A robust mean shift integrating color, GLCM based texture features and frame differencing

Amit Kumar Sharma, Abhinav Malik, Rajesh Rohilla

Abstract - Mean shift is a kernel based widely used algorithm for tracking the location of object robustly. Classical mean shift uses color histogram to represent the object. However, the use of only color restricts the algorithm to track the object only in simple cases and it fails in complex situations like illumination changes, occlusion, and abrupt changes in the location of object. To improve the performance of mean shift, some authors have added some features to basic mean shift. As color based target representation is combined with texture-based target representation, based on spatial dependencies and co-occurrence distribution within interest target region for invariant target description, which is computed through so-scaled Haralick texture features, is an efficient mean shift for target tracking in some real world complex conditions. But it still fails, if there are some abrupt changes in the location of object. So a novel algorithm is presented in this work, which is robust to track the object in above mentioned complex situations. It is the combination of color and gray level co-occurrence matrix based texture features along with the use of frame differencing for abrupt motion changing target detection.

Many experimental results demonstrate the successful of target tracking using the proposed algorithm in many complex situations, where the basic mean shift tracker obviously fails. The performance of the proposed adaptive mean shift tracker is evaluated using the VISOR video Dataset, creative common dataset and also some proprietary videos.

Keywords - Visual tracking, Mean shift tracker, Color histogram, Haralick texture features, Co-occurrence matrix, Frame differencing, Abrupt motion changing object extraction.

———————————— ◆ ————————————

Amit Kumar Sharma is currently working as an Assistant Professor in Electronics & communication Engineering Department in IIMT College of engineering, Greater Noida (India) – email: amitpiyoosh@gmail.com

Abhinav Malik is currently working as an Assistant Professor in Electronics & communication Engineering Department in IIMT College of engineering, Greater Noida (India) – email: abhinav.sky@gmail.com

Rajesh Rohilla is currently working as an Associate Professor in Electronics & communication Engineering department in Delhi Technological University, Delhi (India) –email: rajesh@dce.ac.in

I. INTRODUCTION

Visual object tracking is an important task within the field of computer vision applications such as surveillance [1], [2], [3], perceptual user interface [4], pedestrian protection systems [5], smart rooms [6] and other applications. It aims at locating a moving object or several ones in time using a camera. An algorithm analyses the video frames and outputs the location of moving targets within the video frame. So it can be defined as the process of segmenting an object of interest from a video scene and keeping track of its motion, orientation, occlusion etc. in order to extract useful information by means of some algorithms. Its main task is to find and follow a moving object or several targets in image sequences.

Visual object tracking follows the segmentation step and is more or less equivalent to the "recognition" step in the image processing. Detection of moving objects in video streams is the first relevant step of information extraction in many computer vision applications. There are basically two types of approaches in visual object tracking: Deterministic methods [7] and probabilistic methods [8]. The mean shift tracking is one of the deterministic methods that has gained much our attention in past few years because its robustness and low complexity.

The Mean shift is a non-parametric density estimation method which finds the most similar distribution pattern with a simple pattern by iterative searching. Mean Shift is a powerful and versatile non parametric iterative algorithm that can be used for lot of purposes like finding modes, clustering etc. Mean Shift was introduced in Fukunaga and Hostetler [22] and has been extended to be applicable in other fields like Computer Vision.

Mean shift considers feature space as an empirical probability density function. If the input is a set of points then Mean shift considers them as sampled from the underlying probability density function. If dense regions (or clusters) are present in the feature space, then they correspond to the mode (or local maxima) of the probability density function. For each data point, Mean shift associates it with the nearby peak of the dataset's probability density function. For each data point, mean shift defines a window around it and computes the mean of the data point. Then it shifts the centre of the window to the mean and repeats the algorithm till it converges. After each iteration, we can consider that the window shifts to a denser region of the dataset.

ISSN 2229-5518

Mean shift algorithm climbs the gradient of a probability distribution to find the nearest domain mode (peak).The mean shift is applied in real-time object tracking is published in [9] named kernel based object tracking or mean shift tracking. The size and shape of the interest area is usually described by kernel function. The most used kernel function is Epanechnikov kernel [23] and its kernel profile is

$$k(x) = \begin{cases} \frac{1}{2}\, {c_d}^{-1}(d+2)(1-||x||^2)\ , & if\ ||x|| \leq 1 \\ 0 & ,otherwise \end{cases} \quad (1)$$

The biggest advantage of mean shift tracking is that the computational cost is much cheaper than other matching method because the dense gradient climbing approach is used rather than the brute-force searching approach. Therefore, it became one of the most popular object tracking algorithms for its low computational cost and robustness.

Despite the popularity of the mean shift algorithm, there are still several disadvantage of it. First, the spatial information of the object is not strongly encoded in the representation of the object, thus the scale and orientation information of the object will be lost during tracking. Second, the mean shift tracking algorithm uses a static model of the object which assumes that the object will not change its outlook much which is not true in the real environment. For example, one can easily fail a mean shift tracker by rotating the tracked object to the other side (suppose that the two sizes of the object are different which is true for most of the cases). The third and an important one is that it assumes that the object will not move more than its own size between 2 consecutive frames, thus searching window size is limited to the size of the object. This decreases the computational cost and the distraction of the background, but makes it less robust for the case of abrupt change in object motion.

In our work, we have presented a new mean shift tracking algorithm integrating texture and color features with frame differencing. The proposed frame differencing with color and texture-based target representation based on co-occurrence distribution and discriminant Haralick texture features that are more appropriate for tracking the target in complex situations as like illumination changes, occlusion, abrupt object location changes etc.

II. RELATED WORK

Comaniciu et al. [9] apply the mean shift method to feature space analysis for object tracking. The feature histogram-based target representations are regularized by spatial masking with an isotropic kernel. The masking induces spatially-smooth similarity functions suitable for gradient-based optimization. Hence, the target localization problem can be formulated using the basin of attraction of the local maxima. They employ a metric derived from the Bhattacharyya coefficient as similarity measure, and use the mean shift procedure to perform the optimization. Their semi-automatic method works by searching in each frame for the location of the target region, where the color histogram is similar to the reference color histogram of the tracked target.

Leichter et al. [10] presented an improved mean shift tracking algorithm based on multiple reference color histograms in which authors proposed to update the target model at tracking over time with multiple histograms. In contexts where multiple views of the target are available prior to the tracking, this paper enhances the Mean Shift tracker to use multiple reference histograms obtained from these different target views. This is done while preserving both the convergence and the speed properties of the original tracker. They first suggest a simple method to use multiple reference histograms for producing a single histogram that is more appropriate for tracking the target. Then, to enhance the tracking further, they propose an extension to the Mean Shift tracker where the convex hull of these histograms is used as the target model. Many experimental results demonstrate the successful tracking of targets whose visible colors change drastically and rapidly during the sequence, where the basic Mean Shift tracker obviously fails. This strategy still computationally expensive and insufficient in several real world conditions, especially if the target color is similar to the background components' color at tracking over time.

Azghani et al. [11] suggested an intelligent modified mean shift tracking algorithm using a local search based on a genetic algorithm to improve the convergence procedure. First, a background elimination method is used to eliminate the effects of the background on the target model. The mean shift procedure is applied only for one iteration to give a good approximate region of the target. In the next step, the genetic algorithm is used as a local search tool to exactly identify the target in a small window around the position obtained from the mean shift algorithm. The simulation results prove that the proposed method outperforms the traditional mean shift algorithm in finding the precise location of the target at the expense of slightly more complexity. However, their method still limited around the limitations of genetic algorithms and not robust in several conditions (e.g. light change etc.), because no information has been used to improve the target description, which is the major need of any tracker.

Ming-Yi et al. [12] presented an improved mean shift tracking algorithm using a fuzzy color histogram. A fuzzy color histogram generated by a self-constructing fuzzy cluster is proposed to reduce the interference from lighting changes for the mean shift tracking algorithm. The experimental results show that the proposed tracking approach is more robust than the conventional mean shift tacking algorithm and the cost of increasing computation time is also moderate. Although it is a good idea to avoid the pre-defined color bins. The tracking therefore may fail if the appearance of the object varies substantially.

Peng et al. [13] proposed a target model updating method in the mean shift algorithm, in which authors propose to integrate an adaptive KALMAN filter into the mean shift algorithm to update the target model and to handle temporal appearance changes. They propose a new adaptive model update mechanism for the real-time mean shift blob tracking. Kalman filter has been used for filtering object kernel histogram to obtain the optimal estimate of the object model because of its popularity in smoothing the object trajectory in the tracking system. The acceptance of the object estimate for the next frame tracking is determined by a robust criterion, i.e. the result of hypothesis testing with the samples from the filtering residuals. Therefore, the tracker can not only update object model in time but also handle severe occlusion and dramatic appearance changes to avoid over model update. They have applied the proposed method to track real object under the changes of scale and appearance with encouraging results. However, to update the target model in each frame makes the tracker computationally expensive and sensitive to occlusions and noise.

Jaideep et al. [14] proposed a robust tracking algorithm which overcomes the drawbacks of global color histogram based tracking. They incorporate tracking based only on reliable colors by separating the object from its background. A fast yet robust update model is employed to overcome illumination changes. This algorithm is computationally simple enough to be executed real time and was tested on several complex video sequences.

Object tracking based on Mean Shift (MS) algorithm [15] has been very successful and thus receives significant research interests. Unfortunately, traditional MS based tracking only utilizes the gradient of the similarity function (SF), neglecting completely higher-order information of SF. The paper regards MS based tracking as an optimization problem, and proposes to make use of both the Gradient and Hessian of SF. Specifically, They introduce Newton algorithm with constant, unit step and Newton with varying step lengths, and Trust region algorithm. The advantage of exploiting higher-order information is that higher convergence rate and better performance are achieved.

The standard mean shift algorithm assumes that the representation of tracking targets is always sufficiently discriminative enough against background. Most tracking algorithms developed based on the mean shift algorithm use only one cue (such as color) throughout their tracking process. The widely used color features are not always discriminative enough for target localization because illumination and viewpoint tend to change. Moreover, the background may be of a color similar to that of the target. Wang et al. [16] present an adaptive tracking algorithm that integrates color and shape features. Good features are selected and applied to represent the target according to the descriptive ability of these features. The proposed method has been implemented and tested in different kinds of image sequences. The experimental results demonstrate that our tracking algorithm is robust and efficient in the challenging image sequences.

[17] This paper extends the classic Mean Shift tracking algorithm by combining color and texture features. In the proposed method, firstly, both the color feature and the texture feature of the target are extracted from first frame and the histogram of each feature is computed. Then the Mean Shift algorithm is run for maximizing the similarity measure of each feature independently. In last step, center of the target in the new frame is computed through the integration of the outputs of Mean Shift. Experiments show that the proposed Mean-Shift tracking algorithm combining color and texture features provides more reliable performance than single features tracking.

In spite of all attempts [14–17] to improve the mean shift tracking algorithm, the complex conditions of the real world remain the biggest challenge, which require the use of a very powerful and rich descriptor for better target representation. Until now, the proposed improvement into the mean shift tracker remains in target description by isolated pixels such as the color histogram and texture that lacks of spatial configuration of pixels. These features are insufficient and often invalid in practice, mainly in presence of noise, clutter, illumination change, and local deformation.

Contextual information plays an important role in objects description for objects recognition [18], classification [19]. Inter-frame context matching for object tracking has been proposed recently by Ying and Fan [20], in which the authors propose a new approach to extend the optical flow to contextual flow. In their paper, the authors propose to match not only a brightness information but the visual context of each pixel, where the visual context is defined as a color context and spatial relation between the interest pixel and its neighborhood (e.g. edge direction); although that is a good idea for matching a constant and invariant pattern especially in motion estimation, but still complicated and computationally expensive, especially in real time object tracking applications.

Bousetouane F. et al. [21] believe that one way to improve the tracking in complex conditions is not by using direct information from isolated pixels as the color histogram but through increasing the level of the target description. This level can be described through the exploitation of discriminating and invariant internal targets' properties computed from local dependencies between a set of pixels within the target region, such as: local variation, degrees of texture organization, rate of homogeneity, disorder degrees, edge direction, spatial context, color context, etc. technique is effective in real world conditions but it fails if there is a sudden change in speed of object in some frames as like in the case of abrupt motion change.

III. The basic mean shift tracker

The mean shift algorithm is a simple iterative statistical method; firstly introduced by Fukunaga and Hostetler [22] for finding the nearest mode of a point sample distribution, which produced good results in many applications such as segmentation and classification. Basic kernel based object tracking was presented by Comaniciu et al. [9] , they show that show that by spatially masking the target with an isotropic kernel, a spatially-smooth similarity function can be defined and the target localization problem is then reduced to a search in the basin of attraction of this function. The similarity between the target model and the target candidates in the next frame is measured using the metric derived from the Bhattacharyya coefficient. Here the Bhattacharya coefficient has the meaning of correlation between the reference model and target model. Authors proved that the object center point in the mean shift algorithm could converge to a stable solution. Method is based principally on two steps: target appearance description using color feature and the mean shift tracking procedure to estimate the new target location. The kernel profile is often used to describe the histogram which gives higher weight to pixels near the center of the tracking window. To estimate the new state of the target, we must minimize the distance between the histogram of the reference target at time *t* and the histogram of the candidate target at time $t - 1$. To compute this distance, a popular used distance is the Bhattacharyya coefficient.

A. Target representation using histogram

M-bin RGB color histogram is used to represent the appearance of the target region. The reference target model is represented by its color probability density function $\{q_u\}_{u=1,2,...M}$ which estimates the color distribution of the target in reference frame is given by equation (2). Without the loss of generality, the target model can be considered at the spatial location 0.

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

Where $\{x_i\}_{i=1,2,.......n}$ be the normalized pixel locations in the region defined as the target model. And $k(x)$ is the epanechnikov kernel profile with property that that it is isotropic and it assigns smaller weights to pixels farther from the center and δ is the kronecker delta function. The constant C is the normalization constant, which is derived from the condition $\sum_{u=1}^{M} q_u = 1$ as

$$C = \frac{1}{\sum_{i=1}^{n} k(||x_i||^2)} \tag{3}$$

Where $b(x_i)$ denotes the index associated to the pixel location x_i in the binned feature space.

The candidate target model is represented by the probability density function $\{p_u(y)\}$, centered at y in the current frame. Using the same kernel profile and same bandwidth, the probability of the feature space is given by equation (4)

$$p_u(y) = C_h \sum_{i=1}^{n} k\left(||\frac{y-x_i}{h}||^2\right)\ \delta[b(x_i) - u] \tag{4}$$

Where the normalization constant is derived in same manner as for the $\{q_u\}$, and is represented by equation (5) and the bandwidth h defines the scale of the target.

$$C_h = \frac{1}{\sum_{i=1}^{n} k\left(||\frac{y-x_i}{h}||^2\right)} \tag{5}$$

The similarity measure, Bhattacharya coefficient, is used to measure the similarity degree between reference target histogram and candidate target histogram. The Bhattacharya distance is defined as

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

Where the Bhattacharya coefficient is

$$\rho[p(y), q] = \sum_{u=1}^{M} \sqrt{p_u(y) q_u}$$

B. Target localization: tracking procedure

For finding the new location of target in current frame, the distance in (6) should be minimized and it is equivalent to maximizing the Bhattacharya coefficient. In the current frame, the search for the new target location starts from estimated target location in the previous frame (assuming y_0). Using Taylor series expansion of the Bhattacharya coefficient around $p_u(y_0)$, the linear approximation with some manipulations is

$$\rho[p(y), q] \approx \frac{1}{2} \sum_{u=1}^{M} \sqrt{p_u(y_0) q_u} + \frac{1}{2} \sum_{u=1}^{M} p_u(y) \sqrt{\frac{q_u}{p_u(y_0)}} \tag{7}$$

The approximation is satisfactory when the target does not change drastically from the initial, which is often a valid assumption between two consecutive frames. The centre of the window is recursively moved from the current location y_0 to the new location y_1 according to the equation

$$y_1 = \frac{\sum_{i=1}^{n} x_i w_i g(||\frac{y_0 - x_i}{h}||^2)}{\sum_{i=1}^{n} w_i g(||\frac{y_0 - x_i}{h}||^2)} \tag{8}$$

Where $g(x) = -k'(x)$ and

$$w_i = \sum_{u=1}^{M} \sqrt{\frac{q_u}{p_u(y_0)}}\ \delta[b(x_i) - u]\ . \tag{9}$$

The mean shift will converge in real target location in limited iterations in the current frame.

C. Limitations of the basic mean shift tracker

Despite the popularity of the mean shift algorithm and its robustness in some conditions, this algorithm has many limitations that we can mention:

1. The spatial relation between the set of pixels within interest target region is lost.
2. When the target has a similar appearance to the background, the mean shift tracker converges to a false position.

These defects are due to the use of simple target appearance as a color histogram, which is insufficient and very sensitive to clutter interference, illumination changes, abrupt change in object's location.

IV. TARGET REPRESENTATION USING TEXTURE

As discussed in last section that the basic mean shift tracker, that uses only color as a feature, is failed in many situations like illumination changes in background, clutter interference, occlusion and abrupt location change of object. Bousetouane et. al [21] tried to make the basic mean shift tracking algorithm more robust by combining gray level co-occurrence based texture features to the color in complex real world conditions. Michaela et al. [24] proved that graylevel co-occurrence matrix (GLCM) texture features or cooccurrence distribution is one of the strong techniques used for modeling the spatial context of any area in the scene.They suggested a new texture based target representation for better and discriminant target description. Texture analysis is one of the most used techniques in several areas to describe the spatial context of pixels within a specific region at different levels. Texture analysis makes an important role in medical image analysis (such as distinguishing normal tissue from abnormal tissue, X-ray images, normal), remote sensing, surface inspection, document processing etc. Feature extraction is the first step of image texture analysis. It calculates a characteristic of a digital image able to numerically explain its texture properties. This numerically obtain data are used for texture discrimination, texture classification or object shape determination.

A. Grey Level Co-occurrence Matrix Features:

The GLCM is a well-established statistical device for extracting second order texture information from images. Second order statistics, called Gray Level Co-occurrence Matrix (GLCM). It is a way of extracting 2nd order statistical texture features. A GLCM is a matrix where the number of rows and columns is equal to the number of distinct gray levels or pixel values in the image of that surface. GLCM is a matrix that describes the frequency of one gray level appearing in a specified spatial linear relationship with another gray level within the area of investigation.

Given an image, each with an intensity, the GLCM is a tabulation of how often different combinations of gray levels co-occur in an image or image section. Texture feature calculations use the contents of the GLCM to give a measure of the variation in intensity (i.e., image texture) at the pixel of interest. Typically, the co-occurrence matrix is computed based on two parameters, which are the relative distance between the pixel pair *d* measured in pixel number and their relative orientation θ. Normally, θ is quantized in four directions (e.g., 0º, 45 º, 90 º and 135 º), even though various other combinations could be possible.

If we have an image that contains N_g gray levels from 0 to N_g–1, and if we consider *f(p,q)* is the intensity at sample p, line q of the neighborhood, then we can have the gray level co-occurrence matrix as,

$$p(i,j|\Delta x,\Delta y) = W\ Q(i,j|\Delta x,\Delta y) \tag{10}$$

Where

$$W = \frac{1}{(m-\Delta x)(n-\Delta y)}$$

and $$Q(i,j|\Delta x,\Delta y) = \sum_{p=1}^{m-\Delta x}\sum_{q=1}^{n-\Delta y} A$$

$$A = \begin{cases} 1 & if\ f(p,q) = i\ and\ f(p+\Delta x, q+\Delta y) = j \\ 0 & otherwise \end{cases}$$

Where *(m, n)* is the size of the target, *(p, q)* is the gray level target and *f* is the gray level image. To use this co-occurrence matrix, fourteen metrics have been defined by Haralick [26] , which correspond to global descriptions metrics of texture in specific areas (i.e. these metrics describe the nature of the spatial dependencies between the set of pixels which composes the target). The most useful features [25] out of fourteen features are: angular second moment (ASM), contrast, correlation, entropy, Inverse difference moment and dissimilarity.

Angular second Moment (ASM):

$$ASM = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p_{ij}^2 \tag{11}$$

It is also known as energy, uniformity, and uniformity of energy, returns the sum of squared elements in the GLCM. ASM ranges from 0.0 for an image with many classes and little clumping to 1.0 for an image with a single class.

Contrast:

$$Contrast = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-j)^2 p_{ij}^2 \tag{12}$$

It provides a measure of the intensity contrast between a pixel and its neighbour over the whole image. Contrast is also known as variance and inertia.

Correlation:

$$Corr = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{(i-\mu)*(j-\mu)*p_{ij}}{\sigma^2} \tag{13}$$

Where $\mu = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p_{ij} * i$

and $\sigma = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p_{ij} * (i-\mu)^2$

It returns a measure of how correlated a pixel is to its neighbor over the whole image.

Entropy:

$$ENT = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p_{ij} * \log(p_{ij}) \tag{14}$$

Entropy is a measure of information content. It measures the randomness of intensity distribution. It is a statistical measure of randomness that can be used to characterize the texture of the input image. The normalized GLCM is the joint probability occurrence of pixel pairs with a defined spatial relationship having gray level values of an image.

Inverse Difference Moment: periodic texture in the direction of the translation.

$$IDM = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{p_{ij}}{1+|i-j|^2} \tag{15}$$

Dissimilarity: low value characterizes the homogeneous texture of the target.

$$DISS = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p_{ij} * (i-j) \tag{16}$$

B. Target representation

Target representation with Haralick texture features, proposed by Bousetouane et. al [21], is based on co-occurrence distribution and the spatial dependencies between the set of pixels with in the interest target region. It is represented by a vector of six selected Haralick texture features along with a metric computed by the combination of six selected features. The metric calculated is invariant to rotation, translation and scale change. All the features have been calculated in previous section (from equation

(13) to equation (18)) from gray level co-occurrence matrix (GLCM) are Angular second moment (ASM), Contrast (CONT), Correlation (CORR), Entropy (ENT), Inverse difference moment (IDM), Dissimilarity respectively. The texture base target representation is denoted as

$$V(x,t) = \begin{Bmatrix} ASM(p_{ij},t) \\ CONT(p_{ij},t) \\ CORR(p_{ij},t) \\ ENT(p_{ij},t) \\ IDM(p_{ij},t) \\ DISS(p_{ij},t) \\ M(p_{ij},t) \end{Bmatrix} \quad (17)$$

$$V(x,t+1) = \begin{Bmatrix} ASM(p_{ij},t+1) \\ CONT(p_{ij},t+1) \\ CORR(p_{ij},t+1) \\ ENT(p_{ij},t+1) \\ IDM(p_{ij},t+1) \\ DISS(p_{ij},t+1) \\ M(p_{ij},t+1) \end{Bmatrix} \quad (18)$$

Where the vector $V(x,t)$ aggregates the selected Haralick texture features computed within the reference target region at time t. The target representation vector at time t+1 is computed within the candidate target region denoted by $V(x,t+1)$. In practice, vector model is the normalized with the help of mean and variance of vector as

$$V_i^* = \frac{V_i - \mu}{\sigma}, \ \text{i} = 1,2, \ldots 7 \quad (19)$$

Here μ denotes the mean and σ is the standard deviation of the feature vector.

C. Similarity measurement for texture properties

In basic mean shift, we calculate the similarity measurement between the reference target and the candidate target using Bhattacharya coefficient for the convergence of the classical mean shift tracker. Similarly in [21] Mahalanobis distance [27] has been used to estimate the similarity measurement in the target representation at time $t+1$ the vector $V(x,t+1)$ computed within the target region with reference target representation at time t the vector $V(x,t)$ for convergence of the mean shift tracker.

C.1. Mahalanobis Distance

Mahalanobis distance [28] is a distance measure introduced by P. C. Mahalanobis in 1936. It is based on correlations between variables by which different patterns can be identified and analyzed. It gauges similarity of an unknown sample set to a known one. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant. In order to use the Mahalanobis distance to classify a test point as belonging to one of N classes, one first estimates the covariance matrix of each class, usually based on samples known to belong to each class. Then, given a test sample, one computes the Mahalanobis distance to each class, and classifies the test point as belonging to that class for which the Mahalanobis distance is minimal. Mahalanobis distance and leverage are often used to detect outliers, especially in the development of linear regression models. A point that has a greater Mahalanobis distance from the rest of the sample population of points is said to have higher leverage since it has a greater influence on the slope or coefficients of the regression equation. Mahalanobis distance is also used to determine multivariate outliers. Regression techniques can be used to determine if a specific case within a sample population is an outlier via the combination of two or more variable scores.

C.2. Similarity measurement

Target representation, containing 7 so-scaled Haralick texture features, represents the texture properties of the reference target at time t with vector $V(x,t)$ and at time time $t+1$ the candidate target $V(x,t+1)$. For computing the distance between two targets representation vectors, Mahalanobis distance is given by

$$d(V_t, V_{t+1}) = \sqrt{(V_t - V_{t+1})^T S^{-1} (V_t - V_{t+1})} \quad (20)$$

Where $S = \frac{1}{n}\sum_{i=1}^{n}(V_i - \mu)(V_i - \mu)^T$

and $\mu = \frac{1}{n}\sum_{i=1}^{n} V_i$

Where μ is the mean vector and S is the covariance matrix of size n x n.

With the help of some experiments, Bousetouane et. al [21] showed that the texture vector representation of target is better than color and some other descriptors also, and this is due to the proper exploitation of invariant and discriminant internal properties of the target computed through co-occurrence distribution and Haralick texture features such as: local variation of target intensity, disorder degree, local contrast distribution, organization degree of target texture, etc. All of these properties are discriminant and specific for any objects in the scene. The discriminatory and the invariance ability of the proposed target representation based texture features can be used for target description in other conditions and configuration such as multi-target re-identification and inter-cameras matching in visual sensor network with overlapping or non-overlapping field of views.

Besides all these advantages, the algorithm [21] is unable to track the object if it changes the speed abruptly. So with the help of frame differencing for only those frames in which object's speed has been changed suddenly, we will be able to track the object. In next section we describe frame differencing in detail and the use of it in integrating with the color and texture in tracking algorithm.

V. FRAME DIFFERENCING AND TARGET EXTRACTION

Moving target detection is an image processing procedure for extracting the moving objects which have a relatively apparent movement to background in image sequences based on their characteristics of intensity, edge, texture, and so on. Its purpose is to detect and extract moving foreground targets from their static and dynamic background. The effective detection and segmentation for moving objects area are very important low-level image processing procedures. They are the ground works of post-processing for image such as target classification, identification, tracking, behavior understanding, and so on. For the moving target detection algorithms in a video surveillance system, they can be classified into two classes according to whether there are relative movement between surveillance scene and camera or not. If a camera is fixed, the camera and its surveillance scene remain relatively unchanged position, which may be called static background detection method. If camera and its surveillance scene have relative movement, which may be called moving background detection method. In static background surveillance, the size and position of background pixels will remain unchanged in a different frame of an image sequence. In general, it can employ the difference of the pixels in brightness or color to detect moving region and extract moving objects at the same position in the different frames, which is called frame differencing algorithm. The frame differencing is a particularly efficient and sensitive method for detecting grey level changes between images which are co-registered. Frame differencing algorithm is the common method for moving target detection with a static and dynamic background. The frame differencing [29] is used frame-by-frame to detect a moving object in an efficient manner. The moving object detected by frame differencing is tracked by employing an efficient tracking algorithm. The aim of using frame differencing in mean shift integrating color and texture features is to develop a robust and reliable object tracking in an efficient way to improve security services. This system is highly cost effective and can be used as a surveillance tool in various applications. It is widely used in motion detection, where a fixed or dynamic camera is used to observe dynamic events in a scene.

The task to identify moving objects in a video sequence is critical and fundamental for a general object tracking system. For this approach Frame Differencing technique [32] is applied to the consecutive frames, which identifies all the moving objects in consecutive frames. This basic technique employs the image subtraction operator [31], which takes two images (or frames) as input and produces the output. This output is simply a third image produced after subtracting the second image pixel values from the first image pixel values. This subtraction is in a single pass. The general operation performed for this purpose is given by:

$$DIFF[i,j] = I1[i,j] - I2[i,j] \qquad (21)$$

$DIFF[i,j]$ represents the difference image of two frames. It seems to be a simple process but the challenge occurs is to develop a good frame differencing algorithm for object detection. These challenges can be of any type like

- Due to change in illumination the algorithm should be robust.
- The detection of non-stationary object (like wind, snow etc.) is to be removed.

To overcome such challenges we need to pre-process the $DIFF[i,j]$ image. Pre-processing includes some mathematical morphological operations which results in an efficient difference image. $DIFF[i,j]$ image is first converted into a binary image by using binary threshold and the resultant binary image is processed by morphological operations. The algorithm for object detection is to achieve these challenges and provide a highly efficient algorithm to maintain such task of object tracking. This algorithm provides the position of the moving object.

A. *Algorithm for moving objects detection*

It is assumed few previous frames are stored in a memory buffer and the current frame in video is Fi. Algorithm can be used for static background as well as dynamic background but here we are considering only static background. For static background we can take successive two frames but we are taking three frame differences for better discrimination of object form other background surrounding things.

- Take i^{th} frame (F_i) as input.
- Take $(i\text{-}3)^{th}$ frame (F_{i-3}) from the image buffer.

This image buffer is generally a temporary buffer used to store some of previous frames for future use. Now, perform Frame Differencing Operation on the i^{th} and $(i\text{-}3)^{th}$ frame. The resultant image generated is represented as

$$DIFF_i = F_i - F_{i-3}$$

Frame Differencing performed on i^{th} and $(i\text{-}1)^{th}$ frame has limitation to detect slow moving objects. But the Frame Differencing [29] between i^{h} and $(i\text{-}3)^{th}$ frame removes the limitation to detect slow moving object, which makes it independent of speed of moving object and more reliable.
After the frame differencing operation the binary threshold operation is performed to convert difference image into a binary image with some threshold value and thus the moving object is identified with some irrelevant non-moving pixels due to flickering of camera. And some moving pixels are also there in binary image which corresponds to wind, dust, illusion etc., All these extra pixels should be removed in steps of preprocessing. The binary image (*Fbin*), in which the pixel corresponding to moving object is set to 1 and rest is treated as background which sets to 0. This threshold technique work as, a brightness threshold (T) is chosen with the $DIFF[i,j]$ to which threshold is to be applied

if $DIFF[i,j] >= T$ then
Fbin $[i,j]$ *= 1* ---*for object*
else
Fbin$[i,j]$ *= 0* ---*for background*

This assumes that the interested parts are only light objects with a dark background. But for dark object having light background we use

if $DIFF[i,j] <= T$ then
Fbin $[i,j]$ *= 1* ---*for object*
else
Fbin$[i,j]$ *= 0* ---*for background*

The threshold taken here is not fixed it can vary according to our perception. The use of threshold T is just to separate the objects' pixels from the background. Now, perform the Iterative Mathematical Morphological Operation on this binary image *Fbin*. This is to remove all the small particles present in it and ensures us that the all insignificant moving objects are removed. These small particles may come into account because of illumination changes, occlusion etc., which are irrelevant to our aim and should not be treated as moving objects. The resultant image of morphological operation is represented as *Fmor*.

B. Morphology filtering

Morphology is a branch of studying the structure of animal and plant in biology. For the sake of image analysis and identification, the basic thought of mathematics morphology is to measure and pick up corresponding figure of image by a structure element with special configuration. In general, its operations employ first Opening second Closing, so the mathematics morphology formula is

$$(A \bullet B)oB = \{[(A \ominus \mathrm{B}) \oplus] \oplus \mathrm{B}\} \ominus \mathrm{B} \tag{22}$$

Where, "$\oplus$" is Dilation operator, "$\ominus$" is Erosion operator, "o" is Opening operator, "$\bullet$" is Closing operator, B is a structure element, which is a specified. In our case we are considering disc of size 3.

In these operations, Opening operation is able to eliminate very small objects, separate conjoint objects at their fine joints, and smooth boundary of big objects, and it does not change their area of these objects obviously. Closing operation is able to fill very small apertures and clearances, joint adjacent objects, smooth their boundary, and it also does not change their area of these objects obviously. Therefore, after these morphology operations for the binary image, isolated noise dots and small regions are eliminated in background by Opening operation. Small apertures are filled and small clearances are jointed which all are caused by noises in object region. But there still may be a little white region in background and some black apertures in object region by noise affected after these operations for the binary result image. In order to eliminate these error bigger a little region comparative to very small aperture and clearance, connectivity analyzing is employed. Firstly, compute the area of every conjoint isolated black region surrounding by background pixels. When a black region area is smaller than a given threshold value, we change them into white region which are judged as object pixels. The reason is that moving target in consecutive two frames may appear overlapping, which cause that black apertures (or small region) appear in a big conjoint white region (object region). So this is able to eliminate some small region under a certain size for error judgment. Then compute the area of big conjoint white region. When a white region area is bigger than a given threshold value, it is judged as a moving region.

The next operation is to calculate the Center of Gravity (COG) of the binary objects in image *Fmor*. According to the centroid position a fixed sized rectangular box or a bounding box or perimeter is made for all the binary objects in i^{th} frame F_i. All of the centroid information is stored in a global array. By using threads and producer-consumer concept the centroid information of some objects is transferred to the object tracking module. Now, follow same procedure for other frame.

After applying moving objects detection algorithm, we have extracted all moving objects. Now the objective is to find out our target candidate in all moving objects. If the objects are more than one, then objects can be of similar type in color or the color of objects may match somewhere in background. In this situation it is difficult to find out the target candidate position in frame. Here we have presented a special algorithm in section 6(A) for target candidate extraction in the above discussed situation.

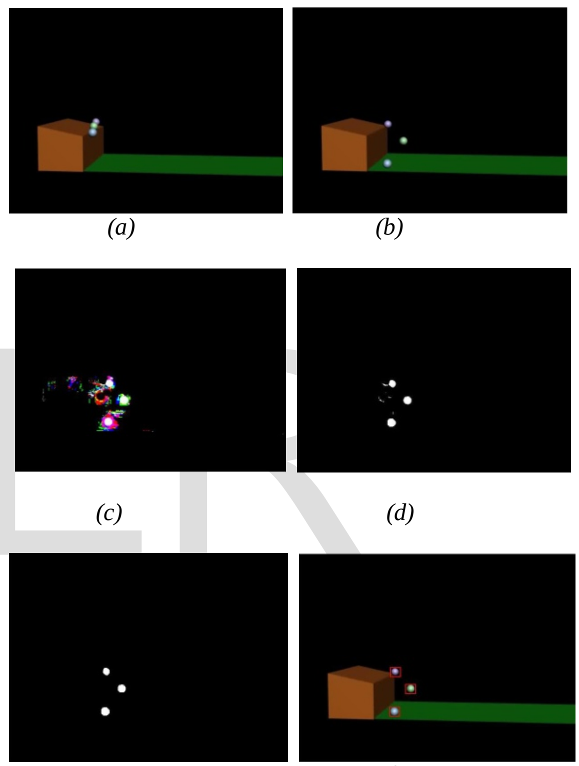


Figure 1: *Moving object extraction by frame differencing (a) Frame 20 (b) Frame 23 (c) Output of two frames difference, (d) Thresolded binary image extracted from difference image, (e) Output of morphological operations on figure (d) by disc structural element of size 3, and (f) Moving objects extracted shown with bounding box.*

VI. PROPOSED MEAN SHIFT INTEGRATING COLOR, TEXTURE AND FRAME DIFFERENCING

With the introduction of frame differencing, and the presented color and texture based target representation, a novel mean shift algorithm integrating color, texture and frame differencing is proposed to enhance the performance of the mean shift in complex real world conditions. As discussed in classical mean shift algorithm, it based on the minimization of color histogram distances of reference target histogram and candidate target histogram according to (6). The mean shift discussed in [21] is based on the minimization of combination of two distances as Bhattacharya distance for color based features of target and Mahalanobis distance for texture based features of target. Bhattacharya distance minimization results in maximization of Bhattacharya coefficient. The convergence criterion, as maximization of Bhattacharya coefficient color based features of target and minimization of color histogram distances of reference target histogram and candidate target histogram, is completely failed if there is sudden changes in speed of object i.e. object is out of window boundaries used to track using mean shift. As we know mean shift works only and only if the object is in the range of window of kernel profile. After one time loss of the object from the window due to abrupt motion change, no density of points will be changed or no mean will be shifted. Therefore tracking window is stuck at any fix place or it is dynamic in more noisy conditions as background is matching up to some extent with the object but object is nowhere in tracking window. As in surveillance case, there can be abrupt changes in objects' speed due to some reasons as like accident.

The proposed modified mean shift algorithm is divided into 7 steps as:

1. Initialization of the mean shift algorithm. In this step, the moving object extraction in the first frame must be performed.
2. Computing the color target features and the texture-based target features in the initial frame, within the extracted target region in the previous step according to (2) and (17); to do so:
   – Compute the co-occurrence matrix within this region (10).
   – Compute the proposed texture-based target representation from the co-occurrence matrix according to (18).
3. Computing the new position of target in current frame. To do so:
   – Compute p_u according to (4).
   – Compute w_i according to (9)
   – Finding the next position of the target y_1 in the current frame according to (8).
4. Computing the color feature description and texture vector at new temporary location y_1.
5. Computing the similarity between the candidate target and reference target feature (color and texture) description:
   – For color, compute the Bhattacharya coefficient using (6)
   – For texture, compute the Mahalanobis distance using (20)
6. Check the presence of the object in window. If not present, apply the algorithm in section 5.3 for target detection. Else Converge of the mean shift (finding the most suitable position for target):
   – Maximize the Bhattacharyya coefficient between color feature description of the reference target and the candidate target.
   – Minimize the Mahalanobis distance between texture vectors of the reference target and the candidate target.
7. If the location of object is same then stop, otherwise return to step 3.

Algorithm: Proposed mean shift algorithm integrating color and texture with frame differencing
Given: Frames of the input video are F_0, F_1 . . . Fn as the input image sequence for algorithm and initial location y_0 of the object.

Begin:

1. In initial frame, compute the color based target feature description q_u using (2) and texture based target feature description $V(y_0)$ at given location.
2. Assuming the location of target as y_0 in current frame, compute the color based target feature description $p_u(y_0)$ using (4), and evaluate the Bhattacharya coefficient as

$$\rho[p(y_0), q] = \sum_{u=1}^{M} \sqrt{p_u(y_0) q_u}$$

3. Derive the weights as

$$w_i = \sum_{u=1}^{M} \sqrt{\frac{q_u}{p_u(y_0)}}\, \delta[b(x_i) - u\,]$$

4. Find the next location of target candidate as

$$y_1 = \frac{\sum_{i=1}^{n} x_i w_i g(||\frac{y_0 - x_i}{h}||^2)}{\sum_{i=1}^{n} w_i g(||\frac{y_0 - x_i}{h}||^2)}$$

5. Compute the color based target feature description $p_u(y_1)$ and texture based target feature description V (y_1) at new temporary location y_1.
6. Compute $\rho[p(y_1), q]$ and $d(V(y_0), V(y_1))$.

   If $\rho[p(y_1), q]$ < Threshold && $d(V(y_0), V(y_1))$ > Threshold
   Estimate y_1 using target detection algorithm, which will be discussed in section 6(A).

   Else
   While $\rho[p(y_1), q] < \rho[p(y_0), q]$ $\wedge$ $d(V(y_0), V(y_1))$ > ε

   Do $y_1 = \frac{y_1 + y_0}{2}$

   End

7. **If** $\| y_1 - y_0 \| <= \varepsilon$ Stop
   Else $y_0 = y_1$ and go to step 3.

End

A. Algorithm for abrupt motion changing target detection

Suppose, after implying moving object extraction algorithm, we have n different objects with n different locations.

1. Compute color based target feature description for all objects using (4) as $p_u(y_i), i = 1,2, \ldots n$.
2. Compute texture based target feature description using (20) as $V(y_i), i = 1,2, \ \ldots . n$.
3. Compute Bhattacharya coefficients with reference target for all objects using (6) as

$$\rho[p(y_i), q] = \sum_{u=1}^{M} \sqrt{p_u(y_i) q_u} \ , \quad for\ i = 1, 2, \ \ . . n$$

4. Compute Mahalanobis distance with reference target for all objects using (23) as

$$d\big(V(y_0), V(y_i)\big), \quad for \ \ i = 1, 2, \ \ . . n$$

5. Find maximum of $\rho[p(y_i), q]$ as $\rho[p(y_k), q]$ and minimum of $d\big(V(y_0), V(y_i)\big)$ as $d\big(V(y_0), V(y_l)\big)$
6. If $k == l$

    Then location of target in current frame $y_1 = y_k = y_l$

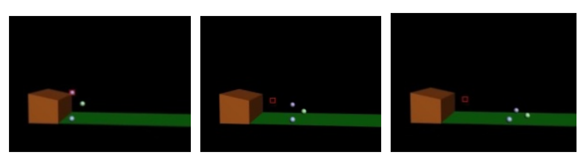
    Else $y_1 = y_k$

    So the y_1 will be the location of target in current frame.

VII. EXPERIMENTAL RESULTS

The results of the proposed mean shift algorithm integrating color and texture features with frame differencing have been presented in detail in this section. Various datasets have been used to show the performance of the proposed mean shift. Using some datasets, comparison of classical, improved by Boustaune et. al. [21] and proposed mean shift has been shown and discussed. Several video image sequences were taken from the common creative common dataset [33], VISOR dataset [34], PETS 2000 dataset [35], other benchmarks, and also some proprietary videos. These sequences consist of environments of variant light, scale change, non-rigid target motions, target occlusion. In these conditions, the proposed improved mean shift tracking algorithm can be fully evaluated. The experiments have been performed in MATLAB 2011b 64-bit on a computer (Intel(R) Core(TM) i5 2410M CPU @ 2.30GHz, RAM 4GB). In all experiments, gray image level, co-occurrence matrix, texture-based target representation, and color density are computed. Moving object detection was performed at over time in the case of abrupt motion change, using frame differencing technique and the algorithm for candidate target detection for tracking window updating. The figures in this section illustrate the tracking results using the proposed mean shift tracking algorithm. The red rectangle area on the object shows the tracked object. The next section will be reserved to detail discussions of the obtained results in different real world complex conditions. To compare the results of the proposed algorithm and the classical mean shift tracker and some other mean shift tracker also and to evaluate the performance of the proposed algorithm against complex situations, we experimented on various video sequences in different scenarios (abrupt motion change, large illumination change, complex occlusion).

Sequence 1: This sequence is presented in figure 2, where we present a set of tracking results obtained, using the proposed algorithm in comparison with classical mean shift tracking algorithm and tracking algorithm in [21] Boustaune et. al.. From this experiment using the basic mean shift tracking algorithm (in figure 2(a) and in figure 2(a)) we are seeing: when the target changes its location abruptly (frames 36, and 61 in figure 2 and frames 30, 41 and 54 in figure 3), the tracking window has a clear shift in comparison with the real object center (i.e. object position). This problem expresses incapability of the classical mean shift tracker as well as tracker presented in [21]. These limitations are caused because of the non consideration of sudden changes in objects location. When we use the proposed modified mean shift tracking algorithm (Figure 2 and figure 3), the red rectangle could track exactly the moving object, whatever the speed of the target (frames 36 and 61 in figure 2(b) and frames 30, 41 and 54 in figure 3(b)) is. Some problems of the classical mean shift tracker are solved by the proposed algorithm, because the proposed mean shift algorithm uses the invariant properties of the target through the proposed combination of color density and texture-based target representation. The proposed combination is more descriptive and discriminative than the classical mean shift using color only and integration of color and texture but without the frame differencing mean shift tracking algorithm. As mentioned earlier, the proposed technique for updating the tracking window of the proposed algorithm at tracking over time does not affect the convergence ability of the proposed algorithm.



Frame_23 *Frame_36* *Frame_61*

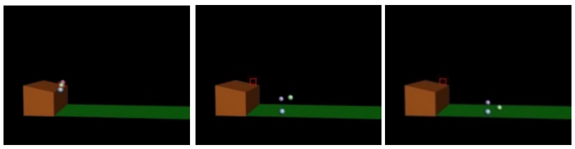
(a) Results of classical mean shift and Improved mean shift by Bousetouane F. et. al. [21] for Ball 1

Frame_23 *Frame_36* *Frame_61*

(b) Results of Proposed mean shift for the same abrupt motion change video for Ball 1.

Figure 2: comparison of the proposed mean shift tracker (b), and classical mean shift as well as improved mean shift [21] (a) for Ball 1

Sequence 2: This sequence presents a comparison between tracking results of classical mean shift algorithm and and proposed mean shift algorithm in illumination changes. Sequence contains moving objects as pedestrians in light and shade changing condition in outdoor environment. Here object color as well as background color is matching approximately due to the lightening condition. In this case



Frame 20 Frame 30 Frame 41

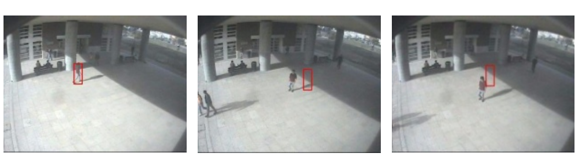
(a) Results of classical mean shift and Improved mean shift by Bousetouane F. et. al. [21] are same for this abrupt motion change video for Ball 2.

Frame 20 Frame 30 Frame 41

(b) Results of proposed mean shift tracker for Ball 2

Figure 3: comparison of the proposed mean shift tracker (b), and classical mean shift as well as improved mean shift [21] (a) for Ball 2

only color based mean shift tracking algorithm fails but texture as different feature have been used in our algorithm, so it efficient as shown in results. From the obtained results using the classical mean shift (Figure 4(a)), we can see a clear shift in the tracking window compared with the real target position. When we use our modified mean shift tracker, the tracking window (Figure 4(b)) tracks exactly the real position of the target. From this experiment, we may conclude that improving target representation using texture is showing better results.



Frame 14 Frame 42 Frame 47

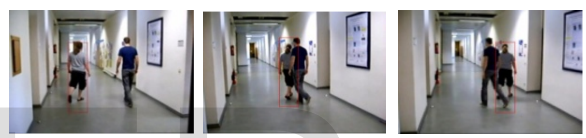
(a) Tracking results using classical mean shift tracker

Frame 14 Frame 42 Frame 47

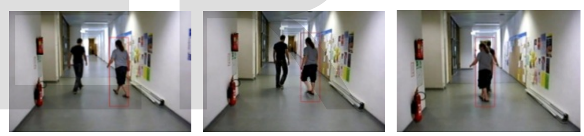
(b) Tracking results using proposed mean shift tracker

*Figure 4**:** comparison between the proposed modified mean shift tracker (b) and the classical mean shift tracker (a)*

Sequence 3: For performance evaluation of proposed mean shift algorithm a very complex background common creative dataset has been used. In this video sequence target faces a large illumination changes in background and complex occlusion as well. This sequence consists of walking objects, where the tracking remains the biggest challenge. The obtained tracking results (Figure 5) using our algorithm are very fostering, although we have used only a gray level histogram and the texture description as a target model.



Frame 71 Frame 114 Frame 131



Frame 267 Frame 277 Frame297

Figure 5: Tracking results using proposed robust mean shift tracking algorithm in presence of large illumination change and complex occlusion.

Sequence 4: This sequence proposed by VISOR dataset [34], which consists of a walking pedestrian target in the presence of complex occlusions almost complete and complex non-rigid target motions in Figure 6. From the tracking results shown in (Figure 6), we can see that the proposed algorithm can track the target perfectly when the occlusion is rather complex (Frames 195 and 226) and when the target is under occlusions. From the obtained results of this sequence, we find that the proposed improvement into mean shift is invariant in several real world complex situations (target rotating, complex occlusion etc.), and this due to the use of color description enhanced by discriminant features texture-based target representation. The proposed technique for updating the tracking window size has proved its effectiveness and has given more simulation of the complex non-rigid target motions.
Sequence 5: The proposed tracking algorithm was finally tested on the creative common dataset [33] (Figure 7). This sequence consists of a moving pedestrian, where objects

faces occlusion by completely same color another object; in this condition, the tracking remains the biggest challenge, because other object overlaps target candidate with same color. From the obtained results (Figure 7), the proposed algorithm tracks perfectly the target whatever the color of this target, and this due to the proposed technique for adapting the research space at over time with the real target mask and the invariance ability of the texture based target representation.

Frame 42 *Frame 85* *Frame 97*

Frame 124 *Frame 195* *Frame 226*

Figure 11: Tracking results using our mean shift algorithm in case of complex occlusion but colors are different

Frame 50 *Frame 72*

Frame 92 *Frame 110*

Figure 12: Tracking results using our mean shift algorithm in case of complex occlusion when the colors are exactly same

More experiments have been performed but not shown here because of the space availability. But due to unavailability of objects' sudden motion change videos, the video sequence 1 has been experimented for all balls. The proposed modified mean shift tracker was experimentally verified in the situation especially in abrupt motion change and complex conditions and scenarios, target in darkness, complex occlusions, color target and direction changes rapidly, very textured background, and also using other capturing modality. For these conditions and scenarios, the classical mean shift tracker and many proposed improvements are obviously insufficient and inappropriate. From these results, we conclude that the proposed frame differencing with integration of texture based target representation and color histogram exploits very well abrupt location change and the internal properties (i.e. invariants texture features computed through co-occurrence distribution and Haralick texture features) of the target and help color-based tracking for the best convergence.

VIII. CONCLUSION

Despite the improved mean shift algorithms presented by many researchers, proved to be robust in many tracking scenarios. However, there are many cases where the target location not defined when it is out of bound of window, or target represented by isolated pixels such as color histogram and texture that lacks of spatial dependencies between pixels, is still insufficient such as light change, non-homogeneous target, textured background, or the target location changes abruptly etc. we have presented a new mean shift tracking algorithm, In this paper, integrating texture and color features with frame differencing. The proposed frame differencing, that is able to find the location of target in case when it is lost due to abrupt motion change, with color and texture-based target representation based on co-occurrence distribution and discriminant Haralick texture features that are more appropriate for tracking the target in complex situations. Many experimental results and data show the effectiveness and the satisfaction of the proposed algorithm even in other capturing modalities. The proposed frame differencing with texture-based target representation and its combination with color distribution makes the classical mean shift more consistent and robust against very complex conditions. However, the proposed algorithm still has some limitations especially when the moving object is too small to be detected or the objects have same size and same color and texture as well. In other case if there is sudden change in location of object and background is dynamic as well, then frame differencing used in this technique fails because moving background is also treated as objects. These limitations can be solved by improving object's location finding algorithm and adding other information to the interested moving object such as a high-level spatial configuration of the tracked target. So, whatever, it would have its own texture model and spatial information that is different from the background and each other's objects. By considering these limitations and implementing some improvements to our algorithm, including the speed up of the processing time, could lead to some improvements in any tracking system. Future work includes making the object extraction algorithm using frame differencing more robust and exploring new criteria to measure the reliability of each feature and extending this framework to multiple object tracking.

REFERENCES

[1]. V. Kettnaker and R. Zabih, “Bayesian Multi-Camera Surveillance,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 253-259, 1999.
[2]. R. Collins, A. Lipton, H. Fujiyoshi, and T. Kanade, “Algorithms for Cooperative ultisensor Surveillance,” Proc. IEEE, vol. 89, no. 10, pp. 1456-1477, 2001
[3]. M. Greiffenhagen, D. Comaniciu, H. Niemann, and V. Ramesh, “Design, Analysis and Engineering of Video Monitoring Systems: An Approach and a Case Study,” Proc. IEEE, vol. 89, no. 10, pp. 1498-1517, 2001.
[4]. G.R. Bradski, “Computer Vision Face Tracking as a Component of a Perceptual User Interface,” Proc. IEEE Workshop Applications of Computer Vision, pp. 214-219, Oct. 1998.
[5]. Bousetouane, F., Dib, L., Snoussi, H.: Robust detection and tracking pedestrian object for real time surveillance applications. Proc. SPIE 8285, 828508 (2011). doi:10.1117/12.913034
[6]. S. Intille, J. Davis, and A. Bobick, “Real-Time Closed-World Tracking,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 697-703, 1997.
[7]. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surv. 38(4), 1–45 (2006)
[8]. Zhang, C., Eggert, J.: A probabilistic method for hierarchical 2D- 3D tracking. In: Proc. IEEE/IJCNN 2010, pp. 1–8 (2010)
[9]. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. 25(5), 564–577 (2003)
[10]. Leichter, I., Lindenbaum, M., Rivlin, E.: Mean shift tracking with multiple reference color histograms. Comput. Vis. Image Underst. (CVIU) 114(3), 400–408 (2010)
[11]. Azghani, M., Aghagolzadeh, A., Ghaemi, S., Kouzehgar, M.: Intelligent modified mean shift tracking using genetic algorithm. In: Proc. 5th International Symposium on Telecommunications, pp. 806–811 (2010)
[12]. Ju, M.-Y., Ouyang, C.-S., Chang, H.-S.: Mean shift tracking using fuzzy color histogram. In: Proc. International Conference on Machine Learning and Cybernetics, pp. 2904–2908 (2010)
[13]. Peng, N., Yang, J., Liu, Z.: Mean shift blob tracking with kernel histogram filtering and hypothesis testing. Pattern Recognit. Lett. 26, 605–614 (2005)
[14]. Jeyakar, J., Babu, R.V., Ramakrishnan, R.K.: Robust object tracking using kernels and background information. In: Proc. IEEE International Conference on Image Processing, pp. 49–52 (2007)
[15]. Xiao, L., Li, P.: Improvement on mean shift based tracking using second-order information. In: Proc. IEEE 19th International Conference on Pattern Recognition, pp. 1–4 (2008)
[16]. Wang, J., Yagi, Y.: Integrating shape and color features for adaptive real-time object tracking. In: Proc. IEEE International Conference on Robotics and Biomimetics, pp. 1–6 (2006)
[17]. Xiang, Z., Dai, Y.M., Chen, Z.W., Zhang, H.X.: An improved Mean Shift tracking algorithm based on color and texture feature. In: Proc. Wavelet Analysis and Pattern Recognition, pp. 38–43 (2010)
[18]. Torralba, A., Oliva, A.: The role of context in object recognition. Trends Cogn. Sci. 11(12), 520–522 (2007)
[19]. Qi, G.-J., Hua, X.-S., Rui, Y., Tang, J., Zhang, H.-J.: Image classification with Kernelized spatial-context. IEEE Trans. Multimed. 12(4), 278–287 (2010)
[20]. Wu, Y., Fan, J.: Contextual flow. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (2009)
[21]. Bousetouane F., Dib L., Snoussi H.: Improved mean shift integrating texture and color features for robust real time object tracking. Springer-Verlag ,Vis comput (2012)
[22]. Fukunaga and Hostetler, "The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition", IEEE Transactions on Information Theory vol 21 , pp 32-40 ,1975
[23]. D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, May 2002
[24]. Benz, U.C., Hofmann, P., Willhauck, G., Lingenfelder, I., Heynen, M.: Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. ISPRS J. Photogramm. Remote Sens. **58**(3/4), 239–258 (2004)
[25]. Haralick, R.M., Shanmugam, K.: Computer classification of reservoir sandstones. IEEE Trans. Inf. Theory **11**(4), 171–177 (1973
[26]. Haralick Robert M., Shanmugam K. and Dinstein I.: “Textural features for image classification”. IEEE transactions on systems, man and cybernatics, Vol. SMC - 3, Nov. 1973
[27]. Mao, K.Z., Wenyin, T.: Recursive Mahalanobis separability measure for gene subset selection. IEEE/ACM Trans. Comput. Biol. Bioinform. **8**(1), 266–272 (2011)
[28]. Wikipedia.org
[29]. Prabhakar N., Vaithiyanathan V., Sharma P. A., Singh A. and Singhal P. : “Object Tracking Using Frame Differencing and Template Matching”, Research Journal of Applied Sciences, Engineering and Technology 4(24): 5497-5501, 2012, ISSN: 2040-7467
[30]. Weng M., Huang G., and Da X. : “A New Interframe Difference Algorithm for Moving Target Detection”, 3rd International Conference on Image and Signal Processing (CISP2010), IEEE
[31]. Rafael, C.G. and E.W. Richard, 2002. Digital Image Processing. 2nd Edn., Prentice Hall International, UK.
[32]. Jain, R. and H. Nagel, 1979. On the analysis of accumulative difference pictures from image sequences of real world scenes. IEEE T. Pattern Anal., 1(2): 206-214.
[33]. Creative common dataset: http://creativecommons.org
[34].Video Surveillance Online Repository, VISOR Dataset Collection. http://imagelab.ing.unimore.it/visor/
[35].PETS 2000 dataset: http://ftp.pets.rdg.ac.uk/PETS2000