solutions

The Mean Shift algorithm is indeed a template matching algorithm. The advantage of the Mean Shift object tracking algorithm over a standard template matching algorithm is the elimination of a brute force search. It usually iterates several times to reach the extreme point. However, there are many factors affecting the Mean Shift tracker, such as occlusion, target rotation, target shape change, clutter and varying environment. All these factors influence the similarity between the candidate and the target. Hence, we discuss the above problems and its solutions below, respectively.

3.1. *Occlusion*

Occlusion can be classified into three categories: self-occlusion, inter-object occlusion and occlusion by the background scene structure.[21] When the target is occluded, its visual features cannot be observed. Therefore, the target cannot be tracked as its features have lost.

A solution to the problem of partial occlusion is to divide the tracking object into a number of parts. When partial occlusion happens, the information of the parts that do not be occluded can be used. Jeyakar *et al.* used a weighted fragment-based approach to handle partial occlusion,[9] but this approach is very time consuming due to the use of too many fragments. Wu *et al.* proposed a dynamic Bayesian network to cope with partial even complete occlusions.[13] Liu *et al.* divided the human body into three parts i.e. head-shoulder, torso and legs, and then used an AdaBoost method to tackle partial occlusion.[22] For complete occlusion, it is possible to use a linear dynamic model or nonlinear dynamic model to model the target motion and keep on predicting the target location till the target reappears in the case of occlusion.[21]

In recent years, a two-step approach has been widely used to deal with the occlusion. The first step detects the occlusion and the second step handles occluded objects. Chen *et al.* defined a threshold T_h to detect whether occlusions happened.[14] If Bhattacharyya coefficient was lower than the threshold T_h, they considered that the tracked object be occluded or disappeared. When the tracking object is not occluded, they use the Mean Shift algorithm to track the target. Otherwise, the Kalman filter is used to estimate the object position. In Ref. 23, both the distance between the targets and the size change of the target were exploited to detect occlusion, and then the author recovered the missing regions of the target by using shape priors consisting of shape level sets during occlusion. Lerdsudwichai *et al.* detected the occlusion using an occlusion grid and used the color distribution of target to achieve occlusion recovery.[24] 15 and 13 established an occlusion detection method based on sub-block detection, which divided the tracking window into the right part and left part and calculated the similarity measure respectively. However, the author did not describe how to model more than two occluded targets. Li *et al.* proposed an improved Mean Shift approach to solve this problem.[16] Occlusion layers were introduced to represent the occlusion relation. The nonocclusion parts of the target are obtained and are used for tracking. During the tracking process, the states of these targets are gradually adjusted one by one to eliminate the effect of occlusion.

3.2. *Rotate and shape change*

In video streams, when an object rotates or changes shape, the appearance of the object changes much. Therefore, the appearance-based object tracking methods such as color-based methods and spatial-based methods may fail. Even though the Mean Shift algorithm has a certain robustness to slight rotation and shape change, it cannot deal with the object tracking with great variation of the pose and shape. The most effective and simple way to overcome the rotation and shape change problem is to update the target model. The model updating method can track the object well when the rotation or shape changes slowly. Hu *et al.* exploited the principal components of the variance matrix to update the orientation of the tracking object. The method can cope with the object deformation problem very well.[25]

Most of time, the shape of a tracking object is asymmetric. Moreover, if the object rotates, its shape changes accordingly. However, the Mean Shift algorithm uses a symmetric kernel. In order to adapt the kernel to the changing shape of the object, an asymmetric kernel that is retrieved from an object mask is used in Ref. 26. A Gaussian Mixture Models (GMM) of the object and a GMM of the surrounding background were also built to segment the object area from the background in Ref. 27 and it can somewhat solve the object rotation problem.

3.3. *Clutter*

As the Mean Shift tracking algorithm usually uses color characteristics of the object as its feature, it is vulnerable to background effects. Therefore, it becomes very difficult to track the target correctly by using the Mean Shift algorithm, especially when the target's color is similar to the background. In summary, background information is very important for the Mean Shift algorithm. There are two main reasons. The first reason is that if some of the target's features are present in the background, their relevance for the localization of the target is diminished. The second reason is that in many applications it is difficult to exactly delineate the target, and its model might contain background features as well. In addition, the improper use of the background information will influence the scale selection algorithm mentioned in Sec. 4, which makes it unreasonable to measure similarity across scales.[8]

In order to reduce the effect of the background features, many researchers used a method named background weighted histogram (BWH) method. Comaniciu *et al.* suggested that background features should be added to the target model in Ref. 8. Let $\{\hat{o}_u\}_{u=1,\ldots,m}(\sum_{u=1}^{m} \hat{o}_u = 1)$ be the histogram of the background in the feature space and $\hat{o}^*$ be its smallest nonzero block. They first extended the region and then collected color features in the background and calculated the background color histogram. Then the weights were defined using the following formula

$$\left\{\nu_u = \min\left(\frac{\hat{o}^*}{\hat{o}_u}, 1\right)\right\}_{u=1,\ldots,m}. \tag{14}$$

The weights are used to calculate the ratio histogram. After that, the target model and the candidate should be modified by adding weight ν_u. Therefore, the more large proportion a color in the background occupy, the lower its weight ν_u is. The transformation diminishes the importance of those prominent features which have low weight ν_u in the background. Jeyakar *et al.* weakened the color feature of the background refer to the method introduced in Ref. 8 and put forward a new BWH method.[9] Using foreground/background to enhance the foreground color feature, the author defined a weight function as Ref. 8. After that, he modified the target model and candidates and implemented the Mean Shift object tracking algorithm. However, Ning proved that BWH in the original Mean Shift tracking algorithm is incorrect in Ref. 28. Moreover, Ning *et al.* proposed a corrected BWH (CBWH), which can truly achieve what the original BWH method wants: reduce the interference of background in target localization.

Babu focused on the discriminative features between the target and the background and proposed a voting strategy to separate the target from its background.[10] Pixels close to the center are regarded as target pixels, whereas a large number of neighboring pixels surrounding the target region are chosen to represent the background. Therefore, the probabilities of ith pixel $h_o(i)$ and $h_b(i)$ which belong to the target and background respectively can be obtained. The resulting log-likelihood ratio of foreground/background region $L_t(i)$ is used to determine target pixels. Then a threshold th_o is set to find the most reliable target pixels. If $L_t(i) > \mathrm{th}_o$, the pixel is regarded as the target pixel, otherwise, the pixel is considered as the background pixel.[10,29] However, it is very difficult to properly set the threshold in practice since the threshold is usually estimated through experience. Haritaoglu *et al.* built a background model by learning method first. Then they updated background model parameters as the environment changes. After that, foreground targets segmented from the background in each frame of the video sequences by four stage processes: thresholding, noise cleaning, morphological filtering, and target detection.[30] This algorithm can generate a set of shape and appearance features for detected target.

Usually, the object region is selected by using a symmetrical window, such as a rectangle or an ellipse, and we assume that the region can represent the object shape. However the symmetrical region always contains several background areas. Therefore, the performance of moving object tracking is dramatically affected when background is complex and changes greatly. Chen used two steps to track moving objects.[11] Firstly, image segmentation is performed for selected object region. Secondly, a new Mean Shift kernel is used to track moving object with segmentation results. This method achieves good performance even background changes dramatically, but it cannot track objects with complex shapes.

If one gets the contours of target, he can well distinguish the target from background. However, it is difficult to obtain the contours of the tracking object. The active contours were ever used to obtain and track the complete object.[31] The active contour models were locked onto nearby edges. It can localize the nearby edges accurately. Then scale-space continuation can be used to enlarge the capture region around features of interest. This method can find edges, lines and subjective contours, but it is a time-consuming algorithm.

3.4. *Environment varying*

Environmental varying includes illumination changes, camera movement, etc. This would lead to false convergence because the optimization becomes difficult. Moreover, the computation time required by the algorithm increased. To resolve the environmental changing problem, the algorithm can be optimized by conducting model updating and adding spatial information. Jeyakar *et al.* used marginal histograms for object tracking,[9] Bhattacharyya coefficients ρ_1, ρ_2 are calculated by using color features and edge features, respectively. Then the location of the target can be obtained by maximizing $\rho_1 \cdot \rho_2$ via the Mean Shift algorithm. There are many other methods to deal with the problem of tracking objects undergoing geometric distortion, changing illumination,[17] but most of them are difficult to implement or time consuming.

In many video streams, moving objects have shadows and the shadows move along with moving objects due to the influence of the high illumination. Shadows can cause serious problems such as merging of objects, distortion of objects' color histogram, shape deformations, false identifications.[32] In order to track object accurately, many methods have been proposed to find shadows. Cucchiara *et al.* used the HSV color space to analyze in Ref. 33 since a shadow cast on a background which does not change significantly its hue. Depending on the distortion of the brightness and difference in the chrominance, Horprasert *et al.* classified a pixel into one of the four categories: foreground, background, shadow, highlight.[34] This algorithm works well on real image sequences of outdoor scenes. However, the limitation of the system is the problem of reflection on highly specular surfaces where the color of a point on such surfaces can change nondeterministically. A two step method is presented in Ref. 35 to remove shadows. First, whether a pixel is a possible shadow pixel or not is determined by evaluating the different components of color variation. Second, the shadow pixels are refined by evaluating their local neighborhood.

The shadow removal mechanism is proved to be effective and adjustable to the different lighting conditions.

4. Improvement of the Mean Shift Algorithm

The Mean Shift algorithm has achieved considerable success in object tracking due to its simplicity and robustness. However, it also has many shortcomings. First, the bandwidth of the tracking window is fixed but the size of the tracking object changes over time. Second, as the region of the tracking object often uses a rectangle or ellipse to represent, it contains much background information. Usually, background information influences the validity of the Mean Shift tracking algorithm. Third, the model of the target is invariable, but it may undergo changes in scale, shape and illumination. Fourth, the kernel function often uses a symmetric kernel, while the shape of the tracking objects is usually asymmetric. There are many other aspects that cause bad tracking effects, such as feature selection, similarity measurement adaptation. Therefore, in this section, we will discuss the recent improvement techniques that are relevant to the Mean Shift object tracking algorithm.

4.1. *Target representation*

In video sequences, we can use many methods to represent objects. Shapes and appearances are the most common way. There are many ways to represent the appearance representations for tracking such as color. The shape representations are employed for tracking, which usually use points, primitive geometric shapes, object silhouette and contour, articulated shape models, skeletal models, etc. Furthermore, we can combine the above two methods into one, such as probability densities of object appearance, templates, active appearance models and multi-view appearance models.[21] In the field of the Mean Shift algorithm, we often use the following ways to represent target.

4.1.1. *Weighted mask*

The traditional Mean Shift tracking algorithm usually uses a rectangular patch or an elliptical patch to represent the tracking object. In this method, each of the pixels in the region has a weight, and the closer it gets away from the center point, the bigger its weight is. This algorithm is a very efficient and simple method to track target, but it is a high time-consuming algorithm.

4.1.2. *Multi patches*

As the method which uses weighted mask to represent the tracking object will lose spatial information, multi patches have been used to represent the target in video sequences. At the first place, the target region should be divided into fragments.

There are a number of ways to divide the target. The patches are divided arbitrarily in Ref. 36, then the author used a voting maps strategy to determine the final position of the tracking object in the current frame. This method can handle partial occlusions or pose changes. However, there are some parameters that have to gain by empirical specified. Maggio and Cavallaro divided the ellipse-based trackers into seven patches,[37] this method increases the robustness of object tracking, but it does not describe a voting strategy and is not resilient to occlusion. For too many fragments, it would increase the processing time for tracking. Then a four overlapping fragments method is used in Ref. 38 to reduce the compute time, and the target localization uses the fragment, which achieves the most similarity, to determine, model update is also employed to overcome illumination changes. But the final position of the target is decided only by one fragment. As it is easy to be affected by the clutter, Wang used Bhattacharyya metric to re-assign the rank of each fragment after each tracking frame, in which the number of fragments is limited in order to reduce the computational cost.[39] This method describes the target better and converges faster.

4.2. *Feature selection*

No feature-based vision system can work well unless good feature can be identified and tracked from frame to frame. Therefore selecting an appropriate feature plays a critical role in tracking and can track more accurately.[40] In general, the most desirable property of a visual feature is its uniqueness, so that the objects can be easily distinguished in the feature space.[21] There are many methods to represent the features of the target, such as color, edges, optical flow and texture.[21] The traditional Mean Shift algorithm uses the color information to represent the feature of the target. Color histograms have been widely used to represent, analyze, and characterize images. They allow significant data reduction, and can be computed efficiently. Moreover, color histograms are robust to noise and local image transformations. In object tracking domain, color histograms are a popular form of target representation, because of their independence from scaling and rotation, and robustness to partial occlusions.[8,37,41,42] Nevertheless the robustness of such model is weakened in challenging tasks due to the lack of spatial information.[37] Therefore, many tracking algorithms use a combination of these features.

As far as color feature is concerned, there are many color spaces which are able to choose from. In order to select the most appropriate color feature, eight color histograms are chosen to describe and evaluate object's feature in Ref. 37, then the Mean Shift tracking algorithm is used to test their effect, respectively. Although some representations based on HSV achieve good results in particular conditions, the average results show that the RGB-based representation outperforms the others. Hence, for a general application with different target classes, we choose RGB. The conventional Mean Shift algorithm uses fixed number of color bins to quantize the RGB color space. The color histogram is generated as the tracked features from the distribution of the target. Such approach may result in unfeasible classification and is sensitive to noisy interference such as lighting changes and quantization errors. Ju *et al.* proposed a fuzzy color histogram for the Mean Shift tracking algorithm.[43] The fuzzy color histogram, which is generated by self-constructing fuzzy cluster, is used to reduce the interference from lighting changes. An adaptive Gaussian mixtures model is described in Ref. 44. As it may contain a large number of void bins, and those void bins limit the capability of representing object color distribution. In order to eliminate the influence of those void bins, Li proposed an adaptive binning color model.[45] In his work, the number of bins can be determined automatically. This method can get better performance than the conventional Mean Shift algorithm. However, when the visible colors of the tracking object change drastically and rapidly during the sequence, the basic Mean Shift tracker based on single color histogram fails to track the target. Leichter *et al.* raised a simple method to use multiple reference histograms for producing a single histogram that is more appropriate for tracking the target first. Then they proposed an extension to the Mean Shift tracker where the convex hull of these histograms is used as the target model.[46] The method is verified very useful in many scenarios where the visible target colors change sharply during the sequence, but it is time consuming.

During the last two decades, two classes of features have been widely considered for tracking and segmentation purposes: color and texture.[23] For tracking algorithm based on color feature, the lack of spatial information is one of its flaws. We cannot separate objects which have the same color histogram, yet we can separate them by spatial information. In this case, the information related to the spatial distribution of the colors is essential for a correct tracking. Moreover, when we need more precise estimation of the target orientation and size, spatial information plays an important role as well. Maggio and Cavallaro divided the ellipse region into multi-part and seven histograms (7MP) are calculated over semi-overlapping regions of the ellipse.[37] Then the Bhattacharyya coefficient is calculated using their mean color histogram. 7MP outperforms the other multi-part representations due to the more complete spatial information is included. This representation maintains the flexibility and robustness to occlusions of the color histogram, and improves the performance of the single-part based tracker. Yilmaz *et al.* fused the color and texture models to produce a semi-parametric statistical model.[23] In this way, pixels are clustered as the target or the background by the "independent opinion polling" strategy. We can observe that the discrimination features are emphasized, and the other features are suppressed. Scale invariant feature transform (SIFT) features are combined with Mean Shift to get a more accurate position in complicated real scenarios.[47] References 48 and 49 integrated color and texture features to make the color based Mean Shift tracking algorithm more robust.

Usually, the traditional Mean Shift tracking algorithm matches and tracks by making use of the rich color information of the color target image, whereas the color information of the gray image is poor. The application of the Mean Shift algorithm is limited since it is difficult to apply the algorithm to gray image. A new

Mean Shift algorithm is proposed in Ref. 50, the author calculated the matching models combined with the image gradient magnitude and direction texture. The image gradient magnitude and direction texture could distinguish the real target from complicated background much better. Orientation histogram of the gradient direction information of the gray image[51] is used to represent target in Ref. 52, this approach can adapt the change of illumination, and tracks low contrast image robustly and in real time.

4.3. *Kernel selection*

Mean Shift is a method of nonparametric probability density estimation, where an isotropic kernel is introduced to assign smaller weights to the pixels farther from the center (using these weights increase the robustness of the density estimation). Note $K(x)$ as a kernel function and $k(x)$ as its profile function, then $K(x) = k(\|x\|^2)$. $k(x)$ is non-negative and nonincreasing, furthermore it is piecewise continuous and $\int_0^\infty k(r)dr < \infty$.[4] A number of kernel functions are available and the commonly used kernel functions are uniform, triangular, biweight, triweight, Epanechnikov, normal, Gaussian and others. The traditional Mean Shift algorithm usually uses symmetric kernels and radically symmetric kernels. The two kinds of kernels are isotropic kernels. Combined with the Epanechnikov kernel and Gaussian kernel, the Mean Shift algorithm can achieve a good tracking effect. Therefore the two kernels become the most widely applied in the traditional Mean Shift algorithm.

The conventional Mean Shift algorithm tracks using isotropic kernels, therefore, it often loses the tracking object when the structure of the tracking object changes over time, especially when it varies fast. Wang *et al.* proposed an anisotropic kernel Mean Shift algorithm, in which the shape, scale, and orientation of the kernels adapt to the changing object structure.[53] In his work, the author replaced a circular kernel with an elliptical kernel. The method obtains a superior behavior on both still images and a short video sequence, but at the same time, it increases the computational complexity. An asymmetric kernel which used a modified form of the implicit level set function is proposed to the Mean Shift tracking algorithm framework in Ref. 54. The proposed method can overcome the scale and orientation changes of the tracked objects. In order to describe the object's shape accurately and eliminate background information inside the object model, an anisotropic asymmetric kernel is introduced into Mean Shift in Ref. 55. It makes the tracking algorithm robust to background clutters by using this kernel. Multiple kernel is also proposed in Refs. 56 and 57.

4.4. *On-line model updating*

As the tracking object moves all the time, it becomes a very crucial issue to update the target model in object tracking. The target model has not been updated since it was selected at the first time in the traditional Mean Shift algorithm. The following candidate model must calculate the Bhattacharyya coefficient with the target model, which is also named reference model. However, when the target suffers from rotation, partial occlusions or change in scale, shape and illumination, the effectiveness of the Mean Shift tracker will become worse and worse. And what's even worse, we may miss the target or track the wrong target. In order to maintain the effectiveness of the Mean Shift tracking, it is essential to update the target model while tracking. We can update the model according to the similarity between the target model and the candidate model. The most simple update strategy is the using of an update rate. First, an update rate α is defined. Assume the current frame is t, then update the model using formula $\mathbf{q}_{-}update = \alpha \cdot \mathbf{q} + (1 - \alpha) \cdot \mathbf{p}_t$, where $\mathbf{q}$ is the target model in frame $t-1$, and $\mathbf{p}_t$ is the model built by the position of the target in frame t.

As the value of the update rate is very difficult to set, Babu *et al.* defined the update rate related to the Bhattacharyya coefficient to update the reference model.[10] This model updating method is a trade-off between adaptation to rapid changes and robustness to changes due to occlusion. We can implement this method easily, it just needs to update the model in the $t + 1$ frame by using the update rate $\exp(-\alpha[1 - \rho(q_t, p)])$, where α is a real positive scalar, which determines the speed of model update. When the similarity between the target model and the candidate model is very high, the update rate becomes larger as well. This model updating method helps the tracker perform well in cluttered background conditions and in the event of appearance changes. In order to avoid unpredictable tracking performance brought by the static template, an on-line histogram updating method is presented in Ref. 18. In the article, the author used a Bayesian inference approaches to generate the template histogram that approximates the observed histogram subject to the manifold constraints imposed by the key appearance histogram. Experiments show that this method performs more robustly and accurately on tracking varying appearances target than the traditional Mean Shift algorithm.

However, there are some negative effects brought by model updating at the same time. The most vital problem is the update speed. If the update is done slowly, targets that change quickly cannot be track faithfully. But on the contrary, if the update speed is very fast, the algorithm may learn the wrong model and the system will track an improper target. For example, when the tracking object is obscured completely or out of the range of lens, the Mean Shift tracker cannot track the target when the target appears again as the model has updated. Therefore, for the condition of low Bhattacharyya coefficient, we do not need to update the model. Jeyakar *et al.* put forward a novel method which used fragment-based tracking and the foreground/background separation to selectively update the model.[9] Because targets generally have an irregular shape and a rectangular or elliptical window is often used to select object, the model contains background colors which creep into the target window. After calculating the number of foreground pixels that the target contains, the model is updated using the current candidate model once the number of foreground pixels over a certain threshold and the Bhattacharyya coefficient is also larger than a given threshold. This scheme was tested and found to faithfully track an object which undergoes severe illumination change where traditional Mean Shift and fragment-based (no adaptation) techniques fail. In term of partial occlusion, Babu *et al.* proposed the same method stated in Part 3 that uses log-likelihood method to extract the background,[10] a threshold th_u is set and if $L_t(i) > \text{th}_u$, update the model, otherwise keep the model unchanged.

4.5. *Weight improvement*

The traditional Mean Shift tracking algorithm uses Eq. (12) to calculate the weights at each pixel. From the formula we know that weights are assigned to histogram bins and pixels belonging to the same bin having the same weights. Therefore, it cannot track an object which has more than one level of intensities. A back projection mechanism is used to enhance the weights of pixels that have a strong discrimination against the background in Ref. 58. This algorithm has better effect when deals with the problem like fast object movement and high background clutter than the traditional Mean Shift tracking algorithm. In order to solve the problem of deformations and partial occlusions, Shi *et al.* raised an adaptive feature-spatial representation (FSR) for the Mean Shift algorithm.[59] When occlusion happens, the blocks which are not occluded will be assigned larger weights. This method achieves an obvious advantage compared with the conventional Mean Shift algorithm in the aspect of weight assigned.

4.6. *Scale adaptation*

According to the Mean Shift object tracking algorithm described above, for a given target model, the location of the target in the current frame maximizes the Bhattacharyya coefficient in the neighborhood of the previous location. The window size of the traditional Mean Shift object tracking algorithm is fixed since it was selected at first. However, the scale of the target often changes over time. So the window size of the target should adapt to the changes accordingly.

A simple scale adaptation algorithm is proposed in Ref. 7, the author modified the radius h of the kernel profile with a fixed parameters Δh(Δh usually uses $\pm 10\%\, h$).[60] h_{prev} denotes the bandwidth in the precious frame. By running the Mean Shift object tracking algorithm three times with bandwidths $h = h_{\text{prev}}$, $h = h_{\text{prev}} + \Delta h$, and $h = h_{\text{prev}} - \Delta h$, respectively, the author calculated the corresponding Bhattacharyya coefficient, and the radius h_{opt} that yields the largest increase in Bhattacharyya coefficient is chosen. The corresponding bandwidth is the tracking object's bandwidth h_{opt} in the current frame. Thus the bandwidth changes along with the scale of the tracking object. However, the algorithm proposed in Ref. 7 does not merge well with the Mean Shift algorithm. The algorithm only use the color histogram, a smaller candidate window may be preferred to give a higher Bhattacharyya coefficient with the target model. In this case, the size of the candidate shrinks gradually by using the above algorithm.[9] In order to avoid over-sensitive scale adaptation, a rate γ is introduced to update the bandwidth h in Ref. 8. The bandwidth associated with the current frame is obtained through the following formula

$$h = \gamma h_{\mathrm{opt}} + (1-\gamma)h_{\mathrm{prev}}. \tag{15}$$

However, this method only reduces the speed of the window shrinking, which cannot change the trend of the window becoming smaller.

Collins adapted Lindeberg's theory of feature scale selection based on local maxima of differential scale-space filters to the problem of selecting kernel scale for the Mean Shift tracker.[61] However, the author chose Epanechnikov kernel as the kernel function, and the derivative of kernel function is a constant. At present, this method is degraded to the method proposed in Ref. 8. Comaniciu *et al.* proposed two kinds of scale adaptation algorithm, they are balloon estimator and sample point estimator.[62] From the experimental results we can find that the performance of the sample point estimator is superior to both fixed bandwidth estimator and balloon estimators. Both of the two methods replace the fixed bandwidth h by a function about bandwidth h, balloon estimators replace bandwidth h by function $h = h(x)$ for each estimation point x, while sample point estimators replace bandwidth h by function $h = h(x_i)$ for each data point x_i. Sample point estimators use Eq. (16) to calculate bandwidth h,

$$h(x_i) = h_0 \left[\frac{\lambda}{f(x_i)}\right]^{1/2}, \tag{16}$$

where h_0 represents a fixed bandwidth, λ is a proportionality constant and $f(x_i)$ is the probability density function at point x_i. Then the Mean Shift algorithm is used to get the location of the object. This algorithm has a certain adaptation for the change of object's scale, but at the same time, it adds the computational complexity in order to compute the probability density function $f(x_i)$ and the bandwidth $h(x_i)$.

A 3D scale space (2 spatial and 1 scale) model is built for the tracking object in Ref. 9. Two kernels are defined, a 2D kernel for the spatial dimensions and a 1D kernel for the scale. A method of selecting the scale of the kernel window is described in Ref. 9, this method uses the information about the spread of pixels inside the target window. The proposed method mainly considers that pixels in the candidate window, which have colors dominate in the target model, have a higher weight, and pixels having colors not present in the target model gain lower weight, hence the weighted deviation ignores them.

$$\alpha = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} x_i w_i, \tag{17}$$

$$\beta = \sqrt{\frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} (x_i - \alpha)^2}. \tag{18}$$

First the author searched the best location of the target spatially, and adjusted the scale depending on the weighted deviation as follows

$$h_{\text{new}} = h_{\text{orig}} \times \frac{\beta}{\beta_{\text{orig}}}. \tag{19}$$

This method can reduce the tracking jitter.

Li *et al.* implemented the Mean Shift procedure via a coarse-to-fine way for global maximum seeking.[63] In his work, the author named this procedure adaptive pyramid Mean Shift, because this method uses the pyramid analysis technique and determines the pyramid level adaptively to achieve convergence with less iteration. The method acquires maximum pyramid level l first, then a new location X^l is obtained using the Mean Shift algorithm. The new location may be a local mode, so the author set $X^l = 2X^l,\ l = l - 1$ and run the Mean Shift algorithm again till $l = 0$. At present, the global mode is gained at point X^0 which is the center point of the tracking object in the current frame. Then the method proposed in Ref. 7 is used with initial position X^0 and scales $h_1 = (1 + a)h, h_2 = h, h_3 = (1 - a)h$ (a is usually very small, for example $a = 0.05$), respectively. The author chose the maximum value as the optimal object point X_{opt} and scale h_{opt}. The adaptive Mean Shift tracker can efficiently and reliably cope with problems of high-speed moving objects, camera motion and vibration during the tracking process.

As the conventional Mean Shift algorithm often uses a rectangle or ellipse box to represent tracking object, their axis are horizontal and vertical. However, the axis of symmetry of the tracking object often has angle with the horizontal and vertical axis. In order to get more information of the target, we let the orientation of the box be defined as the angle between the long axis of bandwidth and the horizontal-axis of the coordinate system.[64–66] Ning *et al.* proposed a scale and orientation adaptive Mean Shift tracking (SOAMST) algorithm in Ref. 67 to address the problem of how to estimate the scale and orientation changes of the target under the Mean Shift tracking framework. He used an ellipse with an angle between the horizontal-axis to represent the target. The target scale and orientation is estimated by employing the weight image that derived from the target model and the target candidate model in the target candidate region. By using this method, we can reduce the interference of background and track the object with high accuracy, but at the same time, it increases the complexity of the Mean Shift algorithm.

4.7. *Optimization function*

In the conventional Mean Shift tracking algorithm, Bhattacharyya coefficient is used as the similarity function between the target model and the candidate model. Then the maximum value of the similarity function is obtained by Mean Shift iteration. Therefore the similarity function is also called optimization function. However, using Bhattacharyya coefficient as the optimization function will cause drift problem in some cases. For instance, when the tracking object is suffering from large translation or the target and the background have the similar color distributions, the traditional

Mean Shift algorithm would be easily influenced by the minor colors.[68] Thus, an area weighted centroid shifting algorithm was proposed in Ref. 68. In that paper, the author defined a novel optimization function

$$\sum_{u=1}^{m} \hat{q}_u \|\hat{y} - M_u\|, \tag{20}$$

where M_u is the centroid of the color bin u. The next location of the tracking object can be obtained by minimizing Eq. (20). This algorithm localizes the target by a direct one step computation, and receives a good tracking effect even in a difficult situation. Moreover, we can propose many other optimization functions, but it will change the base of Mean Shift tracking.

4.8. *Mode optimization*

The aim of the conventional Mean Shift algorithm is to find the extreme point in every frame. The extreme point is regarded as the location of the tracking object in the current frame. However, the Mean Shift is prone to stuck at a saddle point or a local minimum for a local maximum.[19] In Ref. 19, the author proofed that in the case of using piecewise constant kernels, the Mean Shift algorithm needs the same number of iterations to reach the mode as Newton's method, and Mean Shift is a step to the maximum of a quadratic bound. In order to resolve this problem, Shen *et al.* proposed a multi-bandwidth Mean Shift algorithm, called annealed Mean Shift,[69] this algorithm can reliably find the global maximum of a density distribution. However, using this algorithm to find a global maximum consumes much time, so Shen *et al.* also introduced an adaptive over-relaxed accelerated Mean Shift algorithm to accelerate the convergence speed.[69] The algorithm can recover from tracking failures caused by occlusions, illumination changes, etc.

4.9. *Similarity measurement adaptation*

The typical Mean Shift tracking algorithm uses the Bhattacharyya coefficient as the similarity measure. However, there are three shortcomings on this similarity measurement. First, the target's spatial information will lose since the color histogram is used. Second, Bhattacharyya coefficient is not a discriminative measurement. Lastly, Bhattacharyya coefficient is time consuming, but tracking algorithm usually needs real time.[70] A new sample based similarity measure is proposed in Ref. 70. Instead of estimated probability density function, the author used the expectation of the density estimates, which is more accurate and stable in both lower and higher dimensions.[70] The fast Gauss transform is applied to the tracking algorithm to reduce the computational complexity. In contrast to using the gradient information of the similarity function in the traditional Mean Shift algorithm, second-order information is used in Ref. 71. In that paper, the author introduced Newton and Trust region methods to exploit both the Gradient and Hessian of similarity function.[71] As a result, the second-order methods achieve better tracking accuracy than the conventional Mean Shift algorithm.

4.10. *Automated object initialization*

The most commonly used approach to track targets is to detect them using background subtraction first, and then establish correspondence from frame to frame to find the tracks of the tracking object. This method do not need to select the region of the object in the first time.[72,73] The traditional Mean Shift algorithm is simple and fast, but the tracking object should be determined manually or semi-automatically in the initial frame. Therefore, it is inappropriate for unattended supervision. This drawback affects the spread of Mean Shift object tracking algorithm. For this kind of problem, we can use the method of motion detection to select the tracking object. In conventional method, we often use two methods which do not need to select target region initially to detect motion object: (1) tracking by detection and (2) statistical tracking.[74]

Tracking by detection means to localize the position of the object frame by frame. Currently, popular methods of motion detection are optical flow method, frame difference method and background difference method.[75] And there are many other classes described in Refs. 21 and 76. Among those methods, formula for computing optical flow method is not merely complex, but also computationally intensive, so it is not suitable for real-time applications. The frame difference method and the background difference method are widely used in circumstances that demand real-time responses. So we can use those two methods to detect motion object first, after that we conduct the Mean Shift object tracking algorithm. Gang *et al.* adopted background difference,[77] because the background difference method is the most direct and simplest method of the above three kinds of methods. As the common background subtraction must set a threshold to determine whether it is foreground or background, an adaptive background subtraction method is proposed to select the tracking object in Ref. 78. In order to improve the accuracy of the selected object region, the author added a morphological operation after adaptive background subtraction. In most cases, the moving object has the obvious difference in the gradation with the background. The background subtraction method can effectively extract the target from background. Since background subtraction is easy to be affected by the shadows, Porikli and Tuzel added one step to remove shadows.[35] Then they used a model fusion of GMM and Mean Shift to track object. By combining with the FG/BG detection, Chen *et al.* acquired the object's region and position.[14] The Mixture of Gauss or Hide Markov Model is adopted for FG/BG detection. The variables of Kalman filter and the Mean Shift approach are initialized by the region and the position of the detected object, respectively.

Statistical tracking method is based on motion prediction or updated filter, and does not need to execute the detection process in each frame. This method is faster than the continuous detection method. The main statistical techniques are employed in the tracking process such as Kalman filter, particle filter,[79] etc.

In summary, we first get the region of the target by using the above two methods, then conduct the Mean Shift object tracking algorithm. However, automated object initialization also has many problems, such as the detected object region is often very large or small and it regards the background as motion target, it also cannot detect slow movement target. All these drawbacks may inevitably bring about many negative impacts to the Mean Shift tracker. Therefore, it requires further research.[1,3]

5. Hybrid Tracking Method

The Mean Shift tracking algorithm has many defects in the process of tracking the object, and the improvements of the algorithm have been described above. However, the above improvements mostly aim at improving the Mean Shift algorithm itself. Therefore, a great number of hybrid methods combined with Mean Shift have been proposed, and the most widely used methods are particle filter[80–86] and Kalman filter.[87,88] There are many other methods, like SSD,[10,56] GMM,[27] FIFT,[47] Bayesian filtering,[89] RANSAC,[90] are combined into the Mean Shift method. Moreover, the Mean Shift method is usually applied to search the peak of the confidence map in some tracking algorithms recently.[91–93]

5.1. *Combined with particle filter*

The particle filter is an object tracking method which uses an efficient statistical method to estimate the target state. It consists of three operation stages: selection, prediction and observation. The position of the target is predicted according to a motion model. But this method requires much computational cost, so it is not suitable for simultaneous tracking. Using the real-time property of the Mean Shift, a method that combines the Mean Shift algorithm with particle filter is proposed in Ref. 81. This algorithm need only a few samples so that the computational cost to track object with occlusion has been reduced.[81] Shan *et al.* integrated the Mean Shift into particle filter to overcome the degeneracy problem of particle filter,[82] and this method can handle rapid movements and distractors successfully. This method is applied to hand tracking[82] and face tracking.[83] In order to increase the robustness of this method, target model adaptation is used in Ref. 84 during temporally stable image observations. To the problem of sudden motions and distractions, Wang and Yagi proposed an adaptive Mean Shift tracking algorithm with auxiliary particles to make this method robust and efficient.[85]

5.2. *Combined with Kalman filter*

Kalman filter is a very important tool to track moving objects. It is usually used to make predictions for the following frame and to locate the position or to identify related parameters of the moving objects.[87] Combined Kalman filter with the Mean Shift algorithm, Comaniciu used the Kalman filter to predict the next target location and a confidence region first. Then Mean Shift iteration is conducted to get the optimization model.[14,60] This improvement is robust to partial occlusion, clutter, and target scale variations. In Ref. 94, the SVM and Mean Shift are combined with Kalman filter to overcome their respective limitations. This method is used to track eye. As the Kalman filter is mainly used to smooth, Peng *et al.* obtained the optimal estimation of object model by using adaptive Kalman filter to filter object kernel histogram.[95] The author also updated the target model in time according to the result of hypothesis testing. This method can deal with occlusion and object's appearance changes very well. The Mean Shift algorithm is used to perform tracking when Kalman filter fails due to measurement error in Refs. 87 and 88, and it has been applied to human tracking.

5.3. *Combined with classification techniques*

Recently, object tracking has been considered as a binary classification problem by many researches. The object is considered as positive class and the background is regarded as negative class. Then a classifier should be trained to distinguish the object from the background. Given a new video frame, we use the classifier to classify each pixel, and the classification result forms a confidence map. Finally, the Mean Shift method is used to find the mode of the confidence map. We consider the position of the mode is where the object moved to.

It is a common tracking framework to apply the Mean Shift method to a confidence map. Therefore, different classification techniques and feature selection or combination methods can be used to generate the confidence map. Avidan used an ensemble of weak classifiers to classify each pixel in the search window and updated the ensemble with new weak classifiers that are trained on-line during tracking.[91] Similarly, Grabner *et al.* proposed an on-line AdaBoost feature selection method for tracking.[92,93] This algorithm allows to adjust the classifier while tracking owing to its capability of on-line training. It can handle appearance changes of object and runs in real time. In order to best classify object and background, Collins *et al.* proposed an on-line feature selection mechanism in Ref. 96. According to how well the classifier separate sample distributions of object and background pixels, the on-line feature selection mechanism adaptively selects the best discriminative feature for the Mean Shift tracking system. In order to track object in complex conditions, Yang *et al.* first used Bayesian framework to extract the confident region. Then the Mean Shift object tracking algorithm is used to track the confident region.[97] There are many other classifiers for object tracking, such as on-line ensemble SVM classifiers,[98] multi-cues spatial pyramid matching (MSPM),[99] on-line multiple instance learning[100] and semi-supervised on-line boosting.[101]

6. Conclusion

The Mean Shift algorithm has gained wide attention owing to its many advantages over other tracking algorithms. However, the original Mean Shift algorithm has many defects such as the fixed bandwidth, subject to environmental factor like illumination and occlusion. Although many improvements have been proposed, an improvement aims at resolving only one defect. Therefore, it will be significant to comprehensively improve the algorithm, and to design the improvement of the algorithm that can well address the object tracking issue in a complex scene.

Though this article mainly presents the applications of the Mean Shift algorithm in object tracking, this algorithm can also be used in image segmentation,[102] clustering,[4,103,104] Hough transform,[4] and image filter,[105] etc.

References

1. A. Blake and M. Isard, *Active Contours: The Application of Techniques from Graphics, Vision, Control Theory and Statistics to Visual Tracking of Shapes in Motion* (Springer, New York, 1998).
2. P. H. Li, *Methods for Tracking Moving Object in Image Sequences* (Science Press, Beijing, 2010).
3. K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Transactions on Information Theory* **21**, 32–40 (1975).
4. Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**, 790–799 (1995).
5. D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 603–619 (2002).
6. G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *IEEE Workshop on Applications of Computer Vision* (1998), pp. 214–219.
7. D. Comaniciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *IEEE Conf. Computer Vision and Pattern Recognition, 2000*, Vol. 2 (2000), pp. 142–149.
8. D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**, 564–577 (2003).
9. J. Jeyakar, R. V. Babu and K. R. Ramakrishnan, "Robust object tracking with background-weighted local kernels," *Computer Vision and Image Understanding* **112**, 296–309 (2008).
10. R. V. Babu, P. Pérez and P. Bouthemy, "Robust tracking with motion estimation and local kernel-based color modeling," *Image and Vision Computing* **25**, 1205–1216 (2007).
11. X. Chen, S. Yu and Z. Ma, "An improved mean shift algorithm for moving object tracking," *7th World Congress on Intelligent Control and Automation* (IEEE, 2008), pp. 5111–5114.
12. A. Lehuger, P. Lechat and P. Perez, "An adaptive mixture color model for robust visual tracking," *2006 IEEE Int. Conf. Image Processing* (2006), pp. 573–576.
13. Y. Wu, T. Yu and G. Hua, "Tracking appearances with occlusions," in *2003 IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 2003. Proc.*, Vol. 1 (IEEE, 2003), pp. I-789–I-795.

14. Y. Chen, S. Yu, W. Sun and X. Chen, "Object tracking using an improved kernel method," *Int. Conf. Embedded Software and Systems, 2008. ICESS '08* (IEEE, 2008), pp. 511–515.
15. T. Liu and X. Cheng, "Improved mean shift algorithm for moving object tracking," *2010 2nd Int. Conf. Computer Engineering and Technology* (*ICCET*), Vol. 1 (IEEE, 2010), pp. V1-575–V1-578.
16. Z. Li, Q. L. Tang and N. Sang, "Improved mean shift algorithm for occlusion pedestrian tracking," *Electronics Letters* **44**, 622–623 (2008).
17. G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, 1025–1039 (1998).
18. J. Tu, H. Tao and T. Huang, "Online updating appearance generative mixture model for meanshift tracking," *Machine Vision and Applications* **20**, 163–173 (2009).
19. M. Fashing and C. Tomasi, "Mean shift is a bound optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**, 471–474 (2005).
20. T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Transactions on Communication Technology* **15**, 52–60 (1967).
21. A. Yilmaz, O. Javed and M. Shah, "Object tracking: A survey," *Acm Computing Surveys* (*CSUR*) **38**, 13 (2006).
22. H. Liu, Z. Yu, H. Zha, Y. Zou and L. Zhang, "Robust human tracking based on multi-cue integration and mean-shift," *Pattern Recognition Letters* **30**, 827–837 (2009).
23. A. Yilmaz, X. Li and M. Shah, "Contour-based object tracking with occlusion handling in video acquired using mobile cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**, 1531–1536 (2004).
24. C. Lerdsudwichai, M. Abdel-Mottaleb and A. Ansari, "Tracking multiple people with recovery from partial and total occlusion," *Pattern Recognition* **38**, 1059–1070 (2005).
25. J. S. Hu, C. W. Juan and J. J. Wang, "A spatial-color mean-shift object tracking algorithm with scale and orientation estimation," *Pattern Recognition Letters* **29**, 2165–2173 (2008).
26. K. Quast and A. Kaup, "Scale and shape adaptive mean shift object tracking in video sequences," in *Proc. 17th European Signal Processing Conf.* (*EUSIPCO*), Citeseer (2009), pp. 1513–1517.
27. K. Quast and A. Kaup, "Shape adaptive mean shift object tracking using gaussian mixture models," *2010 11th International Workshop Image Analysis for Multimedia Interactive Services* (*WIAMIS*) (IEEE, 2010), pp. 1–4.
28. J. Ning, L. Zhang, D. Zhang and C. Wu, "Robust mean-shift tracking with corrected background-weighted histogram," *Computer Vision, IET* **6**, 62–69 (2012).
29. C. Xue, M. Zhu and A. Chen, "A discriminative feature-based mean-shift algorithm for object tracking," *IEEE Int. Symp. Knowledge Acquisition and Modeling Workshop, 2008. KAM Workshop 2008* (IEEE, 2008), pp. 217–220.
30. I. Haritaoglu, D. Harwood and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 809–830 (2000).
31. M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision* **1**, 321–331 (1988).
32. K. Kim, D. Harwood and L. Davis, "Background updating for visual surveillance," in *Advances in Visual Computing* (Springer, 2005), pp. 337–346.

33. R. Cucchiara, C. Grana, M. Piccardi and A. Prati, "Detecting objects, shadows and ghosts in video streams by exploiting color and motion information," in *11th International Conf. Image Analysis and Processing, 2001. Proc.* (IEEE, 2001), pp. 360–365.
34. T. Horprasert, D. Harwood and L. S. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," *IEEE ICCV*, Vol. 99 (Citeseer, 1999), pp. 256–261.
35. F. Porikli and O. Tuzel, "Human body tracking by adaptive background models and mean-shift analysis," *IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance* (Citeseer, 2003).
36. A. Adam, E. Rivlin and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," *2006 IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Vol. 1 (IEEE, 2006), pp. 798–805.
37. E. Maggio and A. Cavallaro, "Multi-part target representation for color tracking," *IEEE Int. Conf. Image Processing, 2005. ICIP 2005*, Vol. 1 (2005), pp. 729–732.
38. J. Jeyakar, R. V. Babu and K. R. Ramakrishnan, "Robust object tracking using local kernels and background information," *IEEE Int. Conf. Image Processing, 2007. ICIP 2007*, Vol. 5 (IEEE, 2007), pp. V-49–V-52.
39. F. Wang, S. Yu and J. Yang, "A novel fragments-based tracking algorithm using mean shift," *10th International Conf. Control, Automation, Robotics and Vision, 2008. ICARCV 2008* (IEEE, 2008), pp. 694–698.
40. J. Shi and C. Tomasi, "Good features to track," *1994 IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 1994. Proc. CVPR'94* (IEEE, 1994), pp. 593–600.
41. M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision* **7**, 11–32 (1991).
42. K. Nummiaro, E. Koller-Meier and L. V. Gool, "An adaptive color-based particle filter," *Image and Vision Computing* **21**(1), 99–110 (2003).
43. M. Y. Ju, C. S. Ouyang and H. S. Chang, "Mean shift tracking using fuzzy color histogram," *2010 Int. Conf. Machine Learning and Cybernetics* (*ICMLC*), Vol. 6 (IEEE, 2010), pp. 2904–2908.
44. S. J. McKenna, Y. Raja and S. Gong, "Tracking colour objects using adaptive mixture models," *Image and Vision Computing* **17**, 225–231 (1999).
45. P. Li, "An adaptive binning color model for mean shift tracking," *IEEE Transactions on Circuits and Systems for Video Technology* **18**, 1293–1299 (2008).
46. I. Leichter, M. Lindenbaum and E. Rivlin, "Mean Shift tracking with multiple reference color histograms," *Computer Vision and Image Understanding* **114**, 400–408 (2010).
47. H. Zhou, Y. Yuan and C. Shi, "Object tracking using SIFT features and mean shift," *Computer Vision and Image Understanding* **113**, 345–352 (2009).
48. J. Wang and Y. Yagi, "Integrating color and shape-texture features for adaptive real-time object tracking," *IEEE Transactions on Image Processing* **17**, 235–240 (2008).
49. F. Bousetouane, L. Dib and H. Snoussi, "Improved mean shift integrating texture and color features for robust real time object tracking," *The Visual Computer* **29**, 155–170 (2013).
50. Q. Tian, J. Mao, B. Zheng, R. Liang and P. Zhang, "A robust mean shift tracking algorithm combined with gray spatial texture," in *2010 WASE Int. Conf. Information Engineering* (*ICIE*), Vol. 1 (IEEE, 2010), pp. 194–197.
51. F. Ullah and S. Kaneko, "Using orientation codes for rotation-invariant template matching," *Pattern Recognition* **37**, 201–209 (2004).

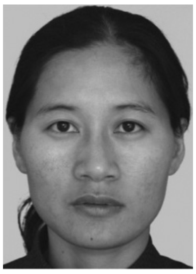
52. X. G. Zhang, E. L. Zhao and Y. J. Wang, "A new algorithm for tracking gray object based on Mean-shift," *Optical Technique* **2**, 226–229 (2007).
53. J. Wang, B. Thiesson, Y. Xu and M. Cohen, "Image and video segmentation by anisotropic kernel mean shift," *Computer Vision-ECCV 2004* (2004), pp. 238–249.
54. A. Yilmaz, "Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection," *IEEE Conf. Computer Vision and Pattern Recognition, 2007 CVPR'07* (IEEE, 2007), pp. 1–6.
55. K. M. Yi, H. S. Ahn and J. Y. Choi, "Orientation and scale invariant mean shift using object mask-based kernel," in *19th Int. Conf. Pattern Recognition, 2008. ICPR 2008* (IEEE, 2008), pp. 1–4.
56. G. D. Hager, M. Dewan and C. V. Stewart, "Multiple kernel tracking with SSD," in *Proc. 2004 IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 2004. CVPR 2004*, Vol. 1 (IEEE, 2004), pp. I-790–I-797.
57. Z. Fan, Y. Wu and M. Yang, "Multiple collaborative kernel tracking," *IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 2005. CVPR 2005*, Vol. 2 (IEEE, 2005), pp. 502–509.
58. I. R. Khan and F. Farbiz, "A back projection scheme for accurate mean shift based tracking," *2010 17th IEEE Int. Conf. Image Processing* (*ICIP*) (IEEE, 2010), pp. 33–36.
59. Y. Shi, H. Liu, Y. Liu and H. Zha, "Adaptive feature-spatial representation for Mean-shift tracker," *15th IEEE Int. Conf. Image Processing, 2008. ICIP 2008* (2008), pp. 2012–2015.
60. D. Comaniciu and V. Ramesh, "Mean shift and optimal prediction for efficient object tracking," in *2000 Int. Conf. Image Processing, 2000. Proc.*, Vol. 3 (2000), pp. 70–73.
61. R. T. Collins, "Mean-shift blob tracking through scale space," in *2003 IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 2003. Proc.*, Vol. 2 (2003), pp. II-234–40.
62. D. Comaniciu, V. Ramesh and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Eighth IEEE Int. Conf. Computer Vision, 2001. ICCV 2001. Proc.*, Vol. 1 (2001), pp. 438–445.
63. S.-X. Li, H.-X. Chang and C.-F. Zhu, "Adaptive pyramid mean shift for global real-time visual tracking," *Image and Vision Computing* **28**, 424–437 (2010).
64. S. Qi and X. Huang, "Hand tracking and gesture gecogniton by anisotropic kernel mean shift," *2008 Int. Conf. Neural Networks and Signal Processing* (2008), pp. 581–585.
65. Z. H. Khan and I. Y. H. Gu, "Joint feature correspondences and appearance similarity for robust visual object tracking," *IEEE Transactions on Information Forensics and Security* **5**, 591–606 (2010).
66. Z. H. Khan, I. Y. H. Gu and A. Backhouse, "Joint particle filters and multi-mode anisotropic mean shift for robust tracking of video objects with partitioned areas," *2009 16th IEEE International Conf. Image Processing* (*ICIP*) (2009), pp. 4077–4080.
67. J. Ning, L. Zhang, D. Zhang and C. Wu, "Scale and orientation adaptive mean shift tracking," *Computer Vision, IET* **6**, 52–61 (2012).
68. L. Suk-Ho, C. Euncheol and K. Moon Gi, "Object tracking based on area weighted centroids shifting with spatiality constraints," in *15th IEEE Int. Conf. Image Processing, 2008. ICIP 2008* (2008), pp. 2632–2635.

69. C. Shen, M. Brooks and A. Van Den Hengel, "Fast global kernel density mode seeking: Applications to localization and tracking," *IEEE Transactions on Image Processing* **16**, 1457–1469 (2007).
70. C. Yang, R. Duraiswami and L. Davis, "Efficient mean-shift tracking via a new similarity measure," *IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 2005. CVPR 2005*, Vol. 1 (IEEE, 2005), pp. 176–183.
71. L. Xiao and P. Li, "Improvement on mean shift based tracking using second-order information," *19th Int. Conf. Pattern Recognition, 2008. ICPR 2008* (IEEE, 2008), pp. 1–4.
72. C. R. Wren, A. Azarbayejani, T. Darrell and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**, 780–785 (1997).
73. C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**, 747–757 (2000).
74. F. Meyer and P. Bouthemy, "Region-based tracking in an image sequence," *Computer Vision — ECCV'92* (Springer, 1992), pp. 476–484.
75. R. T. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa and P. Burt, A system for video surveillance and monitoring, Technical Report, Carnegie Mellon University, the Robotics Institute (2000).
76. P. Turaga, R. Chellappa, V. S. Subrahmanian and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Transactions on Circuits and Systems for Video Technology* **18**, 1473–1488 (2008).
77. Y. Gang, Z. Jun-Sheng, Z. Chun-Hong and Y. Fan, "An approach based on mean shift and background difference for moving object tracking," *2010 6th Int. Conf. Wireless Communications Networking and Mobile Computing* (*WiCOM*), pp. 1–4.
78. M. F. Talu, S. Soyguder and Ö. Aydogmus, "An implementation of a novel vision-based robotic tracking system," *Sensor Review* **30**, 225–232 (2010).
79. R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering* **82**, 35–45 (1960).
80. E. Maggio and A. Cavallaro, "Hybrid particle filter and mean shift tracker with adaptive transition model," in *Proc. IEEE Signal Processing Society Int. Conf. Acoustics, Speech, and Signal Processing* (*ICASSP*) (2005), pp. 19–23.
81. K. Deguchi, O. Kawanaka and T. Okatani, "Object tracking by the mean-shift of regional color distribution combined with the particle-filter algorithms," in *Proc. 17th Int. Conf. Pattern Recognition, 2004. ICPR 2004*, Vol. 3 (2004), pp. 506–509.
82. C. Shan, Y. Wei, T. Tan and F. Ojardias, "Real time hand tracking by combining particle filtering and mean shift," in *Sixth IEEE Int. Conf. Automatic Face and Gesture Recognition, 2004. Proc.* (IEEE, 2004), pp. 669–674.
83. F. Xu, J. Cheng and C. Wang, "Real time face tracking using particle filtering and mean shift," *IEEE Int. Conf. Automation and Logistics, 2008. ICAL 2008* (IEEE, 2008), pp. 2252–2255.
84. K. Nummiaro, E. Koller-Meier and L. Van Gool, "An adaptive color-based particle filter," *Image and Vision Computing* **21**, 99–110 (2003).
85. J. Wang and Y. Yagi, "Adaptive mean-shift tracking with auxiliary particles," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **39**, 1578–1589 (2009).
86. Z. H. Khan, I. Y. Gu and A. G. Backhouse, "Robust visual object tracking using multi-mode anisotropic mean shift and particle filters," *IEEE Transactions on Circuits and Systems for Video Technology* **21**, 74–87 (2011).

87. A. Ali and K. Terada, "A general framework for multi-human tracking using Kalman filter and fast mean shift algorithms," *Journal of Universal Computers Science* **16**, 921–937 (2010).
88. A. Ali and K. Terada, "A framework for human tracking using Kalman filter and fast mean shift algorithms," *2009 IEEE 12th Int. Conf. Computer Vision Workshops* (*ICCV Workshops*) (IEEE, 2009), pp. 1028–1033.
89. Z. Zivkovic, A. T. Cemgil and B. Kröse, "Approximate Bayesian methods for kernel-based object tracking," *Computer Vision and Image Understanding* **113**, 743–749 (2009).
90. S. Haner and I. Y. Gu, "Combining foreground/background feature points and anisotropic mean shift for enhanced visual object tracking," *2010 20th Int. Conf. Pattern Recognition* (*ICPR*) (IEEE, 2010), pp. 3488–3491.
91. S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**, 261–271 (2007).
92. H. Grabner and H. Bischof, "On-line boosting and vision," *IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 2006*, Vol. 1 (2006), pp. 260–267.
93. H. Grabner, M. Grabner and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. BMVC*, Vol. 1 (2006), pp. 47–56.
94. Z. Zhu, Q. Ji, K. Fujimura and K. Lee, "Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination," in *16th Int. Conf. Pattern Recognition, 2002. Proc.*, Vol. 4 (IEEE, 2002), pp. 318–321.
95. N. S. Peng, J. Yang and Z. Liu, "Mean shift blob tracking with kernel histogram filtering and hypothesis testing," *Pattern Recognition Letters* **26**, 605–614 (2005).
96. R. T. Collins, Y. Liu and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**, 1631–1643 (2005).
97. T. Yang, L. I. Jing, P. A. N. Quan and Y. M. Cheng, "Visual tracking with automatic confident region extraction," *International Journal of Image and Graphics* **8**, 369–381 (2008).
98. M. Tian, W. Zhang and F. Liu, "On-line ensemble svm for robust object tracking," *Computer Vision — ACCV 2007* (2007), pp. 355–364.
99. D. Wang, H. Lu and Y. W. Chen, "Object tracking by multi-cues spatial pyramid matching," *2010 17th IEEE Int. Conf. Image Processing* (*ICIP*) (2010), pp. 3957–3960.
100. B. Babenko, M. H. Yang and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **33**, 1619–1632 (2011).
101. H. Grabner, C. Leistner and H. Bischof, "Semi-supervised on-line boosting for robust tracking," *Computer Vision — ECCV 2008* (2008), pp. 234–247.
102. D. Comaniciu and P. Meer, "Robust analysis of feature spaces: Color image segmentation," in *1997 IEEE Computer Society Conf. Computer Vision and Pattern Recognition, 1997. Proc.* (IEEE, 1997), pp. 750–755.
103. K. L. Wu and M. S. Yang, "Mean shift-based clustering," *Pattern Recognition* **40**, 3035–3052 (2007).
104. S. Rao, A. de Medeiros Martins and J. C. Préncipe, "Mean shift: An information theoretic perspective," *Pattern Recognition Letters* **30**, 222–230 (2009).
105. D. Comaniciu and P. Meer, "Mean shift analysis and applications," in *The Proc. Seventh IEEE Int. Conf. Computer Vision, 1999*, Vol. 2 (IEEE, 1999), pp. 1197–1203.

Liwen He is a master's degree candidate in Shenzhen Graduate School, Harbin Institute of Technology (HIT), Shenzhen, China. He received the Bachelor's degree in School of Basic Science from East China Jiaotong University in 2011. His research interests include intelligent video surveillance, object tracking and event detection in crowd scenes.

Yong Xu was born in Sichuan, China, in 1972. He received his B.S. degree and M.S. degree in 1994 and 1997, respectively. He received the Ph.D. degree in Pattern recognition and Intelligence System at NUST (China) in 2005. Now he works at Shenzhen Graduate School, Harbin Institute of Technology. His current interests include image processing, biometrics, face recognition, machine learning and video analysis.



Yan Chen received her B.E. and M.E. degrees in computer science from Northeastern University, China, in 1997 and 2000, respectively, and her Ph.D. in 2010 from University of Technology, Sydney, Australia. Currently, she is a post-doctoral researcher with Harbin Institute of Technology at Shenzhen, China. Her research interests include computer vision and pattern recognition.

Jiajun Wen received the M.S. degree in applied computer technology at Guangdong University of Technology, Guangzhou, China, in 2010. He is currently a Ph.D. candidate in computer science and technology at Shenzhen graduate school, Harbin Institute of Technology (HIT). His current interests include pattern recognition and video analysis.