International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 1 (2018) pp. 252-261

Object Tracking using Orthogonal Learning Particle Swarm Optimization (OLPSO)

Madhavi Gajula

Associate Professor, Mahatma Gandhi Institute of Technology,
Gandipet, Hyderabad, Telangana, India.
Orcid Id: 0000-0003-1919-8603

Dr. A. Jhansi Rani

Professor, Velagapudi Ramakrishna Siddhartha Engineering College,
Vijayawada, Andhra Pradesh, India.

Abstract

Millions of surveillance cameras have been installed in public areas, producing vast amounts of video data every day. It is an urgent need to develop intelligent techniques to automatically detect and segment moving objects. Various approaches have been developed for moving object detection based on background modeling. Most of them focus on temporal information but partly or totally ignore spatial information, bringing about sensitivity to noise and background motion. In order to overcome these issues an improved optimization algorithm is implemented Orthogonal Learning Particle Swarm Optimization (OLPSO) in combination with semantic feature descriptors such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), Principal Component Analysis (PCA) and Gray-Level Co-Occurrence Matrix (GLCM) Experiments and comparisons on PETS 2009 and MIT databases demonstrate the superiority of the proposed model over previous approaches.

Key-words: Gray-Level Co-Occurrence Matrix, Histogram of Oriented Gradients, Local Binary Patterns, Orthogonal Learning Particle Swarm Optimization.

INTRODUCTION

In computer vision and video processing an active research topics are human computer interfaces, robot vision and surveillance system. Here, surveillance has become an essential part in daily activities. The aim behind this topic to design an intelligent surveillance system, because it performs detection, tracking and recognition of objects in commercial, law enforcement, military applications. It is also used to measure the traffic flow, accident detection on highways, monitor pedestrian congestion in public spaces, compile consumer demographics in shopping malls and amusement parks, log routine maintenance tasks at nuclear facilities, and count endangered species. Here, the video surveillance usually involves in high dimensional semantics such as texture, color etc. to be measured for accuracy. The detection of moving object is considered as a low or intermediated level processing stage for various object recognition and analysis applications. The capability of being able to analyze human movements and their activities from video sequences are the most crucial factor in visual surveillance. Here, video based visual tracking is a complex task, consisting in the estimation of the position of a set of targets such as (people, vehicles) moving in the scene. The proposed video analysis method consist of the steps such as, detection of moving objects that are to be analyzed, tracking of such objects from frame to frame, and analysis of tracks to recognize their nature of the objects.

In order to perform these steps, it is necessary to distinguish foreground objects from stationary background in the video sequence. To achieve foreground pixel map at every frame, various combination of techniques, along with low-level image post-processing methods are preferred. Then connect the group of foreground map regions in order to extract the individual object features such as area, perimeter etc., from background region and also it involves in matching of detected foreground objects between consecutive frames using different feature of objects like (motion, color, texture etc.,). Hence, it is the process of tracking the object, by locating its position in every frame of the video sequence. In video surveillance, the previous experiments are done by using multi-dimensional features, it has the difficulties like high complexity, and also it requires more iteration to improve the accuracy. To overcome these difficulties, multi-dimensional feature sets are reduced to low-dimensional feature sets. In real time high resolution video surveillance systems, both methods are not suitable, because it have high computational complexity for the background modeling. To improve the background modeling, a ranking technique employed (OLPSO) Orthogonal Learning Particle Swarm Optimization algorithm. It finds the best feature combination of the particle and provides better combination result.

LITERATURE REVIEW

C. Conde *et al.* [1] presented a new method (HoG) for human detection based on Gabor filters and Histograms of Oriented Gradients. The effect of Gabor preprocessing was analyzed in detail, in particular the improvement experienced by the image's information and the influence exerted over the extracted feature. To compare the performance of the proposed method, several alternative algorithms for human detection were considered. In order to evaluate these techniques in non-controlled environments, a collection of standard databases, well known in the surveillance research community, were used: PETS 2006, PETS 2007, PETS 2009 and CAVIAR. An exhaustive test design was built based on two complementary evaluations: an evaluation oriented to counting people and a novel evaluation oriented to identification. Moreover, with the purpose of studying the performance of the Gabor-based preprocessing, a test adding Gabor filters to other local feature extraction methods, such as Steerable filters and the SIFT method, was implemented. The HoG method achieved a good performance regardless of the difficulty of the images (occlusions, overlapping, carrying baggage, etc.). The proposed method surpassed the alternative techniques in most of the analyzed situations. When the Gabor preprocessing was introduced into other local feature extraction methods, they achieve a better detection of the relevant information by enhancing the human shape. The results showed that using Gabor preprocessing in techniques based on features like orientation or magnitude of gradient improved their performance.

H. Qian *et al.* [2] presented a system framework to recognize multiple kinds of activities from videos by an SVM multi-class classifier with binary tree architecture. The framework was composed of three functionally cascaded modules: (a) detecting and locating people by non-parameter background subtraction approach, (b) extracting various of features such as local ones from the minimum bounding boxes of human blobs in each frames and a newly defined global one, contour coding of the motion energy image (CCMEI), and (c) recognizing activities of people by SVM multi-class classifier whose structure was determined by a clustering process. The thought of hierarchical classification was introduced and multiple SVMs were aggregated to accomplish the recognition of actions. Each SVM in the multi-class classifier was trained separately to achieve its best classification performance by choosing proper features before they were aggregated. Experimental results both on a home-brewed activity data set and the public Scheldt's data set showed the perfect identification performance and high robustness of the system.

Z. H. Zhan *et al.* [3] proposed an orthogonal learning (OL) strategy for PSO to discover more useful information. They named this PSO as orthogonal learning particle swarm optimization (OLPSO). They applied to both global and local versions of PSO, yielding the OLPSO-G and OLPSO-L algorithms, respectively. This new learning strategy and the new algorithms were tested on a set of 16 benchmark functions, and were compared with other PSO algorithms and some state of the art evolutionary algorithms. The experimental results illustrated the effectiveness and efficiency of the proposed learning strategy and algorithms. The comparisons showed that OLPSO significantly improved the performance of PSO, offered faster global convergence, higher solution quality, and stronger robustness.

P. Zhang *et al.* [4] proposed a novel moving people tracking with detection based on (probabilistic) LSA. By employing a novel 'twin-pipeline' training framework to find the latent semantic topics of 'moving people', the proposed detection can effectively detect the interest points on moving people in different indoor and outdoor environments with camera motion. Since the detected interest points on different body parts can be used to locate the position of moving people more accurately, by combining the detection with incremental subspace learning based tracking, the proposed algorithms resolved the problem of tracking drift during each target appearance update process. In addition, due to the time independent processing mechanism of detection, the proposed method was also able to handle the error accumulation problem. The detection can calibrate the tracking errors during updating of each state of the tracking algorithm. Extensive, experiments on various surveillance environments using different benchmark datasets proved the accuracy and robustness of the proposed tracking algorithm. Further, the experimental comparison results clearly showed that the proposed tracking algorithm outperformed the well-known tracking algorithms such as ISL, AMS and WSL algorithms. Furthermore, the speed performance of the proposed method was also satisfactory for realistic surveillance applications.

S. Zhang *et al.* [5] proposed a new approach to detect moving pedestrians aided by motion analysis. Their main contribution was to use motion information in two ways: on the one hand we localize blobs of moving objects for regions of interest (ROIs) selection by segmentation of an optical flow field in a pre-processing step, so as to significantly reduce the number of detection windows needed to be evaluated by a subsequent people classifier, resulting in a fast method suitable for real-time systems. On the other hand, they designed a novel kind of features called Motion Self Difference (MSD) features as a complement to single image appearance features, e. g. Histograms of Oriented Gradients (HOG), to improve distinctness and thus classifier performance. Furthermore, they integrated their novel features in a two-layer classification scheme combining a HOG support Vector Machines (SVM) and a MSD+SVM detector. Experimental results on the Daimler mono moving pedestrian detection benchmark showed that the proposed approach obtained a log-average miss rate of 36 % in the FPPI range [10−2,100], which was a clear improvement with respect to the naive HOG+SVM approach and better than several other state-of-the-art detectors. Moreover, the proposed approach also reduced runtime per frame by an order of magnitude.

PROPOSED METHOD

In recent years, research on video surveillance has grown gradually, especially on moving object detection. In object detection, the background model is a representation model for the scene which includes some objects. The aim behind this topic to improve the intelligent visual surveillance instead of using traditional passive video surveillance and also it is used in a wide range of real world applications like public security systems, traffic controls, robotic service, human-computer interaction, video editing and medical imaging. The main key feature of object detection is object segmentation, because it influences the performance of video processing steps like object tracking and recognition. Many algorithms has been proposed for object detection in video surveillance applications.

In that background subtraction (BS) method is the simplest method to detect the moving object, because it has the capacity to handle lighting changes, repetitive motions from clutter and long term scene changes. The basic idea of BS is to subtract the current frame from the background image and differentiate each pixel as background or foreground by comparing the difference with threshold. Here, PETS 2009 and MIT databases are utilized, because these videos are high dimensional semantics in texture and color and it helps to achieve a high accuracy rate.

In object detection, the video sequence is not directly taken as input, because it makes the procedure slow and ineffective. Here, the video sequence is converted into various frames and the respective frames are given as initial input in the object detection algorithm, because single frame is not sufficient to initiate with segmentation part. Therefore, different image frames are essential in order to show different pixel locations. The frame difference d is mentioned with the time duration as $d(t+1)$. In mathematically it is given by,

$$d(t+1)=|V(a,b,t+1)-V(a,b,t)| \quad (1)$$

Where, $V(a, b, t)$ determines the function of video sequence frames, t is the time dimension, a and b are the spatial location variables of pixel,

The difference between the images would show some intensity for the pixel locations which have changed in the two frames. Though, the background is removed as in the ideal condition, the difference between the intensity levels for background pixel is zero. This approach will work only when the background pixels are stable and all foreground pixels are unstable. To improve the process of BS a threshold Th applies to different image frames. Threshold value is limited on the basis of this condition,

$$|V(a, b, t) - V(a,b,t+1)| > Th \quad (2)$$

Where $V(a, b, t+1)|$ is the background of pixel,

The pixel intensities of the different images are filtered or threshold on the basis of Th value. After calculating the parameters, the output of BS image frames are experienced with semantic feature extraction.

Semantic feature extraction in object background detection

In proposed method by using semantic feature extraction like HOG (Histogram of Oriented Gradients), LBP (Local Binary Patterns), PCA (Principal Component Analysis), and GLCM (Gray-Level Co-Occurrence Matrix) are used to extract the background appearance of objects. Here, these four features are selected, because HOG is invariant to geometric and photometric transformations, for significant normalization. PCA is one of the attractive features in object orientation and LBP provides better illumination result in object detection, because the quality of segmentation is based on both illumination change and object orientation. The identification of specific texture in an image is achieved by GLCM.

Histogram of Oriented Gradients (HOG)

HOG descriptors are local feature descriptors. Hence, they are used in image processing for the determination of object detection. The local appearance of the object easily captures in HOG and it is applied for the invariance of object transformation and illumination. HOG feature vectors are used to calculate the edge and gradient information of the image. Here, a simple gradient operator M is applied to determine the value of gradient. The gradient point of the image is given by (a, b) and the image frames are denoted as I.

$$V_a = M * I(a, b) \text{ and } V_b = M^T * I(a, b) \quad (3)$$

The magnitude of the gradients and edge orientation of the point $(a, b)$is calculated by following the respective conditions,

$$V(a, b) = \sqrt{V_a(a, b)^2 + V_b(a, b)^2} \quad (4)$$

$$\theta(a, b) = \tan^{-1} V_b(a, b)/V_a(a, b) \quad (5)$$

For improved invariance in illumination and noise, a normalization process is performed after the calculation of histogram values. The normalization is helpful for contrast and measurement of local histogram. In HOG four different normalization is used such as L2-norm, L2-Hys, L1-Sqrt and L1-norm. Among this normalization L2-norm gives better performance in object detection. The blocks of normalization in HOG is given by,

$$L_{2-\mathrm{norm}}: f = \frac{x}{\sqrt{||x||_2^2 + e^2}} \quad (6)$$

Where e is the small positive value, only when an empty cell is taken into account, f is a feature extracted value, x is the non-normalized vector in histogram blocks. $||x||_2^2$ Represents 2-norm of HOG normalization.

Local Binary Pattern (LBP)

LBP is a simple texture analysis descriptor that converts an input image into a set of labels. The appearance of the images is described by luminance value. A significant property of LBP is gray-scale invariance, which depends on a texture and local pattern. The neighborhood LBP is denoted as a with radius b, which is derived by using the value of central pixel a_c of a as the threshold to denote the value of m neighborhood pixels are located nearer to central pixela_c. The binary value of the pixels are weighted by using the powers of two and then summed to create a decimal numbers stored on the location of central pixela_c, which is mentioned as $LBP(a, b)$.

$$LBP(a, b) = \sum_{i=0}^{m-1} f(a_c - a_i)2^i, f(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (7)$$

Where a_c and a_i , denoted as the gray value of the central pixel of a local neighborhood,

The basic neighborhood model of LBP is (p-neighborhood model) will form 2^poutputs and it leads to large number possible patterns. So, if the texture analysis descriptor area is small, the LBP histogram is not attractive. In addition, the LBP histogram is affected by noise and slight changes. The binary code loops are obtained by the end-to-end of LBP operators. Hence, it is called as uniform model of LBP, it attain only when the jumping time is maximum. It is measured by using following equation.

$$U(LBP(a, b)) = |f(a_{c-1} - a_i) - f(a_0 - a_i)|$$

$$+ \sum_{i=1}^{m-1} | f(a_c - a_i) - f(a_{c-1} - a_i)| \quad (8)$$

Where u is the maximum jumping time,

Principal component analysis (PCA)

PCA is one of the statistical methods often used to reduce the data dimension. Various techniques are preferred in object detection, but the quality of image segmentation results in PCA, because it is concentrated with explaining the covariance and variance structure of the new variables$a_1, a_2, a_3, \dots, a_p$.Therefore, they are not simple and same functions of the others, some variable magnitudes are much higher than others, hence they will receive heavy weights.

To avoid this reason, the variables are determined on scales with different ranges, otherwise the units of the measurements are not equal. Let, R is the sample correlation matrix calculated from n observations on each principal component p of random variables. The Eigen-value and Eigen-vector pairs of R is represented as (γ_1, e_1), (γ_2, e_2), (γ_3, e_3),….(γ_p, e_p). The ith sample principal component of a vector $\boldsymbol{a} = (a_1, a_2, a_3, \dots, a_p)$ variables is given by,

$$e_i z = e_{i1} z_1 + e_{i2} z_2 + \dots + e_{ip} z_p, \quad i = 1, 2, 3 \dots p \quad (9)$$

Where, $e_i z = (e_{i1}, e_{i2}, \dots e_{ip})$ represents ith eigen value and $z = (z_1, z_2, \dots z_p)$ is the standardized vector observation.

In principal component, the sample variance is represented as n_i and the sample covariance pairs are mentioned as zero. In addition, the total sample variance in all standardized variables are equal to the total sample variance in the principal component. In mathematically the standardized vector observation is expressed as follows,

$$z_k = \frac{a_k - \overline{a_k}}{\sqrt{x_{kk}}}, \qquad k = 1, 2, 3 \dots p \quad (10)$$

Where, $a_k$denotes sample mean andx_{kk} is the sample variance of the variable $\boldsymbol{a}_k$

Gray Level Co-Occurrence Matrix (GLCM)

GLCM approach is used to compute the frequency of pixel pairs, only when the gray level value of images is same. The correlation between the neighboring pixels and the reference pixels are calculated to determine the texture feature of the image. The gray level pixel pair are partitioned by a distance d in ascertain angleθ. In GLCM, the number of rows and columns depends on the gray level values.

The element of co-occurrence matrix V (a, b) specifies the number of conversions between the gray level a and b. Therefore, the size of the co-occurrence matrix is reduced by quantizing the number of gray level intensities. Before calculating the co-occurrence matrix, it is compulsory to describe the relations among the pixels. The following equation (11) is used to measure the number of transitions between the pair of gray levels in the texture.

$$V(a, b) = \{(i, j), (k, l)\} \epsilon \text{ s } |f(i, j) = a \text{ and } f(k, l) = b| \quad (11)$$

Where, s is represented as count,

After the calculation of frequency gray level transition, V (a, b) is placed at the a-th row and b-th column of the matrix. Then the feature descriptors are removed after the normalization procedure. Normalization is calculated by using following equation,

$$V(a, b) = \frac{V(a,b)}{\sum_{i=0}^{R_g} \sum_{j=0}^{R_g} V(i,j)}, \qquad a, b = 0, 1 \dots R_g \quad (12)$$

Where, R_g denotes largest gray level, any kind of matrix or pair of matrices are used to generate a co-occurrence matrix. The common application of co-occurrence matrix is measuring the texture of the image. In this approach the matrix across two different images are also determined. Such matrices are preferred in color mapping. Generally, PSO (Particle Swarm Optimization) is the widely adopted optimization technique in feature combination, but it gives ineffective result in complex search. For the best combination, an orthogonal learning method is included with PSO, which is discussed below.

Orthogonal Learning Particle Swarm Optimization (OLPSO)

This section discussed about OLPSO, it is one of the newly learning strategy in recent decades, because of easy implementation and peak efficiency in various domain. OLPSO is mainly concentrated on three main components of the algorithm such as personal best experience, global best experience and the worst experience of the respective particles i.

In OLPSO the particle i is spread in a dimensional space d. Each particle i is associated with respective position and velocity. In current state the velocity vector of particle is denoted as $\boldsymbol{a_i} = [a_{i1}, a_{i2}, \dots\dots a_{id}]$ and the position vector is denoted as $\boldsymbol{b_i} = [b_{i1}, b_{i2}, \dots\dots b_{id}]$. Moreover, each particle has historically best position vector $\boldsymbol{p_i} = [p_{i1}, p_{i2}, \dots\dots p_{id}]$. The best position of the particle i depends on the position of neighborhood particles $\boldsymbol{p_n} = [p_{n1}, p_{n2}, \dots\dots p_{nd}]$. The vectors $\boldsymbol{a_i}$and $\boldsymbol{b_i}$are modified randomly and updated in the following equations (13) and (14). In order to find the new velocity and position of the particle si.

$$a_{id} = w a_{id} + c_1 r_{1d} (p_{id} - b_{id}) + c_2 r_{2d} (p_{nd} - b_{id}) \quad (13)$$

$$b_{id} = b_{id} + a_{id} (14)$$

Where, w is the inertia weight, it is used to control the exploitation and exploration capabilities of the algorithm. Parameters c_1 and c_2 are acceleration co-efficient, r_{1d} and r_{2d} are the two randomly generated values within the range of [0, 1] in the d dimensional space.

In order to constraint the velocity within the range, a positive value a_{maxd} is used to control the updated velocity. If suppose a_{id} exceeds a_{maxd} then it is set to $sign(a_{id}) a_{maxd}$. Likewise, if the updated position of the particle b_{id} is out of the range, then the position is set to respective bound. The new position of the particle is calculated and it can replace the $\boldsymbol{p_i}$ or $\boldsymbol{p_n}$if it has an improved fitness value. Then the procedure is repeated for the next particles until termination.

Here, OLPSO utilize an OL strategy to associate the information of $\boldsymbol{p_i}$and $\boldsymbol{p_n}$to develop an improved guidance vector $\boldsymbol{p_o}$,the velocity of the particle is adjusted by using below equation,

$$a_{id} = wa_{id} + cr_d\ (p_{od} - b_{id}) \tag{15}$$

Each particle i is associated with its own guidance vector$\boldsymbol{p_o}$, which is mentioned as $\boldsymbol{p_o}$ = $[p_{o1,}p_{o2}\ \ ....,p_{od}]$ and it is assembled from $\boldsymbol{p_i}$and$\boldsymbol{p_n}$as

$p_o = p_i \oplus p_n$(16)

Where, $\oplus$ it is the symbol of OED (Orthogonal Experimental Design) operation,

The OED has been introduced for the resolution of tracking experimental issues. The efficiency of OED depends on the ability of utilizing an orthogonal array (OA) and capability of using a factor analysis (FA) to determine the best combinations by testing only a restricted number of cases. Therefore, OA is mentioned as $K_N(Q^M)$ with M factor and Q levels per factor, Where K denotes OA and N is the number of combinations of test cases. In OLPSO, the OED depends on a two levels of OA factor, because the choice of the dimension from p_{id} and p_{nd} of the dth dimension are considered as two levels. To optimize the problem of d dimension by using OLPSO based on the construction process of $\boldsymbol{p_o}$.

In the construction process, OA is implemented as $K_N(2^d)$, where d is the number of dimensions and N is given by $N = 2^{[\log_2 d+1]}$. Here, N is the test solution for choosing the corresponding value from $\boldsymbol{p_i}$or$\boldsymbol{p_n}$according to OA. If the level value is 1 in OA, then the corresponding factor picks from $\boldsymbol{p_i}$, otherwise from $\boldsymbol{p_n}$. Estimate each test solution and note the best solution with highest fitness value b_b. After the identification of best solution, then analyze the effect of each factor by using FA.

FA is used to determine the contribution of various level on each factor. Let f_n represents the experimental result of the nth combination ($1 \leq n \leq N$) and S_{dq} represents the effect of q th ($1 \leq q \leq Q$) level in the dth ($1 \leq d \leq D$) factor. The determination of the $S_{dq}$to enhance all the f_n level and then divide all the total countz_{ndq}. If f_n is in the q th level of dth factor then z_{ndq} is 1, otherwise z_{ndq} is 0. The total count is divided by using following equations,

$$S_{dq} = \frac{\sum_{n=1}^{N} f_n \times z_{ndq}}{\sum_{n=1}^{N} z_{ndq}} \tag{17}$$

In this manner, the effect of each level is determined and matched. Finally compare b_b and b_p to evaluate the better solution of the vector$\boldsymbol{p_o}$, here, b_p is a predictive solution.

OLPSO for Feature Selection

Particle Representation

In order to select proper features for background detection OLPSO algorithm is preferred. Here, four features such as HOG, LBP, PCA, and GLCM are used to extract the background appearance of objects. These features are numbered as 1, 2, 3, and 4 respectively. The arrangements done with the features are never changed and also compared in the same order. While comparing, the same feature selection and repeated feature particle combinations are not considered. Totally six feature particle combinations are achieved in OLPSO algorithm and they are mentioned as P1, P2, P3, P4, P5, and P6 respectively.

Total Particle Combinations			
~~[1,1]~~	[1,2]	[1,3]	[1,4]
~~[2,1]~~	~~[2,2]~~	[2,3]	[2,4]
~~[3,1]~~	~~[3,2]~~	~~[3,3]~~	[3,4]
~~[4,1]~~	~~[4,2]~~	~~[4,3]~~	~~[4,4]~~

Fitness Function

Fitness function is a specific type of objective function that helps to achieve high accuracy. Here, the fitness function is considered as recognition rate, because it is used to find out the better feature combination. The recognition rate consists of two error rates such as False Reject Rate (FRR) and False Accept Rate (FAR). FRR is the probability that the system incorrectly rejects the authorized person. Here, the false rejection represents the real humans are considered as vehicles.

FAR is the probability that the system incorrectly authorizes the non-authorize person. Here, the false acceptance denotes the vehicles are identified as humans. In mathematically it is given by,

$$\text{FRR} = \frac{P_{FR}}{P_{AA}}, \quad \text{FAR} = \frac{P_{FA}}{P_{IA}} \tag{18}$$

Where, P_{AA} is the human identification and P_{IA} is the vehicle identification in the d dimensional space; P_{FR} and P_{FA} represents the times of false acceptance and false rejection. The formula for recognition rate is given by,

$$\text{RR} = \max(1 - \frac{P_{FR} + P_{FA}}{P_{AA} + P_{IA}}) \tag{19}$$

Implementation of OLPSO for Feature Selection

The implementation using OLPSO, to find the optimal particle combination values for the feature selection, with the help of particle representation and fitness function. Here, particle i represents the object. At first, initialize the velocity $\boldsymbol{a_i}$and position $\boldsymbol{b_i}$of the respective object with random values in their search ranges $[a_{mind},a_{maxd}]$ and $[b_{mind},b_{maxd}]$, respectively. And then set its personal best position $\boldsymbol{p_i}$as$\boldsymbol{b_i}$. Analyze all the particles fitness value and calculate the neighborhood best position $\boldsymbol{p_n}$for each particle**.** Determine *gbest* (Global Best Position) value by comparing all the particles position by using following condition *gbest*= *max*{*fit*($\boldsymbol{p_i}$)}.

OLPSO Combinations	
Particles	Combinations
P1	[1,2]
P2	[1,3]
P3	[1,4]
P4	[2,3]
P5	[2,4]
P6	[3,4]

Update the current velocity and position of the particle in the equations (13) and (14). After evaluate the new position $\boldsymbol{b_i}$, if *fit*($\boldsymbol{b_i}$)>*fit*($\boldsymbol{p_i}$), then set new position as personal best position $\boldsymbol{p_i}$. Otherwise set the previous position as personal best position $\boldsymbol{p_i}$. If *fit*($\boldsymbol{p_i}$)>*fit*($\boldsymbol{p_n}$), then set new position as neighborhood best position $\boldsymbol{p_n} = \boldsymbol{p_i}$ and also if *fit*($\boldsymbol{p_i}$)>*fit*(*gbest*), then set *gbest*=$\boldsymbol{p_i}$. Repeat the same procedure for next particles*i*, if the number of fitness evaluations or number of iterations reaches the maximum, then terminate the process and denote current best position as *gbest* position. Here, best feature particle combination, given as the input for(SVM) Support Vector Machine.

Supervised learning model SVM associated with learning algorithms, mainly concentrated on pattern classification and regression. In SVM a set of training examples are given as input and it builds a model, which assigns new examples to one category or the other. It is denoted as non-probabilistic binary linear classifier. Hence, the performance of SVM in non-linear classification is very efficient. So, SVM classifier is preferred, for training the best feature combination.

EXPERIMENTAL SETUP

In this section, experimental results have been described in detailed. All experiments were implemented on PC with 1.8GHz Pentium IV processor using MATLAB 2015B. Here, the proposed algorithm OLPSO has been tested on two video sequence databases (PETS2009 and MIT) for four different features. The fitness function of OLPSO used to test all the particle combinations. Based on the parameter value, the best feature combination is evaluated. Parameters are defined on the basis of precision, recall and accuracy.

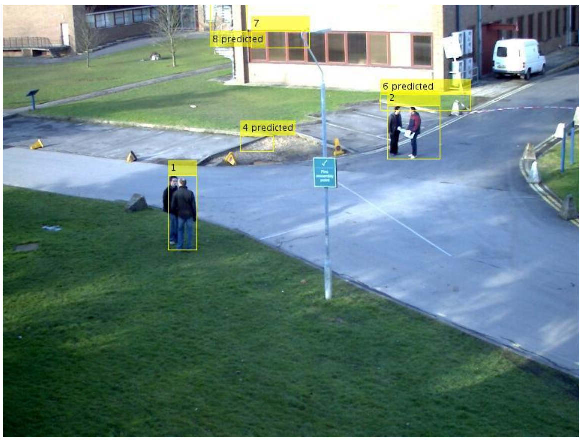
Quantitative Evaluation on Without OLPSO

The following section indicates quantitative evaluation result of without OLPSO. Here, the performance of the uni modal features depends on MOTA (Multiple Object Tracking Accuracy)**.** In uni modal feature extraction**,** the MOTA would be negative number, because the total sum of misses, the number of false positive and the number of mismatch errors for the frames are too many. Hence, it affect the accuracy of object recognition and object localization. To overcome this issue, a learning method can be included, because these features are superior in specific quality like texture, shape, color etc. The combination of features may provide better result in object tracking.

Quantitative Evaluation on With OLPSO

Here, the quantitative evaluation has been done by using OLPSO. This algorithm gives a comparable result in accuracy, it helps us to find the best feature combination. These combination results reduce the number of mismatch errors and number of false positives. Therefore, it is essential to give better accuracy in object recognition and object localization.

Figure 1: Detection of moving objects in PETS2009 Database (Horizontal View)



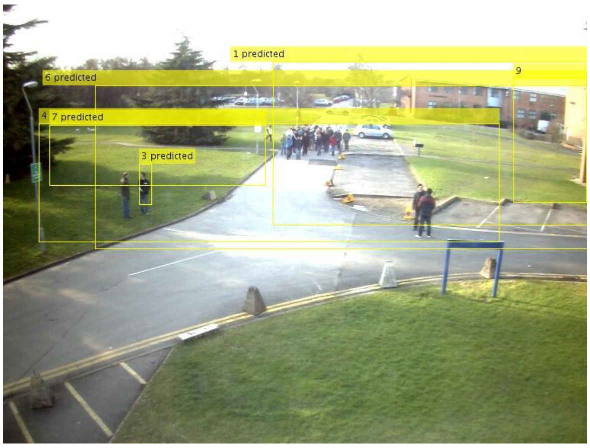


Figure 2: Detection of moving objects in PETS2009 Database (Vertical View)

Figure 3: Detection of moving objects in MIT Database

DISCUSSION

Here, two different views (vertical and horizontal) of PETS2009 database are considered. In above, first four frames represents the vertical view and second four frames represents the horizontal view of the same residence. On other hand, the moving objects are determined as (Humans, vehicles etc.). For instant, on frame 11, vertical view predicts two persons as single prediction, but in horizontal view the two persons are predicted as two. So, it clearly verifies the prediction various based on the direction of camera.It cause more segmentation loss and fails to provide better accuracy. In proposed, after verifying the different camera angles, best combination of camera view is selected. Hence, there is limited in segmentation loss and provides a nearest prediction with better accuracy. Along with camera angles, environmental changes like (wind, rainfall, lights etc.) are considered as major issues in object detection. Whereas, due to bright light, shadow of the objects are predicted before the objects entering into the frame.

In MIT database, it includes two different objects, fast moving and slow moving objects. These objects are predicted based on motion vectors. In motion vector, the fast moving objects are easily predicted than slow moving objects. Here, the prediction rate of fast moving object (vehicles) are higher than the prediction rate of slow moving objects (Humans), because the vector weight count of fast moving objects are more than compared with slow moving objects, In this database, unidirectional camera angle is utilized, the overlapping is high. In order to avoid overlapping factor, multi-directional cameras are preferred.

The performance of detection, on the video sequencePETS2009 and MIT database frames are shown in table1. Compared with the other three methods, our proposed gets a better performance, by analyzing the respective values.

In this experiment, four features are obtained, these features has specific quality in image processing. While applying individual features, it gives reasonable prediction value. Here, a learning algorithm OLPSO is implemented, it finds the best combination of features and this feature combination is given as the input for SVM classifier. Hence, SVM trains 60% then test 40% of feature combination and these respective combinations are individually taken for human, vehicle and the mixture of both vehicle and human. Even after applying OLPSO gives a sufficient prediction level due to weight factor of feature combination and also it affects the segmentation accuracy by referring the value of precision and recall. The following table shows the value of precision, recall and accuracy.

The following table 2, determines the precision, recall and accuracy values for two different databases by comparing the four methods (With OLPSO, Without OLPSO, Method1 and method2). It clearly verifies that our proposed With OLPSO performs significantly better than compared with other methods.

Database	Method	Precision	Recall	Accuracy
PETS 2009	With OLPSO	0.903	0.672	0.856
	Without OLPSO	0.891	0.839	0.743
	Method 1	0.762	0.893	0.692
	Method 2	0.814	0.864	0.712
MIT	With OLPSO	0.871	0.745	0.687
	Without OLPSO	0.745	0.798	0.639
	Method 1	0.716	0.844	0.572
	Method 2	0.670	0.868	0.524
MATLAB	With OLPSO	0.894	0.728	0.745
	Without OLPSO	0.784	0.756	0.697
	Method 1	0.432	0.832	0.643
	Method 2	0.567	0.793	0.673

Database	Sample of Frames	Precision	Recall	Accuracy
PETS 2009	Frame 1	1	0.8	0.883
	Frame 2	0.75	0.75	0.8
	Frame 3	1	0.75	1
	Frame 4	0.5	0.75	0.57
	Frame 5	0.285	0.667	0.27
MIT	Frame 1	1	0.4	0.294
	Frame 2	1	0.57	0.411
	Frame 3	0.8	0.667	0.352
	Frame 4	0.8	0.667	0.352
	Frame 5	0.6	0.6	0.294

CONCLUSION

In this paper, an intelligent video surveillance system for detection of a moving object is implemented by using OLPSO algorithm. Among all the features, OLPSO identifies the best feature combination, based on its fitness function. In OLPSO, PETS2009 Database obtains 11 % of higher accuracy in prediction than other existing methods, because in this Database different camera views of same residence is preferred and also environmental effects are limited in range.

Likewise, in MIT Database, due to its weighting factor the fast moving objects are easily predicts, but the slow moving objects are reasonably predicts. Still, by using OLPSO achieves 6% of higher prediction rate compared with other methods. In future, based on the motion of objects, the activities of the respective objects are determined.

REFERENCES

[1] Ren, Haoyu, and Ze-Nian Li, "Object detection using boosted local binaries",*Pattern Recognition*, Vol.60, pp.793-801, 2016.

[2] Huang, Xiao, Yuanqi Su, and Yuehu Liu, "Iteratively parsing contour fragments for object detection",*Neurocomputing*, Vol.175, pp.585-598, 2016.

[3] Bria, Alessandro, Claudio Marrocco, Mario Molinara, and Francesco Tortorella, "An effective learning strategy for cascaded object detection",*Information Sciences*, Vol.340, pp.17-26, 2016.

[4] Chen, Yingying, Jinqiao Wang, Min Xu, Xiangjian He, and Hanqing Lu, "A unified model sharing framework for moving object detection", *Signal Processing*, Vol.124, pp.72-80, 2016.

[5] Chen, Xiaorong, Chuanli Xi, and Jianghui Cao, "Research on moving object detection based on improved mixture Gaussian model",*Optik-International Journal for Light and Electron Optics*, Vol.126, issue.20, pp.2256-2259, 2015.

[6] Lee, Seungwon, Nahyun Kim, Kyungwon Jeong, Kyungju Park, and Joonki Paik, "Moving object detection using unstable camera for video surveillance systems",*Optik-International Journal for Light and Electron Optics*, Vol.126, issue.20, pp.2436-2441, 2015.

[7] Hu, Wu-Chih, Chao-Ho Chen, Tsong-Yi Chen, Deng-Yuan Huang, and Zong-Che Wu, "Moving object detection and tracking from video captured by moving camera",*Journal of Visual Communication and Image Representation*, Vol.30, pp.164-180, 2015.

[8] Ko, ByoungChul, Ji-Hun Jung, and Jae-Yeal Nam, "View-independent object detection using shared local features",*Journal of Visual Languages & Computing*, Vol.28, pp.56-70, 2015.

[9] Yoshinaga, Satoshi, Atsushi Shimada, Hajime Nagahara, and Rin-ichiro Taniguchi, "Object detection based on spatiotemporal background models",*Computer Vision and Image Understanding*, Vol.122, pp. 84-91, 2014.

[10] Yu, Huapeng, Yongxin Chang, Pei Lu, Zhiyong Xu, Chengyu Fu, and Yafei Wang, "Accurate object detection with a discriminative shape model", *Optik-International Journal for Light and Electron Optics*, Vol.125, issue.15, pp.4102-4107, 2014.

[11] Conde, Cristina, Daniela Moctezuma, Isaac MartíN De Diego, and Enrique Cabello, "HoGG: Gabor and HoG-based human detection for surveillance in non-controlled environments",*Neurocomputing*, Vol.100, pp.19-30, 2013.

[12] Qian, Huimin, Yaobin Mao, Wenbo Xiang, and Zhiquan Wang, "Recognition of human activities using SVM multi-class classifier",*Pattern Recognition Letters*, Vol.31, issue.2, pp.100-111, 2010.

[13] Zhan, Zhi-Hui, Jun Zhang, Yun Li, and Yu-Hui Shi, "Orthogonal learning particle swarm optimization", *IEEE transactions on evolutionary computation*Vol.15, issue.6, pp.832-847, 2011.

[14] Zhang, Peng, Yanning Zhang, Tony Thomas, and Sabu Emmanuel, "Moving people tracking with detection by latent semantic analysis for visual surveillance applications",*Multimedia tools and applications*, Vol.68, issue.3, pp.991-1021, 2014.

[15] Zhang, Shanshan, Dominik A. Klein, Christian Bauckhage, and Armin B. Cremers, "Fast moving pedestrian detection based on motion segmentation and new motion features",*Multimedia Tools and Applications*, Vol.75, issue.11, pp.6263-6282, 2016.