International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 5, May - 2014

Video Based Tracking with Mean-Shift and Kalman Filter

Ashvini Kulkarni
Department of Electronics and Telecommunication
Marathwada Institute of Technology
Aurangabad, India

Manasi Vargantwar
Department of Electronics and Telecommunication
Marathwada Institute of Technology
Aurangabad, India

***Abstract*—tracking object in video sequence is receiving enormous interest in computer vision research. This paper we contrast performance of Mean-Shift algorithm's gradient descent based search strategy with Kalman filter based tracking algorithm used to models the dynamic motion of target object to guide estimate object's position through time. Experimental results of tracking a car demonstrate that the proposed Kalman filter for object tracking is efficient under dynamic environment, robust in occlusion comes at the cost of higher computational requirement, helps to separate object pixel from background pixel for fast moving object.**

Keywords—Object Tracking,Mean-Shift,PDF,Kalman Filter

I. INTRODUCTION

Object tracking on video sequence has many applications such as surveillance systems, public security, visual monitoring and so on. Many surveillance application, the region under video surveillance is simply too large for continuous object observation in video streams. Thus some sort of automation is required to alert the observer to presence of object within the surveillance region and to maintain track of these objects. Continual tracking is most important objective under such application. For continual tracking in video sequence, the tracking algorithm must be able to track object in difficult operation conditions such as low contrast, occlusion of the object being tracked.

The most popular algorithm for object tracking is Mean-shift algorithm [1].This algorithm uses gradient optimum algorithm to realize target location and it can track moving object in video sequence. This algorithm has good ability to track articulated objects such as humans. However it has few weak points such as a gradient descent search strategy, the Mean-Shift is susceptible to converging to similar appearance surround the object being tracked.

To overcome limitation of Mean-Shift tracker Kalman filter is used. Kalman filter has extensive applications in different fields like real time graphics, robotics and computer vision. Kalman filter is optimal estimator using the information from measurements and previous states. Kalman filter has been used in tracking mainly for smoothing the object trajectory. In this paper we present the performance of Mean-Shift and Kalman filter algorithm for fast moving vehicle tracking in video sequence.

The remainder of the paper is organized a follows. Section II describes the basic formulation of an object tracking system. Section III describes Mean-Shift algorithm, gradient descent search. In section IV consists of Kalman filter implementation. Section V consists of description of video sequences and test result in section VI. Conclusion and future work are discussed in section VII.

II. VIDEO-BASED TRACKING

Video based tracking is nothing but process of locating fast or slow moving object. The main difficulty in video tracking is to associate target locations in consecutive video frames, especially when the objects are moving fast relative to the frame rate. Here, video tracking systems usually employ a motion model which describes how the image of the target might change for different possible motions of the object to track. There are two important elements of a video tracking system. Target Representation, Localization and Filtering, Data Association. Out of these two first is Target Representation and Localization. Computational complexity for these algorithms is low. In this initial stage target representation and localization is done on the basis of blob tracking, kernel based tracking and contour tracking algorithms.

Filtering and Data Association, which involves incorporating prior information about the scene or object, dealing with, object dynamics, and evaluation of different hypotheses. The computational complexity for these algorithms is usually much higher. Kalman filtering and Particle filtering are the most popular algorithms in Filtering and data association.

For target representation and localization in given video sequence we have selected one of the method as kernel based tracking [2] and detailed description is provided in section III.

III. MEAN-SHIFT BASED OBJECT TRACKING

A. Probability Density function

The kernel based tracker convergence of the natural mean-shift procedure is mainly determined by the kernel density estimation of the target feature and similarity of the target with the candidate in the tracking or kernel window like Bhattacharyya coefficients [1].However, due to target-shift

Invariance of the colour histogram, the coefficient surface obtained tends to be flat and would make the mean-shift algorithm converge to local extrema and result in imprecise target localization. This situation would further deteriorate when target moves out of the kernel due to target or background motions. To make the colour feature more robust and discriminative for the mean-shift based tracker, spatial information has to be incorporated. The mean shift algorithm is an efficient and nonparametric method for mode seeking based on probability density estimation (PDE).Technically Mean-Shift algorithm estimates the color PDF of image patch using each pixel contribution in weighted form using kernel. This algorithm addresses kernel-window towards possible object location, as follows [2]:

$$m(x) = \frac{\sum_{i=1}^{n} K(x-x_i)x_i}{\sum_{i=1}^{n} K(x-x_i)} \quad (1)$$

Where K(x) is kernel function x and m(x) are current and next Estimated position. Usually the kernel K(x) is a function $\|x^2\|$ of kernel profile k(x) is given by [1]:

$$k(x) = \begin{cases} 1, & if \ \|x\| \le 1 \\ 0, & otherwise \end{cases} \quad (2)$$

There are different variations in kernel type that is Gaussian, Uniform, Epanechnikov, Flat, and Normal. The Gaussian for flat kernel profile is given by:

$$k(x) = \exp(-\|x^2\|) \quad (3)$$

Parzen Window is very popular method to estimate the probability density function of data within the kernel. Parzen-window density estimation is essentially a data-interpolation technique [2].Given an instance of the random sample x_i in d dimensional space for a set of n data points. Parzen windowing estimates the PDF k(x) from which the sample was derived. It essentially superposes kernel functions placed at each observation. In this way, each observation k(x) contributes to the PDF estimate defined as:

$$f`_k(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right) \quad (4)$$

The PDF estimation depends on the type of the kernel window used. Gaussian kernel profile is selected amongst all and it is given as:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{\left(-\frac{1}{2}x\right)} \quad (5)$$

By using this PDF estimation technique, Mean-Shift algorithm models color PDF of target object and converge towards a possible target location area with PDF estimator.

B. Target Localization

The reference target model is representated by its PDF in the selected region space. For example the reference model to be chosen to be the color PDF of the target window. And in subsequent frame target candidate is determined at location y and is characterized by PDF p(y).both PDFs are to be estimated from m-bins and histogram should be used according to color weighted histogram. Thus we have,

Target model:

$$\hat{q} = \{\hat{q}_u\}_{u....m} \qquad \sum_{u=1}^{m} \hat{q}_u = 1 \quad (6)$$

Target candidate:

$$\hat{p}_z(y) = \{\hat{p}_z(y)\} \quad \sum_{u=1}^{m} \hat{p}_u = 1 \quad (7)$$

So measure of the distance between two densities is based on the Bhattacharya Coefficient, whose general form is given by[2]:

$$\rho(y) \equiv \rho[p(y), q] = \int \sqrt{p_z(y)\, q_z}\, dz \quad (8)$$

Properties of Bhattacharya coefficient such as its relation to the fisher measure of information, quality of sample estimate, and explicit forms for various distributions such as discussed in [2].This Bhattacharya coefficient is derived from estimation of densities derived from two PDF samples p and q for which color histogram is to formulate. This density of $\hat{q}_z = \{q_u\}_{u=1}$ (with $\sum_{u=1}^{m} \hat{q}_u = 1$) is computed form m-bin histogram of the target model and target candidate is represented by $\hat{p}_z(y) = \{\hat{p}_z(y)\}$ (with $\sum_{u=1}^{m} \hat{p}_u = 1$) is estimated at given location y from the m-bin histogram of the target candidate. Therefore the sample estimate of the Bhattacharya coefficient is given by:

$$\hat{\rho}(y) \equiv \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u} \quad (9)$$

Based on equation (9) we can define distance between two estimations as

$$d(y) = \sqrt{1 - \rho[\hat{p}_u(y)\hat{q}_u]} \quad (10)$$

C. Weighted Histogram Computation

Target Feature: The pixel location of target feature is denoted by $\{x_i^*\}_{i=1....n}$ Let b:R→ {1….m} be function which associates to the pixel at the location x_i^* the index $b(x_i^*)$ of the histogram bin corresponding to the color of that pixel. The probability of the color u in target feature assigns a smaller weight to the locations that are further from the center of the target. By assuming that the generic coordinates x and y is normalized with h_x and h_y, respectively, and target feature equation in weighted histogram is given by:

$$\hat{q}_u = C \sum_{i=1}^{n} k\left(\|x^*_i\|^2\right) \delta\left[b(x^*_i) - u\right] \quad (11)$$

Where δ is Kronecker delta function and defined by equation as,

$$\delta_{xy} = \begin{cases} 0, & if \ x \ne y \\ 1, & if \ x = y \end{cases} \quad (12)$$

The normalizing constant is given by C and derived by imposing the condition by equation (6).The normalizing constant given by:

$$C = \frac{1}{\sum_{i=1}^{n_h} k\left(\|x^*_i\|^2\right)} \quad (13)$$

Candidate Feature: Candidate feature is centred at y location in current frame and denoted by $\{x_i^*\}_{i=1}$ pixel location of candidate. The probability of the colour u in target feature with same weighting function k is given by

$$p\hat{}_{u\,(y)} = C_h \sum_{i=1}^{nh} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \delta\,[b(x^*_i) - u\,] \quad (14)$$

The scale of the candidate feature is determined by the constant h which plays important role in kernel density estimation. With this scale we obtain normalization constant as

$$C_h = \frac{1}{\sum_{i=1}^{n_h} k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)} \quad (15)$$

Here C_h does not dependent on y; this can be different for given kernel with different values of h.

D. Distance Minimization

The new target location in current frame can be at location $\hat{y}_0$ of the target computed. Thus, the probabilities $\{\hat{p}_u(y_0)\}_{u=1\ldots m}$ of the candidate feature at location $\hat{y}_0$ in the current frame must be computed first.

The minimization of the distance in equation (10) is equivalent to the maximization of the Bhattacharya coefficient (9), these probabilities can be estimated with the Taylor expansion series with these values:

$$\rho[\hat{p}_{(y)}, \hat{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_{u\,(\hat{y}_0)}\,\hat{q}_u} + \frac{1}{2}\sum_{u=1}^{m}\hat{p}_{u\,(y)}\sqrt{\frac{\hat{q}_u}{\hat{p}_{u\,(\hat{y}_0)}}} \quad (16)$$

This equation to be place with the values of the target feature and candidate feature with estimated kernel densities for all values of u = 1…m then this can be given as

$$\rho[\hat{p}_{(y)}, \hat{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_{u\,(\hat{y}_0)}\,\hat{q}_u} + \frac{C_h}{2}\sum_{u=1}^{n_h} w_i\; k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \quad (17)$$

Where

$$w_i = \sum_{i=1}^{m}\delta\,[b(x^*_i) - u\,]\sqrt{\frac{\hat{q}_u}{\hat{p}_{u\,(\hat{y}_0)}}} \quad (18)$$

The second term in equation (17) represents the density estimate computed with kernel profile k(x) at y in the current frame with weighted function given in equation (18).In this procedure kernel is recursively moved from current location y_0 to y_{t+1} which yields to update its position by

$$y_{t+1} = \frac{\sum_{i=1}^{n} x_i\; w_i\; g\left(\left\|\frac{y_t - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} w_i\; g\left(\left\|\frac{y_t - x_i}{h}\right\|^2\right)} \quad (19)$$

Where g(x) = K′(x), the derivation of K(x) with respect to x.

IV. KALMAN FILTER OBJECT TRACKING

In this section, we propose dynamic scheme for Kalman filter as the elements of its state matrix are updated depending upon an evaluation quality of observation. By these mean the tracking procedure may be significantly accelerated.

Kalman filter is a state space estimation method where the dynamic linear model of the target feature is to predict and correct its position through time. This dynamic system can be distributed with noise. This Kalman filter always has prediction and correction stages with all in presence of Gaussian noise and given by following equations:

$$K(k) = P^+(k)\,H^T(k)\,[H(k)\,P^+(k)\,H^T(k) + R(k)]^{-1} \quad (20)$$
$$\hat{x}(k) = x^+(k) + K(k)[z(k) - H(k)x^+(k)] \quad (21)$$
$$\hat{P}(k) = [I - K(k)H(k)]P^+(k) \quad (22)$$
$$x^+(k+1) = \Phi(k)\hat{x}(k) \quad (23)$$
$$P^+(k+1) = \Phi(k)\hat{P}(k)\Phi^T(k) + Q(k) \quad (24)$$

Where K is Kalman gain $\hat{x}$ is state estimate, $\hat{P}$ is state covariance estimate and Φ is the state transition matrix governing propagation of the state forward in time from discrete time k to k+1.H is measurement matrix that relates measurements to the state, R is the priori measurement noise covariance matrix is the priori process noise covariance matrix and z is the measurement vector. The + sign indicates the prediction of state and covariance estimates one step forward in time.

Our object in video sequence is described by its center coordinates (x,y) and its position varies over time by equation (23). we assume with zero mean and variance 0.5 Gaussian noise with covariance matrix Q

Here state model matrix as

$$H = \begin{bmatrix}1&0&0&0\\0&1&0&0\end{bmatrix}$$

State transform matrix as

$$A = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

With these parameters the tracking of object in video sequence takes place with Kalman filter.

V. TEST DESCRIPTION

The test cases are used for evaluation of detected object in video sequences. There is one sets of video sequence to track car in video sequence. These video sequence is having 300 frames at frame rate approximately 29Hz.The video sequence are of 640 pixels by 480 pixels in dimension. In this video sequence, detected cars have their centroid. Figure 1 shows sample video sequence of few frames with a view of cars.

Fig. 1. Sample video sequence

We first compare the Mean-Shift gradient descent search with the car video sequence for selected car. For car selection, one patch is drawn on first frame of the video, and that patch is updated with consequent frames further by Mean-Shift algorithm.

In next part, same video sequence is tested for Kalman filter algorithm where we have detected first car in video sequence and keep on updating detected car position with estimated position of car.

VI. RESULT

To test our implementation we have tried many videos, out of that one common video is put up to give convincing comparison for Mean-shift and Kalman filter algorithm to track cars sucessessfully.

In our implementation car is selected with 3×2 patch for Mean-Shift algorithm and for Kalman filter whole car size in video is taken for reference including its centriods. All of the experiment is carried out on CPU Pentium IV 3.2 GHz PC with 512M memory under MATLAB.

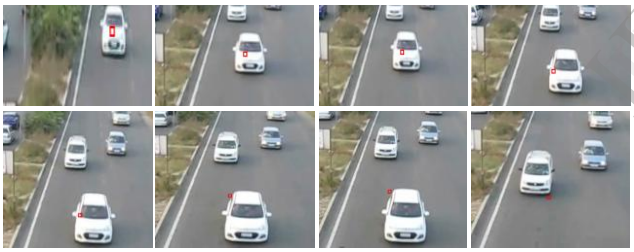

Fig. 2. Mean-Shift tracker

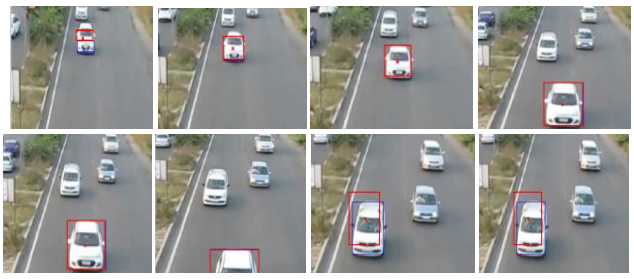

Fig. 3. Kalman filter tracker

Figure 2 illusrates the target car is selected through patch and that selected patch region is estimated with density estimation function and with the next frame according to similarity between current and next frame estimated values patch get updated in Mean-Shift algorithm.

Figure 3 illustartes target car is selected with maximun area in initial frame with centroid that is our current position in first frame which is highlited in blue rectangle.Kalman filter helps to predict position of car in next frame through gain updation given by equation (20).This predicted position is given in red rectangle in figure 3.

TABLE I. PERFORMANCE OF MEAN-SHIFT AND KALMAN FILTER ALGORITHM

Algorithm	**Performance**	
	No. of Frame tracked	***Elapsed time***
Mean-Shift	352	34.082 seconds
Kalman filter	352	35.130 seconds

VII. CONCLUSION

In this paper, we have described and evaluated the use of Kalman filter approach to improve tracking performance of detected object in video sequence. We have shown that the Mean-Shift lacks a segmentation step to separate object pixel from background pixel. Track box is not tight around the object, track lost when object is fast moving, and background changing from light to dark. Furthermore this framework naturally extended to multi-object tracking and get optimize time for object.

REFERENCES

[1] D. Comaniciu and V. Ramesh, "Mean-Shift and Optimal Prediction for Efficient Object Tracking", *Proceedings of the* IEEE Conference on Image Processing, vol. 3, pp. 70-73, Vancouver, Canada, 2000.

[2] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking",*IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-577, May 25th 2003.

[3] A. Mohan, C. Papageprgiou, and T. Poggio, "Example-Based Object Detection in Images by Components", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 349-361, April 2001.

[4] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features", Technical Report CMU-CS-91-132, April 1991.

[5] Mohinder S.Grewal,Angus P.Andrew, "Kalman Filtering:Theory and Prctise Using MATLAB",Second Edition,Published by John Wiley & sons,2001.