Journal of Signal and Information Processing, 2014, 5, 97-103
Published Online November 2014 in SciRes. http://www.scirp.org/journal/jsip
http://dx.doi.org/10.4236/jsip.2014.54012

Rebound of Region of Interest (RROI), a New Kernel-Based Algorithm for Video Object Tracking Applications

Andres Alarcon Ramirez, Mohamed Chouikha

Department of Electrical and Computer Engineering, Howard University, Washington DC, USA
Email: alarconandres2001@gmail.com, mchouikha@howard.edu

Received 2 August 2014; revised 1 September 2014; accepted 27 September 2014

Abstract

This paper presents a new kernel-based algorithm for video object tracking called rebound of region of interest (RROI). The novel algorithm uses a rectangle-shaped section as region of interest (ROI) to represent and track specific objects in videos. The proposed algorithm is constituted by two stages. The first stage seeks to determine the direction of the object's motion by analyzing the changing regions around the object being tracked between two consecutive frames. Once the direction of the object's motion has been predicted, it is initialized an iterative process that seeks to minimize a function of dissimilarity in order to find the location of the object being tracked in the next frame. The main advantage of the proposed algorithm is that, unlike existing kernel-based methods, it is immune to highly cluttered conditions. The results obtained by the proposed algorithm show that the tracking process was successfully carried out for a set of color videos with different challenging conditions such as occlusion, illumination changes, cluttered conditions, and object scale changes.

Keywords

Video Object Tracking, Cluttered Conditions, Kernel-Based Algorithm

1. Introduction

Video object tracking can be defined as the detection of an object in the image plane as it moves around the scene. This topic has a growing interest for both civilian and military applications, such as automated surveillance, video indexing, human-computer interaction (gesture recognition), meteorology, and traffic management system [1]-[3]. Object tracking algorithms strive to detect a determined object through a sequence of images.

How to cite this paper: Ramirez, A.A. and Chouikha, M. (2014) Rebound of Region of Interest (RROI), a New Kernel-Based Algorithm for Video Object Tracking Applications. *Journal of Signal and Information Processing*, **5**, 97-103.
http://dx.doi.org/10.4236/jsip.2014.54012

Regularly, these object tracking algorithms use and correlate many pre-processing tasks, such as motion estimation and image segmentation.

The process of tracking an object in a sequence of frames is directly dependant on the object representation being used. Some representations, for example, use interest points to identify the object to be tracked [4]. These interest points can be detected by using information based on differentiation operators [5] [6], where changes in intensity between two adjacent pixels can emphasize the boundaries of the object of interest in the image. Other object representations use the object's silhouette or object's contour to extract information about the general shape of the object [7] [8].

Cross-correlation [9], on the other hand, was used to implement a face tracking algorithm for video conferencing environment. This method compares a region of the image with a known signal extracted from the object of interest, and then a measure of similarity is used to determine the exact position of the object being tracked in the next frame.

Aggarwal *et al.* [10] presented a methodology for video object tracking that was constituted by four steps, namely, background subtraction, candidate object identification, target object selection, and motion interpolation. Hossein and Bajie proposed a framework [11] for tracking moving objects based on spatio-temporal Markov Random field, and where were taken into account the spatial and temporal aspects of the object's motion. Chun-Te *et al.* [12] used projected gradient to help multiple inter-related kernels in finding the best match during tracking under predefined constraints.

The main advantage of the proposed algorithm is that unlike existing kernel-based methods it is immune to highly cluttered conditions. Our strategy is based on the analyses of the changes that occur within the object being tracked, itself, ignoring the high variability that commonly is presented in the environment that surrounds the object being tracked. This novel strategy makes our algorithm more robust than the existing kernel-based methods in cluttered conditions.

This paper is organized as follows: In Section 2, we present the novel proposed algorithm to track an object through video sequences. In Section 3, it is shown the obtained results. Finally, Section 4 presents the conclusions of this work.

2. Description of the Proposed Algorithm

Two important stages constitute the proposed algorithm, namely, the motion estimation stage and the matching estimation stage. The first stage seeks to obtain an estimation of the direction of the object's motion. The latter stage determines the location of the object being tracked in the next frame by evaluating a function of dissimilarity.

2.1. Motion Estimation Stage

The motion estimation stage seeks to obtain an educated guess of the direction of the object's motion. To do this, it is used a square section called region of interest (ROI) to represent the object being tracked. Additionally, the ROI, which is placed in the object being tracked in the current frame, is also located in the same position but in next frame. Then, it is analyzed the changing pixels in the ROIs to estimate the direction of the object's motion.

As a way of explaining the proposed algorithm, it is shown in the **Figure 1(a)** the current frame of a virtual video where the object to be tracked is represented by a region of interest. In the same way, the **Figure 1(b)**

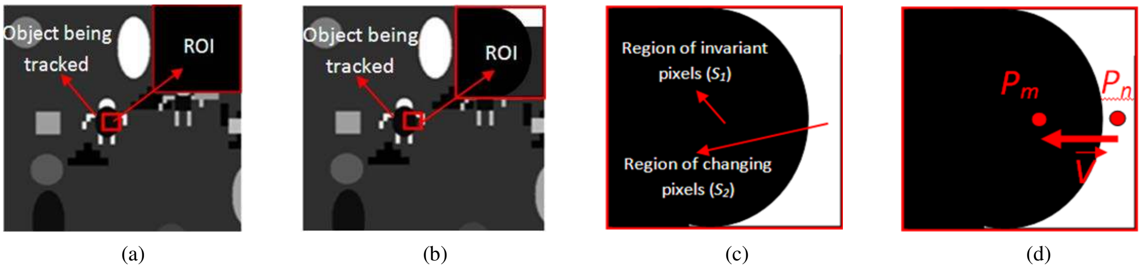

Figure 1. Representation of object being tracked by regions of interest (a) ROI in the current frame; (b) ROI in the second frame; (c) Binary image used to detect the object's motion; (d) Vector that defines the direction of the object's motion.

shows the next frame of the video where the object is moved three pixels to the right. As consequence of the object's motion between these two frames, some of the pixels that initially belong to the object in the first ROI turn into pixels that are part of the background in the second ROI. These pixels are called changing pixels. On the other hand, the pixels that belong to the object in the first ROI as well as in the second ROI are called invariant pixels. The proposed methodology initially seeks to generate a binary image such as the one shown in the **Figure 1(c)**, where it is distinguish the changing pixels from the invariant pixels in the regions of interest. Then, it is obtained an educated guess of the direction of the object's motion by conducting a geometrical analysis over the regions that corresponds to both the changing and the invariant pixels.

To distinguish the changing pixels from the invariant pixels in the regions of interest, it is initially subtracted the second ROI from the first ROI. Thus, it is created a new image where the changing pixels are enhanced with respect to the invariant pixels. In other words, the changing pixels in this new image have greater values in magnitude than the invariant pixels. Additionally, the pixels located in the central zone of the new image have more chances of being part of the invariant pixels than the pixels located in the outer side of the new image. Therefore, it is used the weighted mean as well as the weighted standard deviation, which emphasizes the central pixels and less importance on the outer pixels in the image, to characterize the invariant pixels in the image and obtain a threshold that allows us to distinguish the invariant pixels from the changing pixels in the new image.

To calculate the weighted mean of the image's pixels obtained after subtracting the second ROI from the first ROI, it is used a truncated Gaussian mask that contains the weights that are used to give more importance to the pixels located in the center part of the new image, and less weight to the extreme pixels of this image. The weighted mean is defined by the following equation:

$$\mu = \frac{\sum_{i=1}^{L}\sum_{j=1}^{T} F\left(x_i, y_j\right) * G\left(x_i, y_j\right)}{\sum_{i=1}^{L}\sum_{j=1}^{T} G\left(x_i, y_j\right)} \tag{1}$$

where $F(:)$ represents the new image, and G is a Gaussian mask defined by the following equation:

$$G\left(x_i, y_j\right) = \frac{1}{\sqrt{2\pi} * (L/2)} * \mathrm{e}^{-\left(\left(x^2+y^2\right)/2(L/2)^2\right)} \tag{2}$$

The parameter $(L/2)$, which represents the standard deviation of the Gaussian distribution, is the half of the width of the image. Additionally, it is also calculated the weighted standard deviation such as follows:

$$\sigma^2 = \frac{\sum_{i=1}^{L}\sum_{j=1}^{T} G\left(x_i, y_j\right) * \left(F\left(x_i, y_j\right) - \mu\right)^2}{\sum_{i=1}^{L}\sum_{j=1}^{T} G\left(x_i, y_{ji}\right)} \tag{3}$$

The weighted mean and the weighted standard deviation are used to define the values of two thresholds. These thresholds allow us to classify the new image into two classes, namely, the changing pixels and the invariant pixels. Finally, as result of the classification process, it is created a binary image such as the one shown in the **Figure 1(c)**. Then, this binary image is used to estimate the direction of the object's motion between the current and the next frame. The new binary image is defined by the following Equation:

$$Y\left(x_i, y_i\right) = \begin{bmatrix} 0 & \text{if } -k*\sigma < F\left(x_i, y_i\right) - \mu < k*\sigma \\ 1 & \text{otherwise} \end{bmatrix} \tag{4}$$

where the parameter k, which is a limit that defines the number of standard deviations that are used to discriminate between the two existing classes, is set to 3.

Once we have created the binary image from the Equation (4), we have two sets of pixels, namely, the set of pixels whose intensity values have changed between the current and the next frame, and the set of pixels whose intensity values stay invariant between two consecutive frames. Then, it is calculated the centroids from these two sets in order to determine the direction of the object's motion between the current and the next frame (See the **Figure 1(d)**).

In order to determine the direction of the object's motion between the current and the next frame, it is constituted two sets of pixels such as follows:

$$S_1 = \{(x, y) | Y(x, y) > 0\} \tag{5}$$

where $Y(.)$ is the image obtained from the Equation (4). Thus, the set, S_1, represents the coordinates from the pixels that initially belong to the object being tracked in the first ROI, but that turn into background pixels in the second ROI. The second set is constituted by the coordinates of those pixels that are part of the background in the first and the second region of interest, and it is defined by the following Equation:

$$S_2 = \{(x, y) | Y(x, y) = 0\} \tag{6}$$

The groups of pixels which constitute the sets, S_1 and S_2, are shown in the **Figure 1(c)**.

On the other hand, if we use the Equation (7) to calculate the average of the coordinates that constitute the group, S_1, which was defined by the Equation (5), we will obtain the point, P_n, which represents the centroid of the group, S_1. In the same way, the centroid of the group, S_2, which is represented by the point, P_m, is calculated using the Equation (8). The locations of the centroids, P_n and P_m, in the region of interest are shown in the **Figure 1(d)**.

$$P_n = \left(\overline{x_n}, \overline{y_n}\right) = \frac{1}{N} * \sum_{(x_i, y_i) \in S_1}^{N} (x_i, y_i) \tag{7}$$

$$P_m = \left(\overline{x_n}, \overline{y_n}\right) = \frac{1}{N} * \sum_{(x_i, y_i) \in S_2}^{N} (x_i, y_i) \tag{8}$$

The two points, which correspond to the centroids of the sets, S_1 and S_2, constitute a vector whose direction determines the orientation of the object's motion. In other words, the vector which connects the centroid, P_n, to the centroid, P_m, is an educated guess of the direction of the object's motion between the current and the next frame. This vector, which is shown in the **Figure 1(d)**, is defined by the following Equation.

$$\boldsymbol{V} = P_m - P_n \tag{9}$$

Finally, the angle of the vector which determines the direction of the object's motion is calculated using the Equation (10).

$$\theta = \text{angle}(\boldsymbol{V})_I \tag{10}$$

2.2. Minimization Stage

Once the direction of the object's motion has been established, it is started a minimization process which seeks to determine the location of the object being tracked in the next frame (second frame). To do this, it is used the region of interest defined in the current frame (first frame), R_1. This region of interest is located totally inside the object being. At the same time, it is defined in the second frame a second region of interest, R_*, with the same shape, size, and location of the first region of interest used in the current frame. During the minimization process, the region of interest, R_*, is displaced at discrete steps in the second frame following the direction of the object's motion that was previously estimated in the motion estimation stage. The Equations (11) and the Equation (12) define the movement of the region of interest, R_*, in the second frame.

$$\Delta x = j * \cos(\theta) \tag{11}$$

$$\Delta y = j * \sin(\theta) \tag{12}$$

where, j, is an integer which takes the values of $0, 1, 2, \cdots, S$. The parameter, S, is a constant that represents the maximum possible displacement of the object being tracked; the value of the constant "S" is defined by the user and depends on the nature of the video being processed. Finally, the parameter, θ, is calculated using the Equation (10).

On the other hand, at each iteration of the minimization process that seeks to establish the displacement of the object being tracked, the region of interest, R_*, located in the second frame is compared with the region of interest, R_1, located in the first frame. The Equation (13) presents the function of dissimilarity, $M(\Delta x, \Delta y)$, which is used to compare these two regions.

$$M(\Delta x, \Delta y) = \sum_{i=1}^{L} \left[U\left(R_1(x_i, y_i) - R*(x_i + \Delta x, y_i + \Delta y) \right) \right] \quad (13)$$

where the parameter, L, represents the number of pixels which constitutes the region of interest, R_1. The function, $U(.)$, is defined by the Equation (14).

$$U(x) = \begin{bmatrix} x = 0 & -k*\sigma < x - \mu < k*\sigma \\ x \neq 0 & \text{otherwise} \end{bmatrix} \quad (14)$$

where, the parameters μ and σ are defined by the Equation (1) and the Equation (3) respectively. Additionally, the constant k is set to 3. On the other hand, the function, $M(.)$, depends on the parameters, Δx and Δy, that correspond to the horizontal and vertical displacements of the second region of interest, R_*. Finally, it is selected the pair of values for Δx and Δy that minimizes the function of dissimilarity, $M(.)$.

At the end of the process of minimization, the region of interest in the next frame is updated to the new location defined by the pair of values, Δx and Δy, that minimizes the function of dissimilarity, $M(.)$, such as follows:

$$Y_i = Y_i + \Delta y \quad (15)$$

$$X_i = X_i + \Delta x \quad (16)$$

where the pair of coordinates, X_i and Y_i, corresponds to the location of the center of the region of interest in the next frame.

3. Experiments and Results

The proposed algorithm was tested over more than 2290 frames (of the video sequences "store", "plane", "train", "crowd".) from the PETS 2006 dataset [13] and YouTube videos. The results showed the reliability of the proposed algorithm in a variety of challenging conditions such as occlusion, crowded scenes, illumination changes, and camera movements.

Initially, it is defined a region of interest, *R*, in the first frame of the video. This region is placed in such a way that it is completely inside the object to be tracked. Then, the proposed algorithm updates automatically the location of this region of interest for all the remaining frames of the video sequence. Different urban scenarios were used for testing, and some of the results obtained for these testing videos are shown from the **Figures 2-5**.

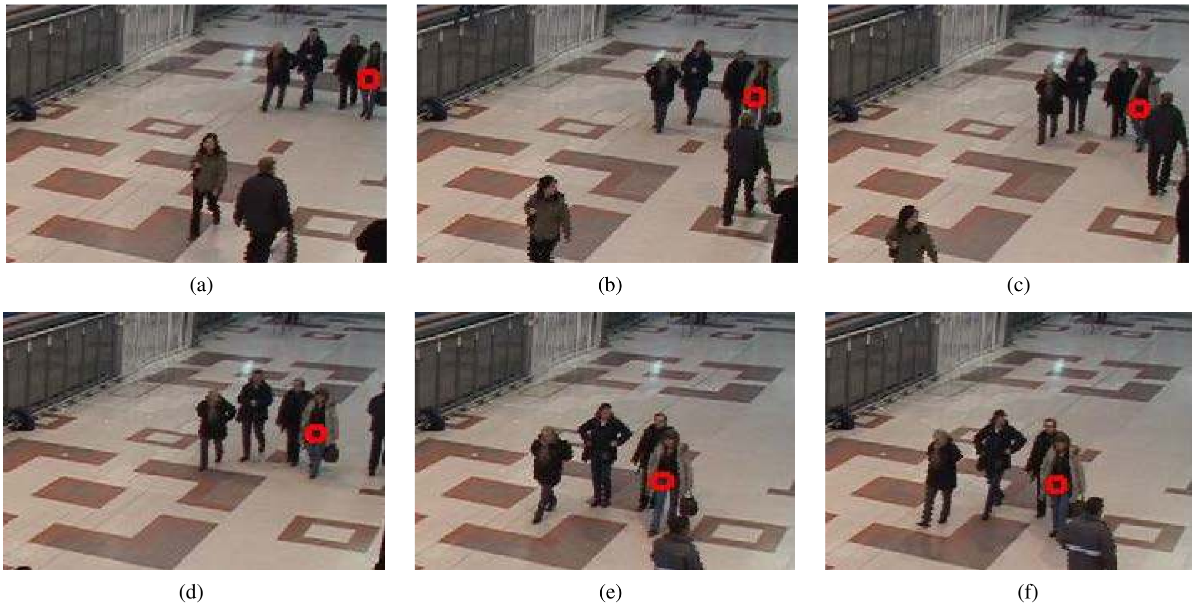

Figure 2. Tracking results for the "Train" sequence. (a) 1th frame; (b) 80th frame; (c) 160th frame; (d) 240th frame; (e) 320th frame; (f) 450th frame.

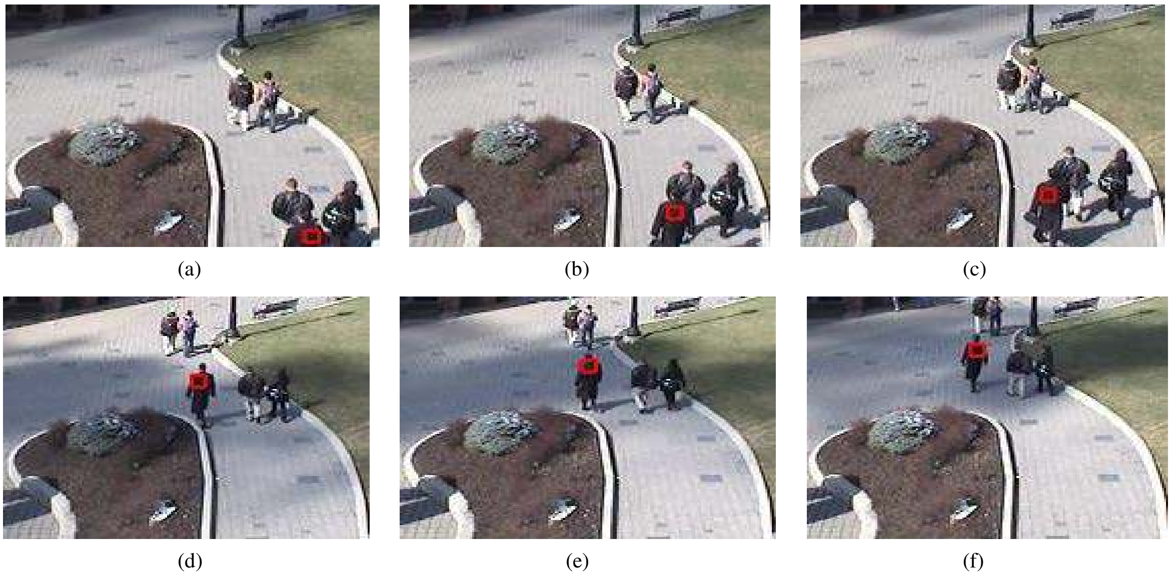

Figure 3. Tracking results for the "Store" sequence. (a) 1th frame; (b) 90th frame; (c) 280th frame; (d) 410th frame; (e) 510th frame; (f) 510th frame.

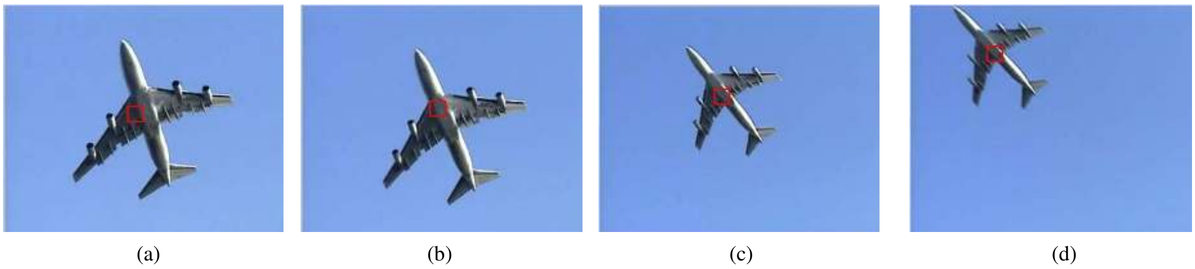

Figure 4. Tracking results for the "Plane" sequence. (a) 1th frame; (b) 250th frame; (c) 450th frame; (d) 550th frame.

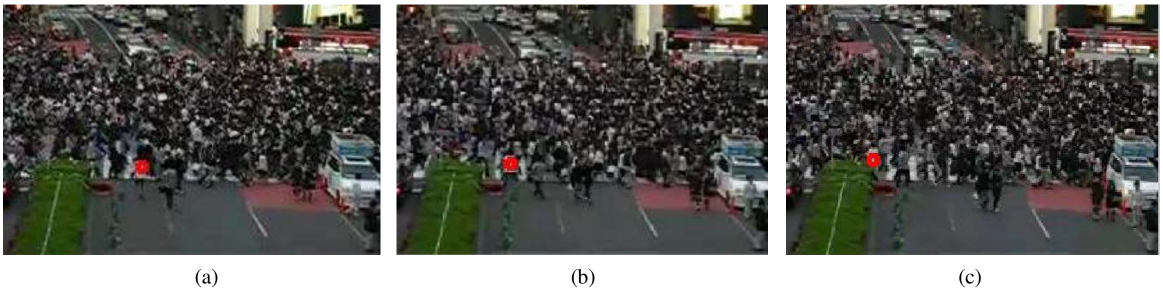

Figure 5. Tracking results for the "Crowd" sequence. (a) 1th frame; (b) 300th frame; (c) 640th frame.

Figure 2 shows a set of frames from a video sequence recorded in a train station with a total number of frames of 450. On the other hand, the pedestrian to be tracked corresponds to a woman who is wandering around. She is selected in the first frame by placing in her a square-shape region of 10 × 10 pixels. The selected pedestrian is occluded by another person who is walking in the opposite direction to her from the 50th to 180th frame in the video sequence.

The next video used for testing was recorded in an urban scenario where different pedestrians are wandering in a complex of stores (see the **Figure 3**). At the beginning of the video, it is selected the pedestrian to be tracked by placing in him a square region of 10 × 10 pixels. Alternatively, the frames from the video present illumination changes that are caused primarily by clouds that occlude the sun light in the scene. The proposed algorithm was tested for a total of 650 frames from this video, and the results showed that the selected pedestrian could be successfully tracked under illumination changes and object-size changes.

The video shown in the **Figure 4** presents a set of frames where the object to be tracked corresponds to an airplane. The size of the region of interest used to track the airplane in the video is of 20 × 20 pixels. The number of frames used to test the proposed algorithm was 550. As can be seen in **Figure 4**, the object is reliably tracked under object scale variation, rotation, and camera movement.

Finally, the **Figure 5**, on the other hand, shows the tracking of a pedestrian who is crossing the street in a crowded intersection in Tokyo, Japan. The number of frames that constitute the video sequence is 640. At the first frame of the video, it is defined a square-shaped region of 7 × 7 pixels that is used to represent the pedestrian to be tracked.

4. Conclusion

This paper proposes a new algorithm for tracking of objects in video sequences. The new method is based on regions of interest that ignore much of the variability in the environment which surrounds the object being tracked. The proposed algorithm was tested under a wide variety of scenarios. Results show that the propose algorithm can reliably track an object under several challenging conditions such as occlusion, camera movements, illumination changes, crowded scenes, and object scale variations.

References

[1] Foresti, G.L. (1999) Object Recognition and Tracking for Remote Video Surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, **9**, 1045-1062. http://dx.doi.org/10.1109/76.795058

[2] Lipton, A.J., Fujiyoshi, H. and Patil, R.S. (1998) Moving Target Classification and Tracking from Real-Time Video. *Proceedings of the Fourth IEEE Workshop on Applications of Computer Vision*, 8-14.

[3] Li, Y., Goshtasby, A. and Garcia, O. (2000) Detecting and Tracking Human Faces in Videos. *Proceeding of the ICPR*'00, **1**, 807-810.

[4] Gabriel, P., Hayet, J.-B., Piater, J. and Verly, J. (2005) Object Tracking Using Color Interest Points. *IEEE Conference on Advanced Video and Signal Based Surveillance*.

[5] Harris, C. and Stephens, M. (1988) A Combined Corner and Edge Detector. *The* 4*th Alvey Conference*, 147-151.

[6] Koenderink, J.J. and Van Doorn, A.J. (1987) Representation of Local Geometry in the Visual System. *Biological Cybernectics*, **55**, 367-375.

[7] Haritaoglu, I., Harwood, D. and Davis, L. (2000) Real-Time Surveillance of People and Their Activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 8. http://dx.doi.org/10.1109/34.868683

[8] Sato, K. and Aggarwal, J. (2004) Temporal Spatio-Velocity Transform and Its Application to Tracking and Interaction. *Computer Vision and Image Understanding*, **96**, 100-128. http://dx.doi.org/10.1016/j.cviu.2004.02.003

[9] Sebastian, P. and Voon, Y.V. (2007) Tracking Using Normalized Cross Correlation and Color Space. *International Conference on Intelligence and Advanced System*.

[10] Aggarwal, A., Biswas, S., Singh, S., Sural, S. and Majumdar, A.K. (2006) Object Tracking**.** Using Background Subtraction and Motion Estimation in MPEG Videos. *ACCV*, *LNCS*, **3852**, 121-130.

[11] Khatoonabadi, S.H. and Bajic, I.V. (2013) Video Object Tracking in the Compressed Domain Using Spatio-Temporal Markov Random Fields. *IEEE Transaction on Image Processing*, **22**, 300-313. http://dx.doi.org/10.1109/TIP.2012.2214049

[12] Chu, C.-T., Hwang, J.-N., Pai, H.-I. and Lan, K.-M. (2011) Robust Video Object Tracking Based on Multiple Kernels with Projected Gradients. ICASSP.

[13] PETS 2006 Benchmark Data. ftp://ftp.pets.rdg.ac.uk/pub/PETS2006/