Scientific Research Publishing

Circuits and Systems, 2016, 7, 3565-3588
http://www.scirp.org/journal/cs
ISSN Online: 2153-1293
ISSN Print: 2153-1285

Multi-Dimension Support Vector Machine Based Crowd Detection and Localisation Framework for Varying Video Sequences

Manoharan Mahalakshmi, Radhakrishnan Kanthavel, Divakaran Thilagavathy Dinesh

Velammal Engineering College, Anna University, Chennai, India
Email: strimaha@gmail.com

How to cite this paper: Mahalakshmi, M., Kanthavel, R. and Dinesh, D.T. (2016) Multi-Dimension Support Vector Machine Based Crowd Detection and Localisation Framework for Varying Video Sequences. *Circuits and Systems*, **7**, 3565-3588.
http://dx.doi.org/10.4236/cs.2016.711303

Received: May 10, 2016
Accepted: May 27, 2016
Published: September 8, 2016

Abstract

In this paper, we propose a novel method for anomalous crowd behaviour detection and localization with divergent centers in intelligent video sequence through multiple SVM (support vector machines) based appearance model. In multi-dimension SVM crowd detection, many features are available to track the object robustly with three main features which include 1) identification of an object by gray scale value, 2) histogram of oriented gradients (HOG) and 3) local binary pattern (LBP). We propose two more powerful features namely gray level co-occurrence matrix (GLCM) and Gaber feature for more accurate and authenticate tracking result. To combine and process the corresponding SVMs obtained from each features, a new collaborative strategy is developed on the basis of the confidence distribution of the video samples which are weighted by entropy method. We have adopted subspace evolution strategy for reconstructing the image of the object by constructing an update model. Also, we determine reconstruction error from the samples and again automatically build an update model for the target which is tracked in the video sequences. Considering the movement of the targeted object, occlusion problem is considered and overcome by constructing a collaborative model from that of appearance model and update model. Also if update model is of discriminative model type, binary classification problem is taken into account and overcome by collaborative model. We run the multi-view SVM tracking method in real time with subspace evolution strategy to track and detect the moving objects in the crowded scene accurately. As shown in the result part, our method also overcomes the occlusion problem that occurs frequently while objects under rotation and illumination change due to different environmental conditions.

Keywords

Multiple Support Vector Machine, Crowd Detection, Motion Blur, Collaborative Model, Gaber Feature

1. Introduction

Tracking is defined as the process of finding the most similar object appearance. The objective of crowd tracking is to determine the states of the desired object in a video sequence. Tracking of moving objects in crowds and motion based detection are important features in many applications such as surveillance system, recognizing the activity of object, human machine interface, monitoring of traffic, motion capture, safety purpose, medical systems and robotics. To determine the location and/or shape of the object in every frame, tracking is used in higher-level applications. An appearance model can be used to depict the desired target in visual tracking. There are three models available for tracking the particular object namely motion model, appearance model and update model. The perfect examples of motion model are Kalman filter and particle filter which are used to depict the probable states of an object. Although many models have been developed, still object tracking seems to be a challenging problem due to the rapid and random changes in the appearance of the object due to abrupt motion in crowd, pose variation, occlusion, varying viewpoints and varying lighting conditions (illumination), compared to all other factors.

Numerous schemes have been proposed for object tracking. The object can be represented in several manners, for example: points [1], articulated models [2], contours [3], or optical flow [4]. In our paper, we have chosen motion and geometric structure based representation of the objects, which can be determined between successive frames. Some of the approaches use model-based techniques which usually assume a priori shape model of a human body represented by stick figures [5] [6] or 2-D contours [7] or volumetric shapes. The appearance-dependent technique uses low-level features to represent the target motion. The motion analysis is based on statistical investigation of these features and/or simple heuristics. The initial frame [8] [9] based static appearance models are used in many object tracking process. But these methods are not capable to cope up with important appearance changes. A key issue in object analysis is finding efficient descriptors for object appearance. Different traditional methods such as Linear Discriminant Analysis (LDA) [10], Supervised Principle Component Analysis and the recent 2D PCA [11] have been studied. The gray-scale invariance is the most important feature for video sequences that have uneven illumination or great variability. Randen and Husoy [12] concluded that the degree of computational complexity of the texture measures is too high in their research. It involves dozens of different spatial filtering methods. For future research the development of powerful texture measures can be extracted and can be done with a low-computational complexity. Several examples of generative tracking techniques are Eigen tracking [13], WSL

tracking [14] and IVT [15]. The object model is frequently updated online in order to adapt to appearance changes as in [15]. The appearance variations are highly non-linear and hence multiple subspaces [16] and non-linear manifold learning methods [17] have been introduced. Apart from category-based methods, exemplar-based methods treat positive samples particularly to avoid the visual-incoherence problem. Chum and Zisserman [18] have developed the exemplar-based classification model to empirically represent object categories. In [19] [20] local distance function learning method by employing triplets, and the focal image which represents the exemplar has been proposed.

Traditional template based tracking algorithms can be divided in two categories. They are offline and online tracking method. In offline method, an object model is either learned offline (by using similar visual examples) or learned by using the first few frames. In both the cases once the object model is generated, immediately a predefined metric model is used to determine the position in adjacent frames. Illustration of this type of tracking algorithms includes Kernel based methods [21] and appearance models [22]. The subspace-based appearance models use the matrices of the pixel values in the image regions that are flattened into vectors and the overall statistical information of every pixel used for finding the vectors is found through PCA. Black and Jepson [23] present a subspace learning-based tracking algorithm. More reliable and accurate schemes like extended dynamic programming [24] are still complex to be employed in problems having maximum number of observations and objects. In spite of several research analyses, detection of crowded objects in many complex situations is still a very good research area [3] [4].

For developing an efficient tracking algorithm, appearance and update model research is concentrated recently. Appearance model can be further divided into discriminative and generative model. The generative model focuses on the knowledge about the object to develop the appearance model whereas the discriminative model simultaneously considers both the knowledge about the background and object. As object tracking can be considered as binary classification task which can be done by extracting the object from its background. The discriminative appearance model is most suitable for the successful implementation of object tracking. Collaborative models are most acceptable for tracking the finer details of the object which are based on above models.

In order to obtain a clear-cut idea about the collaborative model, multi-view learning method has been proposed in this paper. The proposed paper uses collaborative model which is a combination of different sub-models with each having different properties. The collaborative model can be framed by combining two discriminative models or a generative and a discriminative model. In this paper, the first step is to consider three different features namely gray scale value (GSV), histogram of oriented gradients (HOG) and local binary pattern (LBP). Gabor features and gray level co-occurrence matrix (GLCM) are to represent the unique properties of objects from various perspectives. The gray scale feature which can be obtained by vectoring the image regions, gives the basic description of the object image. The histogram of oriented gradients indicates the

edge information and gradient statistic features. HOG feature can also be widely applied in object detection task.

The local binary pattern is used to describe the texture of the object. This can also be used to improve the accuracy in object recognition. These features can be further classified into five complementary views of feature subspaces originally. As these features have different attributes, it inspires us to select them as the multiple views of features. All these features are robust to noise and occlusions since they have structural and local attributes. For further combinations, the sub-models are made by using support vector machines dependent learners. The second step is to determine the weight of each learner. SVM learner gets the confidence score in every frame while tracking the crowd. Each learner's weight has been calculated through the confidence score that is obtained from probability distribution function. To estimate the ambiguity of probability distribution, entropy can be used. Thus entropy can be taken as the measure of weights. The multi view SVMs are combined based on the entropy criterion by a combination strategy that is proposed in this paper. The state-of-the-art techniques are used to estimate the weights depending on the previous performances of the learners whereas the entropy criterion estimates the weights by the current performance of the learners. The flow of the paper next is as follows. Section 2 reviews the literature survey. In Section 3 our proposed algorithm is presented in detail. Experimental results computed using MATLAB software are analysed in Section 4 and conclusion part is included in Section 5.

2. Proposed Methodology

In our proposed paper we intend to build more accurate and robust SVM based detection system. Here we propose a method using fast algorithm for crowd tracking and detection that learns crowd activities and behavior, which finds application in clearing the vehicle traffic during peak hours. To develop crowd detection system that aids fast rescue of lives in times of crises like accident, bomb blast, natural calamities are the motivation behind the research this work. To construct a more comprehensive appearance model using additional feature-Grayscale, Histogram of Oriented Gradients, Local Binary Pattern, Gabor features and Gray Level Co-occurrence Matrix (GLCM) is the major innovation presented in our paper. So that accuracy and robust nature of SVM based crowd detection system is improved. We also present an entropy strategy for the collaboration which determines the weights according to the current performance of each classifier attempting to make a more accurate combination. Finally, we employ FAST (Features from Accelerated Segment Test) algorithm in crowd detection for local features extraction. The block diagram of proposed method is presented in **Figure 1**.

Here, we have combined two algorithms for crowd detection and localization. They are SVM (Support Vector Machine) and FAST (Features from Accelerated Segment Test) algorithms. In this new approach first the input video sequence is converted into 25 to 29 frames per second. Then the frame is segmented to separate the foreground from the background to eliminate the not required area of study in that image. This can

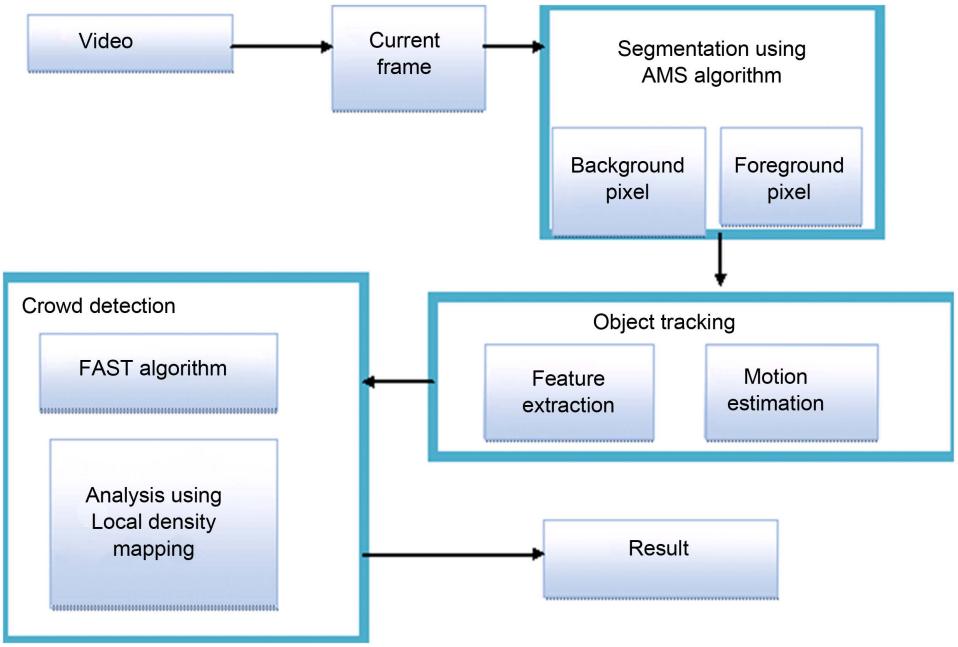

Figure 1. Block diagram of proposed methodology.

be done by using a clustering process called Adaptive mean shift algorithm. The adaptive mean shift (AMS) algorithm is an advanced version of mean shift algorithm. This method is utilized because it converges quickly. After segmentation the crowd or a particular target can be detected and localized using a hybrid algorithm that combines both SVM and FAST algorithm.

2.1. Adaptive Mean-Shift Iterative Segmentation Algorithm Units

It is a non-parametric iterative algorithm. This algorithm is applicable for lot of purposes like finding modes, clustering etc. In our paper we use this algorithm to do segmentation by means of adaptive mean shift iterative process. The feature space is considered as probability density function in adaptive mean shift. The input is considered to be sampled from the probability density function when it is a set of points. The clusters or dense regions are considered as the mode of probability density function when it is present in the feature space. Mean shift is also used to identify the clusters associated with the given mode. Mean shift associates each data point with the nearby peak of the probability density function of the datasets. Mean shift then defines a window around each data point and computes the mean of the data point using the window. The center of the window is shifted to the mean and until it converges the algorithm is repeated. The window shifts to deeper region of the dataset after each iteration.

The algorithm works as follows:

1) A window is fixed around each data point.

2) The mean of data is computed within the window.

3) Then the window is shifted to the mean and the steps are repeated till convergence.

2.1.1. Kernels

A kernel is a function which obeys the following preliminaries,

$$\int R^d \varnothing(x) = 1, \text{ where } \varnothing(x) \geq 0. \tag{1}$$

Some examples of kernels include:

1) Rectangular

$$\varnothing(x) = 1, a \leq x \leq b, \ \varnothing(x) = 0, \text{else}. \tag{2}$$

2) Gaussian

$$\varnothing(x) = \mathrm{e}^{-\frac{x^2}{2\sigma^2}}. \tag{3}$$

3) Epanechnikov

$$\varnothing(x) = \begin{cases} \frac{3}{4}\left(1 - x^2\right), & \text{if } |x| \leq 1 \\ 0, & \text{else} \end{cases}. \tag{4}$$

2.1.2. Kernel Density Estimation and Gradient Ascent Nature of Mean Shift

The non-parametric way to calculate the density function of a random variable is by using Kernel density estimation. This is called as the Parzen window technique. Given a kernel *K*, bandwidth parameter *h*, Kernel density estimator for a given set of d-dimensional points is as follows

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} k\left(\frac{x - x_i}{h}\right). \tag{5}$$

Mean shift is considered to be based on Gradient ascent on the density contour. The generic formula for gradient ascent is,

$$x_1 = x_o + \eta f'(x_o). \tag{6}$$

Applying it to the kernel density estimator we get,

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} k\left(\frac{x - x_i}{h}\right) \tag{7}$$

$$\nabla f(x) = \frac{1}{nh^d} \sum_{i=1}^{n} k'\left(\frac{x - x_i}{h}\right). \tag{8}$$

Setting it d to 0 we get,

$$\sum_{i=1}^{n} k'\left(\frac{x - x_i}{h}\right)\vec{x} = \sum_{i=1}^{n} k'\left(\frac{x - x_i}{h}\right)\vec{x_i}. \tag{9}$$

Finally,

$$\vec{x} = \frac{\sum_{i=1}^{n} k'\left(\frac{x - x_i}{h}\right)\vec{x_i}}{\sum_{i=1}^{n} k'\left(\frac{x - x_i}{h}\right)}. \tag{10}$$

2.1.3. Iterative Mean Shift and Proof of Convergence

The Mean shift considers the points of the feature space as a probability density function. Dense regions in feature space are mapped to local maxima or modes. So, the gradient ascent is performed for each data point on the local estimated density until it convergence. The stationary points obtained via gradient ascent indicate the modes of the density function. All the points associated with the same stationary point belong to the same cluster.

Assuming,

$$g(x) = -k'(x). \tag{11}$$

We have

$$m(x) = \frac{\sum_{i=1}^{n} g\left(\frac{x - x_i}{h}\right)\overrightarrow{x_i}}{\sum_{i=1}^{n} g\left(\frac{x - x_i}{h}\right)} - x. \tag{12}$$

The quantity $m(x)$ is called as the mean shift. So mean shift procedure can be summarized as that for each point x_i, Compute mean shift vector $m\left(x_i^t\right)$ and move the density estimation window by $m\left(x_i^t\right)$, and repeat the process till convergence. Using a Gaussian kernel as an example,

$$y_i^o = x_i \tag{13}$$

$$y_i^{t+1} = \frac{\sum_{i=1}^{n} x_j \mathrm{e}^{\frac{\left|y_i^t - x_j\right|^2}{h^2}}}{\sum_{i=1}^{n} \mathrm{e}^{\frac{\left|y_i^t - x_j\right|^2}{h^2}}}. \tag{14}$$

Using the kernel profile, we have to prove that $f\left(y^{t+1}\right) \geq f\left(y^t\right)$, Using the Equation (14) we can write the above equation as,

$$f\left(y^{t+1}\right) - f\left(y^t\right) \geq \sum_{i=1}^{n} k'\left(\left\|\frac{y' - x_i}{h}\right\|^2\right)\left(\left\|\frac{y^{t+1} - x_i}{h}\right\|^2 - \left\|\frac{y^t - x_i}{h}\right\|^2\right) \geq 0. \tag{15}$$

Thus we have proven that the sequence $\left\{f(j)\right\}_{j=1,2,\cdots}$ is convergent.

2.2. Learning and Training of Multi-View SVMs

We have introduced object tracking based on multi-view SVM (MVS) algorithm. This algorithm is implemented as follows. The MVS tracker which uses different views of features (gray level value, HOG, LBP, Gabor feature, GLCM) are implemented in the particle filter structure. The gathered multi-view SVMs are used for representing the object and implementing the appearance model. Then the entropy strategy is used for combining the multi-view SVMs and is built into the particle filter structure in order to obtain the results of tracking. The subspace evolution strategy is also used in this algorithm for updating, adjusting update rate and also for providing guidance to retrain multi-view SVMs in online. The construction of tracking method is actually based on

the discriminative model. However the discriminative model formulates the tracking as a binary classification problem. For accurate object representation, the multi-view features based SVMs are trained to implement a collaborative appearance model. The fusion strategy is used for building up this collaborative model.

Figure 2 shows how to construct an appearance model. From an image, samples are prepared from positives and negatives samples of the image. The positive samples and the negative samples are selected around the objects. The five features named gray level, HOG, LBP, Gabor features and GLCM are obtained from these samples and they are sent to the respective SVMs to perform training. Let the positive samples be A^p and the negative samples be A^n. The selection would be better if the samples are selected from the cropped image regions. By distance based rule, for example, for m_i, its label is defined as $n_i\left(n_i \in \{-1,1\}\right)$. It is also represented in pairs as $\{m_i, n_i\}$. Let $k(m_0)$ and $k(m_i)$ be the location of target and sample to be trained, respectively. By distance based rule,

$$\text{Case (1):}\ \left\|k(m_i)-k(m_0)\right\| \le d_1,\ m_i \text{ is a positive sample.} \tag{16}$$

$$\text{Case (2):}\ d_2 \le \left\|k(m_i)-k(m_0)\right\| \le d_3,\ m_i \text{ is a negative sample} \tag{17}$$

where d_1, d_2 and d_3 are predefined thresholds

We take $d_1 = 2$ pixels,

$$d_2 = \left(\sqrt{w^2+h^2}\right)\Big/2, \text{ and } d_3 = 2\left(\sqrt{w^2+h^2}\right) \tag{18}$$

where w and h are width and height of the pixels respectively. Practically it is not necessary to select the target region in the video sequence because the training samples are automatically selected in this method. This rule is also updating the training samples in frames. From samples the features are extracted to represent the objects. The gray scale features is obtained by dividing the 2D image region and it gives the basic

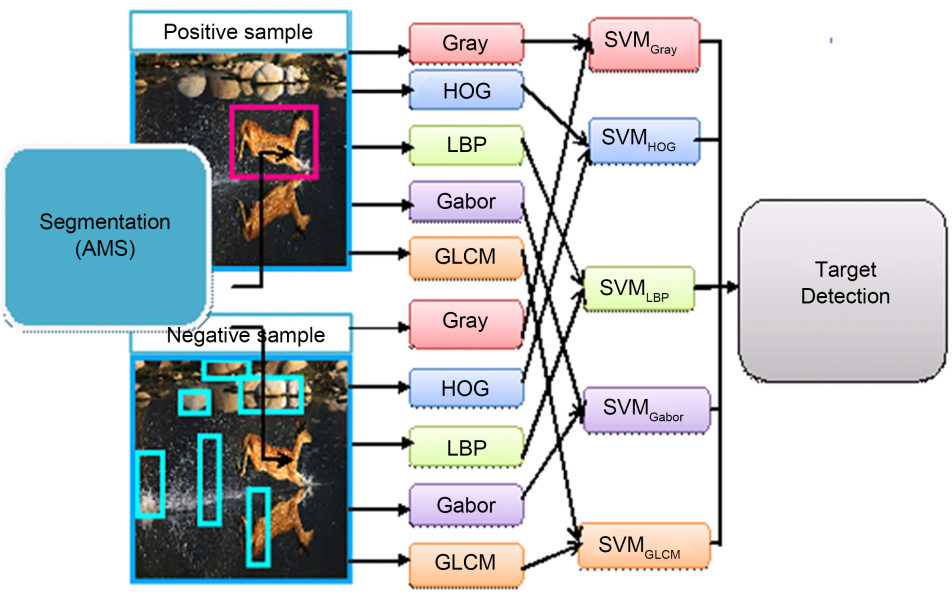

Figure 2. Object tracking based on Multi-View SVM with five different features for detection.

description of the object. The HOG feature has powerful detection capability and it is based on gradient of the image. The LBP feature is used for representing texture of the image. It also has recognition capability. Gabor feature is used for edge detection, texture description and discrimination. The purpose of using gray level co-occurrence matrix is to examine the texture of the object based on spatial relationship between the pixels. The features that are extracted are denoted as y_{gray}, y_{HOG}, y_{LBP}, y_{gaber}, y_{gaber} and y_{GLCM}, respectively. For example the pair of feature vectors is represented as $\left\{y_i^{(k)}, n_i\right\}$. The cropped image is normalized for convenience. These pairs of features are trained into the SVM classifier to build appearance model. The results are obtained with classification problem for which the SVM will use a separate plan done by maximizing the positive and negative image margins. Assume $(k) \in$ {gray, HOG, LBP, GLCM, Gabor} and $i \in \{+1, -1\}$. The optimization problem is represented by,

$$\min_{w,b,\varepsilon} \frac{1}{2}\sum_k \left\|w^k\right\|^2 + \sum_k \sum_i C_k \varepsilon_i^k . \tag{19}$$

Its subjected condition is:

$$n_i\left(\left(w^k\right)^T v^k + b^k\right) \geq 1 - \varepsilon_i^k,\ \ \varepsilon_i^k \geq 0, i = 1, \cdots, M \tag{20}$$

where, C_k trade-parameter and ε_i^k slack variable. Due to independency between the features, the sub-problem can be expressed as,

$$\min_{w,b,\varepsilon} \frac{1}{2}\left\|w^k\right\|^2 + \sum_i C_k \varepsilon_i^k,\ k \in \{\text{gray, HOG, LBP, GLCM, Gabor}\}. \tag{21}$$

The dual problem of lagrangian multiplier algorithm is:

$$\max_\alpha \sum_i \alpha_i^k - \frac{1}{2}\sum_i \sum_m \alpha_i^k \alpha_m^k n_j n_m \left(y_i^k\right)^T y_m^k . \tag{22}$$

Figure 3 explains about training SVMs in particle filter and using entropy criterion to determine the final confidence score. Equation (22) is subjected to,

$$\sum_i \alpha_i^k n_i = 0\ \ \&\ \ 0 \leq \alpha_i^k \leq C_k, i = 1, \cdots, M \tag{23}$$

where $\left\{\alpha_i^k\right\}$ are Lagrange's multipliers. By solving these problems the multiple views of learners represents the appearance model of the object. For a sample m, the confidence score is calculated as,

$$conf\left(y^k\right) = \sum_{i=1}^M \alpha_i n_i \left(y_i^k\right)^T y^k . \tag{24}$$

The final confidence score is:

$$conf\left(m\right) = \sum_k \lambda_k conf\left(y^k\right),\ k \in \{\text{gray, HOG, LBP, GLCM, Gabor}\} \tag{25}$$

λ_k is the weighted parameter of each SVM features.

2.3. Entropy Computation

The particles represent possible state of the object and simulate the state distribution. Hence, the confidence score of each particle is required for the tracking result. The two steps of particle parameters are:

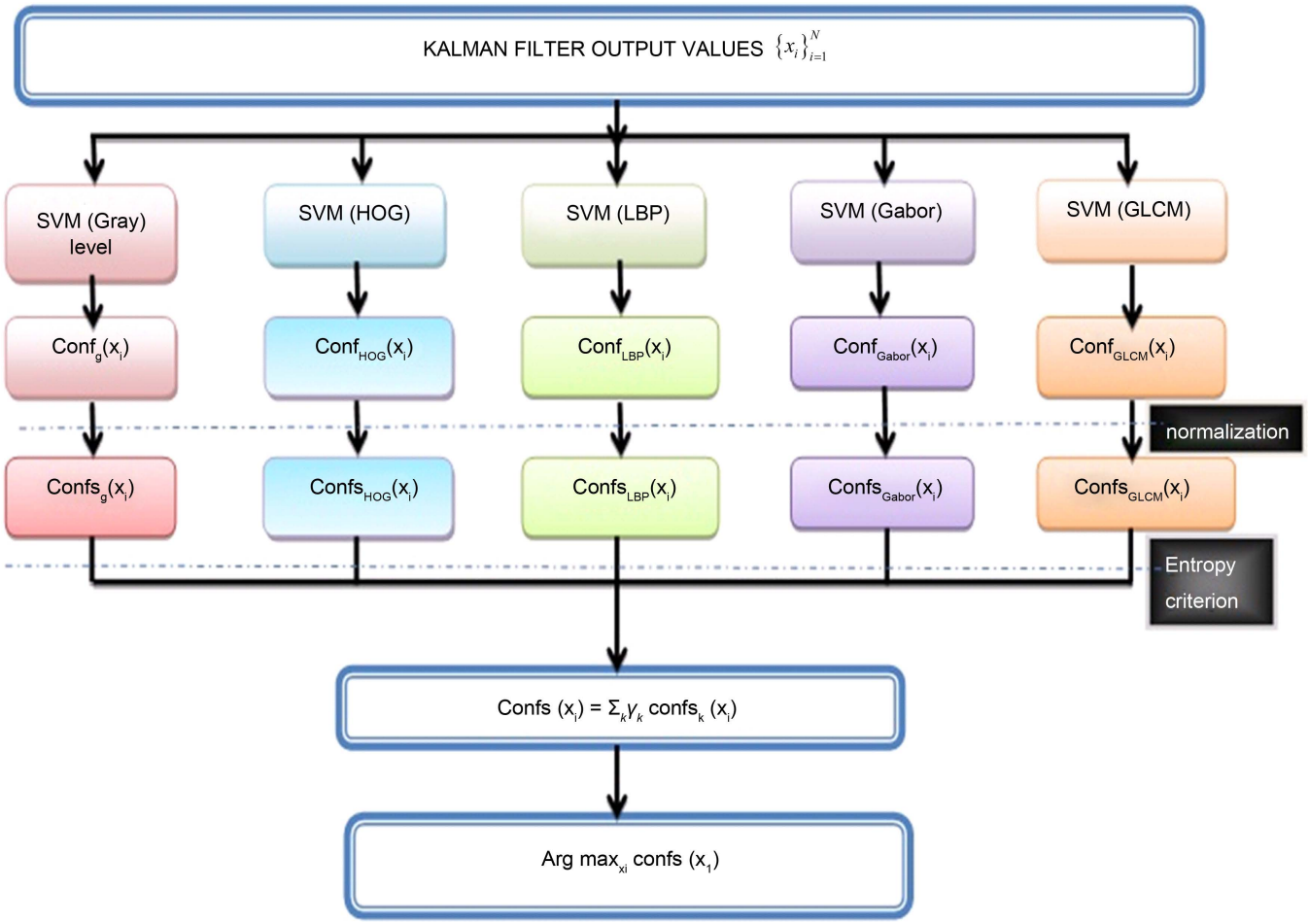

Figure 3. Block diagram of particle filter.

1) Prediction step:

$$q(S_t \mid O_{1:t-1}) \propto \int q(S_t \mid S_{t-1}) q(S_{t-1} \mid O_{1:t-1}) \mathrm{d}S_{t-1}. \tag{26}$$

2) Update step:

$$q(S_t \mid O_t) \propto q(O_t \mid S_t) q(S_t \mid O_{1:t-1}) \tag{27}$$

where $\{O_1, O_2, \cdots, O_t\}$ are observation variables.

The observation model is denoted by $q(O_t \mid S_t)$ and the transition model is represented as $q(S_t \mid S_{t-1})$.

$S = \{p_x, p_y, t_x, t_y\}$, where S is called as state space and p_x, p_y is translation in x and y direction, where t_x, t_y is scale variation in x and y direction.

Consider N number of particle for approximating the state space. The changes in video sequences must be small and random. These particles must obey multi-dimensional Gaussian distribution. Examples: LI and IVT tracking method (same assumptions are taken). Each particle represents candidate sample. Confidence score are calculated over these samples. We normalize the candidate samples m_i ($i = 1, \cdots, n$) to extract all the features and calculate corresponding confidence score. Thus the final con-

fidence score is obtained. However combination is a key problem which can be overcome by considering horizontal and vertical translation. In different sequences the final confidence score which shows that different features of different abilities. The scores of particles by min-max rule in the range (0, 1) is,

$$confs_k\left(x_i\right)=\frac{confk\left(x_i\right)-\min\left(confk\right)}{\max\left(confk\right)-\min\left(confk\right)}. \tag{28}$$

Then the scaled confidence score is:

$$confs\left(x_i\right)=\sum_k \gamma_k confs_k\left(x_i\right) \tag{29}$$

where

$$\gamma_k=\left(\text{normalized } \lambda_k\right)=\frac{1/H\left(k\right)}{\sum_k 1/H\left(k\right)}. \tag{30}$$

The entropy of the confidence scale distribution is,

$$H\left(k\right)=-\sum_{i=1}^{N} p_k\left(i\right)\log p_k\left(i\right). \tag{31}$$

This is determined after calculating entropy of each distribution. The likelihood function is,

$$\mathcal{L}\left(x_i\right)=\frac{1}{\sqrt{2\pi}}\frac{1}{\sigma w}\exp\left(-\left(1-confn\left(x_i\right)\right)/\sigma_w^2\right) \tag{32}$$

where σ_w is the similarity weighted parameter which is used for re-sampling.

By Maximum A Posteriori rule,

$$X_{opt}=\arg\max_{x_i}\left(\mathcal{L}\left(x_i\right)\right)=\arg\max_{x_i} confs\left(x_i\right) \tag{33}$$

where X_{opt} is the optimal candidate sample and its corresponding optimal state S_{opt} is determined.

2.4. Subspace Evolution Used in Update Model

As the background and object often changes during video sequencing, an online update strategy named subspace evolution strategy is used. This controls the updating process in video sequencing. This method involves constructing subspace to the object and evolving it in accordance with the change of the object. It involves two stages, finding the distance between the subspace and optimal sample in the current frame and retraining strategy. We can use the positive and negative samples to span subspace of the object and we construct the new optimal sample X_{opt} based on l1 minimization. The training sample set is $D=\left[v_1^+,\cdots,v_{Np}^+,v_1^-,\cdots,v_{Nn}^-\right]$, solving the problem by LARS algorithm

$$\min_c \left\|Dc-v_{opt}\right\|_2^2+\mu\left\|c\right\|_1 \tag{34}$$

where $c=\left[c_1^+,\cdots,c_{Np}^+,c_1^-,\cdots,c_{Nn}^-\right]^{\mathrm{T}}$ - reconstruction coefficient vector, $\left\|\cdot\right\|_1,\left\|\cdot\right\|_2$ are l1 and l2 norms.

μ is the regularization parameter (0.01). The reconstruction error for positive sam-

ples is,

$$\text{err} = \left\| D^{+}c^{+} - v_{opt} \right\|_2 . \tag{35}$$

This error indicates distance between X_{opt} and subspace of the appearance and decides to whether update or not as per the following cases.

Case (1): If err < Th (threshold), then X_{opt} is represented by existing subspace.

Case (2): If err > Th, then the change of appearance exceeds the representation subspace capacity and hence it has to b evolved. This avoids the occlusion samples and thus posse's robust nature.

2.5. Improved Fast Algorithm for Crowd Detection

There are several feature detectors that are really good, but they are not fast enough. One of the best examples is SLAM (Simultaneous Localization and Mapping) mobile robot in which computational resources are limited. As a solution to this problem, FAST (Features from Accelerated Segment Test) algorithm have been proposed. Here, A pixel P is selected in the image that is to be considered as a required point or not. Let its intensity be I_P. the appropriate threshold value t is selected. A circle of 16 pixels is considered around the pixel under test as shown in Figure 4. (See the image below)

There are 16 pixels chosen from the selected object and from here the corner pixel p is chosen between the pixels $I_P + t$ (maximum brighter value) and $I_P - t$ (maximum darker value). To exclude a large number of non-corners, a high-speed test was proposed. This test analysis is done only on the four pixels at 1, 9, 5 and 13. Suppose P is the corner pixel then there should be 3 pixels between $I_P + t$ and $I_P - t$, otherwise P can never be a corner. The full segment test criterion can then be applied to the successful

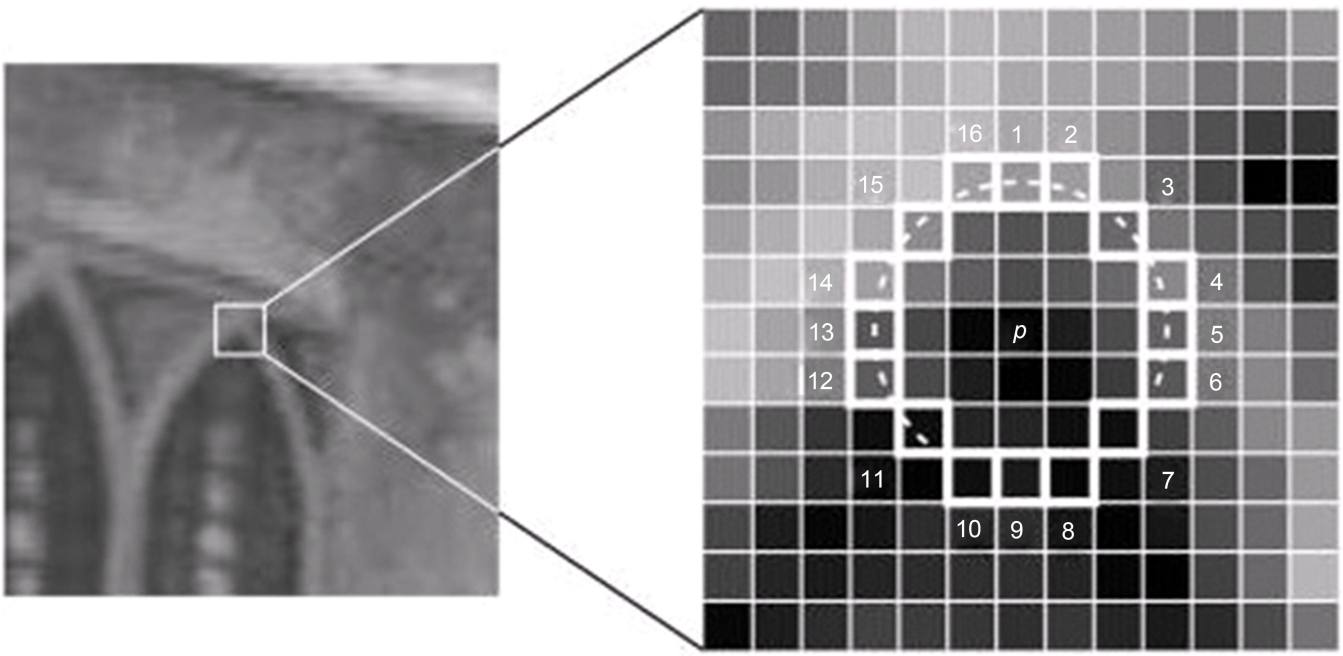

Figure 4. Selection of a correct threshold value from the object using fast algorithm.

candidates by analyzing all pixels in the circle. This detector shows high performance but there are several drawbacks which are lesser optimal performance because its performances are related to the distribution of corner appearances. Multiple features are estimated for the points adjacent to one another. So we are going to machine learning a Corner Detector using improved Fast algorithm. Here we have considered a set of images for training. The algorithm is performed in every image to calculate feature points. The 16 pixels are stored around every feature point as a vector. To get feature vector P, this has to be done for all the images. Each pixel (say x) in these 16 pixels can have one of the following three states:

$$S_{p\to x}=\begin{cases} d, & I_{p\to x}\le I_p - t & (\text{darker}) \\ s, & I_p - t < I_{p\to x} < I_p + t & (\text{similar}) \\ b, & I_p + t \le I_{p\to x} & (\text{brighter}) \end{cases} \tag{36}$$

The feature vector P is further divided into 3 subsets, P_d, P_s, P_b depending on the above states. A new Boolean variable is defined as K_P which is true if P is a corner and false otherwise. By using the ID3 algorithm (decision tree classifier) to examine each subset using the variable k_p to get the knowledge about the true class. It leads to the selection of the x value which gives the basic information about whether the candidate pixel is a corner of the crowd to be detected, which is measured from the entropy of k_p. When the entropy becomes 0, the above process must be stopped and the final resultant frame can be used for fast detection in other images.

3. Experimental Result

In our method, we executed and tested the performance using 8 public challenging video sequences. Firstly, to analyze the performance of our method, we experimented the different sequences with our algorithm under critical conditions. We compare our method with the other state-of-the-art tracking methods and also compared with similar methods. Secondly, we measure the performance of different combinations of the various features in our method. Later, we study the role of the subspace evolution update model and the entropy criterion. Our tracking method is initialized as follows. The first step is to normalize the size of the image region is kept as 30 × 30, for feature extraction. Both the number of positive and negative samples are set to 50. A 5-pixel widow size and 9 orientations are assigned to the HOG descriptor [25]. A 10-pixel window is used for the LBP descriptor [26]. The dimensions of feature vectors are designated as 900, 324, 105, 245 and 522 respectively. It is assumed that both the model complexity and the training errors of each SVM have equivalent mask. The value is mathematically set to 1, where $k \in$ {gray, HOG, LBP, GLCM, Gabor}. The number of the particles in the particle filter is set to 300, where the individual particle depicts the trimmed image region of fixed size 30 × 30. A value of 0.5 is set to the similarity weight, and a value of 0.005 is assigned to the threshold. In all the experiments conducted by us, these parameters are fixed. The 8 video sequences are obtained from different scenes and they also include critical conditions such as pose variations, illumination changes,

occlusions, scale variations, clutter background, etc. The complete information of the different video sequences including the sizes and the frame numbers is given in Table 1.

We have a detailed comparison with several different tracking methods, including Frag [27], IVT [28], MIL [29], OAB [30], L1 [31], VTS [32], TLD [33], MTT [34], CT [35], VTD [36], Struck [37] and PartT [38]. All these trackers apply different representation or inference models. The codes of these trackers are all available in public and we can alter the parameters attentively so that better performance results can be obtained. By analyzing the result of our SVM method both quantitatively and qualitatively, we consider two complementary evaluation parameters for the quantitative analysis. They are, centre location error (CLE), tracking success rate (TSR). By averaging the pixel errors between the obtained centers of tracking results and the ground truth, we can calculate the centre location error. The location error is directly proportional to the centre location error value. *i.e.*, smaller the CLE value, smaller the location error. The reduced value of CLE for our algorithm is shown in Figure 5.

Table 1. Sizes and frame numbers of testing sequences.

Sequence	Size	Frame number
Animal	704 × 400	#1-71
Basket ball	576 × 432	#1-725
Football	624 × 352	#1-362
Girl	320 × 240	#1-502
Human	768 × 576	#1-412
Singer	624 × 352	#1-321
Stone	320 × 240	#1-593
Woman	352 × 288	#1-550

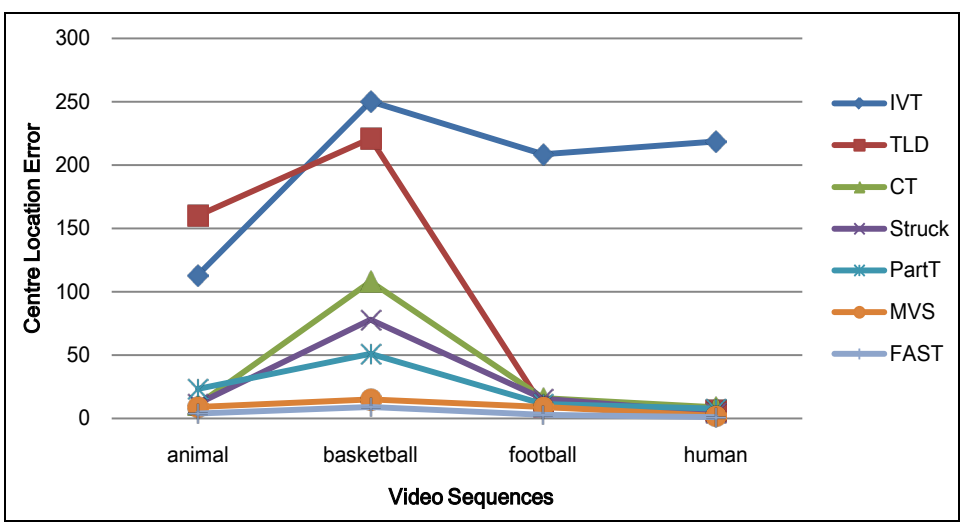

Figure 5. Comparison results of the average center location errors of competing trackers (in pixels).

The best performance of our proposed algorithm through TSR value is shown in Figure 6. Tracking is successful only if the score and TSR gives the ratio of the number of the successful frames and the total. The failure rate (FR) is the third performance parameter, which can be explained as the number of times the tracking is failed in the total video sequence. In our tracking process, to continue the tracking, the tracker will be reinitialized at the failure frame depending on the ground truth and the failure times will be added 1, if and only if the score in current frame. The failure times, which represents the reinitialization times, for the entire video sequences tracking process is set to the value 0.1. The robustness of the trackers can be estimated from the FR value as it is based on the entire sequences. Based on the results obtained from our experiment on different video sequences we can infer that the proposed method performs better in terms of average CLE value as depicted in Table 2 and average TSR value as shown in Table 3. We also observed that the location error is smoother and lower on our input sequences. For most of the sequences our tracking algorithm is able to track the object

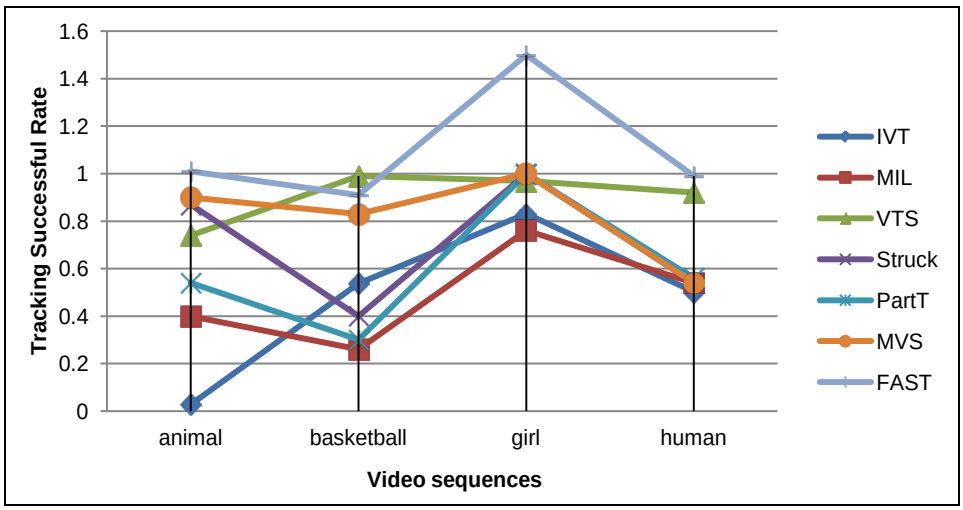

Figure 6. Comparison results of the tracking successful rates of competing trackers.

Table 2. Comparison results of the average center location errors of competing trackers (in pixels).

Image sequence	Frag	IVT	LI	MIL	OAB	VTD	VTS	TLD	MTT	CT	Struck	PartT	MVS	Fast
Animal	102.3	113.1	161.9	67	25	10.4	23	160	26	11	11.7	23.3	6.0	7.2
Basketball	41.4	250.1	129.0	100	32	7.4	7.8	221	292	108	78	-	12.7	11
Football	138.2	208.5	27.0	12	137	4.4	4.7	9.2	12	16	15.5	11.4	5.3	4.2
Human	9.3	218.6	10	2.7	7.4	5.4	5.4	6.4	2.4	8.8	4.7	7.7	2.2	2.1
Girl	24.6	25.2	17.2	25	14.6	13.8	13	45	35.6	32	7.7	6.0	11.9	6.0
Stone	66.3	2.3	7.7	5.0	90.0	25.6	32	6.0	2.2	3.2	2.0	1.9	1.7	2.5
Singer	38.5	103.8	52.2	19	18.6	3.2	3.8	13	23.2	17	14.2	13.1	2.8	2.5
Woman	92	156.7	122.2	121	120	136	136	192	140	103	3.7	2.8	3.0	2.5
Average	66.9	93.9	55.6	40	41.4	26	27.7	67.3	47.7	36.4	14.4	10.0	5.6	4.9

Table 3. Comparison results of the tracking successful rates of competing trackers.

Image sequence	Frag	IVT	LI	MIL	OAB	VTD	VTS	TLD	MTT	CT	Struck	PartT	MVS	FAST
animal	0.028	0.042	0.042	0.40	0.71	0.76	0.74	0.56	0.54	0.85	0.87	0.54	0.90	1.01
basketball	0.537	0.016	0.149	0.26	0.69	0.98	0.99	0.01	0.02	0.24	0.40	-	0.83	0.91
Girl	0.831	0.693	0.842	0.76	0.97	0.94	0.97	0.67	0.71	0.48	1.00	1.00	1.00	1.00
Human	0.502	0.004	0.483	0.54	0.54	0.54	0.92	0.90	0.80	0.99	0.54	0.56	0.54	0.99
Singer	0.246	0.138	0.384	0.24	0.24	1.00	1.00	0.55	0.30	0.24	0.24	0.24	0.90	1.20
Stone	0.294	0.882	0.386	0.83	0.14	0.61	0.62	0.84	0.98	1.00	0.98	0.99	0.98	1.4
Surfer	0.263	0.052	0.039	0.57	0.80	0.92	0.92	0.98	0.21	0.01	0.80	0.86	0.98	0.88
Woman	0.300	0.014	0.221	0.21	0.21	0.15	0.18	0.04	0.21	0.21	1.00	1.00	1.00	1.10
Average	0.347	0.311	0.421	0.48	0.48	0.66	0.70	0.55	0.56	0.54	0.77	0.74	0.95	1.02

successfully. From Figure 7 we can analyze the performance of our tracker in qualitative manner. To deal with occlusion, fast motions, illumination changes, scale variations, complex background and pose variations, our system uses local features.

Example of a human video with crowded objects chosen as below to show the greater performance of our algorithm in detail with multiview SVM features and Improved FAST algorithm.

Figure 7 first shows gray scale feature of the obtained input that is the frame of video represents the crowded objects, second it also represents the local binary pattern of the video frame considered for our analysis.

Figure 8 and Figure 9 represent the HOG and GABOR features that are extracted from the input frame. Figure 10 and Figure 11 calculate the moving pixels obtained by comparing the current frame and the reference frame.

And Figure 12 shows the modified expectation maximization algorithm output. Figure 13 shows the confidence score distribution of the input frame obtained by using SVM classifier.

Figure 14 shows the object tracking output obtained using SVM algorithm and Figure 15 shows the crowd density determination based on crowd movement obtained by using FAST algorithm.

The update model can be used in our method to reduce the effect of occlusions. The IVT method does not perform well on sequences with sudden illumination changes. But our algorithm can overcome this limitation as we use features that are robust to illumination changes. We can deal scale with variation existing in the video sequences by integrating the approaches based on MTP, VTD and VTS. When the object moves fast or having sudden movement, the performance of our method exceeds to that of remaining methods. The SVM proposed in our paper employs discriminative models and various combination strategies so that it can be robust against complex background. Since our method uses suitable on-line learning strategy and collaborative model we

Figure 7. Extracting features from the input frame.

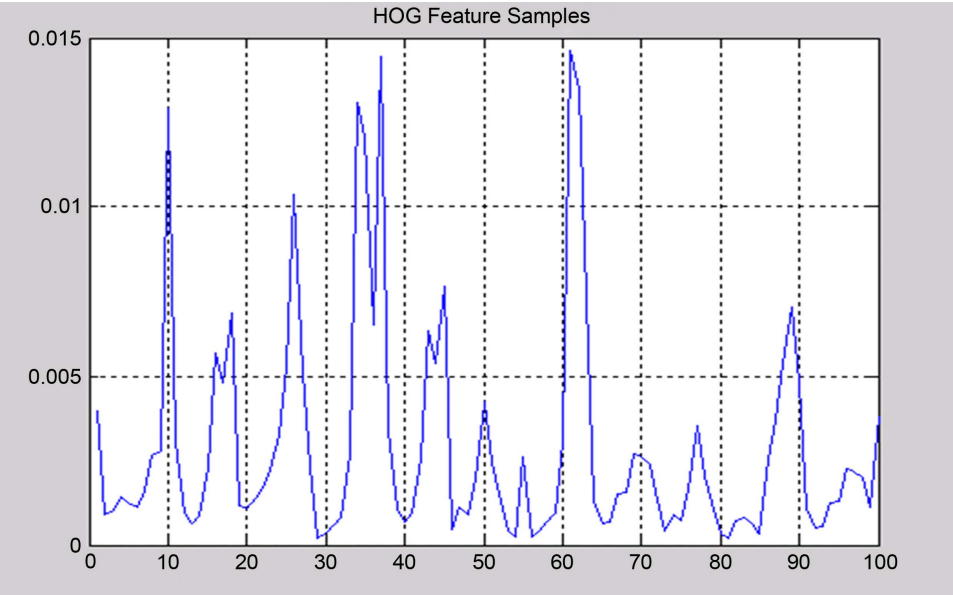

Figure 8. HOG feature.

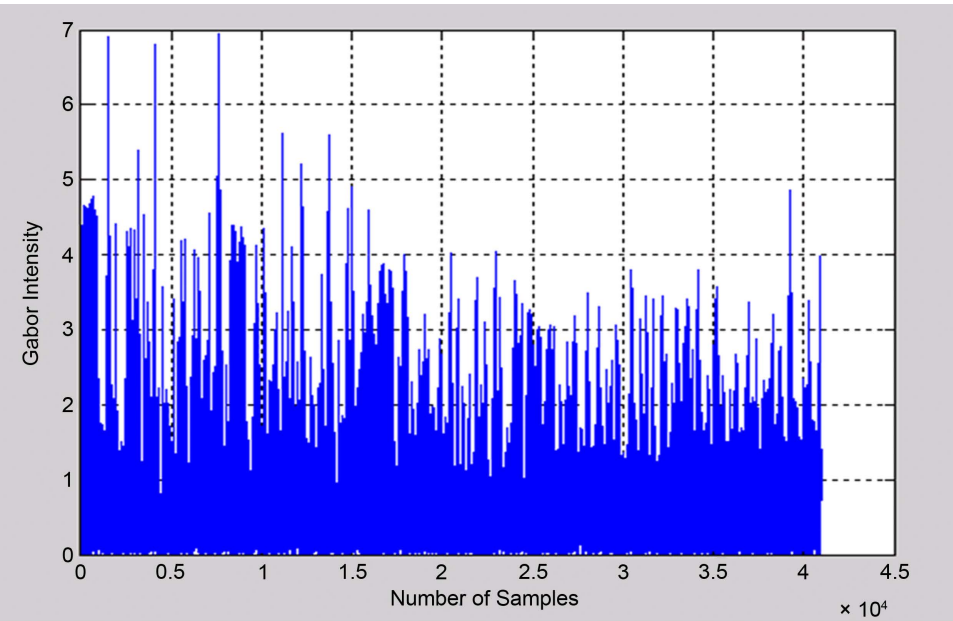

Figure 9. GABOR feature.

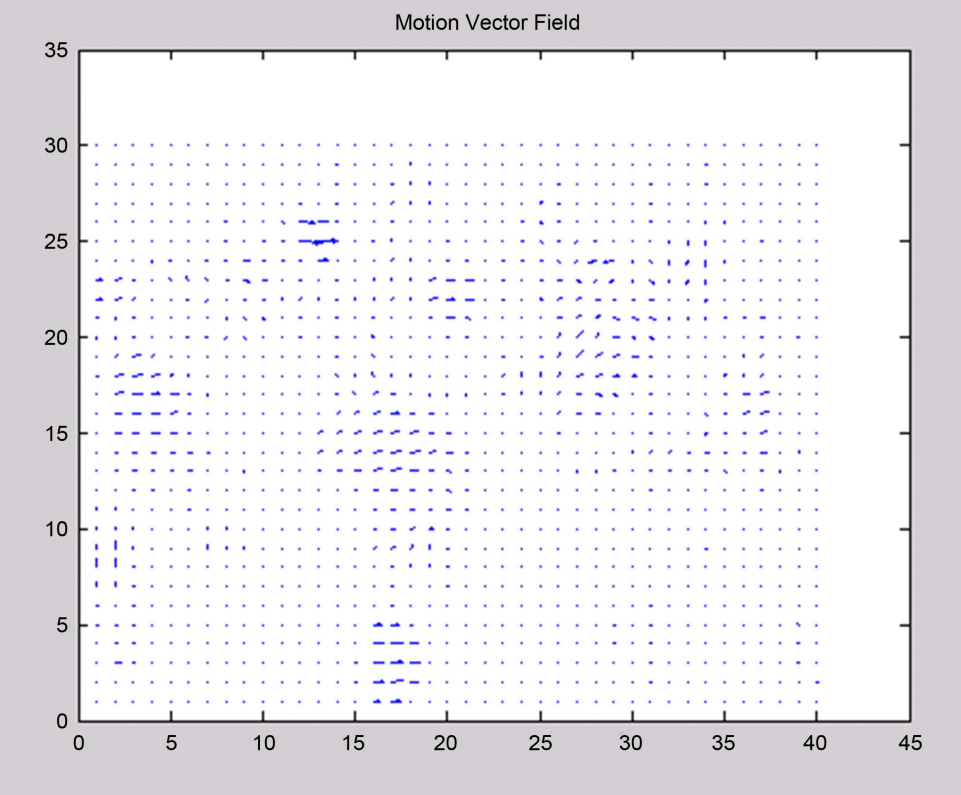

Figure 10. Motion vector field.

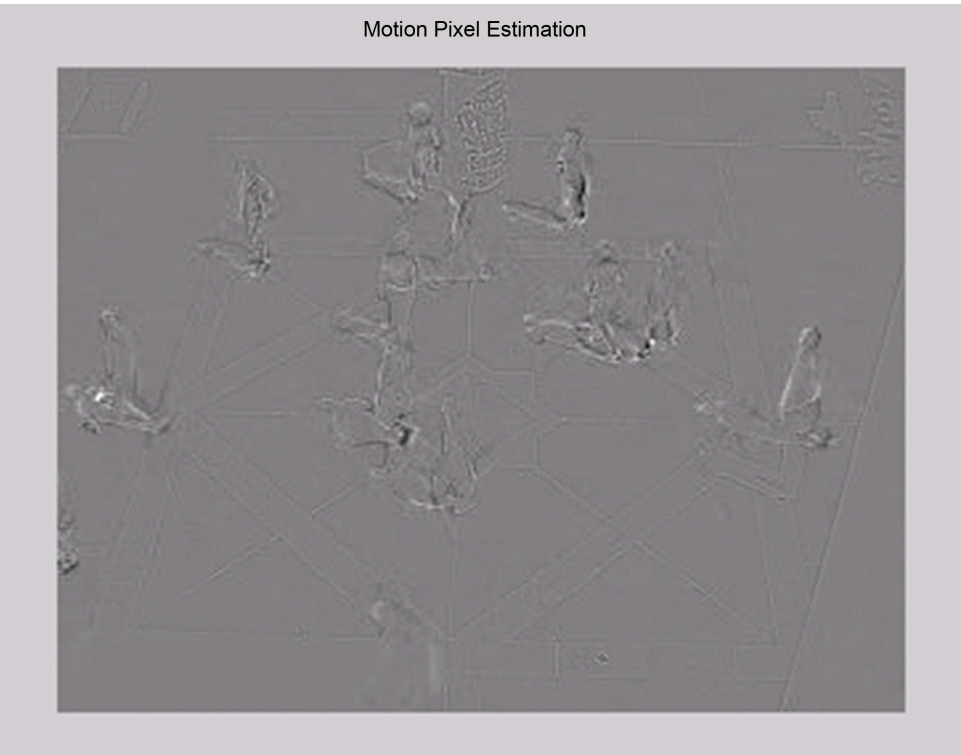

Figure 11. Motion pixel estimation.

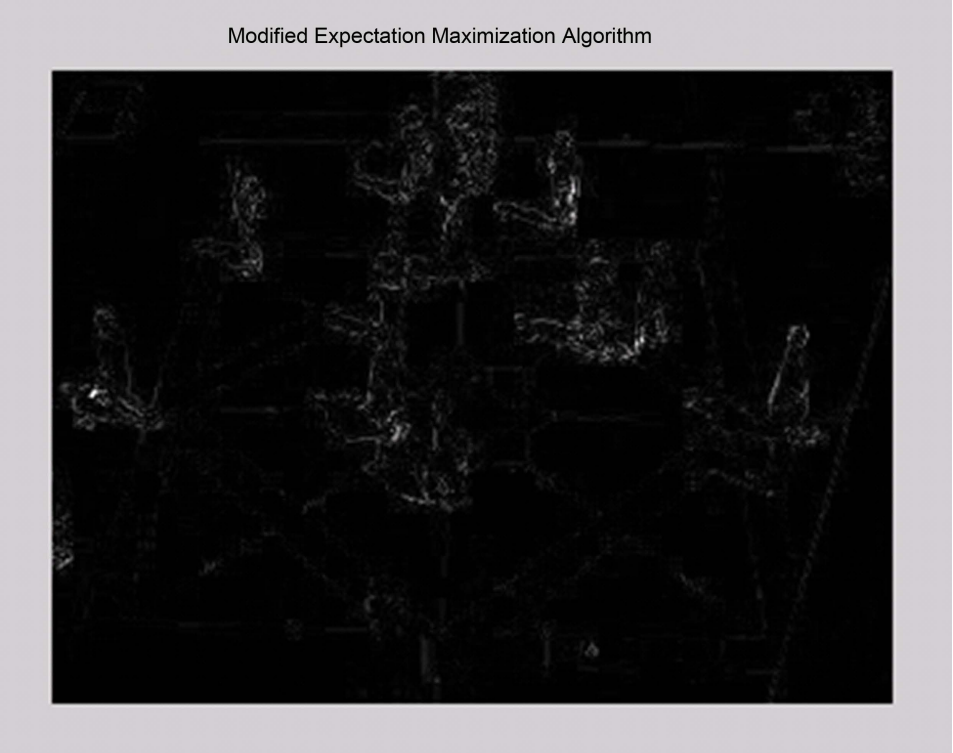

Figure 12. MEM algorithm output.

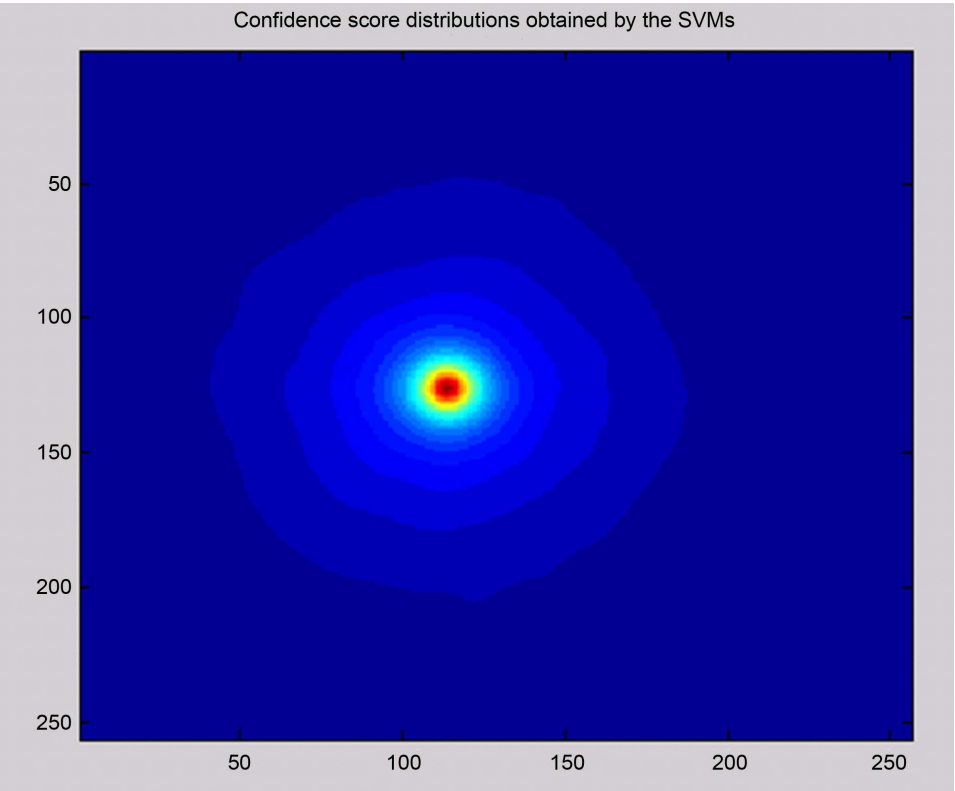

Figure 13. Confidence score.

Figure 14. SVM output.

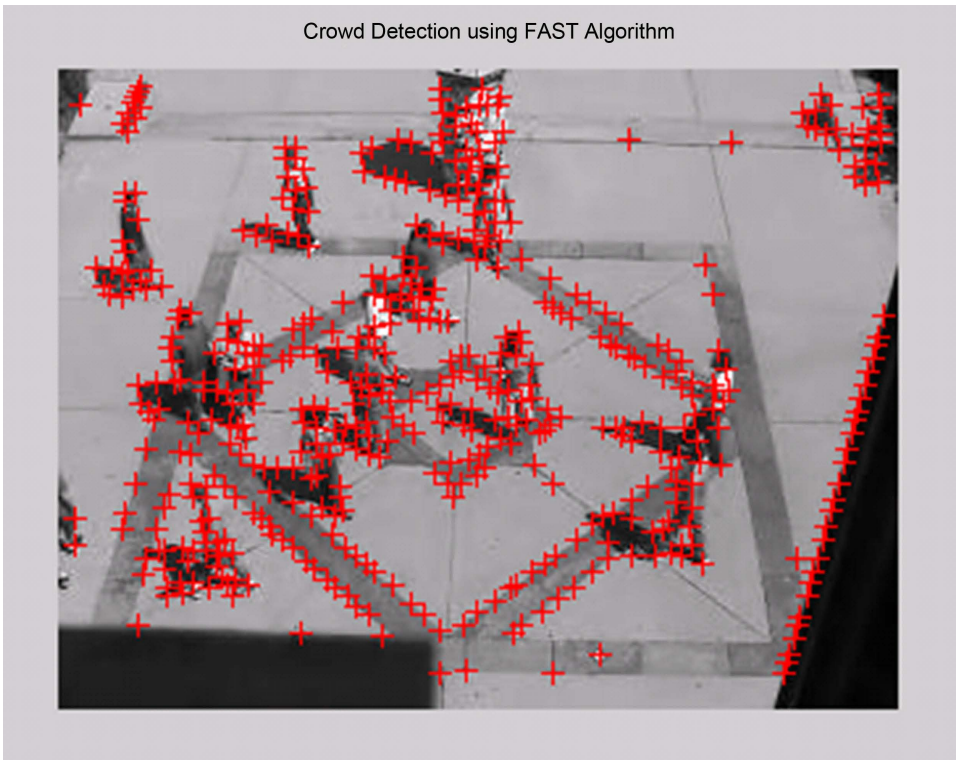

Figure 15. Improved FAST algorithm output.

can successfully track the changes in human's gesture. It is difficult to studying the changes in appearance during tracking for accurate representation. The classification accuracy for rapid appearance changes of the tracked target, cannot reflect the performance of the classifier on current frame, it leads to inaccuracy in the calculation of weights. The sparce collaborative is the model that is been used commonly for these kind of situations in the current scenario. The disadvantages of the stated methods are it cannot be adaptable to different conditions, as it empirically fixes the weight of each model in advance. For learning the appearance online, we present a novel subspace evolution strategy that controls the update rate adaptively. Potential applications of MVS tracking method is multi-media information processing, since the object of interest often needs to be tracked with the preprocessing stage. MVS can also provide a coarse segmentation for editing operations. Additionally MVS can also track each part of the object of interest in pixel domain. The challenge lies on his method further will be removing of full occlusions and dealing with drastic rotations of the object.

4. Conclusion

In this paper, we present a novel MVS tracking method with multiple views of SVMs. The FAST algorithm is used to detect the CROWD. This method follows multi-view learning framework and integrates five views of features. Each view of the features represents a special characteristic of the object and the combination leads to a more comprehensive representation of the appearance. In this paper, we present a novel MVS tracking method with multiple views of SVMs. This method follows multi-view learning framework and integrates three views of features. Each view of the features represents a special characteristic of the object and the combination leads to a more comprehensive representation of the appearance. Furthermore we propose a novel entropy criterion to determine the weights of the multi-view SVMs which make the collaboration more accurate and robust. Besides, we also propose a novel subspace evolution strategy combining with the retraining method to complete the model update. The experimental results demonstrate that our MVS tracking method has the comparable performance with some state-of-the-art methods under various challenging conditions such as avoiding occlusions when objects under movement and illumination changes due to varying environmental conditions. Moreover, the MVS can be considered as a framework in which other views of features with more expressive power can also be embedded in to the hardware such as system on chip (SOC) and can improve the speed and performance in the future.

References

[1] Sand, P. and Teller, S. (2008) Particle Video: Long-Range Motion Estimation Using Point Trajectories. *International Journal of Computer Vision*, **80**, 72-91. http://dx.doi.org/10.1007/s11263-008-0136-6

[2] Buehler, P., Everingham, M., Huttenlocher, D.P. and Zisserman, A. (2008) Long Term Arm and Hand Tracking for Continuous Sign Language TV Broadcasts. *British Machine Vision Conference*, Leeds, 1-4 September 2008, 1105-1114. http://dx.doi.org/10.5244/c.22.110

[3] Bibby, C. and Reid, I. (2010) Real-Time Tracking of Multiple Occluding Objects Using Level Sets. 2010 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), San Francisco, 13-18 June 2010, 1307-1314. http://dx.doi.org/10.1109/CVPR.2010.5539818

[4] Turk, M. and Pentland, A. (2001) Eigen Faces for Recognition, *Journal of Cognitive Neuroscience*, **3**, 71-86. http://dx.doi.org/10.1162/jocn.1991.3.1.71

[5] Huber, E. (1996) 3-D Real-Time Gesture Recognition Using Proximity Space. *Proceedings of the International Conference on Pattern Recognition*, Vienna, 2-4 December 1996, 136-141. http://dx.doi.org/10.1109/acv.1996.572020

[6] Iwasawa, S., Ebihara, K., Ohya, J. and Morishima, S. (1997) Real-Time Estimation of human Body Posture from Monocular Thermal Images. *IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, 17-19 June 1997, 15-20. http://dx.doi.org/10.1109/CVPR.1997.609290

[7] Leung, M.K. and Yang, Y.H. (1995) First Sight: A Human-Body Outline Labeling System. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, 359-377. http://dx.doi.org/10.1109/34.385981

[8] Lepetit, V. and Fua, P. (2006) Keypoint Recognition Using Randomized Trees. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **28**, 1465-1479. http://dx.doi.org/10.1109/TPAMI.2006.188

[9] Adam, A., Rivlin, E. and Shimshoni, I. (2006) Robust Fragments-Based Tracking Using the Integral Histogram. *IEEE Conference on Computer Vision and Pattern Recognition*, **1**, 798-805. http://dx.doi.org/10.1109/cvpr.2006.256

[10] Etemad, K. and Pentland, A. (1997) Discriminant Analysis for Recognition of Human Faces Images. *Journal of the Optical Society of America*, **14**, 1724-1733. http://dx.doi.org/10.1364/JOSAA.14.001724

[11] Randen, T. and Husoy, J.H. (1999) Filtering for Texture Classification: A Comparative Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**, 291-310. http://dx.doi.org/10.1109/34.761261

[12] Black, M.J. and Jepson, A.D. (1998) Eigen Tracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *International Journal of Computer Vision*, **26**, 63-84. http://dx.doi.org/10.1023/A:1007939232436

[13] Jepson, A.D., Fleet, D.J. and El Maraghi, T.F. (2001) Robust Online Appearance Model for Visual Tracking. 2001 *IEEE Conference on Computer Vision and Pattern Recognition*, **25**, 1296-1311. http://dx.doi.org/10.1109/cvpr.2001.990505

[14] Lim, J., Ross, D., Lin, R. and Yang, M. (2004) Incremental Learning for Visual Tracking. *Advances in Neural Information Processing Systems* (*NIPS*), **17**, 793-800.

[15] Lee, K.C. and Kriegman, D. (2005) Online Learning of Probabilistic Appearance Manifolds for Video Based Recognition and Tracking. 2005 *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, 20-25 June 2005, 852-859.

[16] Elgammal, A. (2005) Learning to Track: Conceptual Manifold Map for Closed-Form Tracking. 2005 *IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, 20-25 June 2005, 724-730. http://dx.doi.org/10.1109/CVPR.2005.209

[17] Chum, O. and Zisserman, A. (2007) An Exemplar Model for Learning Object Classes. 2007 *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, 17-22 Jun 2007, 1-8. http://dx.doi.org/10.1109/CVPR.2007.383050

[18] Frome, A., Singer, Y. and Malik, J. (2007) Image Retrieval and Classification Using Local Distance Functions. *Proceedings of Advances in Neural Information Processing Systems*, Vancouver, 4-7 December 2006, 417-424.

[19] Frome, A., Singer, Y., Sha, F. and Malik, J. (2007) Learning Globally-Consistent Local Distance Functions for Shape-Based Image Retrieval and Classification. 11*th International Conference on Computer Vision*, Rio de Janeiro, 14-21 October 2007, 1-8. http://dx.doi.org/10.1109/iccv.2007.4408839

[20] Commaniciu, D., Ramesh, V. and Meer, P. (2003) Kernel-Based Object Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**, 564-577. http://dx.doi.org/10.1109/TPAMI.2003.1195991

[21] Sun, X., Yao, H. and Zhang, S. (2011) A Novel Supervised Level Set Method for Non-Rigid Object Tracking. *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Colorado Springs, 20-25 June 2011, 3393-3400. http://dx.doi.org/10.1109/cvpr.2011.5995656

[22] Black, M.J. and Jepson, A.D. (1998) Eigen Tracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *International Journal of Computer Vision*, **26**, 63-84. http://dx.doi.org/10.1023/A:1007939232436

[23] Wolf, J.K., Viterbi, A.M. and Dixson, G.S. (1989) Finding the Best Set of K Paths through a Trellis with Application to Multi Target Tracking. *IEEE Transactions on Aerospace and Electronic Systems*, **25**, 287-295. http://dx.doi.org/10.1109/7.18692

[24] Hampapur, A., Brown, L., Connell, J., Ekin, A., Haas, N., Lu, M., Merkl, H., Pankanti, S, Senior, A., Shu, C.F. and Tian, Y.L. (2005) Smart Video Surveillance. *IEEE Signal Processing Magazine*, **22**, 38-51. http://dx.doi.org/10.1109/MSP.2005.1406476

[25] Vedaldi, A. and Fulkerson, B. (2008) An Open and Portable Library of Computer Vision Algorithms. http://www.vlfeat.org/

[26] Rivlin, A.E. and Shimshoni, I. (2006) Robust Fragments-Based on Tracking Using the Integral Histogram. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, 798-805.

[27] Ross, D., Lim, J., Lin, R. and Yang, M. (2008) Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision*, **77**, 125-141. http://dx.doi.org/10.1007/s11263-007-0075-7

[28] Babenko, B., Yang, M.-H. and Belongie, S. (2009) Visual Tracking with Online Multiple Instance Learning. *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Miami, 20-25 June 2009, 983-990.

[29] Grabner, H., Grabner, M. and Bischof, H. (2006) Real-Time Tracking via On-Line Boosting. *Proceedings of the British Machine Vision Conference* (*BMVC*), **1**, 47-56.

[30] Mei, X. and Ling, H. (2009) Robust Visual Tracking Using l1 Minimization. *IEEE International Conference on Computer Vision* (*ICCV*), Kyoto, 29 September-2 October 2009, 1436-1443.

[31] Kwon, J. and Lee, K.M. (2011) Tracking by Sampling Trackers. *IEEE International Conference on Computer Vision* (*ICCV*), Barcelona, 6-13 November 2011, 1195-1202.

[32] Kalal, Z., Mikolajczyk, K. and Matas, J. (2012) Tracking-Learning-Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**, 1409-1422. http://dx.doi.org/10.1109/TPAMI.2011.239

[33] Zhang, T., Ghanem, B., Liu, S. and Ahuja, N. (2012) Robust Visual Tracking via Multi-Task Sparse Learning. 2012 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Providence, 16-21 June 2012, 2042-2049. http://dx.doi.org/10.1109/CVPR.2012.6247908

[34] Zhang, K., Zhang, L. and Yang, M.-H. (2012) Real-Time Compressive Tracking. *Proceedings of the European Conference on Computer Vision* (*ECCV*), Florence, 7-13 October

2012, 864-877. http://dx.doi.org/10.1007/978-3-642-33712-3_62

[35] Kwon, J. and Lee, K.M. (2010) Visual Tracking Decomposition. 2010 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), San Francisco, 13-18 June 2010, 1269-1276. http://dx.doi.org/10.1109/CVPR.2010.5539821

[36] Hare, S., Saffari, A. and Torr, P.H.S. (2011) Struck: Structured Output Tracking with Kernels. *IEEE International Conference on Computer Vision* (*ICCV*), Barcelona, 6-13 November 2011, 263-270.

[37] Yao, R., Shi, Q., Shen, C., Zhang, Y. and van den Hengel, A. (2013) Part-Based Visual Tracking with Online Latent Structural Learning. 2013 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Portland, 23-28 June 2013, 2363-2370. http://dx.doi.org/10.1109/CVPR.2013.306

[38] Luka, Matej, K. and Ales, L. (2014) Is My New Tracker Really Better than Yours? 2014 *IEEE Winter Conference on Applications of Computer Vision* (*WACV*), Steamboat Springs, 24-26 March 2014, 540-547.