Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering

Elixir Comp. Sci. & Engg. 65 (2013) 19640-19646

Elixir ISSN: 2229-712X

Local Binary Pattern for Effective Moving Object Detection and Tracking in Video

Nilesh J. Uke[1,*] and Ravindra C. Thool[2]

[1]Department of Information Technology, Sinhgad College of Engineering, Pune University, India.

[2]Department of Information Technology, SGGS Institute of Engineering & Technology, Nanded, India.

ARTICLE INFO

Article history:

Received: 12 October 2013;

Received in revised form:

25 November 2013;

Accepted: 3 December 2013;

Keywords

Object detection,

Object tracking,

Texture feature,

Shape feature,

Forward tracking,

ABSTRACT

Object detection, tracking and recognition currently generated much attention in computer vision community due to their real time applications. The main goal is to detect and track specific instances of an object or broader object classes, such as cars or pedestrians in images sequence. Due The potency of processing video database lies on the search techniques employed in the video processing system. Usage of improper search techniques will make the processing system feeble. This paper proposes unique object detection and tracking system where Local Binary pattern are employed to detect moving object and track them in successive frames. Detecting a specific object from video suing Local Binary Pattern is more suitable when, objects information is not available. Initially, the database video clips are segmented into diverse shots before the feature extraction process. The proposed system comprises two stages: (i) feature extraction and (ii) tracking of object in the video clips. In the feature extraction stage, the color feature is extracted first based on the color quantization. Next, the edge density feature is extracted for the objects present in the query video. Subsequently, the texture feature is extracted using LGXP (local Gabor XOR patterns) technique. Eventually, the object is detected based on those features extracted and the detected object is tracked by utilizing both forward and backward tracking method. The proposed methodology is proved to be more effectual and precise in object detection and tracking.

Introduction

Tracking an object of interest in a video sequence means continuously identifying its position and orientation despite movement of either the camera or the object. Visual tracking is the most salient module for a wide variety of application domains. Robust tracking of features form the primary input to classical vision problems such as structure from motion and registration. As well, tracking has been done in different application areas namely, surveillance, marker-less motion capture, and medical imaging [1]. Recently, visual event recognition has turn into one of the most dynamic topics in computer vision, due to its wide application prospects in video surveillance, video retrieval, and human-computer interaction [2].

This mechanism can be described by a center-surround difference model, which is generally implemented in a variety of feature spaces, such as color, intensity, and texture [3]. The aim of target tracking is to estimate the object's speed and position over time by means of one or more sensors. In case of an electro-optical sensor, the tracking system analyzes a series of digital images or frames. The steps often involved in tracking are (i) target detection and track initiation, (ii) track continuation, (iii) track termination, and (iv) identity declaration or classification. The detection is complicated due to the presence of multiple targets and other objects, referred to as background clutter. Normally, information contained within a single frame is not adequate to differentiate targets from clutter, making track initiation challenging and tricky. In such cases, detection over a sequence of frames (multi-scan detection) can be employed.

Multi-scan detection necessitates solving the data association problem, also known as the matching problem in the video tracking community [4]. Since the objects may have arbitrary shape, color or texture, modern category-specific (e.g., face/car/human) object detection technology that often learns one or more specific classifiers based upon a large set of similar training images, cannot be applied to our scenario. To overcome this detection problem, a framework based on matching a reference and a target video sequences, is used. The reference video is taken through a moving camera when there is no distrustful object in the scene, and the target video is taken through a second camera following a similar trajectory, and observing the same scene where suspicious objects may have been neglected in the interim [5].

Detection (a moving object recognition which is breaking into the field of robot vision) and tracking (similar to the movement until the distance from the object) are the most significant processes in many applications, ranging from games to robot automation of monitoring. In addition, the development of effectual methods for carrying out these tasks represents a challenging test for the integration of different techniques such as image processing, filtering, control theory, and artificial intelligence (AI) approaches [6]. Motion detection approaches based on stereo vision often use traditional stereo vision by dual-camera systems. For ideal dual stereo vision, the system satisfies non-verged geometry and is convenient for stereo rectification and matching [7]. The most widely accepted technique for moving object detection with fixed camera is based on background subtraction.

Tele:

E-mail addresses: njuke.scoe@sinhgad.edu

For frame comparison of video information, a row of measures are exploited as unit for measurement of similarity images [8]. However, the problem of perfection the estimation methods of objects similarity is rather actual, because correlation characteristics of video sequences are far from ideal, and characterized by a significant level of secondary spikes and main spike inaccuracy[8].

Motivation for Local Binary Pattern

The local binary pattern (LBP) texture operator was initially conceptualized as a measure for local image contrast [9], [10]. It describes the neighborhood of the pixel by using binary derivates of the pixel which forms short code to describe the pixel that is theoretically simple yet very powerful method of analyzing textures. The original LBP texture has locally two complementary aspects, a pattern and its strength. The original version of the local binary pattern operator works in a 3 x 3 pixel block of an image which are thresholded by its centre pixel value. Further they are multiplied by powers of two, and then summed to obtain a label for the centre pixel as depicted in Fig 1. As the neighbourhood consists of 8 pixels, a total of 2^8= 256 different labels can be obtained depending on the relative gray values of the centre and the pixels in the neighbourhood.

Example

6	5	2
7	6	1
9	8	7

thresholded

1	0	0
1		0
1	1	1

weight

1	2	4
128		8
64	32	16

pattern: 11110001 LBP=1+16+32+64+128 – 241

Figure 1. The Original LBP

The Local Binary Pattern (LBP) operator, originally introduced for texture analysis, has proved to be a powerful approach to describe local structures of an image.

Because of its powerful capability and low computational complexity, it operator has been widely used in many applications, such as face recognition, motion analysis, pedestrian detection [11]. Here in this paper Local binary pattern (LBP) effective texture description operator is used to measure and extract texture information from the local neighborhood in a gray image, so that moving object is detected and tracked in successive frames.

Related Works in Object Detection and Tracking Moving Object

A numerous researches have been presented in the literature for the detection and tracking of moving object in videos. Almost all existing methods for object detection are expected at finding objects using some image features like colour, edge, texture features. A brief review of some recent use of LBP in their researches is presented here. LBP texture and colour features, combined with particle filtering for dynamic object tracking in [12]. Colour feature extract from the colour image and local binary edge feature extract from the gray image, so the two features can complement each other. [13] presented a novel on-line feature selection method based on mean shift tracking algorithm, which adjusts the weight for each feature and each bin in feature histograms during the tracking process, according to the discrimination between the appearance of object and background with different features. They used use features of gray level, LBP and edge orientation to describe the colour, texture and structure of object which makes performance more strong.[14] uses canny edge detector to get the edge points only for the foreground parts from the foreground detection, and only calculate the LBP value of these edge points for real-time human tracking. [15] Used LBP for multi-target tracking in outdoor environment. LBP, is chosen because DSP is not good at floating-point operations.

In recent years advantage of LBP's tolerance to monotonic illumination variations and its computational simplicity have made it a popular technique for facial feature analysis [16] is taken , especially in facial expression recognition [17]. LBP features are evaluated with weighted chi-square template matching, SVMs, LDA, and Linear Programming classication techniques in regular and low-resolution images, concluding that LBP feature-based face representation outperformed Gabor features for Facial expression recognition [17].

Lin in [18] proposed a novel object tracking method based on the integration of global and local measure under an augmented particle filter framework. They used Colour statistical feature and LBP texture feature to construct the global and local observation model respectively. Also algorithm is accelerated while retaining robustness, by using cascade tow-stage strategy to integrate the colour and texture measures for the weight update process.

The method presented in [19] can tolerate changes of illumination so that robust moving objects detection can be realized. Salient feature points extracted from previous LBP are compared with those features found from the current LBP with a block matching approach so that the corresponding. Further they performed clustering of motion vectors so that , all the moving objects on image frames can be successfully detected and identified. Visual tracking approach based on 'bag of features' (BoF) is described in [20] Initially they used incremental PCA visual tracking (IVT) in the first few frames and collect image patches randomly sampled within the tracked object region in each frame for constructing the codebook. These the tracked object then can be converted to a bag. Later they constructed two codebooks using color (RGB) features and local binary pattern (LBP) features for extracting more informative details.

Proposed Methodology for Designing Efficient Object Detection and Tracking System

The proposed object detection and tracking method involves three stages: Video Segmentation, Feature Extraction, and Tracking. In the initial stage, the database containing video clips are segmented into different frames or shots. Next in the feature extraction stage, three significant features are extracted from the segmented image such as, colour feature, edge density feature, and texture feature. LBP operator has been used to extract objects characteristics. The proposed method is detailed in the following sections.

Shot Segmentation

Today, a giant amount of digital videos have been generated due to the rapid expansion of computing and network infrastructures. Normally, videos are represented by a hierarchical structure, and the shots are the basis units for creating better-quality semantic scenes. Thus, shot boundary detection is an imperative preprocessing step for efficient browsing and further content analysis. A shot contains many consecutive frames that are usually captured by a camera in a single action, and no significant content change takes place between successive frames in the shot [21].

In our proposed method, a clustering technique is applied for video shot segmentation with the assumption that the frames in one cluster constitute a shot. The dominant-set clustering technique is based on the similarity matrix, so it is normally easy to cluster the frames that may not be successive in frame index into the same shot due to their high resemblance, because the clustering technique partially or fully disregards the temporal information. Thus, after performing the standard clustering process, a smoothing and elimination process have been conducted. The dominant-set clustering algorithm used for our segmentation process provides better clustering of the image. The concept of dominant set offers a robust framework for iterative pair-wise clustering.

The algorithm is described below,

Consider the input matrix I

- Initialize I^j, $j=1$ with I.
- Compute the local solution of u^j and $f(u^j)$
- Obtain the dominant set: $D^j = \sigma(u^j)$
- Divide I^j and obtain a new similarity affinity matrix I^{j+1}
- If I^{j+1} is empty, then break,

  Else

  $I^j = I^{j+1}$ and $j = j+1$ then return to step 2

  Output: $U_{n=1}^{j}\{D^n, u^n, f(u^n)\}$

Feature Extraction

We need to select powerful features to avoid many problems when tracking similar objects. Converting the input data into set of features is called feature extraction. Feature extraction is a process, where the image features are extracted to succinctly represent the visual content of an image [22]. In our proposed technique, three features are computed from each region of an individual image in a video shot. The three features are, Color Feature, Edge Density Feature and Texture Feature.

Color Feature Extraction

HSV (Hue, Saturation, Value) color components are more related to human perception and so, the color histogram extraction is based on HSV color space. Based on S component, the color quantization in HSV color space normally separate the gray bins from others and divides the other bins uniformly. Figure 2(a) demonstrates the partition in red SV plane [23]. In spite of this, the nature of HSV color space is that the colors of low V value looks more similar than the colors of high V value with respect to diverse saturations. In the below example, the color similarity between X7 to X9 (i.e., normally black) is greater than X4 to X6.

Figure 2: Color quantization in SV plane (a) Cylindrical space quantization (b) Conical Space quantization

The cylindrical HSV space is converted into cone space to solve the above problem. The cylindrical HSV point $X(H_x, S_x, V_x)$ is related to conical HSV point $Y(H_y, S_y, V_y)$. The transformation is given as,

$$\begin{cases} \mathrm{H}_y = \mathrm{H}_x \\ \mathrm{S}_y = S_x \cdot V_x \\ V_y = V_x \end{cases} \qquad (1)$$

Comparing figure. 1(a) and 1(b), the color of X7 to X9 correspond to Y6 and gray bins. This improves the color quantization of gloomy colors in HSV color space along with the diminution of number of bins used.

Edge Density Feature Extraction

Edge density defines the quality of an image by indicating the regions through the magnitude of the edge of the object available in the image. The resampled regions of the image are subjected to gray scaling operation in order that each region of the segmented image that is in RGB color space is converted to grayscale. Subsequently, the distance between the pixel are computed as,

$$d_1^{(i)} = \left|P_{gr}^{(i)}(x,y) - P_{gr}^{(i)}(x-1,y-1)\right|;\ 1 \le x \le M_r - 1,\ 1 \le y \le N_r - 1 \qquad (2)$$

After the pixel distance calculation, an edge preserving operation is carried out based on the obtained distance and thereby three classes of edges are obtained. Among that, only edge 3 is used. In the following eqn (3), $D_{\max}$ *and* $D_{\min}$ represents the maximum and minimum density value, respectively and D_{HT} represents the higher threshold value for the edge density.

$$E^{(i)}(x,y) = \begin{cases} D_{\max}\ ; if\ d_1^{(i)}(x,y) > D_{HT} \\ D_{\min}\ ; otherwise \end{cases} \qquad (3)$$

$$d_2(x,y) = \left|P_{gr}^{(j)}(x,y+1) - P_{gr}(x,y)\right|;\ 0 \le x \le M_r - 2 \text{ and } 0 \le y \le N_r - 2 \qquad (4)$$

$$E^{(i)}(x,y) = \begin{cases} D_{\max}\ ;\ if\ \ E^{(i)}(x,y) = 0\ \ and\ \ d_{2_i}^{(i)}(x,y) > D_{HT} \\ D_{\min}\ ;\ otherwise \end{cases} \qquad (5)$$

In edge detection process, $E^{(i)}$ determines the edge density. This can be accomplished by determining the edge density matrix as,

$$\rho_E^{(i)}(q,r) = C^{\max}\left(\frac{n_b q}{2} + r\right) + 1;\ 0 \le q \le n_{b/2} - 1,\ 0 \le r \le n_{b/2} - 1 \qquad (6)$$

where, $C^{\max}\left(\frac{n_b q}{2} + r\right)$ is the number of values present in the c^{th} block of the $c = \frac{n_b q}{2} + r$ edge $E^{(i)}$.

It is important to note that $E^{(i)}$ is the edge obtained for the image in a database. Thus, obtained edge is stored as the edge density feature vector of the corresponding region of an image. The aforementioned features are extracted from the regions of each image in the database.

Texture Feature Extraction

Here, the texture features from the image region are extorted by building a color texture histogram using a Local Binary Pattern (LBP). To summarize the local gray level structure of the image, LBP operator is used. LBP operator is defined as a gray scale invariant texture measure, derived from a standard definition of texture in a local neighborhood. Here, a local Gabor XOR pattern is employed for the extraction of texture features. The LBP features are chosen due to their proven effectiveness in the past and they provide robustness to variations in illumination, and are able to provide highly descriminative features due to different levels of locality

LGXP Descriptors

In LGXP descriptors, first the phases are quantized into diverse ranges, and subsequently the LXP operator is applied to the quantized phases of the central pixel and its neighbour [24]. Eventually, the resulted binary labels are combined together as local pattern of the central pixel.

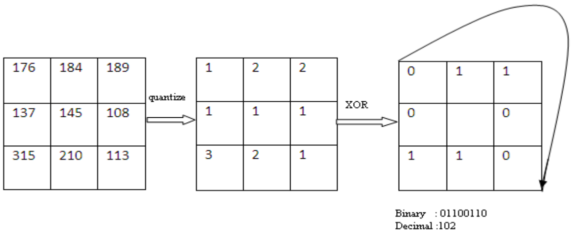


Figure 3. Encoding method of LGXP

The pattern of LGXP in binary and decimal form is given as,

$$LGXP_{\mu,v}(x_i)=\left[LGXP_{\mu,v}^{D},LGXP_{\mu,v}^{D-1},....,LGXP_{\mu,v}^{1}\right]_{binary}=\left[\sum_{j=1}^{D}2^{j-1}.LGXP_{\mu,v}^{j}\right]_{decimal} \quad (7)$$

Where, x_i represents the central pixel position in the Gabor phase map with scale 'v'and orientation 'μ', D represents the size of neighborhood, and $LGXP_{\mu,v}^{j}\ (j=1,2,......,D)$ denotes the pattern computed between x_i and its neighbor x_j, which is given as,

$$LGXP_{\mu,v}^{j}=t(\varphi_{\mu,v}(x_i))\otimes t(\varphi_{\mu,v}(x_j)),\ j=1,2,....,D \quad (8)$$

In eqn (8), $\varphi_{\mu,v}(\ )$ is the phase, $\otimes$ is the LXP operator based on XOR, $t(\ )$ is the quantization operator, which determines the quantized code of phase based on the number of phase ranges.

The number of phase ranges is computed as,

$$g\otimes h=\begin{cases}0\ ;\ if\ \ \mathrm{g}=\mathrm{h}\\ 1;\ else\end{cases} \quad (9)$$

$$t(\varphi_{\mu,v}(\ ))=j; \quad (10)$$

$$\text{if}\ \frac{360*j}{l}\le\varphi_{\mu,v}(\ )<\frac{360*(j+1)}{l},\ j=0,1,....,l-1$$

where l is the number of phase ranges.

For each gabor kernal, one pattern map is determined. Subsequently, each pattern map is splitted into 'k' non-overlapping sub-blocks, and the histograms of all these sub-blocks of all the scales and orientations are combined together to form the proposed LGXP descriptor of the input image.

$$HG=\left[HG_{\mu o,vo,1},....,HG_{\mu o,vo,k};....;HG_{\mu o-1,vo-1,1},.....,HG_{\mu o-1,vo-1,k}\right] \quad (11)$$

Where, $HG_{\mu,v,p}\,(p=1,2,....k)$ represents the histogram of the p^{th} sub-block of LGXP map with scale 'v' and orientation 'μ'. In our proposed technique, Gabor filters of five scales and eight orientations are utilized.

Moving Object Detection and Tracking

The moving objects are detected from each frame. The three different methods employed for the detection of moving object are, Background Subtraction, Temporal Differencing and Optical Flow. The background subtraction is a straightforward technique, used for moving object detection. Here, it is assumed that the background is static, so that the background does not change with the number of frames. First the difference between the object O_k and the background B_k is computed using the below formula,

$$D_k(x,y)=\left|O_k(x,y)-B_k(x,y)\right| \quad (12)$$

Now, threshold the difference using the formula given below,

$$M_k(x,y)=\begin{cases}1\ ,D_k(x,y)>T\\ 0\ ,otherwise\end{cases} \quad (13)$$

Using the gray histogram, the bottom value between the two peaks is taken as the threshold value [25]. Basically, object tracking is used to find the location of the target in different frames in a sequence of images. The main aim of object tracking is to choose the best object characteristics and to use the appropriate searching methods. In our work, an Image Difference Algorithm is employed for Moving Object Detection and Tracking.

Object Tracking

To achieve an excellent detection rate on each shot of a video frame, the detection and tracking are combined and some rules are generated to achieve a successful tracking process. The tracking is of two types: Forward Tracking and Backward Tracking.

Forward Tracking

The forward tracking process is carried out on each frame, starting from frames where the objects have been detected. While tracking, same object may be detected several times in a shot, this leads to multiple tracking of same object, which is time consuming. In order to conquer this problem, some tracking rules are used to find whether the detected objects are multiplied or not. This rule is generally based on the percentage of overlap between the detected object and the one resulting from the forward tracking in the same frame, which is represented as,

$$O(\mathrm{F_T})=\max_j\frac{S_{(F_T\cap B_j)}}{\min\left(S_{B_j},S_{F_T}\right)} \quad (14)$$

Where, S_{B_j} is the area of the j^{th} detection, and S_{F_T} is the area of the forward tracking. As well, $S_{(F_T\cap B_j)}$ represents the area recovered by the detection process.

Backward Tracking

Backward Tracking is carried out on each frame in order to obtain an additional set of object. The backward tracking is very helpful in case an object is not detected at the beginning but in the middle of the frame. The forward tracking provides the objects from the detection frame to the end of the shot whereas backward tracking provides the missing result from the first frame of the shot to the frame wherein the last object detection has been performed. Moreover, the backward tracking is proved to be effective when the forward tracking fails to find the location of an object in a particular frame. This may be due to occlusion, bad lighting or if the tracker sticks to the background.

Object Tracking Process

Here, the tracking process is done by comparing the features extracted in the present frame with the features extracted in the prior frame. Assume we have a total of 'P' number of features extracted in present frame and 'Q' number of features extracted in the prior frame. Thus, a total of P x Q matching is required. For this matching process, a Euclidean Distance measure is utilized. For two feature vectors V_i and Vj where

$i = 1,2,.....P$ and $j = 1,2,......Q$, the Euclidean distance is calculated as,

$$D(\mathrm{V}_i, V_j) = \sqrt{\frac{1}{a}\sum_{m=1}^{t}\left(V_{im} - V_{jm}\right)^2} \qquad (15)$$

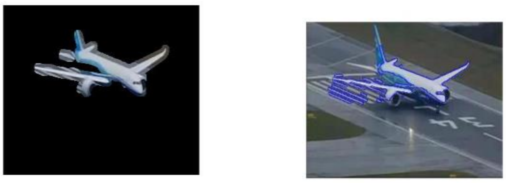
Where, a represents the dimension of the feature vector selected. Once the distances between the features are computed, the minimum distance object is tracked. This is the object which we have to be tracked from the video clip. This process is then carried out for different shots and the movement of the object is found which helps in efficient tracking of the object.

Results and Discussion

The proposed object detection and tracking system using the low level features was implemented in the working platform of MATLAB (version 7.11). The detection and tracking process is tested with different frames of video and the upcoming result of the proposed work has been shown below. Initially, the video are segmented to different shots or frames and then features are extracted followed by the detection and tracking process.

As mentioned in the section 3, the input video clip is shot segmented into number of frames. Then for each frame, various features are extracted using the algorithms mentioned in the proposed methodology. These features help in identifying the object position in each frame. Once the features are extracted, ellipse fitting algorithm is applied to detect and track the objects in the frames. Figure 4 shows the results obtained by the proposed method. The figure 4(a) is the original image of the flight obtained from the shot segmentation. Figure 4(b) is the obtained output after the feature extraction process. Figure 4(c) shows the resultant image after the object detection process and finally, Figure 4(d) shows the tracked image of the object in the frame. Similarly for different frames, the process is repeated and finally the object is tracked. The proposed methodology is proved to be more effective and accurate in object detection and tracking which is shown in figure 5 and 6.

(a) **(b)**



(c) **(d)**

Figure 4. Results of object tracking in first frame: (a) Input frame; (b)Edge feature extraction; (c) Object detection ; (d) Object tracking

(a) **(b)**

(c) **(d)**

Figure 5. Results of object tracking in second frame: (a) Input frame; (b)Edge feature extraction; (c) Object detection ; (d) Object tracking

(a) **(b)**

(c) **(d)**

Figure 6. Results of object tracking in third frame: (a) Input frame; (b)Edge feature extraction; (c) Object detection ; (d) Object tracking

Table 1. Precision and Recall for the proposed method

S.No	**Performance Analysis**			
	Precision		**Recall**	
	Proposed Method	**Existing Method**	**Proposed Method**	**Existing Method**
1	0.8	0.71	0.16	0.15
2	0.75	0.65	0.23	0.24
3	0.68	0.59	0.35	0.35
4	0.6	0.52	0.48	0.46
5	0.56	0.47	0.56	0.52
6	0.5	0.43	0.65	0.57
7	0.44	0.35	0.72	0.65
8	0.35	0.32	0.8	0.69
9	0.26	0.25	0.85	0.75

Performance Analysis:

The precision and recall value for the proposed method are calculated for analyzing the performance. Let the object to be tracked is denoted as O_T and the tracked output is denoted as T_O, then the precision and recall are expressed as,

$$precision = \frac{\{O_T \cap T_O\}}{\{T_O\}} \qquad (16)$$

$$recall = \frac{\{O_T \cap T_O\}}{\{O_T\}} \qquad (17)$$

Precision measures how much of T_O covers the O_T, and recall measures how much of O_T is covered by the T_O. Using Eq. (16) and (17), the precision and recall values for the query image are calculated for the proposed method and also for the existing method. The values obtained from the calculation are given in Table 1. These values are used for the analysis of performance between the proposed and existing methods. Here, the existing method is the vision based object detection and tracking.

The graphical representation of precision and recall corresponding to the above values is shown in the figure 7,

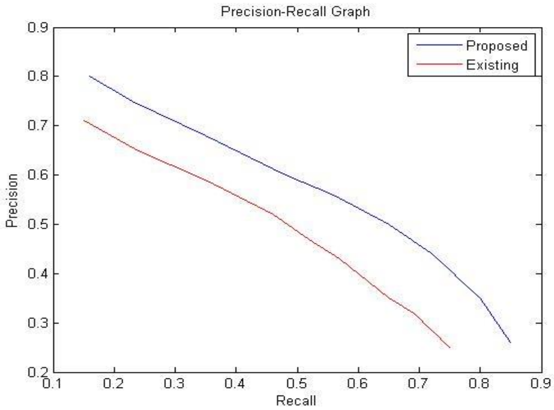


Figure 7. Precision and Recall plot for the proposed method

Conclusion

In this paper, a robust object detection and tracking system was proposed, which is based on low level features. Combination of multiple features can improve the robustness of tracking algorithm, and these three features are very simple and effective access to real-time tracking. Experimental results show that the method proposed can improve tracking accuracy and is strong robustness. In this unique object detection and tracking system, the video segmentation, feature extraction, object detection and tracking processes are combined using a single feature. As compared to previous methods, our proposed methodology is proved to be more efficient and accurate in object detection and tracking. The precision and recall were the two major parameters taken into account for measuring the potency of the proposed method. As shown in the tabular column, the precision and recall values were seems to be improved than the other existing methods of object tracking and detection. Overall, our proposed framework is proved to be an effective and efficient method in the field of object detection and tracking.

Reference

[1] N. Al-najdawi, S. Tedmori, E. Edirisinghe, and H. Bez, "An Automated Real-Time People Tracking System Based on KLT Features Detection," *Int. Arab J. Inf. Technol.*, vol. 9, no. 1, pp. 100–107, 2012.

[2] P. Turaga, R. Chellappa, and V. S. Subrahmanian, "Machine Recognition of Human Activities A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1473–1488, 2008.

[3] W. Zhang, Q. M. J. Wu, S. Member, G. Wang, and H. Yin, "An Adaptive Computational Model for Salient Object Detection," vol. 12, no. 4, pp. 300–316, 2010.

[4] E. Trucco and K. Plakas, "Video Tracking: A Concise Survey," *IEEE J. Ocean. Eng.*, vol. 31, no. 2, pp. 520–529, 2006.

[5] H. Kong, J.-Y. Audibert, and J. Ponce, "Detecting abandoned objects with a moving camera.," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2201–10, Aug. 2010.

[6] L. Dong and X. Lin, "Monocular-Vision-Based Study on Moving Object Detection and Tracking," in *4th International Conference on New Trends in Information Science and Service Science (NISS)*, 2010, pp. 692–695.

[7] M. Z. Brown, D. Burschka, G. D. Hager, and S. Member, "Advances in Computational Stereo," vol. 25, no. 8, pp. 993–1008, 2003.

[8] R. Bogush, S. Maltsev, A. Kastryuk, N. Brovko, and D. Hlukhau, "Motion detection and tracking algorithms in video streams," vol. 25, no. June, pp. 143–151, 2009.

[9] T. Ojala, M. Pietikhenl, D. Harwood, and L. T. Measures, "Performance Evaluation of Texture Measures with Classification Based on Kullback Discrimination of Distributions," in *nternational Conference on Pattern Recognition*, 1994, pp. 582–585.

[10] T. Ojala and M. Pietika, "Unsupervised texture segmentation using feature distributions," *Pattern Recognit.*, vol. 32, pp. 477–486, 1999.

[11] A. Boudissa, J. K. Tan, H. Kim, and S. Ishikawa, "A simple pedestrian detection using LBP-based patterns of oriented edges," *2012 19th IEEE Int. Conf. Image Process.*, pp. 469–472, Sep. 2012.

[12] H. Ying, X. Qiu, J. Song, and X. Ren, "Particle Filtering Object Tracking Based on Texture and Color," *2010 Int. Symp. Intell. Inf. Process. Trust. Comput.*, vol. 1, pp. 626–630, Oct. 2010.

[13] S. Yi, Z. Yao, J. Liu, J. Chen, and W. Liu, "Features," in *2012 Spring Congress on Engineering and Technology (S-CET)*, 2012, pp. 1–5.

[14] J. Xu, A. Beaugendre, and S. Goto, "Real-time Human Tracking by Detection based on HOG and Particle Filter," in *6th International conference on Computer Sciences and Convergence Information Technology (ICCIT), 2011*, 2011, p. 193,198.

[15] Z. Zhao and S. Yu, "A Multi-target Tracking Algorithm using Texture for Real-time Surveillance," in *IEEE International Conference on Robotics and Biomimetics, ROBIO 2008*, 2009, p. 2150,2155.

[16] D. Huang, S. Member, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local Binary Patterns and Its Application to Facial Image Analysis: A Survey," vol. 41, no. 6, pp. 765–781, 2011.

[17] C. Shan, S. Gong, and P. W. McOwan, "Facial expression recognition based on Local Binary Patterns: A comprehensive study," *Image Vis. Comput.*, vol. 27, no. 6, pp. 803–816, May 2009.

[18] L. Gao, Z. Liu, and P. Tang, "Probabilistic Visual Tracking using Global and Local Object's Information," *2008 Int. Conf. Comput. Sci. Softw. Eng.*, no. 1, pp. 1005–1008, 2008.

[19] Y. Huang and Z. Ou, "A Rapid Texture-Based Moving Object Detection Method," in *Proceedings of 2011 8th Asian Control Conference (ASCC) , Kaohsiung, Taiwan*, 2011, pp. 1205–1209.

[20] F. Yang, H. Lu, W. Zhang, and G. Yang, "Visual tracking via bag of features," *IET Image Process.*, vol. 6, no. 2, p. 115, 2012.

[21] L. Li, X. Zeng, W. Hu, P. Zhu, and X. Li, "Video Shot Segmentation Using Graph-based Dominant-Set Clustering Concept of Dominant Set," in *Proceedings of the First International Conference on Internet Multimedia Computing and Service*, 2009, pp. 166–169.

[22] N. M. and F. P. J. Fernández, R. Guerrero, "Multi-Level Paralelism in Image Identification," *Mec. Comput.*, vol. 28, pp. 227–240, 2009.

[23] K. Iqbal, M. O. Odetayo, and A. James, “Content-based image retrieval approach for biometric security using colour, texture and shape features controlled by fuzzy heuristics,” *J. Comput. Syst. Sci.*, vol. 78, no. 4, pp. 1258–1277, Jul. 2012.
[24] S. Xie, S. Shan, X. Chen, and J. Chen, “Fusing local patterns of Gabor magnitude and phase for face recognition.,” *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1349–61, May 2010.
[25] L. Xiao and T. Li, “Research on moving object detection and tracking,” *2010 Seventh Int. Conf. Fuzzy Syst. Knowl. Discov.*, pp. 2324–2327, Aug. 2010