

OBJECT TRACKING IN VIDEOS BY EVOLUTIONARY CLUSTERING AND LOCALLY LINEAR NEURO-FUZZY MODELS

Fatemeh Saadian

Original scientific paper

In this paper a new method based on evolutionary clustering and locally linear neuro-fuzzy (LLNF) models is proposed for the problem of object tracking in videos. This approach utilizes clustering on color feature space to obtain a model of object which is given at the initial frame. To achieve the optimal clustering, evolutionary optimization methods are used. Based on the results of clustering, parameters of LLNF model is determined so it can be used as an identifier of object during the real time video streaming. To track the object, a swarm of weighted evolving linear models are used to estimate the location and size of the object at next frame based on its current and previous states. The performance of the proposed method is evaluated on a benchmark data set and compared to other methods performed on the same data set. The results show that the accuracy of the proposed method is superior to previous methods.

Keywords: clustering; evolutionary computation; Locally Linear Neuro-Fuzzy model; object tracking; swarm optimization

Praćenje predmeta u video snimkama pomoću modela evolucijskog grupiranja i lokalno linearnih neuro-fuzzy modela

Izvorni znanstveni članak

U radu se za problem praćenja objekta u video snimkama predlaže nova metoda koja se zasniva na modelima evolucijskog grupiranja i lokalno linearnim neuro-fuzzy modelima (LLNF). Taj pristup koristi grupiranje na prostoru karakteristike u boji kako bi se dobio model predmeta zadanog u početnom okviru. Za postizanje optimalnog grupiranja, primijenjene su metode evolucijske optimizacije. Na temelju rezultata grupiranja određeni su parametri LLNF modela te se on može rabiti kao identifikator objekta tijekom video prijenosa u realnom vremenu. Za praćenje objekta koristi se roj ponderiranih izvedenih linearnih modela za procjenu lokacije i veličine objekta u sljedećem okviru na temelju njegovog postojećeg i prethodnih stanja. Učinkovitost predložene metode procijenjena je na referentnom nizu podataka i uspoređena s drugim metodama provedenim na istom nizu podataka. Rezultati pokazuju da je po točnosti predložena metoda bolja od prethodnih.

Ključne riječi: evolucijski izračun; grupiranje; lokalno linearni neuro-fuzzy model; optimizacija roja; praćenje predmeta

1 Introduction

The advancements of machine vision technology in recent decades have provided a broad range of new capabilities for human beings. Currently, the applications of machine vision, image and video processing include, but are not limited to, biology and medical applications, space and aeronautics, surveillance and traffic control, robotics technology, sports, arts and media, etc. [1, 2]

One of the most important and challenging issues in machine vision is tracking of moving objects in videos in real time applications. Several aspects of this problem make it relatively hard for system developers: It is often necessary to perform real time computations, even sometimes with a faster output generation than the sensory sampling of the system, which means one needs the output of tracking algorithm before the arrival of next image in the sequence of video recording. Objects with high speeds and variations in their movement are naturally more difficult to be tracked. Changes in the position of camera, the structure, lighting and scenery of the background, and also variations in shape, appearance and orientation of the objects are other possible challenges in a tracking problem.

The next sections of this paper summarize as follows. In section 2 the previous works and literature related to the problem of object tracking in videos are reviewed which include research in either applications or methods. Section 3 introduces the problem of object tracking in a more formal and definitive way. That part also describes several challenges in application of different approaches to this problem. In section 4 of the paper the features as used to model and identify object in an image (frame in

video) are described, and the problem of clustering for the feature space is introduced. Also in that section the evolutionary approach to the clustering problem is described. The concepts of fitness function, genetic algorithm, and particle swarm optimization are presented to be utilized for clustering of points in the feature space. Locally Linear Neuro-Fuzzy (LLNF) model as a main part of the proposed tracking method in this paper is presented in section 5. Section 6 provides a holistic description of the proposed tracking method which combines concepts of evolutionary algorithms, clustering, and LLNF. In section 7 the data used for evaluation, the results of operating the proposed method on data set, and comparison with other approaches and methods are presented. Section 8 provides the concluding remarks.

2 Related works

The problem of object tracking has been studied by many researchers during recent decades, and with a diverse range of approaches [3]. The challenges of this problem for different applications and situations are not the same. Many of previous works have been focused on tracking of human face in videos [4, 5]. Others have been concerned on robotic applications of object tracking [6, 7, 8].

One of the most familiar approaches to object tracking is Kalman filter tracking and its various extended versions [9, 10, 11]. Kalman filters assume linear models for state evolution of a system and try to estimate the underlying dynamics based on sequential observations of outputs. But the need for generalization to nonlinear systems and non-Gaussian noise on observations has

made researchers to propose more complex methods [12÷15]. In the context of object tracking in videos, particle filters have attracted researchers in recent years [16÷18].

One of the challenges in object tracking is to provide a simple but robust model of background and/or the object itself. Many of previous works have tried to model the background of videos [19, 20], but this approach only works well when the background is near steady, for example when camera is not moving itself. In any case, it is almost necessary to segment object in the background. To achieve such segmentation, clustering approaches have been used in previous works [21, 22]. Weng et al. proposed a combination of k-means clustering and extended Kalman filters for object tracking [23].

Recently, utilization of neural networks [24] and neuro-fuzzy models [25] have been also introduced in literature for the problem of tracking. The benefits of these models include their high capability in mapping nonlinear relations and their generality of application. Using boosted learning is also one of the advanced approaches to the problem of object tracking in videos [26].

To implement these general methods for the problem of tracking, one needs to set their parameters as a task of learning. The approach of using evolutionary computation for learning of fuzzy neural networks is one of the most reliable strategies [27].

3 Object tracking problem

Generally, in object tracking tasks one tries to identify, locate and predict the motion of one or several objects in a video signal as a sequence of ordered images (frames). Based on different circumstances, the problem of tracking can be considered in different versions. In some of the tracking tasks, the object of interest is predefined to the algorithm by user, but in other tracking tasks the object may not be predetermined to the system. As an example, in some surveillance systems, the detection of a suspicious motion in the steady background may be considered as a part of the system which determines the moving object automatically, without the need for predetermination by a user. Another issue is the movement of camera, as for the steady camera systems one may have a simpler background model. In single object tracking, the system tries to locate the position of a single object, but in other multi-object tracking problems there could be several objects of interest.

In this section we formalize the definition of object tracking problem that we address in this paper. A video signal is a function of discrete time and two dimensions of position (we do not consider depth in this paper) from points in a subspace of \mathbb{N}^3 to a grey (or color) intensity variable c which is generally in \mathbb{R} (or \mathbb{R}^3) space. We define T as the sequence (set) of natural numbers from 1 to t_f which counts the number of frames. Also consider screen S as a subspace of \mathbb{N}^2 which defines the set of all possible pixel positions in a frame. Then a video V maps an intensity value $I(t, x, y) = c \in C$ to each pixel $(x, y) \in S$ at a time $t \in T$:

$$V = I(t, x, y) : T \times S \rightarrow C \quad (1)$$

where $T \subset \mathbb{N}$, $S \subset \mathbb{N}^2$, and $C \subset \mathbb{R}$ (or \mathbb{R}^3 for RGB color signals).

An object O_t at the time t is a subset of pixels in screen S . Then tracking is to find an estimation \hat{O}_t in a way that

$$\hat{O}_t \rightarrow O_t \text{ i.e. } \hat{O}_t \cup O_t \rightarrow O_t \text{ and } \hat{O}_t \cap O_t \rightarrow O_t \quad (2)$$

it means we want to find the set of pixels \hat{O}_t as much as equivalent to O_t . Practically in implementations, we will define bounding boxes BB_t and $\hat{B}B_t$ for O_t and \hat{O}_t , and try to maximize their mutual overlap.

In addressing such tracking problems, there are several considerations which should be taken into account. In some of the situations one might be able to incorporate some assumptions into the problem to make it easier to solve. On the other hand, in some situations camera movements, the variations in motion pattern, shape and orientation of the object, and also variations in lighting situations may need to be taken into account and the solution method can be more difficult.

In a complete tracking problem, several steps have to be performed. First is the definition or detection of the object. In some systems, the object will be detected by its novelty or distinguishable movement in the background. In most of such situations, a near steady model of background can be defined. Then every region of a frame which is not correlated to that background model can be suspected as a foreground object of interest. In a simpler situation, one may assume a steady background and use the differential image (the difference of two consecutive frames) to find the moving object. But in such assumptions, one has to be careful about issues like temporal stopping of the object, moving in depth, or rotations. Anyway, either by automatic detection or predefinition of object by user, the next step is to identify the object in each frame, by estimating the location of pixel sets assigned to it on the screen set. In this paper, we use modeling of the object based on its features, and then find regions of screen with same feature content to the object. The other important step in tracking is the prediction of the location of the object in the next frame. It is not only important because of its application related necessity, but is important because it can cause a significant improvement in the speed of computations since it reduces the search space in the screen for the object at the next frame. As an example for application necessity of location prediction (in many works this step is actually called the tracking step) consider designing a robot arm with task of grasping a moving ball.

4 Clustering in colour feature space using evolutionary methods

In this section we describe the procedure of extracting a model for the object, which is assumed to be given by user in our considerations in this paper. Therefore, at the beginning of the tracking procedure, a region in the first frame (or a bounding box around it) is determined by user of the object. Then automatically, the system extracts a model based on several features of the object in order to be utilized in next frame as the identification tool of the

object. In the context of image analysis, one can have a very broad range of features of the signal. These include frequency domain features like Fourier coefficients or wavelet coefficients, Hough transforms, color features, geometric features, etc. The selection of appropriate feature representation for each specific problem is a very sensitive, important, and almost intuitive task. A feature representation very useful for one situation may be no good for another one at all. In addition, trying to use an optimally fewer number of features in modeling the object is not only effective in reduction of computation time, but also may increase the generalization of the model. A good feature representation has to be robust to variations of lighting, shape, and orientation of the object.

In this paper we address the tracking of objects in color videos, so the use of color features to extract a model of object seems straightforward. Anyway, in this paper we develop a method for tracking of objects based on clustering and LLNF models, and then in other applications with other appropriate feature representations the same approach can be used. In fact, as we introduce a general procedure of clustering, then this would be non-relevant that what kinds of features one tries to use for modeling, as long as it can be locally defined on regions and be used in clustering.

4.1 Local colour features

Local features are extracted by considering small neighborhoods around pixels, and calculation of some quantities from these tiny images. In this paper we use color features of such local neighborhoods. For each pixel in the RGB color image, there are three values of intensity for each component $\{c1, c2, c3\} = \{r, g, b\}$. We can also represent pixels in their HSV representation $\{c4, c5, c6\} = \{h, s, v\}$ [28]. Then consider a rectangular d -neighborhood $D(x, y)$ around pixel (x, y) :

$$D(x, y) = \{(x - d, y - d), (x - d + 1, y - d), \dots, (x, y), \dots, (x + d, y + d - 1), (x + d, y + d)\} \quad (3)$$

Then we can assign average values of $c1$ to $c6$ for $D(x, y)$ as the feature representation assigned to (x, y) . Other quantities like variances, or even all the c values in $D(x, y)$ can be also considered as features.

4.2 Clustering of samples

Clustering is used in the proposed method of this paper for learning (setting the parameters) of LLNF as the model of object. In this regard, on an initial frame, for which the object is determined by user, several random positions are selected, and the feature values assigned to them are calculated. Consider a random selection of N sample points (x_i, y_i) on screen where $i = 1, 2, \dots, N$. Then there are N feature vectors $f(x_i, y_i)$ with length 6. Also for each point (x_i, y_i) a label L is assigned based on the user determination of the object O_0 in initial frame:

$$f(x_i, y_i) = \begin{cases} 1, & (x_i, y_i) \in O_0 \\ 0, & (x_i, y_i) \notin O_0 \end{cases} \quad (4)$$

The goal of clustering is to partition the whole set of sample points into several subsets (cluster), and assign a cluster center to each cluster in a way minimizing an objective function which is mainly the summation of distances of points in each cluster to its assigned center. Other kinds of objectives like maximization of distances between centers may be incorporated.

Many methods of data clustering have been introduced in literature. The most famous one of clustering methods is k-means algorithm [23]. This algorithm is fast and useful to many problems, but it has several drawbacks. In fact, it may be incapable of finding the globally optimum solution to the clustering problem in some cases, and be not very accurate confronting highly mixed datasets.

The approach of this paper to clustering is using evolutionary methods. In recent decades, evolutionary computation has been grown majorly and found a very diverse range of applications in engineering, science, management, economics, and even arts. Generally, in evolutionary methods, one generates initial solutions to a problem almost randomly at the beginning. Then by means of several appropriate evolution operators performing iteratively on the solutions, tries to find better solutions based on the fitness (objective) function of the problem. Most of famous evolutionary methods have been inspired from biological processes in nature like genetic evolutions, or flocking of organisms. In the next subsection, main ideas of evolutionary methods of genetic algorithms and particle swarm optimization are described, and their use in clustering of data is explained.

4.3 Evolutionary clustering

For any evolutionary method to be used, we need to define an appropriate representation of the possible solution for the problem, and also define a fitness function which maps each possible solution to a real value as the measure of fitness of that solution to be the best one for the problem in hand. In most of cases the solution representation is an array of values, with each element representing one aspect or dimension of that solution [29-32]. In genetic algorithm (GA) [33] such array is called chromosome, and in particle swarm optimization (PSO) [34] it is called particle position. Let $X = (X_1, X_2, \dots, X_m)$ represent such array in any of those algorithms, where m is the dimension of the problem. Fitness function is a real value $F(X)$, which has to be minimized or maximized by the algorithm. In minimization problem, one wants to find \bar{X} for which $F(\bar{X}) < F(X)$ for all X in the domain of the problem. The specific thing to each algorithm is the way they evolve the initial solutions iteratively to find \bar{X} or at least a good estimation of it.

In conventional GA, there are two types of genetic operators used to evolve chromosomes: crossover, and mutation. Crossover operator combines a partial chromosome with the complementary parts of another chromosome to generate a new offspring chromosome (parent1 \otimes parent2 \rightarrow child). For mutation operator, one or several genes (elements of chromosome array) are selected and changed randomly. The selection of parent chromosomes is done based on their assigned selection probabilities which depend on their fitness values. In this

way, being better a chromosome, the more is its chance to produce new generation. The next population of chromosomes would be composed of the best chromosomes among parent, offspring, and mutated chromosomes according to their corresponding fitness values.

The evolution of solutions in PSO is inspired from the flocking of birds (namely particles). Initial solutions are considered as position vectors of several particles with some initial random velocities v_i . Based on fitness function, a globally best position g_{best} among all particles in iteration is found. Also each particle i has its own memory of best experienced position p_{best}_i from beginning to the current iteration. Then the velocities of all particles are updated as

$$v_i(t) = c_1r_1(g_{best} - X_i) + c_2r_2(p_{best} - X_i) \tag{5}$$

Then the updated position of particle is $X_i(t + 1) = X_i(t) + v_i(t)$. This procedure utilizes the tendency of particles toward the best experienced position, and so during their movement they may find new better positions.

To use evolutionary methods in our clustering problem of sample points in the initial frame of the video, we take the positions of cluster centers in color features space (6-dimensional in our choice of features) as the solution array (chromosome or particle position). The fitness function that we try to minimize is

$$F = w_1 \sum_{j,i} |G_j - f_{j,i}| - w_2 \sum_{j,k} |G_j - G_k| + w_3 \sum_j \sigma_j^2 \tag{6}$$

The first term in fitness function is the sum of distance between a cluster centers G_j to its connected sample points $f_{j,i}$. Each sample point is considered as connected to its closest cluster center. The second term is the sum of distances between cluster centers, and it has negative sign because we want it to be maximized. The third term is the sum of variances of labels in clusters. In fact, the third term is considered to ensure the maximized uniformity of labels within clusters. The weights w_i are introduced to make all terms in same order of magnitudes; these are set by intuitive considerations on numbers of clusters and data points.

5 Locally linear neuro-fuzzy model

The goal of neuro-fuzzy systems is to combine and take the advantages of artificial neural networks with fuzzy inference. Artificial neural networks (ANN) are mathematical models capable of representing a mapping between inputs and outputs. To obtain such mapping, several learning algorithms have been introduced in literature. There are also diverse types of ANNs including feed-forward networks and radial basis function (RBF) networks. Fuzzy inference systems (FIS) are methods used for decision about a value by means of several rules based on fuzzy logic in an environment for which the data can be uncertain. The concept of fuzzy membership functions helps to generalize the notion of membership of elements in sets. In fact, in fuzzy logic an element can be

a member of a set by membership degree of u (with $0 < u < 1$), and at the same time be a non-member of it with degree $1 - u$.

Locally linear neuro-fuzzy (LLNF) systems [35] are RBF like neural networks with each neuron accompanied with a set of (usually Gaussian) fuzzy membership functions and a linear function of inputs. It is locally linear because the fuzzy functions act as selective weighting functions for the linear part, and weigh high for some regions of input space and weigh low for other parts.

For a n -dimensional vector of inputs x , the output of the r^{th} neuron is calculated by rules decision A_r and with weight w_r :

$$A_r = a_r x + a_{0r} \tag{7}$$

$$w_r = \exp[-(x - c_r)^T \sum_r^{-1} (x - c_r)] \tag{8}$$

where the n -vector a_r and number a_{0r} define the linear rule; c_r is the mean n -vector, and the Σ_r is the $n \times n$ variance matrix of the n dimensional Gaussian function. The overall output of the network can be calculated by normalized weighted summation

$$y = \frac{\sum_r w_r A_r}{\sum_r w_r} \tag{9}$$

The learning of system means to find the appropriate values for parameters c_r , Σ_r , a_r , and a_{0r} in a way providing the minimum error in mapping between inputs and outputs. The approach of this paper to set those parameters is using evolutionary algorithms and clustering, with the goal of finding the best model for the object presented by user in the first frame of the video. The proposed method is described in the next section of paper.

6 Proposed method

In order to construct a model of object to distinguish it from the background based on its color features, in this paper the clustering of sampled points from the initial frame is done by means of evolutionary optimization of the clustering fitness function. Then, the result of clustering is used to set the parameters of LLNF model. The output of LLNF for the object sample points has to be equal to 1, and for the background points to be 0. Then during the tracking task when video is streaming, the trained LLNF model is used to decide for each sample point in a frame is a part of object or it is not. It is emphasized that this method decides a point being foreground (object) or not, but it does not decide that a point is in background. In other word, a background model is not constructed in this method, because the background generally can be varying.

The way the results of clustering define the parameters of LLNF is described as follows. Let's assume we have R clusters found by means of evolutionary clustering as described in previous sections. Then we

consider R neurons for the LLNF model. The mean vector c_r of Gaussian function of r -th neuron is set to be equal to the center point of that cluster. The Σ_r matrix is considered as a diagonal matrix with the i^{th} diagonal element equal to the square of maximum distance in the i^{th} dimension within the cluster from its center point. For linear rule coefficients, at first a_r is set equal to zero vector, and $a_{0,r}$ is equal to L_r , which is the label of cluster i.e. the average label of points in that cluster. If the results are satisfactory, there would be no need to complicate the model with incorporation of non-zero a_r . However, it is always possible to turn those coefficients on and try to set them optimally as a part of evolutionary optimization step.

For the tracking of the object in video stream, two tasks have to be done for each frame; predicting the location and size of the object in the new frame before its observation, and identifying the object when the frame is observed. For the second part, the LLNF as trained by clustering is used by its operation on random sample points on the screen of the new frame. For the other task, predicting the behavior of the object, a combination of weighted linear predictors with evolutionary search is used in this paper. If the location and size state of the bounding box of the object at time t is, $X_t = (x, y, w_x, w_y)$ then the next state is calculated by a linear predictor being

$$X_{t+1} = B_0 + B_1 X_t + B_2 X_{t-1} \tag{10}$$

where B_0 , B_1 and B_2 are 4×1 , 4×4 , and 4×4 matrices respectively. To achieve the goal of evolving predictor, we use a swarm of those linear predictors and assign a weight to each. The weight of each model depends on its performance in predicting the object. In fact, at each frame, the best linear predictor obtains the highest weight. Then the coefficient parameters of the i^{th} model (the elements of B_0 , B_1 and B_2), if all represented in one 1×36 array P_i will be updated as

$$P_i(t+1) = P_i(t) + \delta \left(\frac{|w_i(t) - w_{\text{best}}(t)|}{\sum_i w_i} \right) (P_{\text{best}}(t) - P_i(t)) \tag{11}$$

where the index "best" represents the current best one of linear predictors with highest weight, and δ is a random coefficient. For the next time step, the new B matrices are obtained again from the new P arrays. This procedure is

similar to the idea of PSO algorithm, and it provides the evolution of predictor model towards better performance in predicting the objects behavior.

7 Results

The dataset of videos used for evaluation of the proposed method is named BoBoT (Bonn Benchmark on Tracking) [26]. This data set is purposefully generated for testing object tracking algorithms. It includes cases with color videos of moving objects, moving cameras, low and fast movements, rigid and non-rigid objects, indoor and outdoor sceneries.

At first a distribution of sample points in color feature space obtained from one frame of a video, which is shown in Fig. 1, is generated by random spreading sampler windows on the image. The sample points in space of 3-features $\{r, g, b\}$ and in 3-features $\{h, s, v\}$ are shown in Fig. 2. The points from object and from background are marked with different markers. It is clear that many points from object have very distinctive features from the points of background. Of course, there are some overlaps between the two classes, but the idea of the proposed method is to weigh more on distinctive regions by means of LLNF. It should be noted that in next frames there may be several points in background with close features to some of object points, and then be classified as foreground during the tracking procedure. This effect is reduced by utilization of predictor model to limit the search region on the screen. In addition, by sampling enough points, the aggregated results will eliminate effects of such false foreground estimation points.

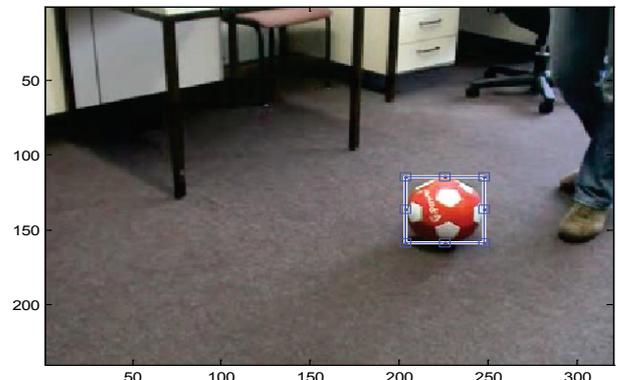


Figure 1 A frame from one of the videos of data set, object shown in bounding box [26]

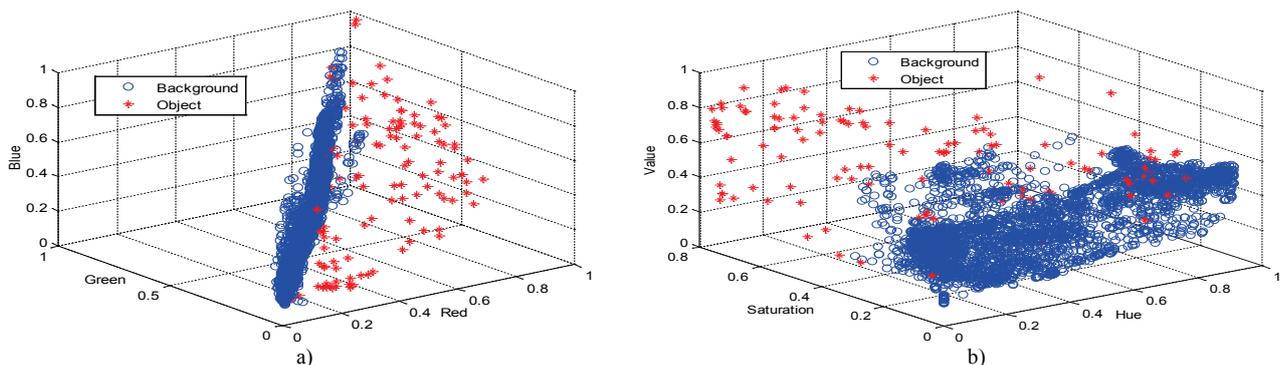


Figure 2 Sampled points from initial frame shown in (a) RGB colour space, and (b) HSV colour space. Foreground (object) and background points are distinguished with different markers.

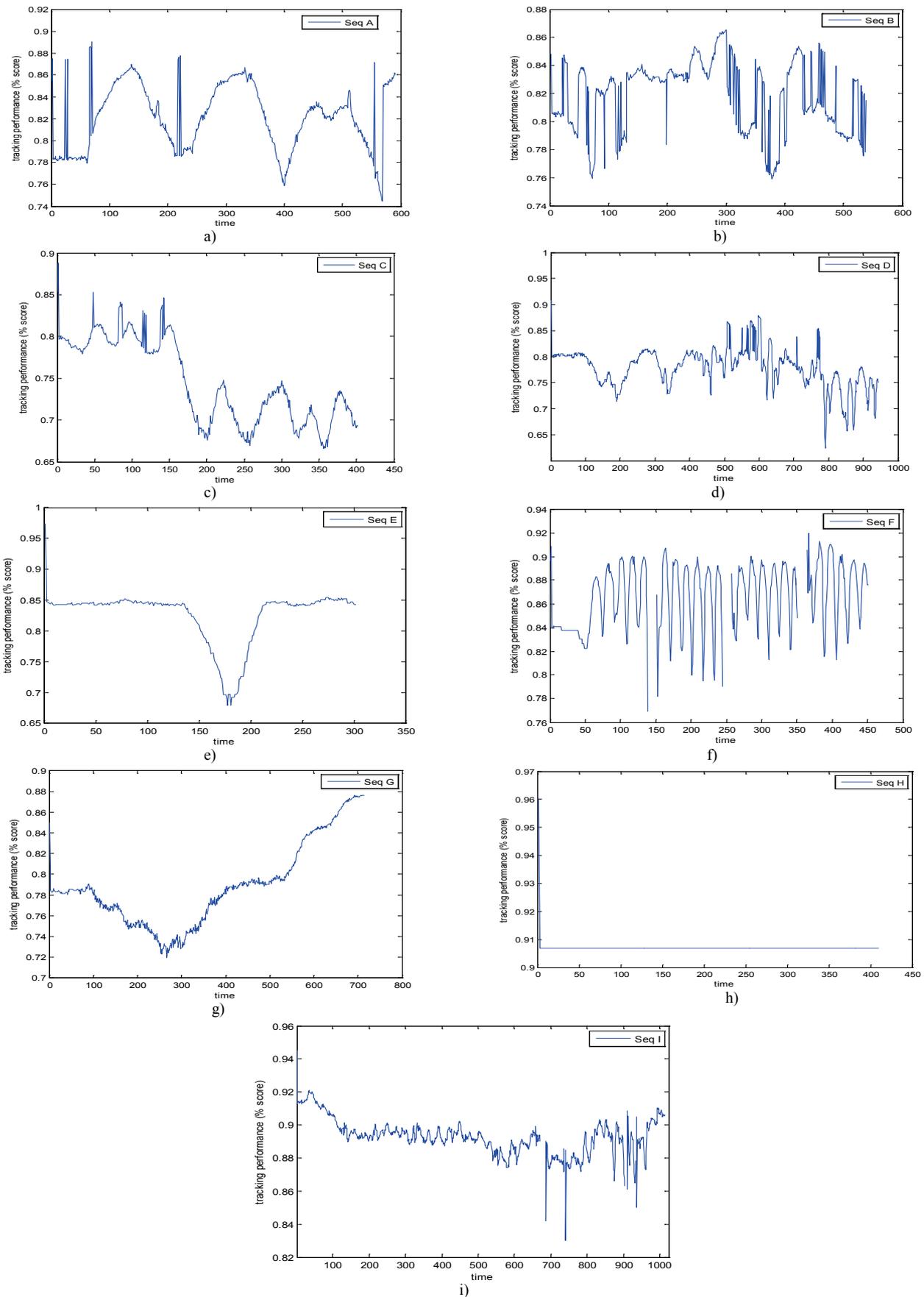


Figure 3 The tracking performance results in the form of overlapping percent of estimated and actual bounding boxes around the objects for sequences A to I from the data set.

To compare the results of tracking with other approaches, a performance score is used which calculates the percent of overlapping between tracked (estimated)

and actual bounding box of the object. It should be noted that, for frames with full occlusion effects, this score should be calculated as non-existing. For the simulations,

the following settings have been used. For all sequences, 10 clusters have been found for 5000 sample points of initial frame. To find optimal clustering, PSO is used with 30 particles and 20 iterations. GA is not used in final results, because PSO was faster in this problem. There was no need to introduce coefficients of LLNF rules more than a_0 , based on the satisfactory results. When tracking the object during the video stream, 1000 sample points from the screen were used for each frame, and 8 linear predictors have been utilized accompanied with co-evolution of initial random parameters as described.

For several videos of the dataset, the tracking results in the form of performance scores versus time are shown in Fig. 3. Various behaviors on plots are obtained based on the various situations of corresponding videos. For example, in video sequence F, there are several full occlusions, and on the plot it is seen as non-existing values for the performance score. As for sequence H, there is no actual movement of object or camera, then the predictor model converges to a stable position and then performance trend with time is constant.

To compare the results, in Tab. 1 the average performance for each sequence is presented for several methods compared to the results of the proposed method. The results show that the performance of the proposed method is better than other approaches, for most of the sequences. Only for 3 of 9 sequences, the results of the proposed method are slightly lower than other methods. Based on the average of all results, the higher accuracy of proposed method is concluded.

Table 1 Comparison of tracking performances for various methods [26]

Sequence	Average performance score of methods (%)					
	Histogram	Multi-comp	N. ad. hcs	Adapt. hcs	Adapt. P. part. hcs	Proposed method
A	70,73	63,24	38,35	65,06	59,35	82,18
B	67,02	50,73	6,02	79,01	77,38	81,85
C	47,58	63,71	89,33	90,66	91,33	74,56
D	63,35	76,39	62,78	71,12	75,21	77,87
E	78,21	77,42	83,12	84,49	86,32	82,61
F	44,43	40,02	63,99	60,82	68,32	86,66
G	46,27	49,62	34,34	77,30	71,16	78,87
H	62,19	86,50	95,79	94,41	94,47	90,71
I	68,94	47,63	48,97	75,02	56,33	89,30
Average	60,97	61,70	58,08	77,54	75,54	82,73

8 Conclusion

The object tracking method proposed in this paper combines the ideas of clustering, evolutionary computation, and neuro-fuzzy models. The evolutionary optimization is used to obtain a good clustering of sample points on the initial frame accompanied with the determination of the object. The clustering results set the parameters of locally linear neuro-fuzzy model in a simple way. The resulted identifier is used to segment the object based on random sample points during the video stream. Another swarm of linear models co-evolve based

on their performances for estimating the behavior of object and predict the next location. The performance analysis and comparison to previous methods on same data set showed that the accuracy of the proposed method is higher than that of previous methods. For the extension of this work, other neuro-fuzzy models, clustering, and evolutionary computation of object tracking can be used. Future studies would extend the efficient optimization models or the problem of object tracking in videos.

9 References

- [1] Hu, W.; Tan, T.; Wang, L.; Maybank, S. A survey on visual surveillance of object motion and behaviors. // *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on. 34, 3(2004), pp. 334-352. DOI: 10.1109/TSMCC.2004.829274
- [2] Koller, D.; Daniilidis, K.; Nagel, H.-H. Model-based object tracking in monocular image sequences of road traffic scenes, // *International Journal of Computer. 10, 3(1993)*, pp. 257-281.
- [3] Wu, Y.; Lim, J.; Yang, M.-H. Online object tracking: A benchmark. // in *Computer vision and pattern recognition (CVPR), 2013 IEEE Conference on*. 2013, pp. 2411-2418.
- [4] Vadakkepat, P.; Lim, P.; De Silva, L. C.; Jing, L.; Ling, L. L. Multimodal approach to human-face detection and tracking. // *Industrial Electronics, IEEE Transactions on*. 55, 3(2008), pp. 1385-1393. DOI: 10.1109/TIE.2007.903993
- [5] Liu, M.; Yao, H.; Gao, W. Real-Time human face tracking in color images. // *Chinese Journal of Computers-Chinese Edition*. 21, (1998), pp. 527-532.
- [6] Allen, P. K.; Timcenko, A.; Yoshimi, B.; Michelman, P. Automated tracking and grasping of a moving object with a robotic hand-eye system. // *Robotics and Automation, IEEE Transactions on*. 9, 2(1993), pp. 152-165. DOI: 10.1109/70.238279
- [7] Papanikolopoulos, N. P.; Khosla, P. K. Adaptive robotic visual tracking: Theory and experiments. // *Automatic Control, IEEE Transactions on*. 38, 3(1993), pp. 429-445. DOI: 10.1109/9.210141
- [8] Rai, N.; Rai, B.; Rai, P. Computer vision approach for controlling educational robotic arm based on object properties. // in *Emerging Technology Trends in Electronics, Communication and Networking (ET2ECN), 2014 2nd International Conference on*. 2014, pp. 1-9.
- [9] Shantaiya, S.; Verma, K.; Mehta, K. Multiple Object Tracking using Kalman Filter and Optical Flow. // *European Journal of Advances in Engineering and Technology*. 2, 2(2015), pp. 34-39.
- [10] Li, X.; Wang, K.; Wang, W.; Li, Y. A multiple object tracking method using Kalman filter. // in *Information and Automation (ICIA), 2010 IEEE International Conference on*. 2010, pp. 1862-1866.
- [11] Li, X.; Zhang, T.; Shen, X.; Sun, J. Object tracking using an adaptive Kalman filter combined with mean shift. // *Optical Engineering*. 49, 2(2010), pp. 020503-020503. DOI: 10.1117/1.3327281
- [12] Ristic, B.; Arulampalam, S.; Gordon, N. Beyond the Kalman filter. // *IEEE Aerospace and Electronic Systems Magazine*. 19, 7(2004), pp. 37-38. DOI: 10.1109/MAES.2004.1346848
- [13] Lippiello, V.; Siciliano, B.; Villani, L. Adaptive extended Kalman filtering for visual motion estimation of 3D objects. // *Control Engineering Practice*. 15, 1(2007), pp. 123-134. DOI: 10.1016/j.conengprac.2006.05.006
- [14] Liu, C.; Shui, P.; Wei, G.; Li, S. Modified unscented Kalman filter using modified filter gain and variance scale factor for highly manoeuvring target tracking. // *Journal of*

- Systems Engineering and Electronics. 25, 3(2014), pp. 380-385. DOI: 10.1109/JSEE.2014.00043
- [15] Chen, S. Y. Kalman filter for robot vision: a survey. // *Industrial Electronics, IEEE Transactions on*. 59, 11(2012), pp. 4409-4420. DOI: 10.1109/TIE.2011.2162714
- [16] Nummiaro, K.; Koller-Meier, E.; Van Gool, L. An adaptive color-based particle filter. // *Image and Vision Computing*. 21, 1(2003), pp. 99-110. DOI: 10.1016/S0262-8856(02)00129-4
- [17] Gonczarek, A.; Tomczak, J. M. Manifold regularized particle filter for articulated human motion tracking. // In *Advances in Systems Science / Springer International Publishing*. 2014, pp. 283-293
- [18] Lucena, M.; Fuertes, J. M.; de la Blanca, N. P. Optical flow-based observation models for particle filter tracking. // *Pattern Analysis and Applications*. 18, 1(2015), pp.135-143. DOI: 10.1007/s10044-014-0409-3
- [19] KaewTraKulPong, P.; Bowden, R. An improved adaptive background mixture model for real-time tracking with shadow detection. // In *Video-based surveillance systems / Springer US*. 2002, pp. 135-144
- [20] Rittscher, J.; Kato, J.; Joga, S.; Blake, A. A probabilistic background model for tracking. // In *Computer Vision—ECCV 2000 / Springer Berlin Heidelberg*. 2000, pp. 336-350
- [21] Gu, B.I. Mu, A.; Tekalp, A. M. Temporal video segmentation using unsupervised clustering and semantic object tracking. // *Journal of Electronic Imaging*. 7, 3(1998), pp. 592-604. DOI: 10.1117/1.482613
- [22] Heisele, B.; Kressel, U.; Ritter, W. Tracking non-rigid, moving objects based on color cluster flow. // In *Computer Vision and Pattern Recognition, Proceedings 1997 IEEE Computer Society Conference on*. 1997, pp. 257-260
- [23] Weng, S. K.; Kuo, C. M.; Tu, S. K. Video object tracking using adaptive Kalman filter. // *Journal of Visual Communication and Image Representation*. 17, 6(2006), pp. 1190-1208. DOI: 10.1016/j.jvcir.2006.03.004
- [24] Yang, F.; Paindavoine, M. Implementation of an RBF neural network on embedded systems: real-time face tracking and identity verification. // *IEEE Transactions on Neural Networks*. 14, 5(2003), pp. 1162-1175. DOI: 10.1109/TNN.2003.816035
- [25] Angelov, P.; Sadeghi-Tehran, P.; Ramezani, R. An approach to automatic real-time novelty detection, object identification, and tracking in video streams based on recursive density estimation and evolving Takagi–Sugeno fuzzy systems. // *International Journal of Intelligent Systems*. 26, 3(2011), pp. 189-205. DOI: 10.1002/int.20462
- [26] Klein, D. A.; Schulz, D.; Frintrop, S.; Cremers, A. B. Adaptive real-time video-tracking for arbitrary objects. // In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2010, pp. 772-777. DOI: 10.1109/IROS.2010.5650583
- [27] Leng, G.; McGinnity, T. M.; Prasad, G. Design for self-organizing fuzzy neural networks based on genetic algorithms. // *IEEE Transactions on Fuzzy Systems*. 14, 6 (2006), pp. 755-766. DOI: 10.1109/TFUZZ.2006.877361
- [28] Cheng, H. D., Jiang, X. H.; Sun, Y.; Wang, J. Color image segmentation: advances and prospects. // *Pattern recognition*. 34, 12(2001), pp. 2259-2281. DOI: 10.1016/S0031-3203(00)00149-7
- [29] Simunovic, G.; Svalina, I.; Simunovic, K.; Saric, T.; Havrlisan, S.; Vukelic, D. Surface roughness assessing based on digital image features. // *Advances in Production Engineering & Management*. 11(2016), 2, pp. 93-104. DOI: 10.14743/apem2016.2.212
- [30] Tang, M.; Gong, D.; Liu, S.; Zhang, H. Applying multi-phase particle swarm optimization to solve bulk cargo port scheduling Problem. // *Advances in Production Engineering & Management*. 11, 4(2016), pp. 299-310. DOI: 10.14743/apem2016.4.228
- [31] Chandrasekaran, M.; Devarasiddappa, D. Artificial neural network modeling for surface roughness prediction in cylindrical grinding of Al-SiCp metal matrix composites and ANOVA analysis. // *Advances in Production Engineering & Management*. 9, 2(2014), pp. 59-70, DOI: 10.14743/apem2014.2.176
- [32] Saric T.; Simunovic G.; Simunovic K.; Svalina I. Estimation of Machining Time for CNC Manufacturing Using Neural Computing. // *International Journal of Simulation Modelling*. 15(2016), 4, pp. 663-675. DOI: 10.2507/IJSIMM15(4)7.359
- [33] Goldberg, D. E.; Holland, J. H. Genetic algorithms and machine learning. // *Machine learning*. 3, 2(1988), pp. 95-99. DOI: 10.1023/A:1022602019183
- [34] Kennedy, J. Particle swarm optimization. // in *Encyclopedia of Machine Learning*, Springer, 2010.
- [35] Nelles, O. Nonlinear system identification: from classical approaches to neural networks and fuzzy models. Springer Science & Business Media, 2001. DOI: 10.1007/978-3-662-04323-3

Author's address**Fatemeh Saadian,**

Sama Technical and Vocational Training College,
Islamic Azad University, Nour Branch, Nour, Iran
E-mail: fatemeh.saadian@chmail.ir