



Prediction Based Object Tracking Algorithm Using Kalman Filtering

¹ Ms.Vani Biradar G , ²Prof.B.N.Veerappa B.E,M.TECH

¹P.G.Student

²ASSOCIATE PROFESSOR

Department of Studies in Computer science and engineering

UBDT College of Engineering, Davanagere

Abstract -Object tracking have been broadly useful to robotics control, video retrieval, traffic surveillance and homing technologies. A bundle of object tracking algorithms have been reported in literatures, but the region is still incomplete with an efficient algorithm which cannot only track the objects but at the same time capable to identify the direction and movement of object. In this, an efficient object tracking system is proposed based on Modified mean shift tracking (MMST) algorithm. In this project essentially deal by how to tackle the problem to estimation the scale and also direction change of the target over the mean shift tracking framework. In the original mean shift tracking algorithm, the location of the target be able to well estimated, whereas the scale and direction changes cannot be adaptively estimated. Consider that the weight image originated by the target model and the candidate model can characterize the opportunity that pixel belong to the target, in this project work it will illustrate that the original mean shift tracking algorithm can be derived by means of the zeroth and the first order moment of the weight image. With the zero moment and the Bhattacharyya coefficient among the target model and candidate model, a easy and efficient method is projected to estimate the range of target. Subsequently an approach, which utilizes the approximate area and the second order centre moment, is planned to adaptively estimate the width, height and direction changes of the target.

Keywords— Object Tracking, Video Retrieval, Modified mean shift tracking algorithm, Bhattacharyya coefficient

I.INTRODUCTION

1.1 Overview

Moving object tracking plays important role in video surveillance, and has been used in many application such as military and civil area. It would be difficult to track the objects when background, occlusion and light change. Therefore, how to track the moving objects accurately, quickly and steadily in complicated environments is a challenge. Mean shift algorithm is widely used in object tracking with the characteristics of no parameters and fast pattern matching. This algorithm is good at tracking object movement such as deformation and rotation, and strong robustness to noise. However, the algorithm requires the overlap between two adjacent frames areas of the object searching model, and does not use moving direction and velocity in space; therefore it has a limit to track the object with suddenly variable velocity in real-time. Kalman filtering, with less computational complexity and real time calculation, predicts the position of the objects based on linear minimum variance estimation of dynamic system state sequences.

The accuracy of the object model based on mean shift algorithm plays a vital role in moving object tracking in complicated environment. It is focuses on the improved mean shifting algorithm based on generating and updating object model. To eliminate the interference of the most similar features between tracking object and background, the coefficient ratio of the object to surrounding environment is first imported to generate the object model.To make sure the

accuracy of updating object model, the effective way that combines similarity evaluation and Kalman filtering prediction is then applied for judge whether the tracking object is sheltered by other objects or background.

There are three key steps in video analysis, detection interesting moving objects, tracking of such objects from each and every frame to frame, and analysis of object tracks to recognize their behaviour. Therefore, the use of object tracking is pertinent in the tasks of, motion based recognition. Automatic detection, tracking, and counting of a variable number of objects are crucial tasks for a wide range of home, business, and industrial applications such as security, surveillance, management of access points, urban planning, traffic control, etc. these applications are not still playing an important part in consumer electronics.

The main reason is that they need strong requirements to achieve satisfactory working conditions, specialized and expensive hardware, complex installations and setup procedures, and supervision of qualified workers. In this respect, video surveillance usually utilizes electro-optical sensors (video cameras) to collect information. Therefore, surveillance systems must be automated to improve the performance and eliminate such operator errors. Ideally, an automated surveillance system should only require the objectives of an application, in which real time interpretation and robustness is needed. Then, the challenge is to provide robust and real-time performing surveillance systems at an affordable price. With the decrease in costs of hardware for sensing and computing, and the increase in the processor speeds, surveillance systems have become commercially available, and they are now applied to a number of different applications, such as traffic monitoring, airport and bank security.

There are some challenging problems within the surveillance algorithms, such as background modeling, feature extraction, tracking, occlusion handling and event recognition. Moreover, machine vision algorithms are still not robust enough to handle fully automated systems and many research studies on such improvements are still being done. This work focuses on developing a framework to detect moving objects and generate reliable tracks from surveillance video. The problem is most of the existing algorithms works on the gray scale video. But after converting the RGB video frames to gray at the time of conversion, information loss occurs. The main problem comes when background and the foreground both have approximately same gray values. Then it is difficult for the algorithm to find out which pixel is foreground pixel and which one background pixel. Sometimes two different colors such as dark blue and dark violet, color when converted to gray scale, their gray values will come very near to each other, it can't be differentiated that which value comes from dark blue and which comes from dark violet. However, if color images are taken then the background and foreground color can be easily differentiated. So without losing the color information this modified background model will work directly on the color frames of the video.

A. Objectives

Automatic tracking of objects can be the foundation for many interesting applications. An accurate and efficient tracking capability at the heart of such a system is essential for building higher level vision-based intelligence. Tracking is not a trivial task given the non-deterministic nature of the subjects, their motion, and the image capture process itself. The objective of video tracking is to associate target objects in consecutive video frames. The association can be especially difficult when the objects are moving fast relative to the frame rate. The main objectives are,

1. To set up a system for automatic segmentation and tracking of moving objects in stationary camera video scenes, which may serve as a foundation for higher level reasoning tasks and applications.
2. To make significant improvements in commonly used algorithms.
3. To track object using mean shift in complicated environment.
4. To analyze the performance with different filters.

B. Problem statement

Tracking is usually performed in the context of higher –level application that require the location and or shape of the object in every frame. Object tracking can be described as tracking the motion of an object of interest by consistently assigning tags to the object throughout consecutive video frames of scene. Difficulties in object tracking can arise due to abrupt object motion, changing appearance patterns of both the object and the scene, occlusion, non-occlusion situation. In many cases the problem is arise, such as partial background change, motion blur, object's occlusion, temporal movement, merging of objects etc.

Hence, in general object tracking is a challenging problem. and disadvantages of the earlier method is that objects consisting of different colors, will not be found as one object. And also if object consisting of same colors in that time

All Rights Reserved, @IJAREST-2016

also general algorithm fails to track the moving object. Here a new strategy is proposed to improve the tracking ability of mean shift algorithm, in which the contrast between object and background along with similarity evaluation are applied for generating and updating object model.

To eliminate the interference of the most similar features between tracking object and background, the coefficient ratio of the object to surrounding environment is first imported to generate the object model. the effective way that combines similarity evaluation and Kalman filtering prediction is then applied for judge whether the tracking object is sheltered by other objects or background. This algorithm is good at tracking object movement such as deformation and rotation, and strong robustness to noise, and also updating the object model timely by using the rest information of object during occlusion.

C. Existing system

Object tracking can be described as tracking the motion of an object of interest by consistently assigning tags to the object throughout consecutive video frames of scene. Difficulties in object tracking can arise due to abrupt object motion, changing appearance patterns of both the object and the scene, occlusion, non-occlusion situation. In many cases the problem is arise, such as partial background change, motion blur, object's occlusion, temporal movement, merging of objects etc.

Hence, in general object tracking is a challenging problem. and disadvantages of the earlier method is that objects consisting of different colors, will not be found as one object. And also if object consisting of same colors in that time also general algorithm fails to track the moving object.

D .Proposed system

Mean shift algorithm fails to track the moving object in complicated environment. Here a new strategy is proposed to improve the tracking ability of mean shift algorithm, in which the contrast between object and background along with similarity evaluation are applied for generating and updating object model. To eliminate the interference of the most similar features between tracking object and background, the coefficient ratio of the object to surrounding environment is first imported to generate the object model.

To make sure the accuracy of updating object model, the effective way that combines similarity evaluation and Kalman filtering prediction is then applied for judge whether the tracking object is sheltered by other objects or background. The experimental results have shown that the proposed method can track the moving object stably. This method can not only detect occurrence of occlusion exactly, but also updating the object model timely by using the rest information of object during occlusion. This algorithm is good at tracking object movement such as deformation and rotation, and strong robustness to noise.

II.LITERATURE SURVEY

Along with different methods of tracking, the mean shift tracking algorithm be a trendy one suitable to its effectiveness and simplicity. The mean shift algorithm was initially derived by Fukunaga and Hostetler [2] for analysis of data. In their paper a nonparametric density gradient estimation with a generalized kernel method is adapted. Various circumstances based on the kernel functions are estimated by consistency, guarantee asymptotic unbiasedness and identical reliability of the estimates. The outcomes are comprehensive to acquire a simple mean-shift estimate so as to it will be extensive in a nearest-neighbour method. Applications of efficient evaluation to pattern identification are offered using intrinsic dimensionality problems and clustering along with the crucial aim of provided that additional considerate of these trouble in terms of density gradients.

Cheng [3] was established Mean shift algorithm to the area of computer vision. In their paper he was discussed in brief about Mean shift, Mean Shift is a easy interactive process that shift each data point to the standard of data points in its neighbourhood is universal and analyzed. This simplification makes several k-means like clustering algorithms its particular cases.

It is exposed that mean shift is a mode-seeking procedure lying on the plane construct with a "shadow" kernel. For Gaussian kernels, mean shift is a gradient mapping. Divergence is considered for mean shift iterations. Cluster investigation if treated as a deterministic difficulty of result a fixed point of mean shift that characterize the data. Uses in

clustering and Hough transform were established. Mean shift is as well measured as an evolutionary scheme that perform multistate global optimization.

Bradski [6] was introduced modified Mean Shift Algorithm and also estimated the Continuously Adaptive Mean Shift (CAMSHIFT) algorithm used for face tracking. As a primary step in the direction of a perceptual user interface, a computer vision algorithm of color tracking was estimated and useful towards tracking human being faces. Computer vision algorithms that are planned to outline part of a perceptual client interface should be quick and well-organized. They should be capable to track in real time so far not attract a most important share of computational property: other tasks should be proficient to run whereas the visual interface is used. The new algorithm derived was based on a strong non-parametric procedure for hiking density gradients to discover the mode (peak) of probability distributions is known as the mean shift algorithm. In his case, they desire to locate the method of a color distribution inside a video scene. Hence, the mean shift algorithm was adapted to compact with enthusiastically changing color probability distributions originated from video frame sequence. The modified algorithm was specified as the Continuously Adaptive Mean Shift (CAMSHIFT) algorithm. CAMSHIFT's tracking accuracy was distinguished against a Polhemus tracker. Tolerance to noise, performance and distracters was calculated. CAMSHIFT was subsequently used as a computer interface for overprotective commercial computer games and for explore immersive 3D graphic worlds.

Comaniciu and Meer were effectively applied mean shift algorithm to object tracking and image segmentation [7]. They have derived a innovative method for real time tracking of non-rigid objects seen from a stirring camera was proposed. The essential computational part is dependent on the mean shift iterations and calculates the more feasible target location in the existing frame. The variation between the target model (its color distribution) and the target candidates were articulated by a metric originated from the "Bhattacharyya" coefficient. The hypothetical investigation of the approach exposed, that it related to the Bayesian framework whereas given that a practical, efficient solution and robust. The capacity of the tracker to hold in real time partial occlusions, target scale variations and significant clutter were established for numerous image sequences.

III.METHODOLOGY

The explanation of the proposed Modified Mean shift tracking algorithm is discussed in this section. Fig 3.1 represents the block diagram of proposed methodology.

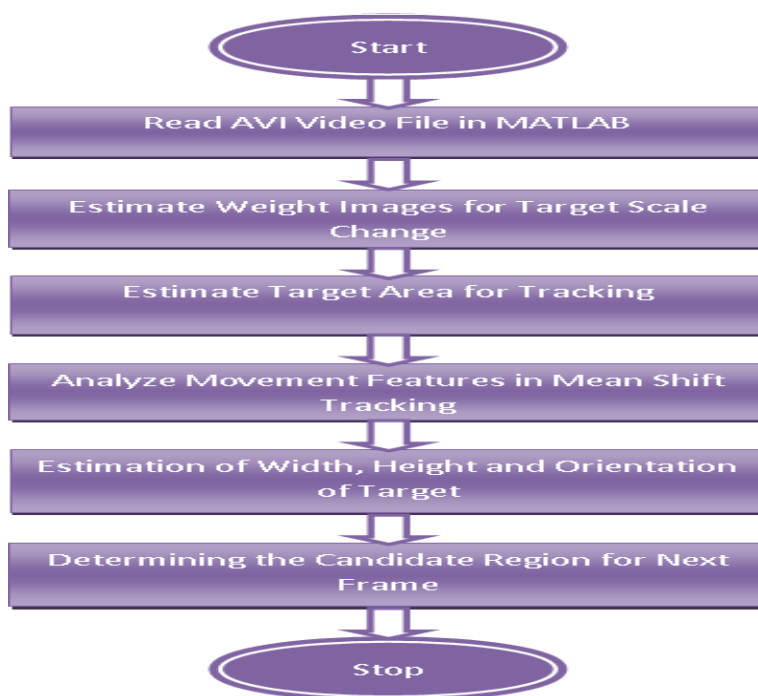


Figure 3.1: Project Methodology

3.1 Mean shift tracking algorithm

3.1.1 Target representation

In object tracking, a target is typically represented as an ellipsoidal or a rectangle area in the image. Presently, a broadly used target demonstration is the color histogram because of its self-determination of rotation and scaling and its toughness to biased occlusions. Denote by $\{X_i\}_{i=1 \dots n}$ the normalized pixels in the target region, which is invented to the centred at the source point and have n pixels. The probability of the feature u ($u=1, 2 \dots m$) in the target model is computed as.

$$\begin{cases} \hat{q} = \{\hat{q}_u\}_{u=1 \dots m} \\ \hat{q}_u = C \sum_{i=1}^n k(\|x_i^*\|^2) \delta[b(x_i^*) - u] \end{cases} \quad (3.1)$$

Where, $\hat{q}$ is the target model, $\hat{q}_u$ is the probability of the u^{th} element of $\hat{q}$, $b\{X_i\}$ associates the pixel X_i to the histogram b_{in} , δ is the Kronecker delta function and $k(x)$ is an isotropic kernel profile. Constant C is a normalization function represented by

$$C = 1 / \sum_{i=1}^n k(\|x_i^*\|^2) \quad (3.2)$$

Similarly, from the candidate area centred at location y , the probability of the feature u in the target candidate model is given by

$$\begin{cases} \hat{p}(y) = \{\hat{p}_u(y)\}_{u=1 \dots m} \\ \hat{p}_u(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u] \end{cases} \quad (3.3)$$

$$C_h = 1 / \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \quad (3.4)$$

Where, $\hat{p}(y)$ is the target candidate model, $\hat{p}_u(y)$ is the probability of the u^{th} element of $\hat{p}(y)$. $\{X_i\}_{i=1 \dots n_h}$ 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 . 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 $\hat{p}(y)$ and $\hat{q}$ as follows

$$\rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y) \hat{q}_u} \quad (3.5)$$

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

$$d[\hat{p}(y), \hat{q}] = \sqrt{1 - \rho[\hat{p}(y), \hat{q}]} \quad (3.6)$$

Minimizing the distance $d[\hat{p}(y), \hat{q}]$ in Eq. (3.6) is equivalent to maximizing the Bhattacharyya coefficient $\rho[\hat{p}(y), \hat{q}]$ in Eq. (3.5). The optimization process is initialized with the target place and is an iterative process, denoted by y_o in the previous frame. By using the Taylor expansion around coefficient $\hat{p}_u(y_o)$, the linear approximation of the Bhattacharyya in Eq. (3.5) can be obtained as:

$$\rho[\hat{p}(y), \hat{q}] \approx \frac{1}{2} \sum_{u=1}^m \sqrt{\hat{p}_u(y_0) \hat{q}_u} + \frac{C_h}{2} \sum_{u=1}^{n_h} w_i k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \quad (3.7)$$

Where,

$$w_i = \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \delta[b(x_i) - u] \quad (3.8)$$

Since the first term in Eq. (3.7) is independent of y , to minimize the distance in Eq. (3.6) is to maximize the second term in Eq. (3.7). In the mean shift iteration, the estimated target moves from y to a new position y_1 , which is defined as

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y - x_i}{h}\right\|^2\right)} \quad (3.9)$$

When we choose the kernel $k(x)$ with the Epanechnikov profile, there is $g(x) = k(x) = 1$, and Eq. (3.9) can be reduced to.

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i} \quad (3.10)$$

By using Eq. (3.10), the mean shift tracking algorithm finds in the new frame the most similar area to the object. From Eq. (3.10) it can be observed that the key parameters in the mean shift tracking algorithm are the weights w_i . In this we will centre of attention on the analysis of w_i with which the orientation and scale of the tracked target can be well estimated, and then a scale and orientation adaptive mean shift tracking algorithm can be developed.

3.2 Modified mean shift tracking for scale and orientation of target

In this section, we initially study about how to compute adaptively the orientation and scale of the target in sub-sections 3.2.1 ~ 3.2.5, then in sub-section 3.2.6, a modified mean shift tracking (MMST) algorithm for orientation and scale of target is offered.

The shrinking or enlarging of the target is typically a steady process in successive frames. Therefore we can suppose that the scale change of the target is smooth and this statement hold practically well in more video sequences. No general tracking algorithm can track it efficiently, if the scale of the target changes suddenly in adjacent frames. Among this supposition, we can invent a little adjustment of the original mean shift tracking algorithm. Assume that we have evaluated the region of the target (the region estimation will be discussed in sub-section 3.2.2) in the earlier frame, in the existing frame we allow the area or window size the of the target candidate area be a slight higher than the evaluated region of the target. Hence, no issue how the orientation and scale of the target change, it must be still in this higher target candidate area in the existing frame. Now the difficulty turns to how to approximate the orientation and real area from the target candidate area.

3.2.1 The Weight images for target scale changing

In the mean shift tracking algorithms and the CAMSHIFT, the evaluation of the target position is essentially gained by using a weight image. In CAMSHIFT, a hue-based object histogram is derived by the weight image,

where probability of its hue in the object model is the weight of a pixel. Although the weight image is represented by Eq. (3.8) in the mean shift tracking algorithm, where the weight of a pixel is the square root of the ratio of its color probability in the target candidate model to its color probability in the target model. Furthermore, it is not exact to use the weight image by CAMSHIFT to approximate the position of the target, and the mean shift tracking algorithm can have improved evaluation outcomes. So that, the weight image in the mean shift tracking algorithm is more consistent than that in the CAMSHIFT algorithm.

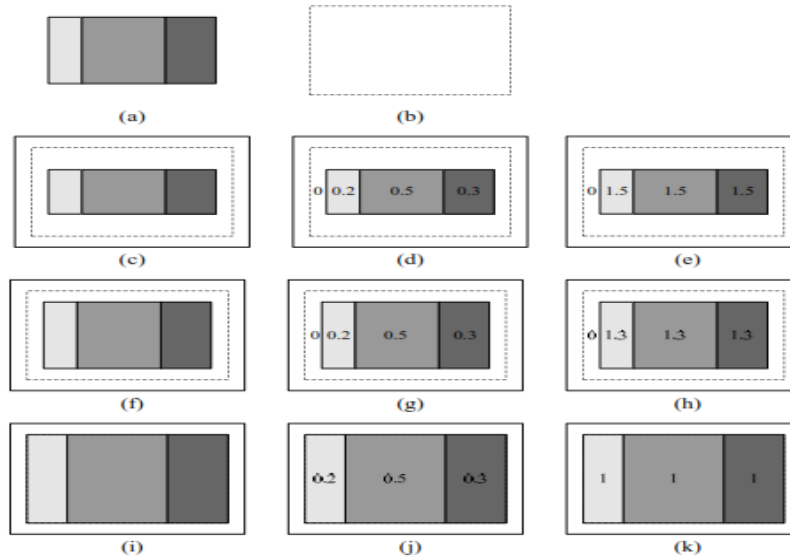


Fig. 3.2: Weight images in CAMSHIFT and mean shift tracking algorithms when the object scale changes.

(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. As in the CAMSHIFT algorithm, in the MMST scheme to be developed, the scale and orientation of the target will be estimated by using the moment features 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 MMST algorithm.

As mentioned at the beginning of Section 4, 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 3.2-(a) shows a synthesized target that has three gray levels. Figure 3.2-(b) shows the candidate region that is a little bigger than the target. Figures 3.2-(c), (f) and (i) are the tracking results when the scale of the synthesized target decreases, keeps invariant and increases, respectively. Figures 3.2-(d), (g) and (j) illustrate the weight images calculated by the CAMSHIFT algorithm in the three cases, while Figures 3.2-(e), (h) and (k) illustrate the weight images calculated by the mean shift tracking algorithm in the three cases.

3.2.2 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:

$$M_{00} = \sum_{i=1}^n w(x_i) \quad (3.11)$$

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. (3.8) 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 3.2-(e), 3.2-(h) and 3.2-(k).

On the other hand, the Bhattacharyya coefficient (referring to Eq. (3.5)) is an indicator of the similarity between the target model q and the target candidate model $p(y)$. A smaller Bhattacharyya coefficient means that there are more

All Rights Reserved, @IJAREST-2016

features from the background and fewer features from the target in the target candidate region, vice versa. If we take M_{00} as the estimation of the target area, then according to Eq. (3.11), 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 3.2 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 increase (e.g., the estimation error will decrease). M_{00} as the target area will also 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:

$$A = c(\rho)M_{00} \quad (3.12)$$

Where $c(\rho)$ is a monotonically increasing function with respect to the Bhattacharyya coefficient $\rho(0 \leq \rho \leq 1)$. As can be seen in Figures 3.2-(e), 3.2-(h) and 3.2-(k) and Table 3.1, 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:

$$c(\rho) = \exp\left(\frac{\rho-1}{\sigma}\right) \quad (3.13)$$

From Eqs. (3.12) and (3.13) 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 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 3.1 The area estimation (pixels) of the target under different scale changes by the project method

Tracking Result		Fig. 3.2 (e)		Fig. 3.2 (h)		Fig. 3.2 (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	+0.5	195	+30%	240	0%
	$\sigma = 1.5$	118.42	+18.42%	169.59	+13.06%	240	0%
	$\sigma = 1$	105.22	+5.22%	158.16	+5.44%	240	0%

Table 3.1 lists the area estimation results of the target by using Eq. (3.12) under different scale changes in Figures 3.2-(e), 3.2-(h) and 3.2-(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.

3.2.3 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:

$$M_{10} = \sum_i^{n_k} w_i x_{i,1} \quad M_{01} = \sum_i^{n_k} w_i x_{i,2} \quad (3.14)$$

$$M_{20} = \sum_i^{n_k} w_i x_{i,1}^2, M_{02} = \sum_i^{n_k} w_i x_{i,2}^2, M_{11} = \sum_i^{n_k} w_i x_{i,1} x_{i,2} \quad (3.15)$$

Where, pair $(x_{i,1}, x_{i,2})$ is the coordinate of pixel i in the candidate region. Comparing Eq. (3.10) with Eqs. (3.11) and (3.14), we can find that y_1 is actually the ratio of the first order moment to the zeroth order moment:

$$y_1 = (\bar{x}_1, \bar{x}_2) = (M_{10} / M_{00}, M_{01} / M_{00}) \quad (3.16)$$

Where $(\bar{x}_1, \bar{x}_2)$ 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. (3.10), (3.11), (3.15) and (3.16), we can convert Eq. (3.9) to the second order center moment as follows

$$\mu_{20} = M_{20} / M_{00} - \bar{x}_1^2 \quad \mu_{11} = M_{11} / M_{00} - \bar{x}_1 \bar{x}_2 \quad \mu_{02} = M_{02} / M_{00} - \bar{x}_2^2 \quad (3.17)$$

Eq. (3.17) can be rewritten as the following covariance matrix in order to estimate the width, height and orientation of the target:

$$Cov = \begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} \quad (3.18)$$

3.2.4 Estimating the width, height and orientation of the target

By using the estimated area (sub-section 3.2.2) and the moment features (sub-section 3.2.3), the width, height and orientation of the target can be well estimated. The covariance matrix in Eq.(3.18) can be decomposed by using the singular value decomposition (SVD) as follows

$$Cov = U \times S \times U^T = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \times \begin{bmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}^T \quad (3.19)$$

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 Eigen values of Cov. The vectors $(u_{11}, u_{21})^T$ and $(u_{12}, u_{22})^T$ 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 height of the target, which is actually improper? 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 that the ratio of λ_1 and 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

All Rights Reserved, @IJAREST-2016

$$k = \sqrt{A / (\pi \lambda_1 \lambda_2)} \quad (3.20)$$

$$a = \sqrt{\lambda_1 A / (\pi \lambda_2)} \quad b = \sqrt{\lambda_2 A / (\pi \lambda_1)} \quad (3.21)$$

Now the covariance matrix becomes

$$Cov = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \times \begin{bmatrix} a^2 & 0 \\ 0 & b^2 \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}^T \quad (3.22)$$

The adjustment of covariance matrix Cov in Eq. (3.22) is a key step of the proposed algorithm. It should be noted that the EM-like algorithm by Zivkovic and Krose 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 4.2, we listed the estimated width, height and orientation of the synthetic ellipse sequence in Figure 4.2 together with the relative estimation error by using the Developed MMST algorithm. It can be seen that the estimation accuracy is very satisfying.

3.2.5 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. (3.22), we define the following covariance matrix to represent the size of the target candidate region in the next frame

$$Cov_2 = U \times \begin{bmatrix} (a + \Delta d)^2 & 0 \\ 0 & (b + \Delta d)^2 \end{bmatrix} \times U^T \quad (3.23)$$

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

$$(x - y_1) \times Cov_2^{-1} \times (x - y_1)^T \leq 1 \quad (3.24)$$

3.2.6 Implementation of the MMST Algorithm

Based on the above analyses in sub-sections 3.2.1 ~ 3.2.5, the scale and orientation of the target can be estimated and then a scale and orientation adaptive mean shift tracking algorithm, i.e. the MMST algorithm, can be developed. The implementation of the whole algorithm is summarized as follows.

Algorithm of Modified Mean Shift Tracking

- 1) **Initialization:** calculate the target model $\hat{q}$ 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 $\hat{p}(y_0)$ in the current frame.
- 4) Calculate the weight vector $\{w_i\}_{i=1 \dots n}$ using Eq. (3.8).
- 5) Calculate the new position y_1 of the target candidate model using Eq. (3.10).
- 6) Let $d \leftarrow \|y_1 - y_0\|$, $y_0 \leftarrow y_1$. Set the error threshold ε (default 0.1) and the maximum Iteration number N

All Rights Reserved, @IJAREST-2016

(default 15).

If ($d < \varepsilon \vee k \geq N$)
Otherwise

Stop and go to step 7;

$k \leftarrow k+1$, and go to step 3.

- 7) Estimate the width, height and orientation from the target candidate model using Eq. (3.22).
- 8) Estimate the initial target candidate model for next frame using Eq. (3.24).

IV.EXPERIMENTAL RESULTS AND DISCUSSIONS

This section evaluate the improved MMST algorithm in contrast with the inventive mean shift algorithm, i.e., mean shift tracking along with a permanent scale, the EM-shift algorithm and the adaptive scale algorithm. These two algorithms are two representative schemes to deal with the orientation and scale changes of the targets under the mean shift framework. It is prone to errors in estimating the orientation and scale of the object, since the weight image approximate by CAMSHIFT is not consistent. Therefore CAMSHIFT is not used in the experiments. We selected particular RGB color space like as the feature space and it was quantized into $16 \times 16 \times 16$ bins for a reasonable contrast among several algorithms. It must be illustrious that other color space such as the HSV can also be used in MMST. Three real video sequences and one artificial video sequence are used in the experiments.

4.1 Experiments on real video sequences

The improved MMST algorithm is subsequently analyzed by using 4 real video sequences. Torch sequence recorded in home is the initial video (Figure 4.1) whereas the object has obviously orientation and scale changes. To illustrate the efficiency of improved MMST algorithm as shown in figure 4.1 contains successive frames 20, 40, 80.



Fig. 4.1: Tracking results of the torch sequence for MMST algorithms. The frames 20, 40 and 80 are displayed.

4.2 Experiments on a synthetic sequence

To validate the efficiency of the proposed MMST algorithm, we initially use a synthetic ellipse sequence. As shown in Figure 4.2-(d), the window size of the primary target (blue ellipse) is 59×89 . We choose $\Delta k = 10$ in the improved MMST algorithm as a result that the window size of the initial target candidate area (red ellipse in Fig 4.2-(b)) in frame 1 is 79×109 . In the MMST outcomes for other frames, the external ellipses correspond to the target candidate region, which are used for estimation of the real targets, i.e., the inner ellipses. The experimental outcomes represents that the improved MMST algorithm may possibly consistently track the ellipse along with orientation and scale changes. For the moment, the experimental outcomes provide the fixed-scale mean shift is not superior for the reason that remarkable orientation and scale changes of the object. The adaptive scale algorithm does not evaluate the target direction change and has shocking tracking outcomes. The EM-shift algorithm fails to accurately approximate the orientation and scale of the synthetic ellipse, even though the target in this sequence is extremely easy.

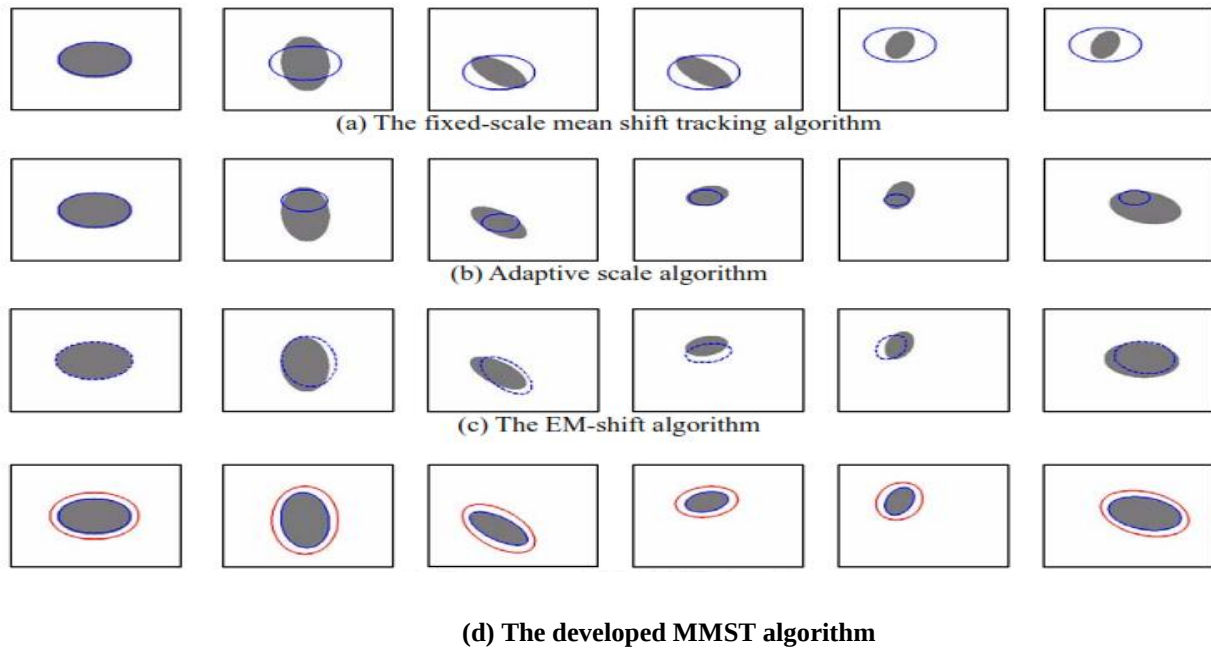


Fig. 4.2: Tracking outcomes of the synthetic ellipse sequence by altered tracking algorithms.

The red ellipses show the target candidate area whereas the blue ellipse shows the approximate target area. The frames 1, 20, 30, 40, 50, 70 are displayed. Table 4.1 lists the estimated width, height and orientation of the ellipse in this sequence used by MMST method. Evaluation of orientation is the angle between the x-axis and major axis. To describe the rest frames and the target model were used for testing; the initial frame of the sequence was used. It can be proved that the improved MMST technique gains excellent estimation accurateness of the orientation and scale of the target.

Table 4.1: The estimation outcomes and accuracy of the height, width and orientation of the ellipse by the improved MMST technique.

Frame No.	Semi Major Length a			Semi Mior 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			3.5			2.81			1.47

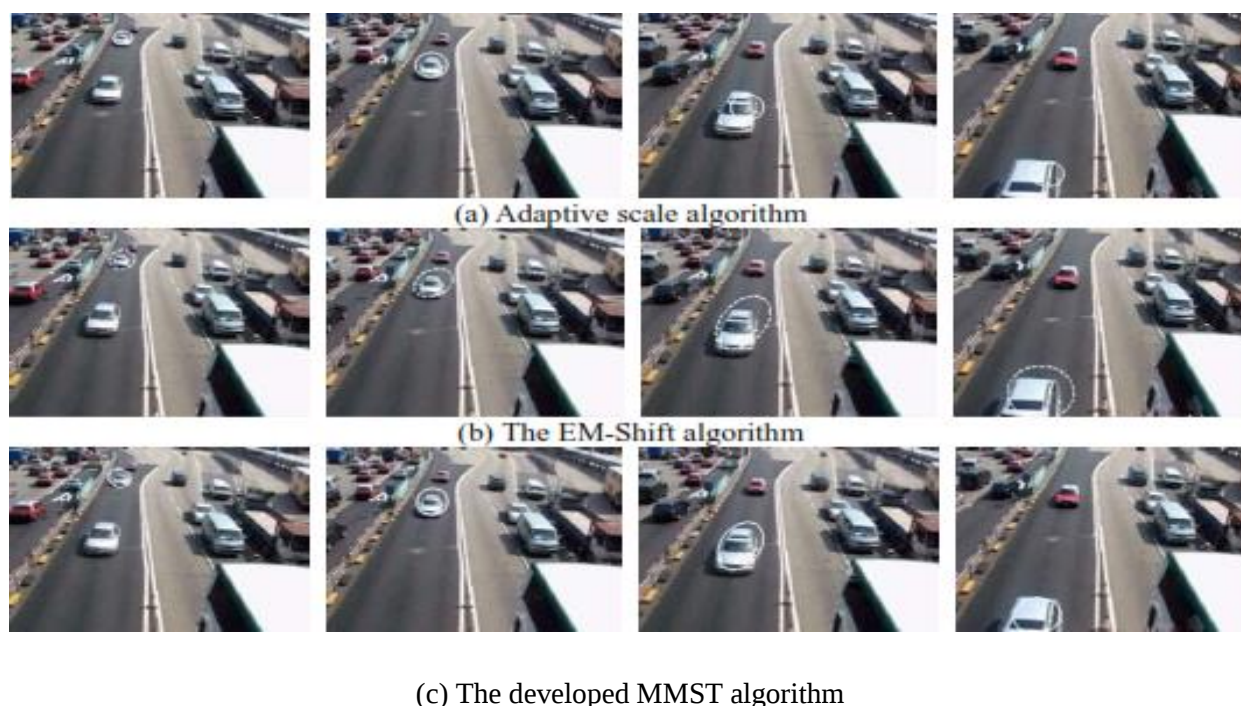


Fig. 4.3: Tracking results of the car sequence by various tracking algorithms. The frames 15, 40, 60 and 75 are displayed.

A car sequence where the scale of the object is the third video (a white car) increases progressively as shown in Figure 4.3. The experimental outcomes represent that the improved MMST algorithm approximates more exactly the scale changes than the EM-shift algorithms and the adaptive scale.

Table 4.2. The average number of iterations by different methods on the four sequences.

Methods	Fixed-scale mean shift	Adaptive scale	EM-shift	MMST
Synthetic ellipse	2.34	13.62	6.27	2.59
Car sequence	3.82	11.25	6.34	3.34

Table 4.2 lists the average numbers of iterations used by various methods on the 4 video sequences. The original mean shift algorithm with fixed scale equal to that of the average number of iterations of the improved MMST is approximately. Because it runs mean shift algorithm three times so the iteration number of the modified scale algorithm is the maximum. The foremost factors which causes on the MMST algorithms and the convergence speed of the EM-shift are the calculation of the covariance matrix. EM-shift evaluates it in each iterations whereas MMST only evaluates it one time for each frame. Therefore MMST is quicker than EM-shift.

V. CONCLUSION AND FUTURE ENHANCEMENT

By studying the moment characteristics of the weight image of the “Bhattacharyya” coefficients and target candidate area as well as we improved an orientation and scale adaptive mean shift tracking (MMST) algorithm. It is capable of resolve the difficulty of how to estimate vigorously the orientation and scale changes of the target under the mean shift tracking framework.

All Rights Reserved, @IJAREST-2016

The probability can represent by the weight of a pixel in the candidate area of belong to the target, although the weighted area of the candidate area can represent by the zeroth order moment of the weights image. The “Bhattacharyya” coefficients and the zeroth order moment are used among the candidate and the target model, a easy and efficient technique to estimation of the target region was proposed. Subsequently a new method, which is depends on the corrected second order and the center moments region of the target, it was projected to adaptively estimation of the height, width and direction changes of the target.

The improved MMST technique inherits the advantages of mean shift tracking such as efficiency, simplicity, and robustness. Wide-ranging experiments were performed and the outcomes proved that MMST can consistently track the objects along with orientation and scale changes, which is complex to accomplish by other state-of-the-art techniques. In the future research, we will attention on how to identify and utilize the accurate shape of the target, as an alternative of a rectangle or an ellipse model used for a further robust tracking.

ACKNOWLEDGEMENT

I am highly obliged to Department of studies in computer science and engineering, UBTD college of engineering. And I am highly grateful and thankful to my guide Prof .B.N.Veerappa for his valuable instructions, guidance, corrections in my project work and presentation.

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.

