DOI: 10.21917/ijivp.2011.0039

HUMAN OBJECT TRACKING IN VIDEO SEQUENCES

S. Saravanakumar[1], A. Vadivel[2] and C.G. Saneem Ahmed[3]

[1, 2]*Department of Computer Applications, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India*

E-mail: [1]somansk@yahoo.com, [2]vadi@nitt.edu

[3]*Department of Electronics and Communication Engineering, National Institute of Technology, Tiruchirappalli, Tamil Nadu, India*

E-mail: saneem89@gmail.com

Abstract

The object representation and tracking is one of the important tasks in computer vision. The object can be represented in various ways and in this paper the objects are represented using the properties of the HSV color space. Adaptive k-means clustering algorithm was applied to cluster objects centroids color values and co-ordinates were sent to next frame for clustering. After clustering, for comparing the objects present in both the reference frame and the target frame, a similarity measure was proposed which uses position, color and size of the objects for comparison. Based on the similarity value, the objects were detected and tracked. The performance of the proposed approach was verified with human objects and the same was effectively tracked. The comparison was carried with similar methods and the results are encouraging.

Keywords:

Object Tracking, HSV Color Space, Human Object Tracking, Similarity Matching

1. INTRODUCTION

In Computer Vision, object tracking is considered as one of the important tasks. Various methods have been proposed and reported both in academia and industry for large number of real-time applications. All the object tracking methods may broadly be categorized as template-based, probabilistic and pixel-wise. While the template-based method represents the object in a suitable way for tracking, the probabilistic method uses intelligent searching strategy for tracking the target object. Similarly, the similarity matching techniques are used for tracking the target object in pixel-based methods. However, among all the above said approaches, the template-based approach is found to be suitable for many real-time applications [1], [2]. In this category of tracking methods, similarity of the predefined target is being calculated with the object translation. However, for object transformations such as translation, rotation and scaling this method often fails. This is due to the fact that the procedures of selection of target object as constant size templates. For handling this unwanted situation, varying templates are used. The inclusion of background pixels into the template introduces the problem of positioning error and the positioning error continuously getting added while updating the template.

In template-based approach category, mean-shift method [3] and Kernel-based tracking method [4] have been proposed, where the color histograms of the target object is constructed using a Kernel density estimation function. Since, the color histogram is invariant feature for rotation, scaling and translation, it is considered as one of the suitable features for handling the problem of change in the scale, rotation and translation of target object. The object tracking is carried out by comparing the color histogram of the template and the target object. However, mean-shift method is not suitable for 3-D target object and monochromatic object. In case of monochromatic target object, even small variation in illumination, produces narrow histogram pattern and tracking often fails.

In object tracking problem, the object representation is the one of the important and difficult aspects. Various ways of representing or describing target object have been proposed such as object appearance [1], [2], image features [5], [6], target contour [7], [8] and color histogram [4]. In both appearance-based and color histogram based approaches, the region of the object has to be defined for describing the target. Thus, if some of the background pixels are mixed with the defined region, the tracking may fail.

While tracking non-rigid objects, the probabilistic based tracking methods have given better performance. Some of the approach in this category can be found in [9] - [12]. In one of the probabilistic methods [9], the factors such as motion detector, region tracker, head detector and active shape tracker have been combined for tracking the pedestrian. The assumption made in this method is that there are no people moving in the background. Since, this method uses contour as one of the feature, initial contour definition is difficult for the complicated contour target object.

Object tracking is also performed by predicting the object position from the past information and the predicted current position. These types of methods combine both statistical computation and the parameter vector [13] - [16]. However, for real-time object tracking systems, it has been found to be difficult for constructing the proper feature vectors. This method has been extended by Khan, et al. [13], for dealing with the problem of interacting targets. The Markov Random Field (MRF) has been used for modelling the interactions. This has been achieved by adding an interaction weighted factor. However, in this method the tracking fails while there is an overlap between targets.

In contrast to model-based tracking methods, the pixel-wise tracking methods are data-driven methods. In pixel-wise tracking method, prior model of the target is not required. A parallel K-means clustering algorithm [17] has been used by Heisele, *et. al.* [18], [19] for segmenting the color image sequence and moving region is identified as target. However, the method is computationally expensive due to large number of clusters. Similarly, another K-means based autoregressive model has been proposed and the clustering is performed only to the positive samples. Thus, the tracking failure can't be detected and the failure recovery may not be possible. For tracking, the image pixels are divided as target and non-target pixels and K-means clustering algorithm is applied on these pixels [20]. However, this method can't deal with the appearance changes of the target

object such as size, pose, etc. In addition, the computational cost is proportional to the number of non-target points.

It is understood from the above discussion that pixel-based methods are robust against the background interfusion methods. In this kind of method, the failure detection and automatic failure recovery can be carried out effectively.

A very fundamental and critical task in computer vision is the detection and tracking of moving objects in video sequences. Possible applications are as follows; 1) Visual surveillance: A human action recognition system process image sequences captured by video cameras monitoring sensitive areas such as bank, departmental stores, parking lots and country border to determine whether one or more humans engaged are suspicious or under criminal activity. 2) Content based video retrieval: A human behavior understanding system scan an input video, and an action or event specified in high-level language as output. This application will be very much useful for sportscasters to retrieve quickly important events in particular games. (3) Precise analysis of athletic performance: Video analysis of athlete action is becoming an important tool for sports training, since it has no intervention to the athletic.

In all these applications fixed cameras are used with respect to static background (e.g. stationary surveillance camera) and a common approach of background subtraction is used to obtain an initial estimate of moving objects. First perform background modeling to yield reference model. This reference model is used in background subtraction in which each video sequence is compared against the reference model to determine possible variation. The variations between current video frames to that of the reference frame in terms of pixels signify existence of moving objects. The variation which also represents the foreground pixels are further processed for object localization and tracking. Ideally, background subtraction should detect real moving objects with high accuracy and limiting false negatives (not detected) as much as possible. At the same time, it should extract pixels of moving objects with maximum possible pixels, avoiding shadows, static objects and noise.

In the detection of shadows the foreground objects are very common, producing undesirable consequences. For example, shadows connect different people walking in a group, generating a single object (typically called blob) as output of background subtraction. In such case, it is more difficult to isolate and track each person in the group. There are several techniques for shadow detection in video sequences [21] – [23].

The main objective of this paper is to extract features of objects present in video frames using the properties of the HSV color space and track the same object in subsequent video frames by considering human as target object.

In this paper we developed two steps, first adaptive k-means clustering, it is sent to next frame cluster objects centroids color values and co-ordinates for clustering current frame. Second step, after clustering current frame, for comparing the objects present in both reference fame and target frame, we propose a similarity measure, which uses position, color and size of the objects for comparison. Based on the similarity value, the objects are detected and tracked.

The rest of the paper is organized as follows. In section 2 the object segmentation of the video frames from the HSV color space is discussed. The similarity measure is presented in section 3. The experimental results are given in section 4, and we conclude the paper in the last section.

2. OBJECT SEGMENTATION OF VIDEO FRAMES FROM THE HSV COLOR SPACE

2.1 HSV COLOR SPACE PROPERTIES

A three dimensional representation of the HSV color space is a hexacone, with the central vertical axis representing intensity. Hue is defined as an angle in the range [0, 2π] relative to the red axis with red at angle 0, green at $2\pi/3$, blue at $4\pi/3$ and red again at 2π. Saturation is the purity of color and is measured as a radial distance from the central axis with values between 0 at the center to 1 at the outer surface. Any color in the HSV space can be transformed to a shade of gray by sufficiently lowering the saturation. The value of intensity determines the particular gray shade to which this transformation converges. Saturation gives an idea about the depth of color and human eye is less sensitive to its variation compared to variation in hue or intensity. We, therefore, use the saturation of a pixel to determine whether the hue or the intensity is more pertinent to human visual perception of the color of that pixel and ignore the actual value of the saturation. For low saturation, a color can be approximated by a gray value specified by the intensity level while for higher saturation, the color can be approximated by its hue. The saturation threshold that determines this transition is once again dependent on the intensity. For low intensities, even for a high saturation, a color is close to the gray value and vice versa. it is observed that for higher values of intensity, a saturation of about 0.2 differentiates between hue and intensity dominance. Assuming the maximum intensity value to be 255, we use the following threshold function to determine if a pixel should be represented by hue or intensity as its dominant feature.

$$th_{sat}(V) = 1.0 - \frac{0.8V}{255} \quad (1)$$

Thus, we treat each pixel in an image either as a "true color" pixel – a pixel whose saturation is greater than $th_{sat}(V)$ and hence, its hue is the dominant component or as a "gray color" pixel – a pixel whose saturation is less than $th_{sat}(V)$ and hence, its intensity is the dominant component. This method of separating true color pixels from gray color pixels using saturation is a novel concept and it achieves image segmentation that is useful for object tracking. First of all, the sensitivity to intensity variation, which is a drawback of most of the pixel domain object tracking techniques, is reduced to a great extent. Secondly, since temporally close video frames have high object level similarity, except when there is an intervening shot boundary, an object-level representation of the video frames gives more robust method for object comparison for tracking [24]. Finally, this method is similar to the way humans perceive object presents and shot changes in video. Human eyes perceive a change in object movement only when objects present in a frame differ considerably from its previous frame.

2.2 FEATURE EXTRACTION

The visual properties of the HSV color space was effectively used and described in the above section for object

representation. Each frame can be represented as a collection of its pixel features as follows,

$$F \equiv \{(pos, [t/g], val)\} \quad (2)$$

here each pixel is a triplet where pos denotes the position of the pixel, $[t/g]$ denotes whether the pixel is a "true color" pixel or a "gray color" pixel and val denotes the "true color" value or the "gray color" value. Thus, $val \in [0, 2\pi]$ if $[t/g]$ takes a value of t and $val \in [0, 255]$ if $[t/g]$ takes a value of g. The feature of a pixel is the pair ($[t/g]$, val) – whether it is a "true color" pixel or a "gray color" pixel and the corresponding hue or intensity value.

2.3 PIXEL GROUPING BY K-MEANS CLUSTERING ALGORITHM

Once we have extracted each pixel feature in the form of ($[t/g]$, val), a clustering algorithm is used to group similar feature values. The clustering problem is to represent the frame as a set of n non-overlapping partitions as follows,

$$F \equiv \{O_1 \mid O_2 \mid O_3 \mid \ldots \mid O_n\} \quad (3)$$

Here each $O_i \equiv ([t/g], val, \{pos\})$, i.e., each partition is either a "true color" partition or a "gray color" partition and it consists of the positions of all the image pixels that have colors close to val. We use K-Means clustering for pixel grouping. The "true color" and the "gray color" pixels are clustered separately. In the K - Means clustering algorithm, we start with K = 2 and adaptively increase the number of clusters till the improvement in error falls below a threshold or a maximum number of clusters is reached. The maximum number of clusters is determined by the resolution of human eye and can be derived from the NBS distance [25].

2.4 POST PROCESSING

After initial K-Means clustering, we get different cluster centres and the pixels that belong to these clusters. In Fig.1(a), we show a frame from a video. In Fig.1(b), we show the transformed image after feature extraction and K-Means clustering. It is observed that the clustering algorithm has determined five "true color" clusters, namely, Blue, Green, Orange, Yellow and Red for this particular image and three gray clusters – Black and two other shades of gray.

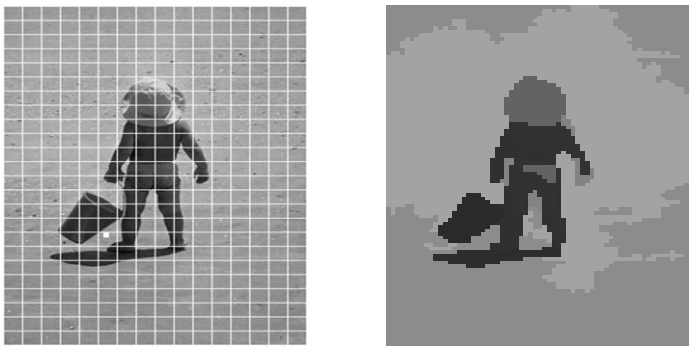


Fig.1(a) Original frame (b) Clustering and connected component analysis

However, these clustered pixels do not yet contain sufficient information about the various objects in the frame because it is not yet known if all the pixels that belong to the same cluster are actually part of the same object or not. To ascertain this, connected component analysis [26] of the pixels was performed to determine the different objects in a frame. During this process, the connected components whose size is less than a certain percentage (typically 1%) of the size of the frame was detected. These small regions are to be merged with the surrounding clusters in the next step.

3. OBJECT LEVEL FRAME-TO-FRAME SIMILARITY MEASURE AND TRACKING

A complete video V may be represented as a collection of frames and it may be represented as $V = \{F_i; i = 1, 2, 3, \ldots, M\}$ where F_1, F_2, $F_3, \ldots, F_M$ are the I-frames, M being the total number of I-frames in the video V. Once the frames are decomposed into small object to represent object-level information using the method described in the previous section, similarity between objects in the current frame and the objects in the successive frames is determined based on the object color, size and position difference. Let us consider two frames F_1 and F_2 containing n_1 and n_2 number of objects, respectively. Out of n_1 objects in frame F_1, let n_{1t}, be the number of objects of true color and let n_{1g}, be the number of objects of gray color so that $n_1 = n_{1t} + n_{1g}$. Similarly, n_{2t} and n_{2g} are defined for frame F_2. Let the objects of the two frames be named as O_{11}, O_{12}, O_{13}, $\ldots, O_{1n1}$ and O_{21}, O_{22}, O_{23}, $\ldots$, O_{2n2} respectively. It is possible that more than one object of a frame has the same true color or the same gray color value. Without loss of generality, we assume that the objects of frame, F_1, i.e., O_{11}, O_{12}, O_{13}, $\ldots$, O_{1n1} are sorted in descending order of object size S_{1i}, $i = 1, 2, \ldots, n_1$; i.e., $S_{1k} >= S_{1m}$ for $k < m$.

A standard approach for matching objects in two images is the use of the Integrated Region Matching (IRM) method [27]. In this method, each object of one image is matched with each object of the second image. However, in this approach, there is an averaging effect that often results in two completely dissimilar images being matched during object tracking retrieval. Every frame contains similar objects as its previous frame unless there is an intervening new object. So it is important that the objects in a frame are matched only with the corresponding objects in the next frame. The similarity between two frames can thus be measured as the degree of matching between their objects. It should, however, be noted that due to variation in lighting condition and object movement, some parts of an object may get obscured by another foreground object in an adjacent frame even though a new object is entered. Thus, while trying to match objects, it is possible that an object is actually broken into multiple objects or multiple objects may get merged into a single object in successive frames. Our frame matching approach takes into consideration these special characteristics of a video. The complete algorithm for object-level frame matching is shown in Fig.2, which works as follows,

For each object O_{1i} of frame F_1, we first determine the objects of F_2 that are similar in color (true or gray). To do this, objects of F_2 are sorted in descending order of their color difference from O_{1i}. Objects whose colors do not differ significantly from the color of O_{1i} are candidates for matching with O_{1i}. Out of all the candidate objects of F_2, we next consider only those objects whose centres are close to the centre of O_{1i}.

This is done to ensure that two distinctly different objects with the same color are not matched with one another. Thus, two objects are considered for matching only if their colors are less than COL_THRESHOLD separation and their centres are less than CEN_THRESHOLD separation. Typical values of these parameters are 80% and 85%, respectively. Further, if an object of F_2 is already matched with an object of F_1, it is not considered again for matching with another object of F_1. However, if an object of F_2 is partially matched with an object of F_1, the remaining part can be considered for matching with another object of F_1. Similarly, an object of F_1 may be matched with more than one object of F_2 through partial matching with each one of them. In the algorithm, any object of frame F_1, which has been matched, with objects of F_2 by a fraction of MAX_MATCH or more, is considered to be matched. This is done because two objects cannot always be matched exactly due to small camera movement or light variations. A typical value of MAX_MATCH is 90%. The amount of matching is measured as the percentage of the area of O_1 that can be matched to one or more objects of F_2.

The frame-to-frame matching between F_1 and F_2 are the sums of the object-to-object matching between F_1 and F_2. The number of pixels in each frame is fixed and has been denoted by number of frame pixels in Fig.2.

Function Object_Similarity (Frame F_1, Frame F_2)
Frame-to-frame-similarity= 0.0
for i=1 to n_1
matched_frame_1[i] = 0.0 // F_1 has n_1 objects – All unmatched initially
for j=1 to n_2
matched_frame_2[j] = 0.0 // F_2 has n_2 objects – All unmatched initially
sort objects of F_1 in descending order of size
let the sorted sequence of objects be O_{11}, O_{12}, …, O_{1n1}
for i= 1 to n_1
let count_1[i] denote the number of pixels in object O_{1i}
sort objects of F_2 in ascending order of their color similarity with O_{1i}
let the sorted sequence of objects be O_{21}, O_{22}, …, O_{2n2}
for j = 1 to n_2
let count_2[j] denote the number of pixels in object O_{2j}
if (matched_frame_1[i] < MAX_MATCH && matched_frame_2[j] < MAX_MATCH)
if ((col[O_{1i}] – col[O_{2j}])< COL_THRESHOLD)
if ((cen[O_{1i}] - cen[O_{2j}])<CEN_THRESHOLD)

$$ovelap = \frac{(1.0 - \text{matched\_frame\_2[j]}) * \text{count\_2[j]}}{(1.0 - \text{matched\_frame\_1[i]}) * \text{count\_1[i]}}$$

if (overlap < MAX_OVERLAP)
matched_frame_1[i]=matched_frame_1[i]+

$$\frac{1.0 - matched\_frame\_2[j] * count\_2[j]}{count\_1[i]}$$

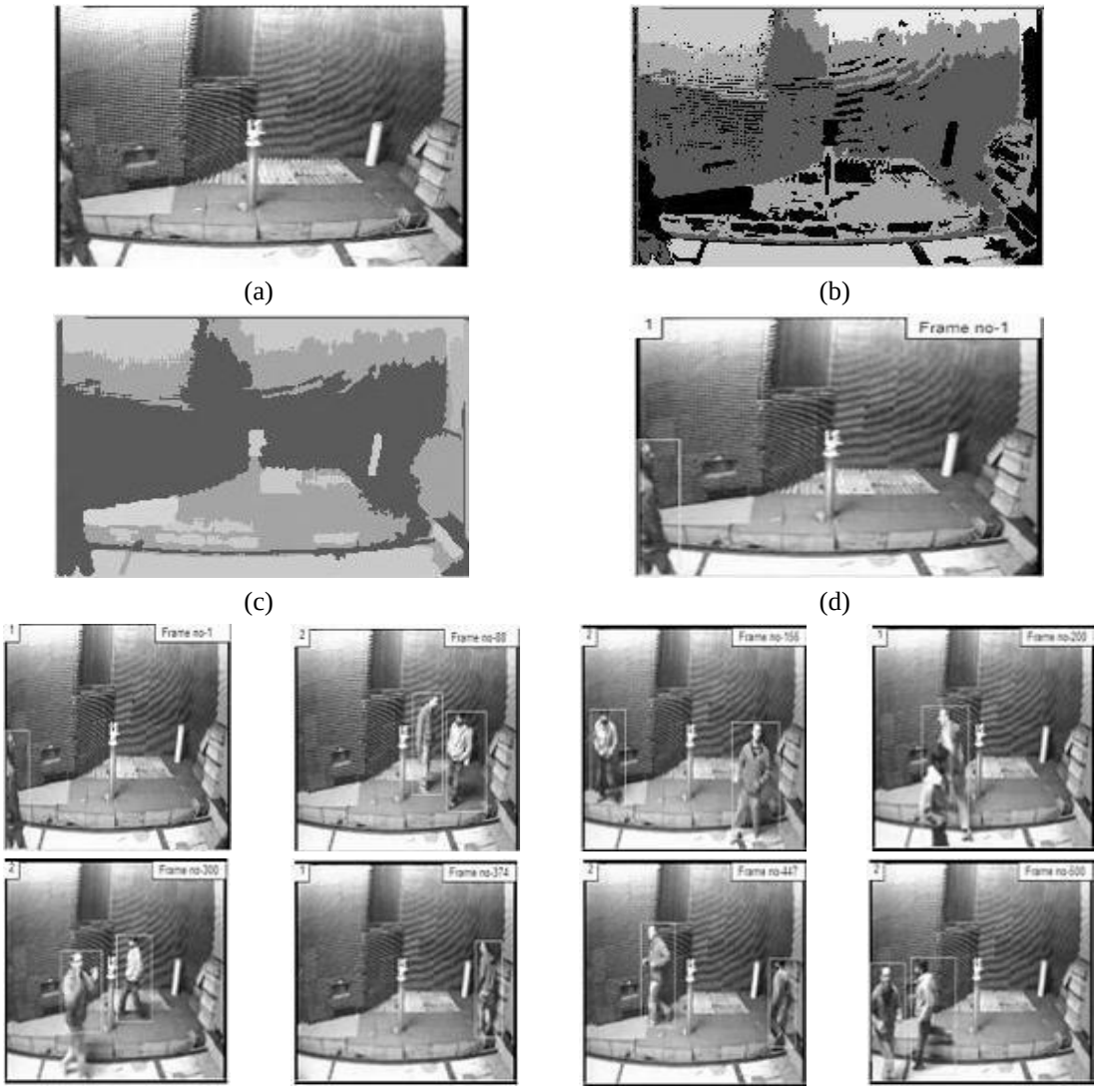
matched_frame_2[j]=1.0
else
matched_frame_2[j]=matched_frame_2[j]+ $\frac{1.0 - matched\_frame\_1[i] * count\_1[i]}{count\_2[i]}$
matched_frame_1[i]=1.0
frame-to-frame-similarity=frame-to-frame-similarity+matched_frame_1[i]*count_1[i]
frame-to-frame-similarity= frame-to-frame-similarity/NO_OF_FRAME_PIXELS
return frame-to-frame-similarity

Fig.2. Algorithm for object similarity calculation

In our approach, we considering a fixed area as background and is treated as reference frame. The feature of the object present in reference frame is extracted by using the procedure mentioned in section 2. While a new object, say for example, human entered into the frame and thus an overlap of a new object over the reference frame. This causes a significant change in the content of the current frame and thus there is a change in the content of the frame. The feature of the object present current frame and the reference frame is extracted and the similarity between these two frames is measured using the method described in the previous section. The new objects extracted from the current frame is bounded by a rectangle and tracked in consecutive frames. Subsequently, objects present in n^{th} and $(n-1)^{th}$ frames are compared for tracking the objects continuously. During comparison, as described in section 3, for measuring the similarity value, the centre position, color and the size of the objects in the nth and $(n-1)^{th}$ frames are calculated. Based the difference between the centre value of the objects, the direction of the movement of the objects is estimated and the object is tracked.

4. EXPERIMENTAL RESULTS

In this section, the experimental results were presented and the proposed method was detecting and tracking the moving objects exactly. For evaluating the performance of the proposed method, human as moving object was used for tracking. For experiments, benchmark video sequence SPEVI video dataset were used. In addition, some of proprietary video sequences were used. During tracking, the objects centre color values were passed and co-ordinates to the consecutive frames for clustering, to detect and track the human objects in the video sequences. In Fig.3, the tracking results are presented, for evaluating the performance of the proposed approach with target object moving very fast, SPEVI benchmark video sequence was used with 35 sec (25 frames per second) as duration and the result is given in Fig.3.



(a) (b) (c) (d) (e)

Fig.3. Tracking target object from the video sequence (a) Input video frame (b) After clustering (c) After connected component analysis (d) Object tracking and (e) Intermediate frames

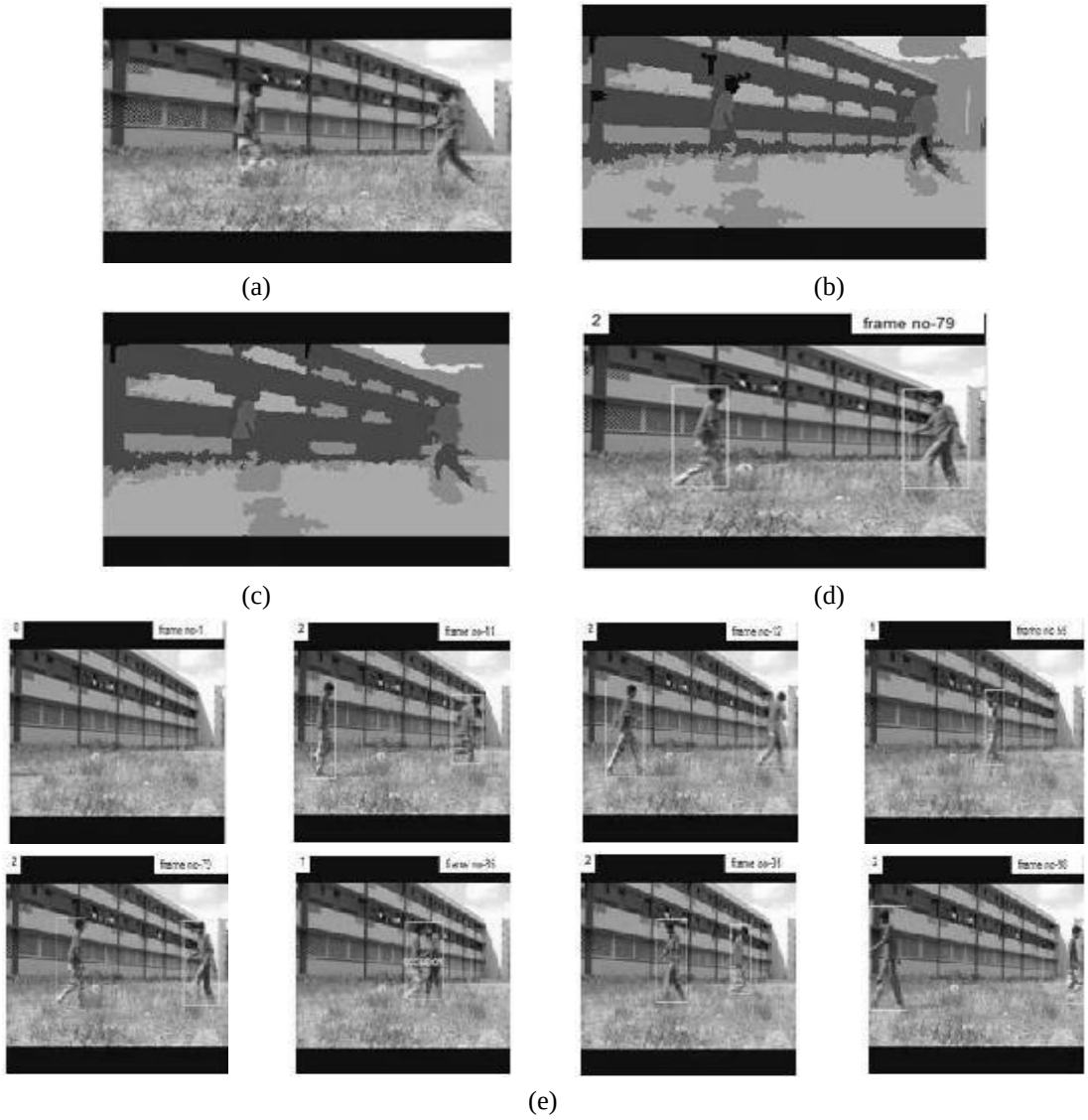
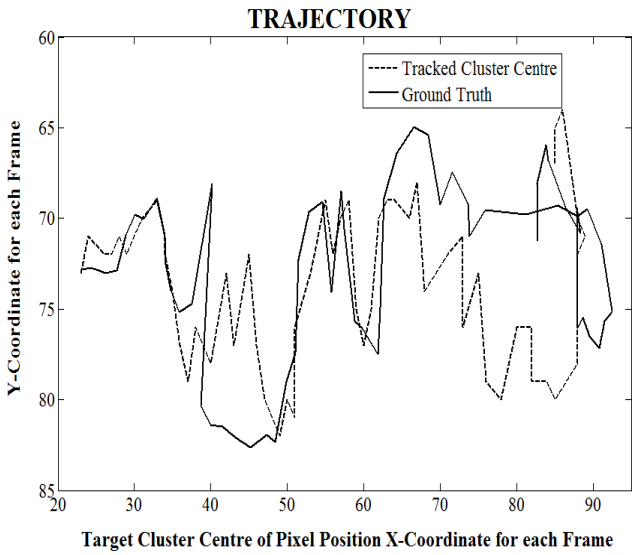


Fig.4. Tracking target object from the video sequence (a) Input video frame (b) After clustering (c) After connected component analysis (d) Object tracking and (e) Intermediate frames

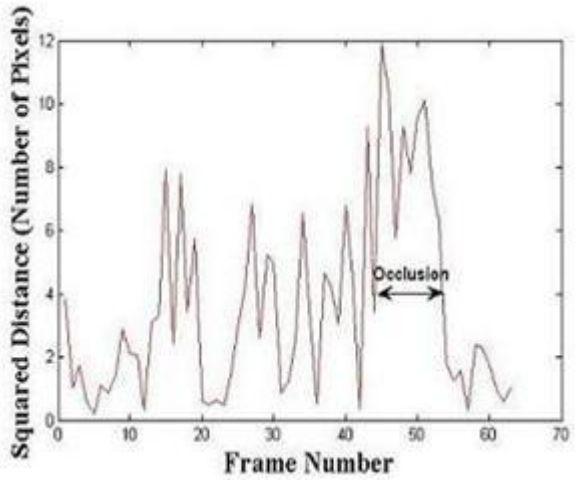
In Fig.3 and Fig.4(a), the captured sample video is shown. The reference clustered frame and CCA applied frame are shown in Fig.3 and Fig.4(b) and Fig.3 and Fig.5(c). After clustering, the CCA is applied for removing small clusters and combining them with nearest clusters. In our experiment, we have merged the small clusters with less than 1% of sum of frame pixel size. Initially, human enter into the reference frame area and thus causes change in the frame content. By detecting the change in content, the object in the reference frame and the current frame is extracted and the similarity is calculated for identifying the newly entered human object. Based on the similarity value, the object tracking is carried out. For the first time, the position of the new object is identified and the rectangular boundary of the new object is drawn for visual feeling. This is shown in Fig.3 and Fig.4(d). For drawing the boundary, we initially cover all the objects and the boundary size is large. However, after passing some number of frames, the boundary size will be conversing exactly over the human target object and can be viewed from Fig.3 and Fig.4(d) and Fig,3 and Fig.4(e).

In Fig.5, we show the target object trajectory for the experimental video sequence. It is observed from the figure that the proposed approach, tracks the target object effectively. For measuring the performance of the proposed approach, we have measured the ground truth of the sample video sequences. The trajectory value of the target object is compared with the ground truth value and is shown in Fig.5(a). It may be noticed that the trajectory value of the target obtained by proposed approach aligns with the ground truth values. In Fig.5(b), the squared distance interms of number of pixels is given. In this figure, it is noticed that the pixel wise difference is also very low and the maximum pixel difference is only 12 for the frame number 46.

The sample video is captured with 576 X720 RGB streams at a speed of 25 frames/sec, considered 875 sample video frames of with 300 X 240 (resized) and processed every i^{th} frame. The experiment is conducted in system with Intel(R) Core(TM)2 Duo CPU E7400@ 2.80GHZ processor, 2.0GB memory and 32-bit windows operating system.



(a)



(b)

Fig.5. Target object tracking in a sample video sequence
(a) Target tracked cluster centre and ground truth value
(b) Squared difference between algorithm output and ground truth value

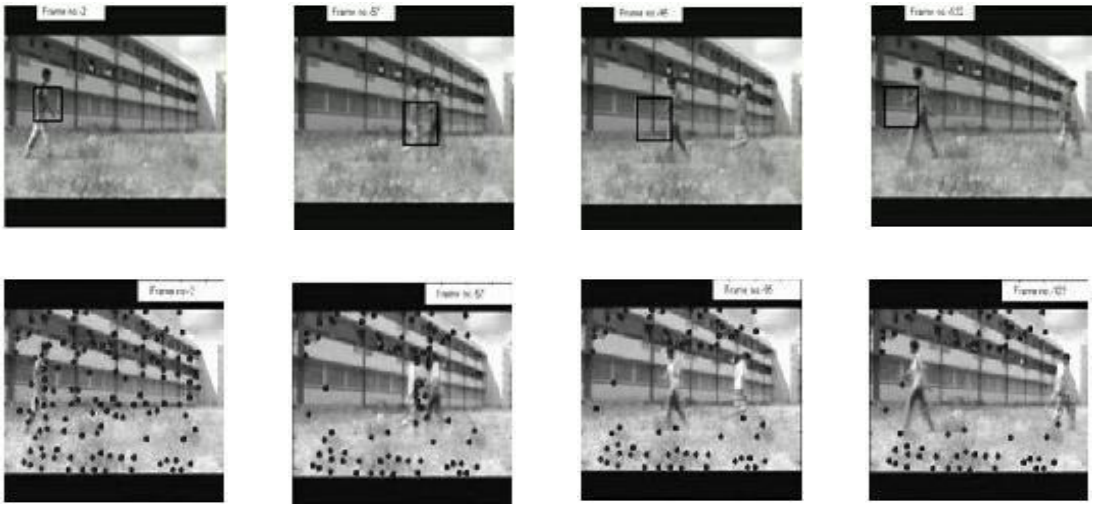
To evaluate the performance of the proposed method, we have carried out experiments and compared the results with various similar methods such as Mean-shift object tracking [3], Feature-based tracking [5] and K-means tracking [27] and result is shown in Fig.5.

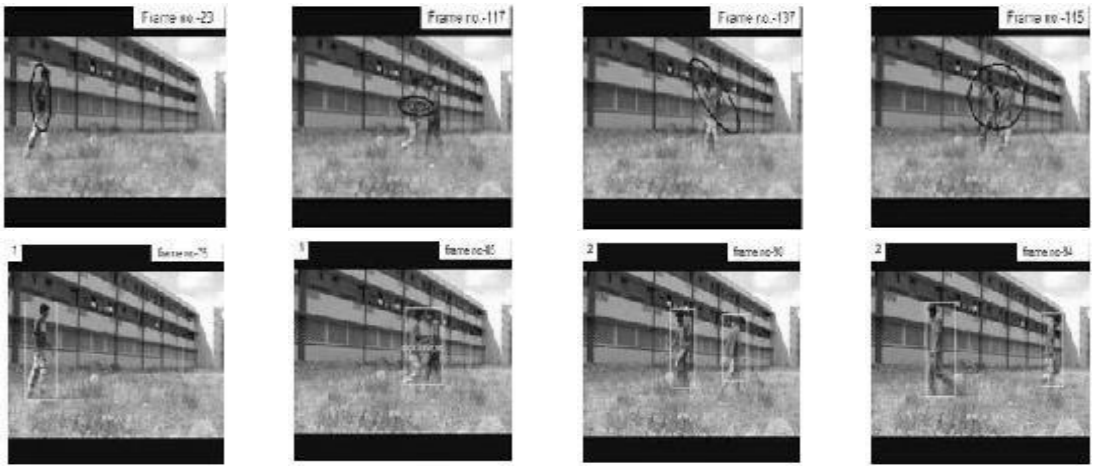
4.1 PERFORMANCE COMPARISON

To evaluate the performance of the proposed method experiments were carried out and the results were compared with various recently proposed methods and such as Mean-shift object tracking [3], Feature-based tracking [5], Particle filter [28] and K-means tracking [20] and the results are shown in Fig.6 and Fig.7.

The performance of mean-shift algorithm is shown in the first row. The second row of the figures depicts the performance of feature-based tracking, the performance of K-means tracking is shown in the third row and the performance of the proposed method is shown in last row. The mean-shift object tracking method fails after frame number 85 and the feature based method also fails after the frame number 96. This is due to the fact that, the feature points are moving away from the target object and unable to represent the object and track. Similarly, the K-means tracking method fails after 137th frame of video due failure in bounding the target object. In contrast to all these approaches, the proposed method represents and tracks the target effectively till the end of the video sequence.

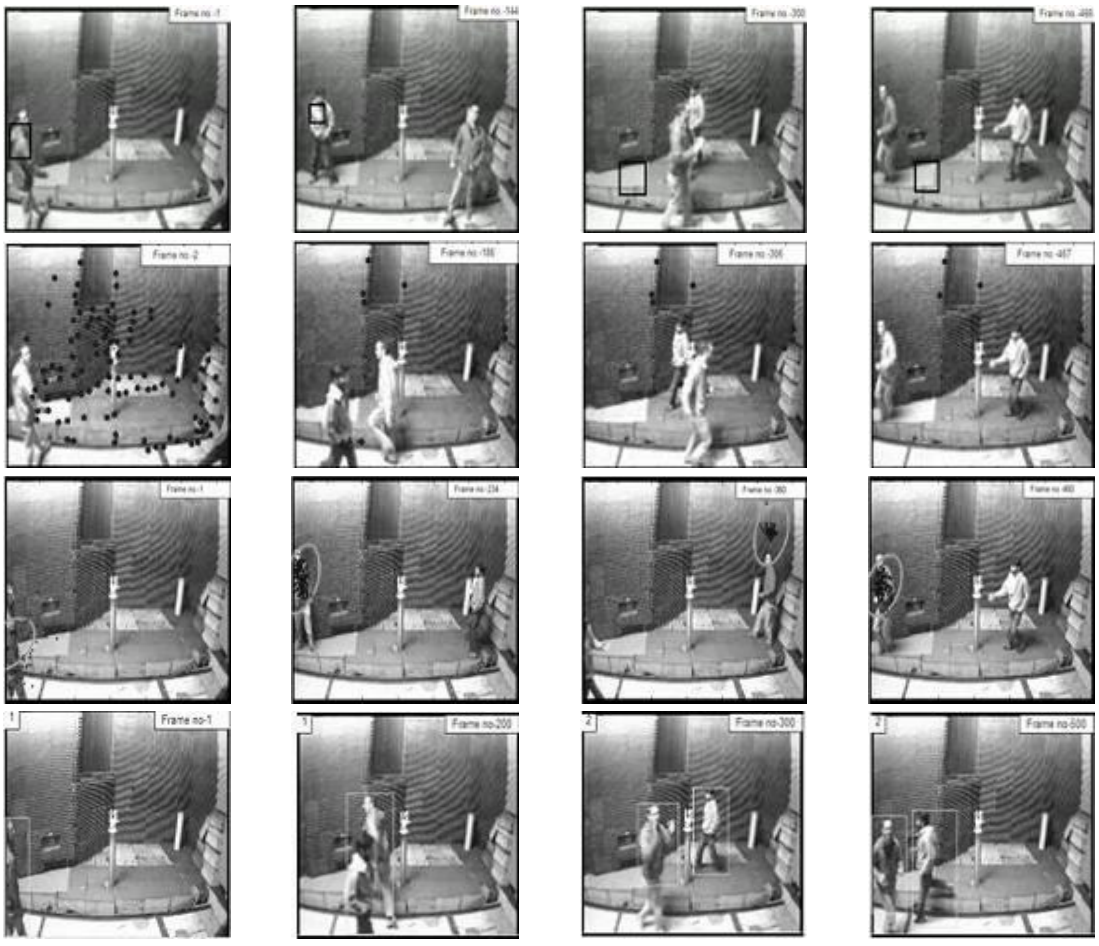
In Fig.7, the performance the proposed approach is compared with Mean-Shift object tracking [3], Feature-based tracking [5] and Particle filter [28]. In each row, the performances of the methods are shown, say for example in first row, the result of the performance of Mean-Shift object tracking method was shown. In second and third rows, the performance of feature based and particle filter based are shown. The performance of the proposed approach is depicted in the final row. While observing the performance of the all other methods, the proposed approach tracks the human objects in all the frames ever there is an intermediate failure.





(a) (b) (c) (d)

Fig.6. Comparative experiments with high-speed moving human objects (a) Frame 002, (b) Frame 057, (c) Frame 96, (d) Frame 122



(a) (b) (c) (d)

Fig.7. Comparative experiments with high-speed moving human object (a) Frame 001, (b) Frame 200, (c) Frame 300 and (d) Frame 500

Table.1 present quantitative results of the proposed approach on all datasets. The Multiple Object Tracking Accuracy *(MOTA)* takes into account false positives, missed targets and identity switches. The Multiple Object Tracking Precision *(MOTP)* is simply the average distance between true and estimated targets. Furthermore, the metrics proposed in [29] was computed, which counts the number of mostly tracked *(MT)* and partially tracked *(PT)*.

$$MOTA = \left(1 - \frac{error}{totalerror}\right) * 100\% \quad (4)$$

$$MOTP = \sum_{t=1}^{N_{frames}} \left[\frac{G^{(t)} \bigcap D^{(t)}}{G^{(t)} \bigcup D^{(t)}} \right] \quad (5)$$

$$MT > Threshold \quad (6)$$

Therefore, threshold is 50% target pixels within ellipse boundary.

$$PT < Threshold \quad (7)$$

Table.1. Quantitative results of our methods

Sequence/ Duration	Object	MOTA	MOTP	MT in No. of Frames	PT in No. of Frames
SPEVI 12.6sec to 23.8sec	A	92.6 %	96.3%	567	33
	B	97.8%	95.8%	583	17
Proprietary 1.0sec to 19.5sec	A	80.6%	88.6%	426	26
	B	80.8%	83.4%	437	15

5. CONCLUSION

In this paper, feature extraction of the objects present in video frames for representing and tracking was proposed. These features of the objects were compared for tracking the same and a novel similarity measure was proposed. The proposed feature extraction method uses the properties of the HSV color space and the changes due to illumination is effectively considered. As a future work, multiple objects will be tracked and the similarity measure will be extended accordingly.

ACKNOWLEDGMENT

The work done by Dr. A. Vadivel is supported by research grant from the Department of Science and Technology, India, under Grant DST/TSG/ICT/2009/27 dated 3rd September 2010.

REFERENCES

[1] Crane, H.D. and Steele, C.M, "Translation-tolerant Mask Matching using Noncoherent Reflective Optics", *Pattern Recognition*, Vol. 1, No. 2, pp.129-136, 1968.

[2] Grassl C, Zinsser T and Niemanr H, "Illumination Insensitive Template Matching with Hyperplanes", *Proceedings of 25th Pattern Recognition Symposium (DAGM '03)*, Vol. 2781, pp. 273-280, 2003.

[3] Comaniciu D, Ramesh V and Meer P, "Real-time Tracking of Non-rigid Objects using Mean Shift", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00)*, Vol. 2, pp. 142-149, 2000.

[4] Comaniciu, D., Ramesh, V. and Meer, P, "Kernel-Based Object Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 5, pp. 564-577, 2003.

[5] Collins, R. and Liu, Y, "Online Selection of Discriminative Tracking Feature", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, pp. 1631-1643, 2005.

[6] Nguyen H.T and Semeulders A, "Tracking aspects of the foreground against the background", *Proceedings of 8th European Conference on Computer Vision*, Vol. 3022, pp. 446-456, 2004.

[7] Kass M, Witkin A, and Terzopoulos D, "Snakes: active contour models", *International Journal of Computer Vision*, Vol. 1, No. 4, pp. 321-331, 1988.

[8] Isard M and Blake A, "Contour tracking by stochastic propagation of conditional density", *Proceedings of 4th European Conference on Computer Vision*, Vol. 1, pp.343-356, 1996.

[9] Siebel, N.T. and Maybank, S, "Fusion of Multiple Tracking Algorithms for Robust People Tracking", *Proceedings of 7th European Conference on Computer Vision*, Vol. IV, pp. 373-387, 2002.

[10] Ying Wu, Gang Hua, Ting Yu, "Switching Observation Models for Contour Tracking in Clutter", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. I-295 – I-302, 2003.

[11] Li, P and Zhang T, "Visual Contour Tracking Based on Particle Filter", *Image and Vision Computing",* Vol. 21, No. 1, pp. 111-123, 2003.

[12] Yilmaz A, Li X and Shah M, "Object Contour Tracking Using Level Sets", *Asian Conference on Computer Vision (ACCV)*, pp. 1-7, 2004.

[13] Khan, Z, Balch, T and Dellaert, F, "An MCMC-based Particle Filter for Tracking Multiple Interacting Targets", *Proceedings of 8th European Conference on Computer Vision*, Vol. 4, pp. 279-290, 2004.

[14] Tao Zhao and Ram Nevatia, "Tracking Multiple Human in Crowded Environment", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, Vol. 2, pp. 406-413, 2004.

[15] Vermaak, T, Perez, P, Gangnet, M and Blake, A, "Towards Improved Observation Models for Visual Tracking: Selective Adaptation", *Proceedings of 7th European Conference on Computer Vision*, Vol. 1, pp. 645-660, 2002.

[16] Sidenbladh, H and Black, M. J, "Learning Image Statistics for Bayesian Tracking", *IEEE International Conference on Computer Vision (ICCV)*, Vol. 2, pp. 709-716, 2001.

[17] Hartigan, J and Wong, M, "Algorithm AS136: A K-Means Clustering Algorithm", *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 28, No. 1, pp.100-108, 1979.

[18] Heisele B, "Motion-based Object Detection and Tracking in Color Image Sequence", *4th Asian Conference on Computer Vision*, pp. 1028-1033, 2000.

[19] Heisele, B, Kressel, U and Ritter, W, "Tracking Non-Rigid Moving Objects Based on Color Cluster Flow", *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 253-257, 1997.

[20] Hua, C, Wu, H, Chen, Q, and Wada, T, "K- means Tracker: A General Algorithm for Tracking People", *Journal of Multimedia*, Vol. 1, No. 4, pp. 46-53, 2006.

[21] Chien, S.Y, Ma, S.Y, and Chen, L.G, "Efficient moving object segmentation algorithm using background registration technique", *IEEE Transaction on Circuits and Systems for Video Technology*, Vol. 12, No. 7, pp. 577-586, 2002.

[22] Cucchiara, R, Grana, C, Piccardi, M, and Prati, A, "Detecting moving objects, ghosts, and shadows in video streams", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 25, No. 10, pp. 1337-1342, 2003.

[23] Grest, D, Frahm, J.M, and Koch. R, "A color similarity measure for robust shadow removal in real time", *Proceedings of Vision, Modeling and Visualization*, pp. 253-260, 2003.

[24] Vadivel, A, Mohan, M, Sural Shamik and Majumdar, A. K, "Object Level Frame Comparison for Video Shot Detection", *IEEE Workshop on Motion Video Computing*, Vol. 2, pp. 235 – 240, 2005.

[25] Gong, Y, Proietti, G, and Faloutsos, C, "Image Indexing and Retrieval based on Human Perceptual Color Clustering", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp.578-583, 1998.

[26] Stockman, G, and Shapiro, L, "*Computer Vision*", Prentice Hall, 2001.

[27] Wang, J.Z, Li, J, and Wiederhold, G, "SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 9, pp.947-963, 2001.

[28] Arulampalam, M.S, Maskell, S, Gordon, N, and Clapp, T, "A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking", *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, pp. 174–188, 2002.

[29] Li, Y, Huang, C and Nevatia, R, "Learning to associate: 'HybridBoosted multi-target tracker for crowded scene", *Proceedings of IEEE Conf. Computer Vision and Pattern Recognition (CVPR'09)*, Vol. 2, pp. 142-149, 2009.

[30] http://www.eecs.qmul.ac.uk/~andrea/spevi.html