

# **1. 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.

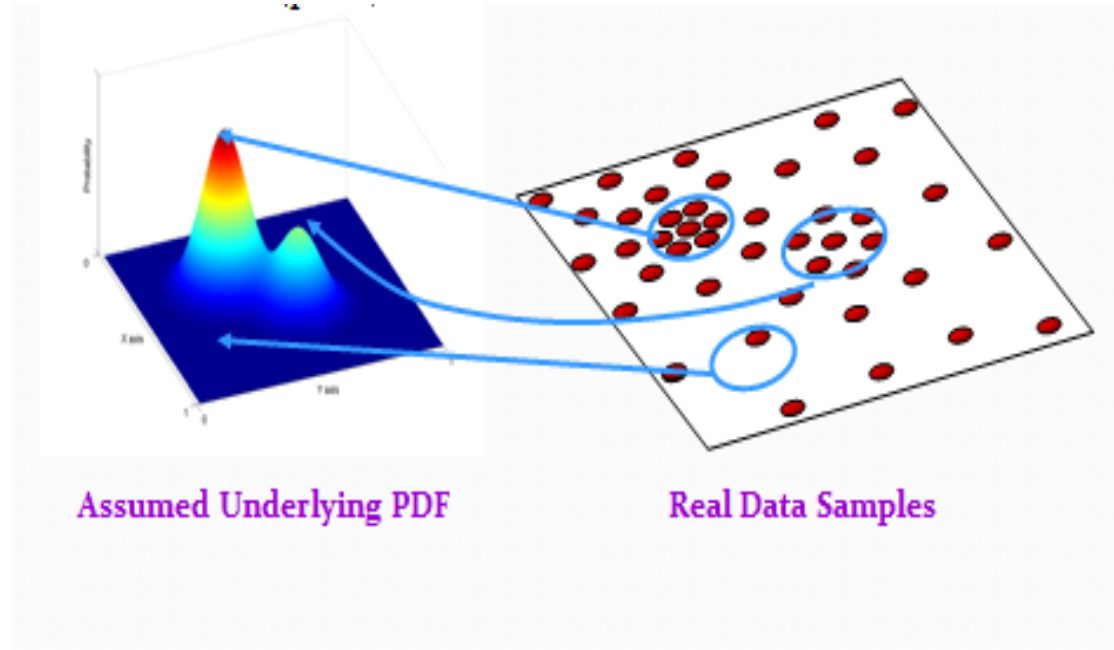
Visual object tracking follows the segmentation step and is more or less equivalent to the "recognition" step in the image processing. Detection of moving objects in video streams is the first relevant step of information extraction in many computer vision applications. There are basically two types 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.

Mean shift algorithm climbs the gradient of a probability distribution to find the nearest domain mode (peak)



**Figure 1:** Gradients of probability density function corresponding to the densities of points.

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

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

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 disadvantages 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.

## **2. 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 tracking 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.



### **3. 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 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 Bhattacharyya 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.

#### **3.1 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  $\{q_u\}_{u=1,2,\dots,M}$  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.

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

Where  $\{x_i\}_{i=1,2,\dots,n}$  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  $\delta$  is the kronecker delta function. The constant  $C$  is the normalization constant, which is derived from the condition

$$\sum_{u=1}^M q_u = 1 \text{ as}$$

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

And  $b(x_i)$  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)

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

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.

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

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

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

Where the Bhattacharya coefficient is

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

### 3.2 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

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

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

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

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

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

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

### 3.3 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.

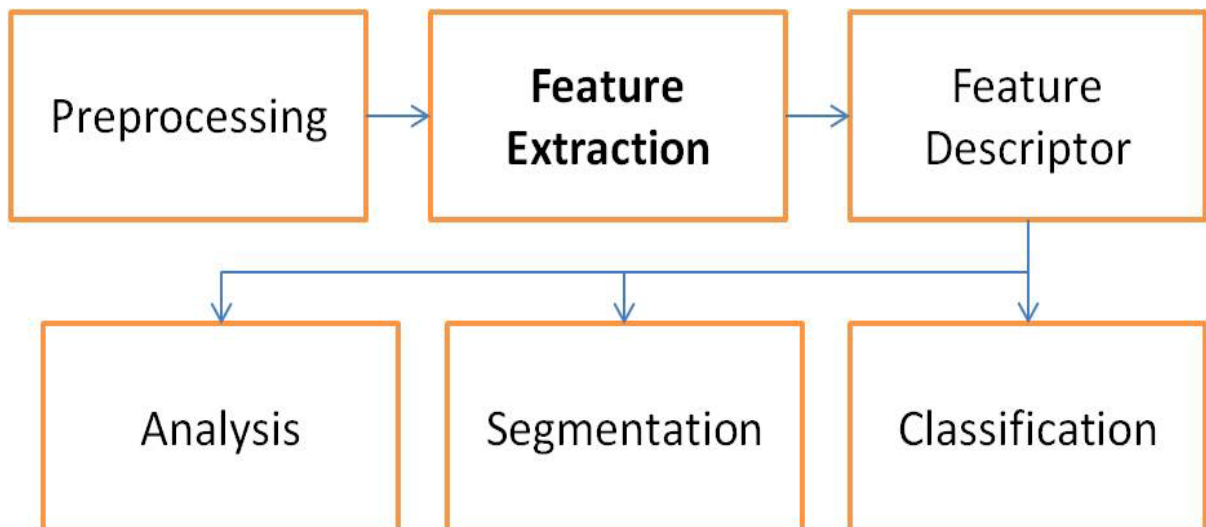
These defects are due to the use of simple target appearance as a color histogram, which is insufficient and very sensitive to clutter interference, illumination changes, abrupt change in object's location.

## 4. 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. 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.

### 4.1 Texture analysis:

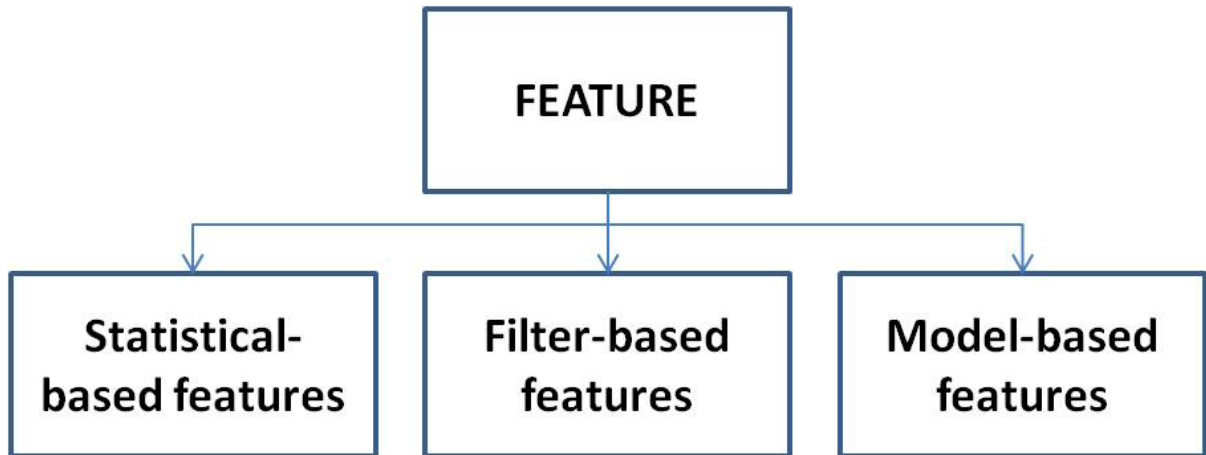
Texture analysis [24] is very important for machine vision system. It has no specific definition to identify. Vision researchers define texture definition in different way. Texture analysis makes an important role in medical image analysis (such as distinguishing normal tissue from abnormal tissue, X-ray images, normal), remote sensing, surface inspection, document processing etc.



**Figure2:** A simple flow-diagram that depicts the position of feature extraction for various applications

For texture analysis we have to consider step by step process like feature extraction, texture discrimination, texture classification and shape from texture. Feature extraction is

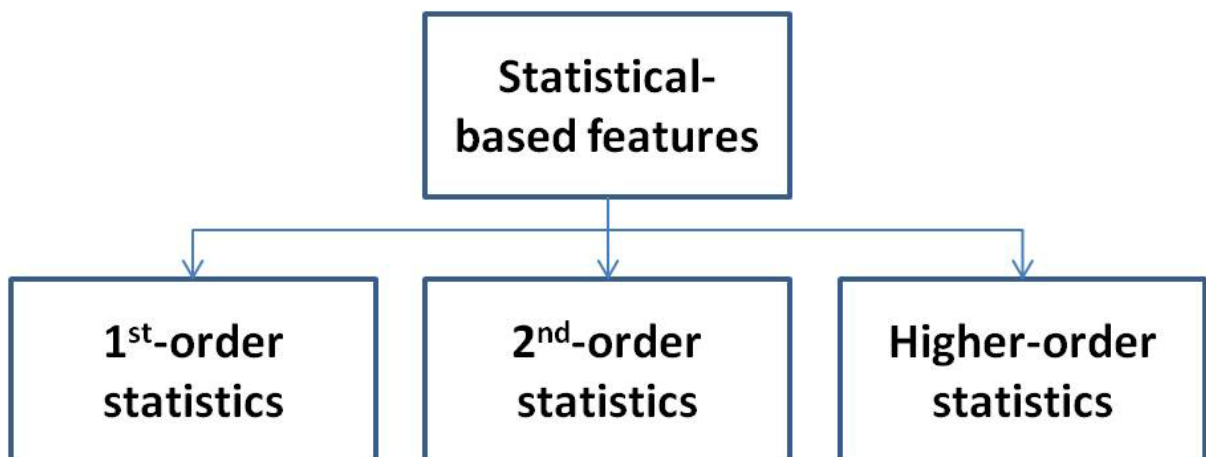
the first step of image texture analysis. It calculates a characteristic of a digital image able to numerically explain its texture properties. This numerically obtain data are used for texture discrimination, texture classification or object shape determination.



*Figure 3: Classification of features into three areas*

#### 4.1.1 Statistical methods:

Statistical-based features can be arranged in three different types, namely – 1st-order statistics, 2nd-order statistics and higher-order statistics (Fig. 3). In this section, we present key statistically-produced features.



*Figure 4: Statistical-based features.*

##### 4.1.1.1 First Order Histogram Based Features:

First Order histogram provides different statistical properties such as 4 statistical moments of the intensity histogram of an image. They depend only on individual pixel values and not on the interaction or co-occurrence of neighbouring pixel values. Four first order

histogram statistics are Mean, Variance, Skewness and Kurtosis. A histogram  $h$  for a gray-scale image  $I$  with intensity values in the range  $I(x, y) \in [0, K-1]$  would contain exactly  $K$  entries, where for a typical 8-bit gray scale image,  $K = 2^8 = 256$ . Each individual histogram entry is defined as,  $h(i) =$  the number of pixels in  $I$  with the intensity value  $I$  for all  $0 \leq i < K$ . We can redefine histogram as,

$$h(i) = \text{card}\{(x, y) | f(x, y) = i\} \quad (10)$$

Where  $\text{card}\{\dots\}$  denotes the number of elements (“cardinality”) in a set. The standard deviation (s.d.) and skewness of the intensity histogram are defined as:

$$s. d. = \sqrt{\frac{\sum (f(x,y) - m)^2}{N}} \quad (11)$$

$$skewness = \frac{\sum (f(x,y) - m)^3}{N(s.d.)^3} \quad (12)$$

#### 4.1.1.2. Second Order 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  $\theta$ . Normally,  $\theta$  is quantized in

four directions (e.g., 0°, 45 °, 90 ° and 135 °), 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,

$$p(i,j|\Delta x, \Delta y) = W Q(i,j|\Delta x, \Delta y) \quad (13)$$

Where

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

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

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):*

$$ASM = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{ij}^2 \quad (14)$$

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:*

$$Contrast = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - j)^2 p_{ij}^2 \quad (15)$$

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:*

$$Corr = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{(i-\mu)*(j-\mu)*p_{ij}}{\sigma^2} \quad (16)$$

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:*

$$ENT = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{ij} * \log(p_{ij}) \quad (17)$$

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.

$$IDM = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{p_{ij}}{1+|i-j|^2} \quad (18)$$

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

$$DISS = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_{ij} * (i - j) \quad (19)$$

#### 4.1.1.3 Gray Level Run Length Matrix Features:

The grey level run-length matrix (RLM) is defined as the numbers of runs with pixels of gray level  $i$  and run length  $j$  for a given direction. GLRLM generate for each sample image fragment. A set of consecutive pixels with the same gray level is called a gray level run. The number of pixels in a run is the run-length. In order to extract texture features gray



level run length matrix (GLRLM) are computed. Each element,  $ij$  of the GLRLM represents the number of runs of gray level  $i$  having length  $j$ . GLRLM can be computed for any direction. Five features derived from the GLRLM. These features are: Short Runs Emphasis (SRE), Long Runs Emphasis (LRE), Grey Level Non-Uniformity (GLNU), Run Length Non-Uniformity (RLNU), and Run Percentage (RPERC). They are quite improved in representing binary textures.

#### **4.1.1.4. Local Binary Pattern (LBP) Features:**

Local binary pattern (LBP) operator is introduced as a complementary measure for local image contrast. The LBP operator associate statistical and structural texture analysis. LBP describes the texture with smallest primitives called *textons* (or, histograms of texture elements). For each pixel in an image, a binary code is produced by thresholding its neighbourhood (for instance, the closest 8 pixels) with the value of the center pixel. A histogram is then assemble to collect up the occurrences of different binary codes representing different types of curved edges, spots, flat areas, etc. This histogram is arrangement as the feature vector result of applying the LBP operator. The LBP operator considers only the eight nearest neighbours of each pixel and it is rotation variant, but invariant to monotonic changes in gray-scale can be applied. The dimensionality of the LBP feature distribution can be calculated according to the number of neighbours used. LBP is one of the most used approaches in practical applications, as it has the advantage of simple implementation and fast performance.

#### **4.1.1.5. Autocorrelation Features:**

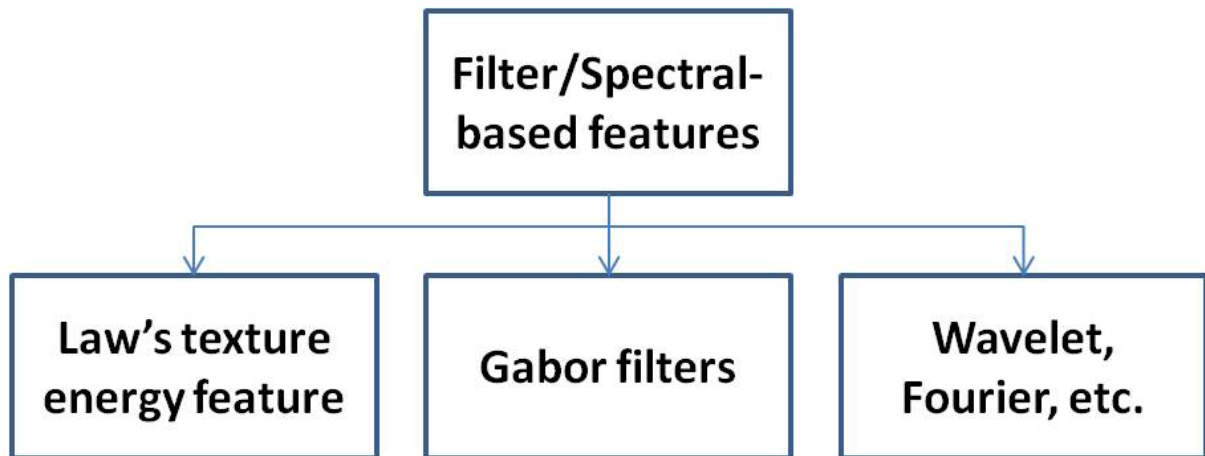
An important characteristic of texture is the repetitive nature of the position of texture elements in the image. Based on observation the autocorrelation feature is computed that some textures are repetitive in nature, such as textiles. The autocorrelation feature of an image is used to evaluate the fineness or roughness of the texture present in the image. This function is related to the size of the texture primitive for example the fitness of the texture. If the texture is rough or unsmooth, then the autocorrelation function will go down slowly, if not it will go down very quickly. For normal textures, the autocorrelation

function will show peaks and valleys. It has relationship with power spectrum of the Fourier transform. It is also responsive to noise interference.

#### 4.1.2 Filter based methods:

##### 4.1.2.1. Law's Texture Energy Features:

Laws presented his novel texture energy approach to texture analysis. According to the method proposed by Laws, textural features that certain operators such as Laplacian and Sobel operators accentuated the underlying microstructure of texture within an image. This is the basis for a feature extraction scheme based a series of pixel impulse response arrays obtained from combinations of 1-D vectors.



*Figure 5: Filter or spectral-based features.*

The Law's masks are constructed by convolving together just 3 basic 1x3 masks:

$$L3 = [1 \ 2 \ 1]$$

$$E3 = [-1 \ 2 \ -1]$$

$$S3 = [-1 \ 2 \ -1]$$

The initial of these masks indicate local averaging, Edge detection, and Spot detection.

These basic masks span the entire 1x3 subspace and form a complete set. Similarly, the 1x5 masks obtained by convolving pairs of these 1x3 masks together from a complete set:

$$L5 = [1 \ 4 \ 6 \ 4 \ 1]$$

$$E5 = [-1 \ -2 \ 0 \ 2 \ 1]$$

$$S5 = [-1 \ 0 \ 2 \ 0 \ -1]$$

$$R5 = [1 \ -4 \ 6 \ -4 \ 1]$$

$$W5 = [-1 \ 2 \ 0 \ -2 \ 1]$$

In principle, nine masks can be formed in this way, but only five of them are distinct. Here the initial letters are as before with the addition of Ripple detection and Wave detection. These masks are subsequently convolved with a texture field to highlight its microstructure giving an image from which the energy of the microstructure arrays is measured together with other statistics.

#### 4.1.2.2. Gabor Filter-based Texture Features:

The 2-D Gabor filters have been proved to be an important tool in texture analysis. They consist of a sinusoidal plane wave of some frequency and orientation modulated by Gaussian envelope. A Gabor filter is a band-pass filter which can be used to extract a specific band of frequency components from an image. Gabor functions appear to share many properties with the human visual system. Bank of Gabor filters are used to extract local image features. An image is convolved with a 2D Gabor function to obtain a Gabor feature image, and by varying spatial frequency and orientations, a bank of different Gabor filters can be produced. There are various Gabor features that can be exploited for feature analysis, e.g., linear Gabor features (symmetric, anti-symmetric), thresholded Gabor features (symmetric, anti-symmetric), Gabor-energy features, etc. An input image  $f(x, y)$ ,  $x, y \in \Omega$  ( $\Omega$  - the set of image points), is convolved with two-dimensional Gabor function  $g(x, y)$ ,  $x, y \in \Omega$ , to obtain a Gabor feature image  $r(x, y)$  as follows:

$$r(x, y) = \iint_{\omega} f(\alpha, \beta) g(x - \alpha, y - \beta) d\alpha d\beta \quad (20)$$

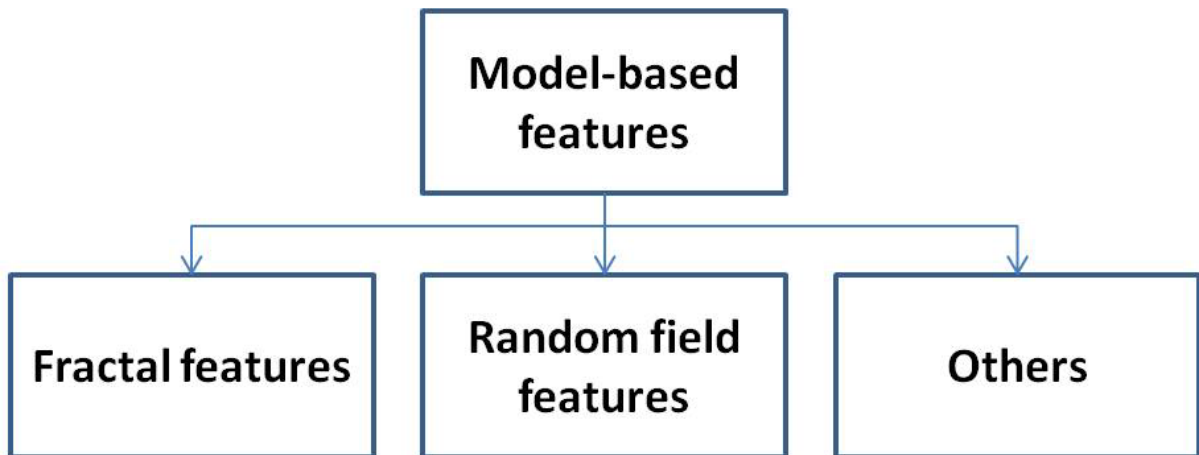
The filter responses that result from the application of a filter bank of Gabor filters can be used directly as texture features, though none of the approaches described in the literature employs such texture features.

#### 4.1.2.3. Wavelet-based Feature:

Some transforms are developed to characterize localized frequencies specific to each pixel location based on various wavelet transform methods. Wavelet transform characterizes multi scale frequency content, called, wavelet coefficients, at each spatial location of an image. It can decompose image texture scale, i.e., it can characterize textures at multiple scales. The multi-resolution properties of Wavelet transforms are beneficial for

accomplishing segmentation, classification and subtle discrimination of texture. However, the Wavelet transforms are usually computationally taxing. A tree-structured wavelet transform is proposed for texture analysis and classification. A simple Discrete Wavelet Transform representation can capture small differences in rotation or scale. Some other representations on Wavelet transform can be Discrete Wavelet Frame Transform (DWFT), Discrete Wavelet Packet Transform (DWPT), Modulus Maxima of a Continuous Wavelet Transform (MMWT), Multi-Orientation Wavelet Pyramid, Dual-Tree Complex Wavelet Transform, or scale, rotation, and translation invariant wavelets.

#### 4.1.3. Model based methods:



*Figure 6: Model Based Features*

##### 4.1.3.1. Random Field Feature:

In some cases, image textures are modeled as a Markov random field of pixels gray level. In this approach, relationships between the gray level of neighboring pixels are statistically characterized. For example, a Gauss-Markov random field-based probabilistic texture model is developed to characterize hyper spectral textures. Another similar model is proposed for texture classification and texture segmentation. For Markov random field (MRF) models, the sufficient statistics of each sub-window can be considered as the feature vector. There are few other approaches based on random fields. However, usually it is noticed that feature-based approaches are less computationally-expensive as well as more effective than Markov random field-based approaches.

#### **4.1.3.2 Fractal Feature:**

Fractal features can be considered to develop discriminative and invariant features for texture classification, especially in cases where scale changes are prominent in textures. Such variations can be handled by choosing local interest points and selecting their characteristic scales. But, apart from the problems related to have suitable local interest points, it is difficult to have an optimized characteristic scale that can be suitable for different textures of varied scales. It is found that fractal features demonstrate promising results while handling scaling constraint. However, fractal-based classification approaches do not consider statistical characterizations of textures by using global features. Moreover, they cannot differentiate between key texture elements (e.g., edges, corners, etc.). These issues can be overcome by evaluating densely local fractal features. It is shown that the local fractal dimension is invariant to local bi-Lipschitz transformations whereas its extension is able to correctly distinguish between fundamental texture primitives. Fractal models can classify irregular textures too. A multi-fractal analysis is also proposed for texture descriptor.

#### **4.2 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 within 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

$$V(x, t) = \left\{ \begin{array}{l} ASM(p_{ij}, t) \\ CONT(p_{ij}, t) \\ CORR(p_{ij}, t) \\ ENT(p_{ij}, t) \\ IDM(p_{ij}, t) \\ DISS(p_{ij}, t) \\ M(p_{ij}, t) \end{array} \right\} \quad (20)$$

$$V(x, t + 1) = \left\{ \begin{array}{l} ASM(p_{ij}, t + 1) \\ CONT(p_{ij}, t + 1) \\ CORR(p_{ij}, t + 1) \\ ENT(p_{ij}, t + 1) \\ IDM(p_{ij}, t + 1) \\ DISS(p_{ij}, t + 1) \\ M(p_{ij}, t + 1) \end{array} \right\} \quad (21)$$

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

$$V_i^* = \frac{V_i - \mu}{\sigma}, \quad i = 1, 2, \dots, 7 \quad (22)$$

Here  $\mu$  denotes the mean and  $\sigma$  is the standard deviation of the feature vector.

### 4.3 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.

#### 4.3.1. 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.

#### 4.3.2 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  $t + 1$  the candidate target  $V(x, t + 1)$ . For computing the distance between two targets representation vectors, Mahalanobis distance is given by

$$d(V_t, V_{t+1}) = \sqrt{(V_t - V_{t+1})^T S^{-1} (V_t - V_{t+1})} \quad (23)$$

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  $\mu$  is the mean vector and  $S$  is the covariance matrix of size  $n \times n$ .

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.



## **5. 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. Kettner 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 multisensor 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]. Shahera Hossain, Seiichi Serikawa: "Features for Texture Analysis". In: *Proc. SICE Annual Conference 2012*

- [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., Vaithyanathan 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>