Chaotic target representation for robust object tracking

Marjan Abdechiri, Karim Faez, Hamidreza Amindavar, Eleonora Bilotta

www.elsevier.com/locate/image

PII: S0923-5965(17)30020-6
DOI: http://dx.doi.org/10.1016/j.image.2017.02.004
Reference: IMAGE15175

To appear in: *Signal Processing : Image Communication*

Received date: 10 August 2016
Revised date: 11 February 2017
Accepted date: 12 February 2017

Cite this article as: Marjan Abdechiri, Karim Faez, Hamidreza Amindavar and Eleonora Bilotta, Chaotic target representation for robust object tracking, *Signal Processing : Image Communication* http://dx.doi.org/10.1016/j.image.2017.02.004

Chaotic target representation for robust object tracking

Marjan Abdechiri[1], Karim Faez[1*], Hamidreza Amindavar[1], Eleonora Bilotta[2]
[1] Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran
[2] Physics Department, University of Calabria, Via Pietro Bucci, 87036 Rende, Italy
[*]*Corresponding authors: kfaez@aut.ac.ir*

Abstract

In this paper, a new object representation method is introduced as an appearance model based on chaos theory. For robust object tracking, the theory is used to extract a deterministic model from irregular patterns of pixel amplitudes in a target region. The object tracking algorithm that accompanies the proposed method involves two steps. First, fractal theory is applied to a compressive sensing method intended to embed an image into a two-dimensional state space during tracking by detection. After an object representation is extracted from an instance, the fractal dimension of the state space is assigned to the importance weight of the instance for efficient online multiple-instance learning. Second, a chaotic map approach is adopted to update the appearance model. Such updating is a crucial step in selecting discriminative and robust features. To highlight the advantages of the algorithm put forward in this work, its accuracy is validated on a large dataset. Results show that the proposed algorithm is more efficient than state-of-the-art tracking algorithms, with the former outperforming the latter under rotation, illumination, and scale changes.

Keywords: Chaos theory, fractal theory, online multiple-instance learning, object tracking.

1. Introduction

Object tracking, which is a highly practical approach to motion analysis, traffic monitoring, and video analysis, is a topic of considerable interest in computer vision research. The field of visual object tracking has recently attained substantial achievements, such as the development of the context-aware exclusive sparse tracker [1], adaptive multi-task feature learning [2], and sparse appearance model [3]. Nevertheless, designing a robust tracking system remains a challenging task because of appearance-related variations, including fast motion, occlusion, illumination, scale, and rotation [4]. Under these challenging conditions, a key component in object tracking is visual representation, which is categorized into two schemes: local and global representation [5]. Global representation reflects an object's global statistical characteristics, including raw pixels, optical flows, histograms, covariances, wavelets, and active contours [6]. This scheme is a simple and efficient method for real-time object tracking, but it is sensitive to global appearance changes, such as variations in illumination, deformation shape, rotation, and partial occlusion [7]. Local representation, on the other hand, extracts local structural appearance characteristics as interest points that are used to encode appearance information. These local characteristics are classified on the basis of local templates, segmentation, scale-invariant feature transformation, maximally stable external regions, speeded up robust features, corners, and feature pool-based [8]. Despite the robustness of local representation against global appearance changes, however, representations suffer from noise disturbance and background distraction. The use of local feature extraction methods also increases computational costs [9]. To reduce complexity and noise disturbance, compressive sensing methods [10] are used to obtain a low-dimensional representation on the basis of correlation reduction in object tracking. The drawback of these methods is that they do not consider the dynamic information contained in data for compression, thereby causing the loss of important samples for signal recovery.

A target region is a complex interaction of pixel amplitudes and variations—an interaction that presents difficulties in detecting and recognizing a target with irregular patterns under different situations. Extracting a deterministic model of irregular patterns on the basis of chaos theory aids efficient compressive sensing for object tracking. Chaos theory, which is aimed at an understanding of nonlinear dynamics, has been successfully used to model variations in time series [11]. It has also been actively adopted in research on signal and image processing [12] and mathematics [13]. Chaotic characteristics may facilitate the high-quality extraction of dynamic time series information, which can be applied to image processing and object tracking. In this study, compressive sensing and chaos theory are used to develop a feature extraction method that extracts a low-dimensional feature space on the basis of dynamic information. To address the weaknesses of local and global representation in object tracking, fractal theory is used to extract the global dynamics of irregular patterns in state space on the basis of local information regarding pixel correlation.

The resultant representation balances local and global features for object tracking. A new online weighted multiple-instance learning (MIL) method is also established to maximize bag likelihood function on the basis of a chaotic approximation model.

To evaluate the performance of the chaotic representation, the proposed method is applied to 10 video sequences, for which objects are detected using an MIL methods, and 20 video sequences (characterized by different challenging conditions), for which objects are detected using the latest tracking methods. The performance of the aforementioned methods on a large dataset is then compared. Experimental results demonstrate that the representation derived with the proposed approach is robust to partial occlusion, illumination changes, scale changes, and rotation variations. The rest of the paper is organized as follows. Section 2 introduces the chaos theory in nonlinear dynamics. Section 3 represents the basic concept of online MIL algorithms. In section 4, the architecture of chaotic representation and details of MCIL method are given. Experimental results are presented in section 5 with various examples and quantification analyses. Finally, section 6 concludes this paper.

2. Chaos theory in nonlinear dynamics

Chaos theory is used to process nonlinear dynamic systems, thereby improving our understanding of such systems. Although a chaotic time series is random-like with complex dynamics, it has a deterministic attractor in state space [14]. The evolution of a dynamic system can be described by using a trajectory in state space; this trajectory is determined on the basis of chaos theory [15]. State space reconstruction can be used for prediction, recognition, and modeling in nonlinear systems, and state space features are used to learn dynamic time series models, such as speech recognition [16] and disease detection models [17]. In this work, Takens' embedding theorem is used to generate a map from a one-dimensional space to an m-dimensional space [18]. The state space reconstruction expands a vector $(x_0, x_1, \dots, x_{n-1})$ into multi-dimensional space $x_n(m,\tau) = [x_n, x_{n+\tau}, \dots x_{n+(m-1)\tau}] \in \mathbb{R}^m$, where τ represents the time delay, and m denotes the embedding dimension [19]. The dynamics of the multi-dimensional space can be represented by m-dimensional map $x_{n+1} = F(x_n, n)$ in state space [20]. Constructing a state space model of time series data necessitates the selection of an appropriate dimension m and time lag τ. The mutual information is used to compute the minimum lag τ between x_i and $x_{i+\tau}$ and thereby guarantee topological equivalence with original and reconstructed attractors [21]. Mutual information is computed to measure the linear dependence between values [22]. Embedding dimension m is estimated using the false nearest neighbor algorithm because of dimension reduction [23]. The embedding dimension of a time series is computed on the basis of the average number of false nearest neighbors [24].

Chaotic sparse representation is implemented to extract the chaotic dynamics of a nonlinear time series. Compressive sensing is applied to reduce the number of coherent samples, and fractal theory is used to directly extract the chaotic dynamics of streaming signals [25]. This study proposes a new compressive sensing method that extracts the state space of an image without the need to compute mutual information and use the false nearest neighbor algorithm. Given these features, the method may reduce complexity. The fractal function is described by

using power exponent distribution as $N(r) = cr^{-D}$ where $N(r)$ is observation data in the time r [26]. The fractal dimension D is

$$D = \frac{ln\left(\frac{N_A}{N_B}\right)}{ln\left(\frac{r_B}{r_A}\right)} \quad (1)$$

$$c = N_A r_A^D = N_B^D \quad (2)$$

where c is constant value.

3. Online MIL trackers

Object tracking algorithms can generally be categorized into generative [27] and discriminative algorithms [28], which estimate object location by using classifiers as trackers that are based on a detection method. Generative algorithms perform well under slight changes because of the rich features of objects, whereas discriminative algorithms are robust under complicated situations because they use negative instances to avoid the drifting problem. Online MIL is a discriminative model that is based on positive and negative instances [28]. MIL can be extended in object tracking task, multi-label learning [29], and object detection. The tracker uses a bag of instances instead of an instance for visual tracking. As shown in Fig. 1, a bag is positive when one or more of the instances in it are positive [30]. For each new frame, we crop out a set of patches $X^s = \{x | \|l(x) - l_{t-1}^*\| < s\}$ with the radius s of the current tracker location to compute $p(y|x)$ for $x \in X^s$. The location of image patch x is $l(x)$. At each time t, the tracker maintains object location l_t^*.

Similarly, we use greedy search to update tracker location $l_t^* = l\big(argmax\, p(y|x)\big)$. The motion model determines the location of the target in frame *t* on the basis of the tracker location at time *(t-1)* and radius *s* [30]:

$$p(l_t^*|l_{t-1}^*) \propto \begin{cases} 1 & if\ \|l_t^* - l_{t-1}^*\| < s \\ 0 & otherwise \end{cases} \quad (3)$$

To update our appearance model, a set of image patches is randomly selected using $X^r = \{x | \|l(x) - l_t^*\| < r\}$ and $X^{r,\beta} = \{x |\, r < \|l(x) - l_t^*\| < \beta\}$ for positive and negative instances, respectively. In these equations, $l(x)$ denotes the location of image patches x, and r and β represent integer radii. We consider all instances positive with $r > 1$ (Fig. 1). In the training phase, the inputs are *N* instances as the training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, and the label assigned to instance $x_i \in \boldsymbol{X}$ is $y_i = \{-1,1\}$.

The MIL developed in this research uses *N* training bags $D = \{X_i, Y_i\}_{i=1}^N$ and bag label y_i, where $X_i = \{x_{i1}, x_{i2}, \ldots, x_{ij}\}$ [28]. The bag labels are defined as $y_i = \max_j(y_{ij})$. Consequently, the classifier is updated by positive and negative samples $H(x) = \sum_{i=1}^K \alpha_k h_k(x)$, where $h_k(x)$ is a weak classifier [28].

For predictive purposes, a binary function $\sigma(H(x))$ is used, where $\sigma(x)$ is a sigmoid function $\sigma(x) = 1/(1 + e^{-x})$. At every time step *t*, all *M* candidate classifiers are updated by using

$$h_k(x) = \log\left[\frac{p_t(y=1|f_k(x))}{p_t(y=0|f_k(x))}\right] \quad (4)$$

where $p_t(f_k(x)|y = 1) \sim N(\mu_1, \sigma_1)$ with a Haar-like feature vector f_k. The maximum log likelihood of bags $h_k = \text{argmax}_{h \in \{h_1, \ldots, h_M\}} \mathcal{L}(H_{k-1} + \alpha h)$ is then computed to select *k* weak classifiers, where *M*>>*k*, and $\mathcal{L}(.)$ is the log likelihood of bags.

$$\mathcal{L}(H) = \sum_i y_i \log(p_i) + (1 - y_i) \log(1 - p_i). \quad (5)$$

Conditional distributions are computed using a Gaussian function $p(f_k(x_{ij})|y_i = 1) \sim N(\mu_1, \sigma_1)$ and $p(f_k(x_{ij})|y_i = 0) \sim N(\mu_0, \sigma_0)$. The updating rules are $\mu = \eta\mu + (1-\eta)\frac{1}{N}\sum_{j|y_i=1} f_k(x_{ij})$ and $\sigma = \eta\sigma + (1-\eta)\sqrt{\frac{1}{N}\sum_{j|y_i=1}(f_k(x_{ij}) - \mu)^2}$, where η is the learning rate, and N represents the number of positive instances [28]. Bag probability is a noisy model $p_i = P(y_i = 1|X_i) = 1 - \prod_j(1 - p_{ij})$, and instance probability is $p_{ij} = P(y_{ij} = 1|x_{ij}) = \sigma(H(x))$. The MIL method selects the discriminative weak classifier by maximizing the log likelihood of bags as follows:

$$h_k = arg \max_{h=\{h_1,h_2,\ldots,h_M\}} L(H_{k-1} + h). \quad (6)$$

The model above requires the computation of instance probability and bag probabilities during online classification. Although online MIL is preferable for real-time tracking, it presents two intractable problems: appearance representation and the online updating model. In appearance representation, Haar-like features are sensitive to highly complex illumination changes during online updating. To obtain effective representation, the histogram of oriented gradients (HOG) [31] and Distribution Fields (DFs) [32] are introduced to describe a robust target representation for object tracking. DF layers represent the spatial information of an image through the probability distribution of each pixel; the use of this distribution, in turn, generates a smaller pool of features. Although HOG features can overcome the influence of illumination, however, the features are sensitive to the spatial location of an object, thus causing drifting in long video sequences [33]. HOG- and DF-based methods are also sensitive to partial occlusions and illumination changes.

Local representations cannot describe a target with robust features under complex appearance changes in video sequences. A robust tracking system requires a high-dimensional space of local features to select important target features from a pool of features for tracking. In the online updating model, the online MIL tracker may select less important positive instances and thus cause tracking failure. Bag likelihood function is computed directly with instance and bag probabilities, but this computation increases the complexity of online MIL. To address this problem, the online weighted MIL tracker takes advantages of instance weights in selecting fewer features in a learning procedure [34]. On the basis of the weights, the optimized bag likelihood function is computed using first-order Taylor approximation, which then reduces the complexity of online MIL. Some drawbacks of online MIL and other MIL algorithms are listed in Table 1. The visual representation and likelihood function of the proposed method are introduced using a chaotic system (Table 1).

4. Proposed object representation method

The object representation method introduced in this research capitalizes on chaos theory to enhance our understanding of nonlinear systems for object tracking. A new chaotic extraction scheme is applied to reduce the dimension of feature space for tracking with invariant features. The irregularities and patterns of a target are represented by using chaotic representation in state space. To separate the target from its background, online weighted MIL is used to update the appearance model on the basis of the fractal dimension of instance representation and the instance distance from the position predicted in the previous frame. The framework of the proposed object representation method is illustrated in Fig. 2.

4.1. Chaotic representation

State space reconstruction can be used for feature extraction in image processing. This study puts forward a new method for reducing the complexity of reconstruction. The method can embed noisy data in a low-dimensional trajectory, which is a deterministic model of variations that can be applied to feature extraction. To represent the variations and patterns in a leaf image, the image can be considered a time series (Fig. 3). In the state space reconstruction for the image, its embedding dimension is $m = 5$, which is estimated using the false nearest neighbor algorithm. The embedding dimension of the bounding box in the first frame can be regarded as the optimal

dimension over frames, but this consideration diminishes the performance of the trackers. Considering an image as a time series requires computing the optimal embedding dimension for each instance in the MIL process to reduce the speed of object tracking. To solve these deficiencies, we present a method for directly extracting the low-dimensional feature space of image variations. This extraction method does not require the computation of optimal delay and dimension.

An image or instance can be converted to one-dimension time series as $x_1, x_2, \dots, x_n$. A compressive sensing method that considers fractal dimensions is then used to directly find an embedding space in the image. In the algorithm, the columns of an image are considered time-varying signals. Subsequently, fractal dimensions are computed for successive samples by using Eq. (1). The fractal dimensions aid the selection of important samples with chaotic characteristics and a low-dimensional state space.

In the theory of compressive sensing, an unknown sparse signal $x \in R^N$ is k-sparse, given that it has k nonzero components ($N \gg k$). The linear measurement $y = \Phi x$ is applied to obtain a limited number of observations, where $y \in R^M$ and $\Phi \in R^{M \times N}$ and $M \ll N$ [35]. A convex optimization algorithm using l_1-norm is applied in order to minimize the function

$$min\|x\|_1 \;\; such\ that\ y = \Phi x. \tag{7}$$

In streaming signal, the active interval Π is shifted by removing few oldest y_t and adding a few new ones [36]. In the active interval, $\bar{x}$ is a small number of x_t. The compact form is $\bar{y} = \bar{\Phi}\bar{x} + \bar{e}$, where $\bar{y}$ denotes the measurements y_t. The signal is transformed by using $\bar{x} = \bar{\Psi}\bar{\alpha}$. In the system model, the system is separated $\bar{x}$ into two matrices as follows:

$$\bar{x} = \bar{\Psi}\bar{\alpha} = \left[\breve{\Psi}\ \tilde{\Psi}\right]\begin{bmatrix}\breve{\alpha}\\ \tilde{\alpha}\end{bmatrix} = \breve{\Psi}\breve{\alpha} + \tilde{\Psi}\tilde{\alpha}, \tag{8}$$

where $\breve{\Psi}$ is the first columns in $\bar{\Psi}$. The system equation modifies with removing α_{l-1} as $\tilde{y} = \bar{\Phi}\tilde{\Psi}\tilde{\alpha} + \tilde{e}$. In recovery process of streaming signal, the l_1-norm minimization problem estimates coefficients $\tilde{\alpha}$ as

$$minimize_{\alpha}\ \|W\alpha\|_1 + \frac{1}{2}\left\|\bar{\Phi}\tilde{\Psi}\alpha - \tilde{y}\right\|_2^2. \tag{9}$$

Fractal theory is applied to the algorithm to extract a low-dimensional trajectory in the state space of the image. In the proposed extraction method, the minimization problem (i.e., Eq. (9)) is modified as follows:

$$minimize_{\ \alpha=\{\alpha_1,\alpha_2,\dots,\alpha_m\}} \sum_{k=1}^{m}\|W_k\alpha_k\|_1 + \frac{1}{2}\|\Phi_k\Psi_k\alpha_k - y_k\|_2^2 + \frac{\lambda_k}{2}\|\hat{z} - \Psi_k\alpha_k\|_2^2 \tag{10}$$

where $\lambda_k > 0$ are regularization parameters. The third term in the equation is used to recover a signal on the basis of the minimum error of fractal prediction. To predict succeeding samples within compressive sensing, grey theory is applied to sparse samples. The theory is also used to determine optimal samples with a two-dimensional state space at a minimum prediction error. The fractal dimensions of samples $D = \{d_1, d_2, \dots, d_{m-1}\}$ are computed using Eq. (1). Grey theory is computed for D by using cumulative-sum series $D1_i = \sum_{j=1}^{i} D_j$ and $D2_i = \sum_{j=1}^{i} D1_j$. The best sparse samples are found within the data transformed with grey theory, after which the inverse accumulating generation operator is used to predict state as $\widehat{x'}^{i}(m) = \widehat{x'}^{i+1}(m) - \widehat{x'}^{i+1}(m-1)$ at a minimum prediction error e. l_1-homotopy optimization is used [36] to dynamically and iteratively update the solutions of Eq. (10) for the purpose of estimating the $\tilde{\alpha}$ of the optimization problem.

$$minimize_{\ \alpha}\ \ \|W\alpha\|_1 + \frac{1}{2}\left\|\bar{\Phi}\tilde{\Psi}\alpha - \tilde{y}\right\|_2^2 + \frac{\lambda}{2}\left\|\hat{z} - \tilde{\Psi}\alpha\right\|_2^2. \tag{11}$$

The third term aides to select chaotic samples within the compressive sensing. The pseudo code of the proposed method is explained in Algorithm 1.

4.2. Online weighted multiple-chaotic instance learning

In the training phase, the feature vector of each sample is extracted using the proposed object representation method, in which two bags $\{X^+, X^-\}$ are considered for training. The positive bag comprises N positive instances x_{1j} and L negative instances x_{0j}. Classifier $H_k(.)$ is described as

$$H_k(x) = \log \frac{p(f(x)|y=1)\, p(y=1)}{p(f(x)|y=0)\, p(y=0)} = \sum_k h_k(x) \tag{12}$$

where $y = 1$ is the label of a positive instance, and $y = 0$ is the label of a negative instance. In online updating, a fractal dimension is computed for each instance as a weight. The weight of instances is applied as

$$w = \frac{1}{2\pi} e^{-\frac{\left(D(x)-D(x^*)\right)^2}{2}} e^{-\frac{\|l(x)-l(x^*)\|}{2}} \tag{13}$$

where the optimal tracked position is $l(x^*)$ in the current frame, $l(.) \in R^2$ denotes the location function, and D represents the fractal dimension. The weights ease the selection of fewer features for updating the appearance model. The instances near the tracking location with the same fractal dimension contribute to bag probability. Hence, the instances are contingent on the weights in a bag's label. The bag probability is the weighted model, as indicated below:

$$p(y = 1|X^+) = \sum_{j=0}^{N-1} w_j\, p(y_1 = 1|x_{1j}) \tag{14}$$

The bag's log-likelihood function is computed by deploying the new weighted model.

$$l(H) = \sum_{s=0}^{1} y_s \log(p(y_i = 1|X^+) + (1 - y_s) \log\big(p(y_i = 0|X^-)\big). \tag{15}$$

We use a dynamic system that creates a chaotic map [37, 38] by using the steepest descent method to optimize objective function Eq. (6). We use the steepest descent method to convert Eq. (6) into a chaotic map thus:

$$\begin{cases} y_{k+1}(t+1) = y_k(t) - \gamma\, (-\nabla_{x_k} l(H_k + h)) \\ x_k(t+1) = S\big(y_k(t+1)\big) \end{cases}. \tag{16}$$

The generated chaotic sequence of the map converges to a local minimum. The solution is based on the initial condition. γ is sufficiently small and can generate a sequence that is chaotic in the sense of Li-Yorke chaos [37]. The search model can determine solutions in the descending direction. An appropriate transformation function $S: \Re^n \longrightarrow X$ is $S(y_k) = x_k$ or $y_k = S^{-1}(x_k)$. The gradient of loss function is

$$\nabla l(H)\big(x_{ij}\big) = \frac{\partial l\left(H + \delta 1_{x_{ij}}\right)}{\partial \delta}|_{\delta=0}. \tag{17}$$

$$= \left(y_i \log\left(\sum_{m=0}^{N-1} w_{m0}\sigma\left(H(x_{1m}) + \delta 1_{x_{ij}}\right)\right) + (1 - y_i) \log\left(\sum_{m=N}^{N+L-1}\left(1 - \sigma\left(H(x_{0m}) + \delta 1_{x_{ij}}\right)\right)\right)\right) \tag{18}$$

$$= y_i \frac{w_{j0}\sigma(H(x_{ij}))(1-\sigma\left(H(x_{ij})\right))}{\sum_{m=0}^{N-1} w_{m0}\sigma(H(x_{im}))} - (1 - y_i) \frac{\sigma(H(x_{ij}))(1-\sigma\left(H(x_{ij})\right))}{\sum_{m=N}^{N+L-1}\left(1-\sigma\left(H(x_{im})\right)\right)} \tag{19}$$

The discriminative classifiers are computed using the chaotic map:

$$\begin{cases} y_{k+1}(t+1) = y_k(t) + \gamma\, \nabla l(H_k) \\ x_k(t+1) = S\big(y_k(t+1)\big) \end{cases}. \tag{20}$$

The model above is the chaotic gradient descent and comes with the advantage of chaotic transformation in identifying optimal points. Our optimization method (i.e., Eq. (20)) is not indispensable to directly computing bag probabilities and enables fast optimization that maximizes log-likelihood function. The feature selection in Eq. (20) is more efficient than that in Eq. (6) in the MIL tracker.

4.3. Chaotic representation under certain challenges

The robustness of the proposed object representation method under three main challenges is illustrated in Figs. 4 to 6. The target patch of the frame is cropped to represent the chaotic model of the target. In Fig. 4, the representations of the target are shown under translation in the instances for the David video sequences. The proposed method generates a sparse representation of the true target location. On the basis of these observations, we note that the precision localization of the tracked target can be determined by penalizing the chaotic characteristic of the model. A penalty is used as the weight of the instances in our tracking algorithm. The histograms of instances demonstrate the probability distribution of three different instances in state space. The key contribution of this study is its use of a low-dimensional dynamic model of instances in describing a target in video sequences. Fig. 5 indicates that our appearance model is of a scale-invariant nature given that the state space generates a scale-free model for targets in a histogram. Fig. 6 shows that the chaotic representation is rotation invariant. Eliminating the noise components and the reduction in dimensionality are the main advantages of the object representation method put forward in this work. It uses fractal theory to concentrate selected pixels into a low-dimensional attractor in state space. The obtained chaotic attractor of a target region maintains the ergodicity of chaotic attractors on the basis of self-similarity [39]. Consequently, the proposed method is invariant under rotation and translation variations [40].

5. Experiments and numerical results

To verify the performance of the proposed method, we conduct three experiments that involve online MIL trackers, several state-of-the-art algorithms, and a large tracking benchmark (TB) [41-42]. The parameter settings of the methods are presented in this section.

5.1. Parameter settings and evaluation metrics

The proposed object tracking algorithm is compared with MIL-based trackers, namely, MIL [28], weighted MIL (WMIL) [34], DFMIL [32], online MIL underlain by an improved appearance model (I-MIL) [48], and online informative feature selection (OIFS) [49]. It is also evaluated against some detection-based and stochastic algorithms. The algorithms are implemented on a computer with a dual-core CPU at 2.2 GHz frequency and 2 GB RAM running under Windows 7. The parameter settings are presented in Table 2. The results of some of the algorithms agree with those reported in previous research.

As shown in Table 2, the MIL algorithm sets the number of discriminative classifiers at K=50 and the number of candidate weak classifiers at M=250, whereas our algorithm establishes these values at K=10 and M=130, respectively. In our algorithm, the radius of search s is set to 30 pixels to identify object location. Radius $s = 4$ is used to crop positive instances, and inner and outer radii $r = 8$ and $\beta = 30$ are used to randomly select 50 negative instances. The learning rate is set to $\lambda = 0.85$. Our algorithm independently runs five times on each dataset, and we then compute the average of the five runs.

Two metrics are employed to quantitatively evaluate the performance of the algorithms. The center location error and the success rate are used as basis in determining the precision and tracking robustness of the trackers. The center location error (CLE) can be used to measure the Euclidean distance between the central tracked position and the position of an object in ground truth [50]. The center location error (CLE) is defined as

$$error = dist\big(center(t), gt(t)\big), \tag{21}$$

where t is the number of frames, $dist$ denotes the Euclidean distance, $center(t)$ represents the center location of a tracked object, and $gt(t)$ is the center location of the object in ground truth. The overlap score is defined as follows:

$$S = \frac{|B_R \cap B_T|}{|B_R \cup B_T|} \tag{22}$$

where a tracked bounding box and the ground truth bounding box are denoted as B_R and B_T, respectively, and the intersection and union of two regions are depicted as $\cap$ and $\cup$, respectively. The number of pixels in the region is $|.|$. The success rate considers the threshold $S > 0.5$ as a correct result [50].

5.2. Comparison of MCIL with MIL trackers

To illustrate the effectiveness and robustness of the MCIL algorithm, it is compared with other MIL algorithms via two metrics: average center error and success rate. Table 3 shows the average center errors and success rates of the compared algorithms on 10 video sequences. The average center error of the MCIL algorithm is 7.11 pixels as can be seen in Table 3. The results confirm that the proposed algorithm obtains the best success rates in nine out of 10 video sequences and reaches the second best success rate in the Dollar sequences. The OIFS method requires high-dimensional multi-scale image feature vectors to project these onto a low-dimensional space, whereas MCIL can directly explore low-dimensional state space for object representation. The proposed object representation method can effectively extract patterns of object regions.

The error curves of the video sequences are shown in Fig. 7. The plots indicate that the MCIL algorithm achieves the best or second best performance in all the sequences. The MIL and weighted MIL algorithms perform good tracking only in the first frames, and they suffer from drifting. Our algorithm achieves good performance in the Coke11 sequences (Fig. 8). In the Tiger 1 sequence, our algorithm tracks a tiger with significant rotation and fast motion (sequences #41 to #80). It can also effectively handle occlusion and is robust against illumination changes and rotation (sequences #88 to #100). It achieves good performance in most of the David and Occluded Face sequences. Fig. 8 displays some tracking results for four video sequences.

The proposed algorithm exhibits good performance under illumination changes (frame #375 of david and frames #730 and #1000 of Sylvester) and overcomes partial occlusions because of the robust representations in frame #442 of the Girl sequences and frame #225 and #779 in occluded face (Fig. 9). Under rotation and fast motion, it performs very well, whereas MIL and WMIL drift within frames #121 and #242 of the Girl sequences and frame #730 of the Sylvester sequences (Fig. 9). Table 4 lists the average speed of the algorithms. The online MIL tracker involves more complexity in directly computing bag probabilities and object tracking instances. The speed of the online MIL tracker is lower than those of the other algorithms.

WMIL and DFMIL compute the likelihood function using Taylor approximation, whereas MCIL computes this function using a chaotic map. The aforementioned trackers are more efficient than the online MIL tracker. The MIL and WMIL trackers quickly calculate Haar-like features by using an integral image, but they require a high-dimensional pool of features in an online process.

Although DFMIL and the proposed method entail a slightly longer time in feature extraction, they require only a few weak classifiers M and K. Generally, the speed of the proposed method is comparable to that of WMIL. A trade-off exists between the number of selectors K and the number of weak classifiers M. A large number of M and K reduces the speed of the MIL trackers but enables them to acquire more discriminative features. Figs. 9 and 10 illustrate the average center errors of the proposed algorithm under different numbers of M and K. It exhibits good performance at $M = 100$ and robustness at $K = 10$. Note that a large number of K can decrease the performance of MCIL because a strong classifier can be learned using some weak classifiers.

5.3. Comparison of MCIL with state-of-the-art trackers

In this section, the performance of our algorithm is compared with that of detection-based and stochastic algorithms, namely, VTD [46], Struck [47], TLD [48], CT [49], fast compressive sensing (FCT) [50], transductive learning with sample matching costs (TL) [51], and sparse hashing (SH) [52]. Table 5 indicates that the performance of the proposed algorithm is superior to that of the other algorithms in most of the video sequences. Its center location error is comparable to the best results of SH in the Caviar1 and Face sequences. The success rates of the proposed algorithm are comparable to those of FCT in the Animal, Singer1, and Jumping sequences.

The findings show that the compressive sensing method based on the chaotic dynamics of an image more effectively executes object tracking than does FCT. The TLD and VTD trackers frequently fail because of holistic representations in a complex background. By contrast, the MCIL tracker successfully selects features for the separation of the foreground from the background through chaotic updating. MCIL also performs well in fast motion situations, such as that in the OWL sequence, because it uses the dynamic model of the target to distinguish it from the background and adopts the local model to match target details.

Nevertheless, although the proposed algorithm efficiently deals with partial occlusion and illumination changes, it cannot handle full occlusion and out-of-view attributes (Fig. 11). MCIL is based on a discriminative appearance model or MIL, which is unable to estimate the global motion of objects. The proposed object representation method is an effective feature extraction approach for object tracking, but it is unsuitable for small targets (Fig. 11). In this case, it cannot extract the dynamic information of a target.

5.4. Comparisons of performance on the large dataset (TB)

The proposed algorithm is applied to the TB dataset [41, 42], which comprises 50 video sequences and 29 tracking algorithms. The success rates are the robustness metrics for tracking methods. A metric represents the percentage of correctly predicted targets within location error thresholds. The success plots reveal success rates for thresholds 0 to 1. The area under the curve (AUC) of a success plot is used to rank the algorithms. Three metrics are deployed for evaluation: one-pass evaluation (OPE), temporal robustness evaluation (TRE), and spatial robustness evaluation (SRE). Both temporal and spatial robustness levels are evaluated on the basis of overlap precision. TRE evaluates the temporal robustness of the algorithms at 20 runs with different starting frames, whereas SRE evaluates sensitivity to the spatial initialization errors of trackers. For the metrics, different shifts (four in the center and corner) and scales (four scale variations) are employed to the ground truth bonding box of an object. In OPE, more than 29,000 frames are used for comparison; in SRE, more than 350,000 bounding boxes are generated; and in TRE, more than 310,000 frames are tested, with each sequence consisting of 20 parts. We compare 30 trackers on the TB dataset, but we show only the AUCs of the top 10 algorithms in figures for clarity in each plot, wherein we use the same color to denote the same rank in all the plots. The plots in Fig. 12 depict the performance of the top 10 algorithms on video sequences under attributes. As can be seen in Fig. 12, the success rate of MCIL is higher than those of ASLA [53] and SCM at a small threshold. The success rate of MIL is comparable to those of ASLA and SCM, but it is better than that of Struck. The results also show that MCIL can handle scale and shift variations. Our representation is ranked the highest under rotation, fast motion, and illumination variations. Under occlusion, Struck, MCIL, and SCM outperform the other algorithms because of the learning process and rich representations presented by the two aforementioned trackers.

As can be observed in Fig. 13, the robustness of the chaotic representations in TRE and SRE is higher than that in OPE. The chaotic representation in the first two evaluations can extract sparse feature vectors and thus describe the global and local information of the target. As a result, our algorithm performs well when evaluated against TRE and SRE metrics. The chaotic model can represent appearance changes, and the results confirm that our algorithm is more effective than Struck and other sparse representations such as (SCM, ASLA [53], and LSK). The SRE of our algorithm indicates that it is robust against translation and scale variations. It can extract the invariant features of sparse state space, thereby preserving significant information on the deterministic dynamics of targets.

The proposed algorithm is also compared with CNT [54], KCF [55], TGPR [56], and 29 algorithms on the TB dataset under 11 challenges, namely, low resolution (LR), in-plane rotation (IPR), out-of-plane rotation (OPR), scale variation (SV), occlusion (OCC), deformation (DEF), background clutter (BC), illumination variation (IV), motion

blur (MB), fast motion (FM), and out-of view (OV) attributes (Tables 6 and 7). Table 6 and Table 7 show that the proposed algorithm outperforms the others in most of the cases. Its superior performance indicates that chaotic object representation based on chaos theory can handle four challenges: OPR, SV, IV, and FM. The tables also show that the proposed algorithm and CNT perform well under low-resolution attributes with the help of the obtained dense feature space of the proposed algorithm. Under rotation and scale variations, our algorithm uses invariant features with local and global feature extraction. Under occlusions, it deals with partial occlusion by using a deterministic model of the sub-region of a target, whereas CNT, KCF, TGPR, and SCM extract local information without a global model of a target. The proposed algorithm also performs well under illumination variations. Furthermore, it can represent the pixel variations in a target region, which is insensitive to illumination changes. The chaotic representation extracts significant information on a target region for object tracking.

6. Conclusion and future work

This study proposes an efficient tracker with an appearance model based on chaotic representation and a chaotic map method for online updating. In the algorithm, the variations in intensity values are embedded in state space by using the chaotic characteristics of a target region. The fractal dimension of state space is then employed as an importance weight of an instance for efficient online learning. Moreover, chaotic approximation is applied in the online learning of a classifier through the use of chaotic approximation to maximize the bag likelihood function for online MIL. The chaotic representation is robust to any object appearance changes, including partial occlusion, illumination changes, scale changes, and rotation variations. The essential idea of chaotic representation is to simultaneously recover the deterministic and sparse dynamics of a target. To illustrate the effectiveness and robustness of MCIL, we compare it with online MIL trackers and the latest state-of-the-art algorithms. The experimental results reveal that the proposed algorithm is a more effective tracker than stochastic and online MIL algorithms. Nevertheless, it suffers from some limitations in object tracking. For instance, it cannot accurately localize a small target given that the proposed object representation method requires a sufficient number of target region pixels to extract the dynamic information of a target. It also fails to handle full occlusion given that it is an MIL-based method. Therefore, chaotic representation is applicable to a motion-based method in precisely localizing targets.

Acknowledgement
The authors would like to thank three anonymous reviewers for their constructive comments. We are thankful to Dr. Javad Alikhani koupaei who provided expertise that greatly assisted the research.

References

[1] T. Zhang, B. Ghanem, S. Liu, C. Xu, N. Ahuja, Robust Visual Tracking via Exclusive Context Modeling, IEEE Transactions on Cybernetics 46 (2016) 51-63.
[2] P. Chen, X. Zhang, A. Mao, J. Xiong, Visual tracking via adaptive multi-task feature learning with calibration and identification, Signal Processing: Image Communication 49 (2016) 17-24.
[3] L. Wang, T. Liu, G. Wang, K. L. Chan, Q. Yang, Video tracking using learned hierarchical features, IEEE Transactions on Image Processing 24 (2015) 1424-1435.
[4] H. Bao, M. Lin, Z. Chen, Robust visual tracking based on hierarchical appearance model, Neurocomputing 221 (2016) 108-122.
[5] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, A. V. D. Hengel, A survey of appearance models in visual object tracking, ACM transactions on Intelligent Systems and Technology 4 (2013) 1-42.
[6] M. Hassaballah, A. A. Abdelmgeid, H. A. Alshazly, Image Features Detection, Description and Matching, In Image Feature Detectors and Descriptors, pp. 11-45. Springer International Publishing, 2016.
[7] S. Zhang, Q. Tian, Q. Huang, W. Gao, Y. Rui, USB: ultrashort binary descriptor for fast visual matching and retrieval, IEEE Trans. Image Process. 23 (2014) 3671–3683.

[8] T. Tuytelaars, K. Mikolajczyk, Local invariant feature detectors: a survey, Found. Trends Comput. Graph. Vis. 3(2007) 177–280.
[9] W. Xiong, L. Zhang, B. Du, and D. Tao, Combining local and global: Rich and robust feature pooling for visual recognition, Pattern Recognition 62 (2017) 225-235.
[10] X. Xie, Y. Xu, Q. Liu, F. Hu, T. Cai, N. Jiang, H. Xiong, A study on fast SIFT image mosaic algorithm based on compressed sensing and wavelet transform, Journal of Ambient Intelligence and Humanized Computing 6 (2015) 835-843.
[11] E. Zeraoulia, Models and Applications of Chaos Theory in Modern Sciences, CRC Press, 2011.
[12] J. Alikhani Koupaei, and S. M. M. Hosseini, A new hybrid algorithm based on chaotic maps for solving systems of nonlinear equations, *Chaos, Solitons & Fractals* 81 (2015) 233-245.
[13] M. E. Farmer, A chaos Theoretic analysis of motion and illumination in video sequences, Journal of Multimedia 2 (2007) 53 -64.
[14] R. K. Upadhyay, S. R. Iyengar, Introduction to mathematical modeling and chaotic dynamics, CRC press, 2013.
[15] C. B. Gan, H. Lei, A new procedure for exploring chaotic attractors in nonlinear dynamical systems under random excitations, Acta Mechanica Sinica 27 (2011) 593-601.
[16] A. Jafari, F. Almasganj, Using nonlinear modeling of reconstructed phase space and frequency domain analysis to improve automatic speech recognition performance, International Journal of Bifurcation and Chaos 22 (2012): 1250053.
[17] S. Mangiarotti, M. Peyre, and M. Huc, A chaotic model for the epidemic of Ebola virus disease in West Africa (2013–2016), Chaos: An Interdisciplinary Journal of Nonlinear Science 26 (2016) 113112.
[18] F. Takens, Lecture Notes in Mathematics, 898 (Springer, Berlin), 366 (1981).
[19] F. Takens, Detecting strange attractors in turbulence, Springer Berlin Heidelberg, 1981, pp. 366-381.
[20] M. Casdagli, Nonlinear prediction of chaotic time series, Physica D: Nonlinear Phenomena 35 (1989) 335-356.
[21] A. M. Fraser, H. L. Swinney, Independent coordinates for strange attractors from mutual information, Physical review A 33 (1986) 1134-1140.
[22] M. B. Kennel, R. Brown, H. D. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction, Physical review A 45 (1992) 3403-3411.
[23] A. H. Jiang, X. C. Huang, Z. H. Zhang, J. Li, Z. Y. Zhang, H. X. Hua, Mutual information algorithms, Mechanical Systems and Signal Processing 24 (2010) 2947-2960.
[24] X. Li, G. Ouyang, Nonlinear similarity analysis for epileptic seizures prediction, Nonlinear Analysis: Theory, Methods & Applications 64 (2006) 1666-1678.
[25] M. Abdechiri, K. Faez, H. Amindavar, E. Bilotta, The chaotic dynamics of high-dimensional systems, Nonlinear Dynamics (2016). doi:10.1007/s11071-016-3213-3, pp. 1-14.
[26] J. Wu, J. Lu, J. Wang, Application of chaos and fractal models to water quality time series prediction, Environmental Modelling & Software 24 (2009) 632-636.
[27] I. Leichter, M. Lindenbaum, E. Rivlin, Mean shift tracking with multiple reference color histograms, Computer Vision and Image Understanding 114 (2010) 400-408.
[28] B. Babenko, M. H. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 1619-1632.
[29] M. Abdechiri, K. Faez, Efficacy of utilizing a hybrid algorithmic method in enhancing the functionality of multi-instance multi-label radial basis function neural networks, *Applied Soft Computing* 34 (2015) 788-798.
[30] B. Babenko, M. H. Yang, S. Belongie, Visual tracking with online multiple instance learning, The IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 983-990.
[31] Y. Xie, Y. Qu, C. Li, W. Zhang, Online multiple instance gradient feature selection for robust visual tracking, Pattern Recognition Letters 33 (2012) 1075-1082.
[32] J. Ning, W. Shi, S. Yang, P. Yanne, Visual tracking based on distribution fields and online weighted multiple instance learning, Image and Vision Computing 31 (2013) 853-863.
[33] L. Zhang, L. Van Der Maaten, Preserving structure in model-free tracking, IEEE transactions on pattern analysis and machine intelligence 36 (2014) 756-769.
[34] K. Zhang, H. Song, Real-time visual tracking via online weighted multiple instance learning, Pattern Recognition 46 (2013) 397-411.
[35] A. M. Tillmann, M. E. Pfetsch, The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing, IEEE Transactions on Information Theory 60 (2014) 1248-1259.
[36] M. S. Asif, J. Romberg, Sparse Recovery of Streaming Signals Using l_1-Homotopy, IEEE Transactions on Signal Proce. 62 (2014) 4209-4223.

[37] K. Tatsumi, T. Tanino, A sufficient condition for chaos in the gradient model with perturbation method for global optimization, International Journal of Bifurcation and Chaos 23 (2013) 1350102.
[38] K. Tatsumi, Y. Obita, T. Tanino, Chaos generator exploiting a gradient model with sinusoidal perturbations for global optimization, Chaos, Solitons & Fractals 42 (2009) 1705-1723.
[39] E. Lindenstrauss, Invariant measures and arithmetic quantum unique ergodicity, Annals of Mathematics (2006) 165-219.
[40] E. Hopf, Some Topics of Ergodic Theory, In Sistemi dinamici e teoremi ergodici, pp. 47-112. Springer Berlin Heidelberg, 2011.
[41] Y. Wu, J. Lim, M. H Yang, Object tracking benchmark, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (2015) 1834-1848.
[42] Y. Wu, J. Lim, M. H. Yang, Online object tracking: A benchmark, The IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2411-2418.
[43] J. Ning, W. Shi, S. Yang, P. Yanne, Improved appearance updating method in multiple instance learning tracking, IET Computer Vision 8 (2014) 118-130.
[44] H. Song, Robust visual tracking via online informative feature selection, Electronics Letters 50 (2014) 1931-1933.
[45] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: An experimental survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2014) 1442-1468.
[46] J. Kwon, and K. M. Lee, Visual tracking decomposition, The IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 1269-1276.
[47] S. Hare, A. Saffari, P. H. Torr, Struck: Structured output tracking with kernels, Computer Vision (ICCV), 2011.
[48] Z. Kalal, K. Mikolajczyk, and J. Matas, Tracking-learning-detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (2012) 1409-1422.
[49] K. Zhang, L. Zhang, M. H. Yang, Real-time compressive tracking, Computer Vision, ECCV 2012, Springer Berlin Heidelberg, 2012, pp. 864-877.
[50] K. Zhang, L. Zhang, M. H. Yang, Fast compressive tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (2014) 2002-2015.
[51] P. Zhang, T. Zhuo, Y. Zhang, D. Tao, J. Cheng, Online tracking based on efficient transductive learning with sample matching costs, Neurocomputing 175 (2016) 166–176.
[52] L. Zhang, H. Lu, D. Du, L. Liu, Sparse hashing tracking, IEEE Transactions on Image Processing 25 (2016) 840-849.
[53] X. Jia, H. Lu, M. H. Yang, Visual tracking via adaptive structural local sparse appearance model, The IEEE conference on Computer vision and pattern recognition (CVPR), 2012, pp. 1822-1829.
[54] K. Zhang, Q. Liu, Y. Wu, M. H. Yang, Robust Visual Tracking via Convolutional Networks Without Training, IEEE Transactions on Image Processing 25 (2016) 1779-1792.
[55] J. F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (2015) 583-596.
[56] J. Gao, H. Ling, W. Hu, J. Xing, Transfer learning based visual tracking with gaussian processes regression, In European Conference on Computer Vision, pp. 188-203. Springer International Publishing, 2014.

Figure 1. MIL tracker. The green bounding box is considered to select positive instances for training phase with radius r. The yellow bounding box is selected by radius β for negative samples. In the frame $t+1$, the radius s or red bounding box of current position to crop out a set of instances to compute $p(y|x)$.

Figure 2. The proposed method consists of the motion model and appearance model updating.

Figure 3. The leaf image is converted to a time series. The time series shows the pixel amplitudes in time *t*. The time series is created with raw or column scanning of image.

Figure 4. The state space reconstruction of the target and the histogram of state space with different patches around the target for David video sequences.

Figure 5. The state space reconstructions and the histograms in different scale (size target in First sample is 91*71 and in second sample is 64*54.

Figure 6. The state space reconstruction and the histograms for rotation out of view and resolution change.

Figure 7. Error plots of MIL, WMIL, DFMIL, and proposed method for Coke11, Tiger1, David, Occluded face, Girl, and Surfer.

Figure 8. The results of tracking for (a) David, (b) Girl, (c) Occluded Face, and (d) Sylvester sequences. The green bounding box is the obtained result of MIL. The obtained results of WMIL are blue bounding boxes. The red and white bounding are the results of DFMI and the proposed method respectively.

Figure 9. The average center errors of MCIL tracker under different numbers of weak classifiers (M).

Figure 10. The average center errors of MCIL tracker under different numbers of selectors (k).

Figure 11. Qualitative results on sequences with small target and full occlusion. The results of the proposed method are shown on Walking and Coke_c.

Figure 12. The success plots of top 10 algorithms in different challenges with AUC in the bracket.

Figure 13. The OPE, SRE, and TRE plots.

Algorithm 1. The proposed tracker for new frame (t)

1. Crop out a set patches, $X^s = \{x | \|l(x) - l_{t-1}^*\| < s\}$.
2. Compute chaotic representation for each patch.
3. Estimating $p(y = 1|x)$ for $x \epsilon X^s$ based on chaotic map.
4. Updating motion model $l_t^* = l\big(argmax\ p(y|x)\big)$.
5. Crop out two set of positive $X^r = \{x | \|l(x) - l_t^*\| < r\}$ and negative $X^{r,\beta} = \{x |\ r < \|l(x) - l_t^*\| < \beta\}$ samples and compute chaotic representation of each sample.
6. Updating appearance model using the bags.

Table 1. More details of MIL algorithms and main differences in visual representations and likelihood function.

	MIL [28]	WMIL [34]	DFMIL [32]	HOGMIL [31]	Our algorithm
Feature	Haar-like	Haar-like	Distribution fields	Gradient	Chaotic state space
Size of pool	Large	Large	Small	Large	Small
Drawbacks	Computational complexity	Computational complexity	Computational complexity	Computational complexity	-
Sensitive to	Illumination Blur image	Illumination Blur image	Complex appearances	Complex appearances	-
Weight	×	Euclidean distance	Euclidean distance	×	Euclidean and self-similarity
Bag probability	Noisy-OR	Arithmetic mean	Geometric mean	Noisy-OR	Arithmetic mean

Likelihood function	Original Eq. (5)	Gradient descent	Gradient descent	Original Eq. (5)	Chaotic map
Drawbacks	Computational complexity	Local minimum	Local minimum	Computational complexity	
Matlab code & data	http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml http://code.google.com/p/online-weighted-miltracker http://www.youtube.com/user/zhkhua/videos				

Table 2. Parameter settings

Algorithms	Parameters
MIL	$s = 30, \alpha = 4, \zeta = 8, \beta = 30, \lambda = 0.8$, K=50, M=250
WMIL	$s = 30, \alpha = 4, \zeta = 8, \beta = 30, \lambda = 0.8$, K=15, M=150
DFMIL	$s = 30, \alpha = 4, \zeta = 8, \beta = 30, \lambda = 0.8$, DF layer=16
Our algorithm	$s = 30, \alpha = 4, \zeta = 8, \beta = 30, \lambda = 0.85$, K=10, M=130

Table 3. Average center location errors (in pixels) and the success rates of MIL-based methods and the proposed method

Video	MIL[28]		WMIL[34]		DFMIL[32]		OIFS [49]		I-MIL[43]		The proposed method	
	CLE	SR	CLE	SR	CLE	SR	CLE	SR	CLE	SR	CLE	SR
David	26.34	56.87	22.71	88.18	14.90	91.77	8.32	**98.92**	11.23	**98.92**	**5.13**	95.50
Dollar	15.15	90.76	21.32	86.06	11.24	98.46	8.18	97.00	13.74	**100**	**6.55**	98.39
Sylvester	15.48	69.76	18.23	61.68	12.84	71.24	11.65	83.61	9.71	81.04	10.04	**93.05**
Coke11	20.85	17.93	20.90	64.21	12.18	75.06	12.00	63.76	16.08	49.15	**5.57**	**78.19**
Girl	35.77	53.20	40.67	69.40	24.51	84.11	34.76	76.32	26.79	76.24	**11.41**	**89.92**
Tiger 1	20.63	41.11	13.01	64.76	9.22	90.46	9.30	81.55	8.33	76.06	**5.08**	**90.76**
Tiger 2	21.70	44.89	15.17	68.58	7.40	92.39	10.81	83.21	10.63	72.60	**5.17**	**92.80**
Surfer	15.51	54.09	21.29	70.37	7.63	95.64	11.97	87.84	11.23	59.21	**6.11**	**96.78**
Occluded Face	29.55	75.30	31.31	75.32	11.89	98.01	21.69	93.00	25.24	80.34	**5.88**	**99.55**
Occluded Face 2	23.80	76.51	18.20	90.16	11.90	97.10	11.48	98.10	13.58	**100**	**10.19**	98.30
Average	22.47	58.04	22.28	73.87	12.37	89.52	14.01	86.33	14.65	79.35	**7.11**	**93.32**

(*) average on videos in original paper.

Table 4. Average speed of algorithms.

Algorithms	MIL [28]	WMIL [34]	DFMIL [32]	Our algorithm
Speed (FPS)	2	27	16	25

Table 5. Average center location errors (in pixels) and the overlap score of the proposed method and state-of-the arts.

Video	VTD [46]		Struck [47]		TLD [48]		CT [49]		FCT [50]		TL [51]		SH [52]		MCIL	
	CLE	SR	CLE	SR	CLE	SR	CLE	SR	CLE	SR	CLE	SR	CLE	SR	CLE	SR
Animal	10.4	58.32	11.7	60.00	160.4	49.07	10.7	N/A	15.4	**81.54**	N/A	N/A	N/A	N/A	**8.1**	78.45
Boy	7.6	63.91	3.8	77.64	4.5	67.92	9.0	60.93	7.5	81.32	5.7	70.29	6.7	70.86	**2.2**	**82.70**
Car4	12.3	73.22	7.7	49.42	N/A	64.79	234	24.27	10.3	93.33	6.0	72.96	11.4	72.69	**2.0**	**97.43**
Caviar1	**3.9**	7.75	2.5	54.71	5.5	78.58	42.1	47.35	14.7	54.23	N/A	60.00	**1.2**	60.05	5.6	**91.85**
Couple	104	32.54	11.3	69.66	2.5	41.07	36.4	69.34	10.9	58.84	10.4	64.13	8.0	64.68	**2.2**	**93.74**

Crossing	26.1	41.00	2.8	29.00	24.3	10.98	3.6	31.42	4.21	81.11	4.4	61.59	4.2	61.11	**2.1**	**76.07**
David3	66.7	60.11	107	71.30	N/A	59.27	88.7	54.05	13.2	78.32	10.7	82.44	13.1	82.29	**8.0**	**90.67**
Dog1	11.0	87.79	5.7	83.39	4.2	65.00	7.0	60.54	6.3	86.35	6.3	90.11	3.5	90.68	**2.2**	**91.33**
Face	11.1	12.66	6.9	19.17	15.4	60.02	30.7	10.69	12.4	84.93	N/A	77.65	**4.2**	77.35	5.4	**88.57**
Owl	86.8	79.32	71.9	35.74	N/A	79.06	150	34.95	10.3	69.48	N/A	79.97	7.1	79.56	**3.0**	**86.40**
Singer1	**4.1**	16.43	21.9	66.11	30	19.49	19.4	58.82	6.9	**80.64**	6.8	73.07	6.3	73.05	**4.1**	79.00
Subway	141	62.98	4.5	59.50	N/A	46.05	11.1	1.58	11.9	73.43	3.8	72.98	3.5	72.48	**2.9**	**86.65**
Walking	5.8	15.08	4.6	61.54	10.2	60.17	1.9	17.74	4.5	76.34	N/A	76.09	2.1	76.60	**1.5**	**81.23**
Woman	137	36.75	4.3	34.01	N/A	29.86	113	15.98	21.0	60.86	17.7	58.77	4.3	58.00	**3.1**	**81.77**
Freeman1	10.3	16.69	14.3	17.90	6.7	77.93	119	18.16	13.8	79.59	N/A	78.49	9.3	78.43	**4.8**	**80.36**
Jogging	83.3	12.19	62.1	63.12	6.7	65.46	92.5	N/A	10.5	66.64	N/A	N/A	6.1	N/A	**5.9**	**70.86**
Jumping	43.6	58.31	6.0	60.00	47.9	49.49	7.7	N/A	8.6	**80.00**	N/A	N/A	N/A	N/A	**5.2**	78.46

Table 6. The AUC values of success plots with 11 challenges. The obtained results of 10 top trackers are listed.

Attribute	MCIL	CNT	TGPR	KCF	SCM	Struck	TLD	ALSA	CXT	VTS
LR	0.420	**0.437**	0.351	0.312	0.279	0.372	0.309	0.157	0.312	0.168
IPR	**0.503**	0.495	0.487	0.497	0.458	0.444	0.416	0.425	0.452	0.416
OPR	**0.510**	0.501	0.507	0.495	0.470	0.432	0.420	0.422	0.418	0.425
SV	**0.574**	0.508	0.443	0.427	0.518	0.425	0.421	0.452	0.389	0.400
OCC	0.512	0.503	0.494	**0.514**	0.487	0.413	0.402	0.376	0.372	0.398
DEF	0.533	0.524	**0.556**	0.534	0.448	0.393	0.378	0.372	0.324	0.368
BC	**0.545**	0.488	0.543	0.535	0.450	0.458	0.345	0.408	0.338	0.428
IV	**0.505**	0.456	0.486	0.493	0.473	0.428	0.399	0.429	0.368	0.429
MB	0.420	0.417	0.440	**0.497**	0.298	0.433	0.404	0.258	0.369	0.304
FM	**0.500**	0.404	0.441	0.459	0.296	0.462	0.417	0.247	0.388	0.300
OV	0.451	0.431	0.431	**0.550**	0.361	0.459	0.457	0.312	0.427	0.443

Table 7. The AUC values of precision plots with 11 challenges and 10 top trackers.

Attribute	MCIL	TGPR	KCF	CNT	Struck	SCM	TLD	VTD	VTS	CXT
LR	0.550	0.539	0.381	**0.557**	0.545	0.305	0.349	0.168	0.187	0.371
IPR	**0.731**	0.706	0.725	0.661	0.617	0.597	0.584	0.599	0.579	0.610
OPR	**0.745**	0.741	0.729	0.672	0.597	0.618	0.596	0.620	0.604	0.574
SV	**0.711**	0.703	0.679	0.662	0.639	0.672	0.606	0.597	0.582	0.550
OCC	**0.750**	0.708	0.749	0.662	0.564	0.640	0.563	0.545	0.534	0.491
DEF	0.656	**0.768**	0.740	0.687	0.521	0.586	0.512	0.501	0.487	0.422
BC	0.647	**0.761**	0.753	0.646	0.585	0.578	0.428	0.571	0.578	0.443
IV	**0.749**	0.687	0.728	0.566	0.558	0.594	0.537	0.557	0.573	0.501
MB	**0.662**	0.578	0.650	0.507	0.551	0.339	0.518	0.375	0.375	0.509
FM	**0.635**	0.575	0.602	0.500	0.604	0.333	0.551	0.352	0.353	0.515
OV	**0.581**	0.431	0.550	0.439	0.459	0.361	0.457	0.462	0.443	0.427

Highlights

- In this paper, a chaotic sparse representation is introduced to balance the local and global features for object tracking based on a deterministic model of irregularity of intensities.
- The fractal dimension and the position distance are simultaneously used to provide an efficient online learning procedure.
- This method improves the effectiveness and efficiency of the online classifier in the online MIL tracker.

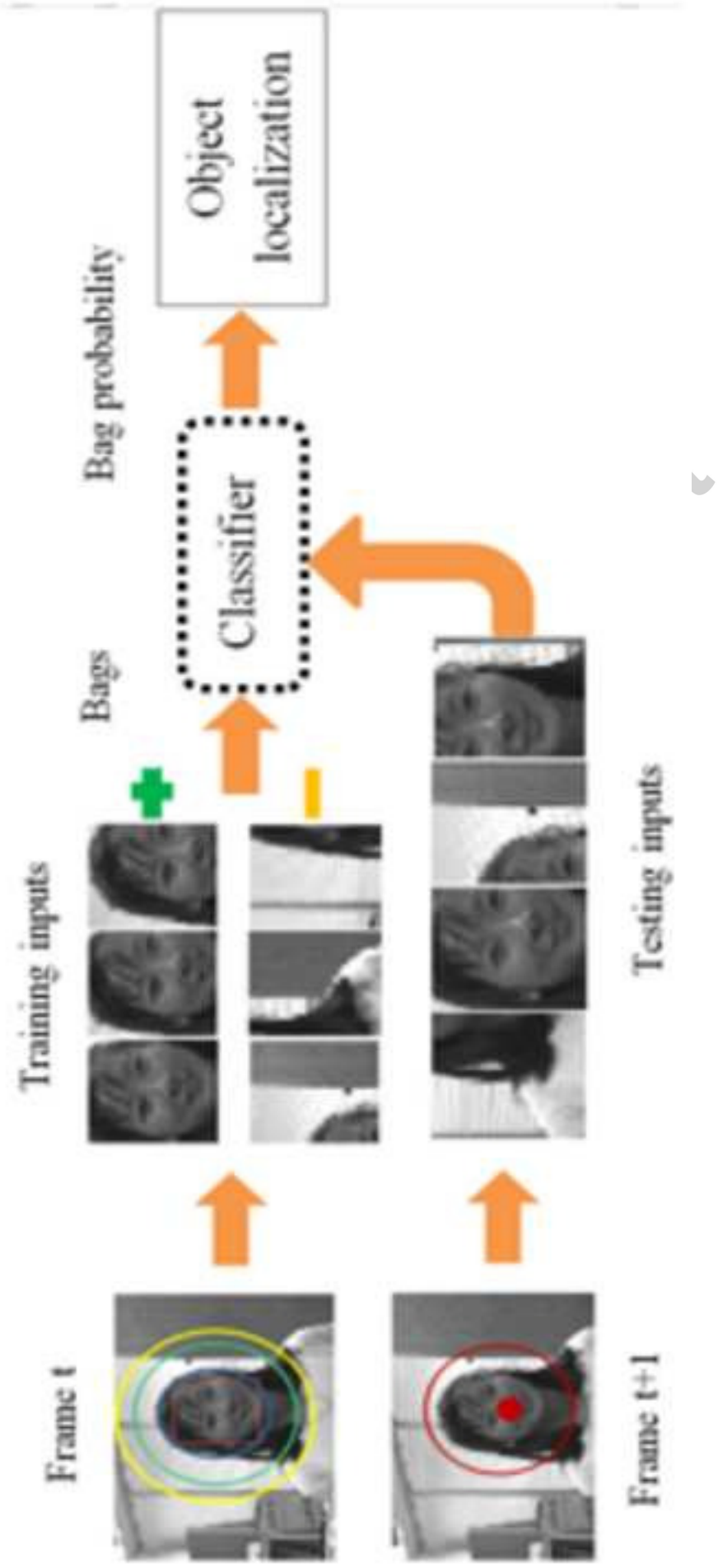
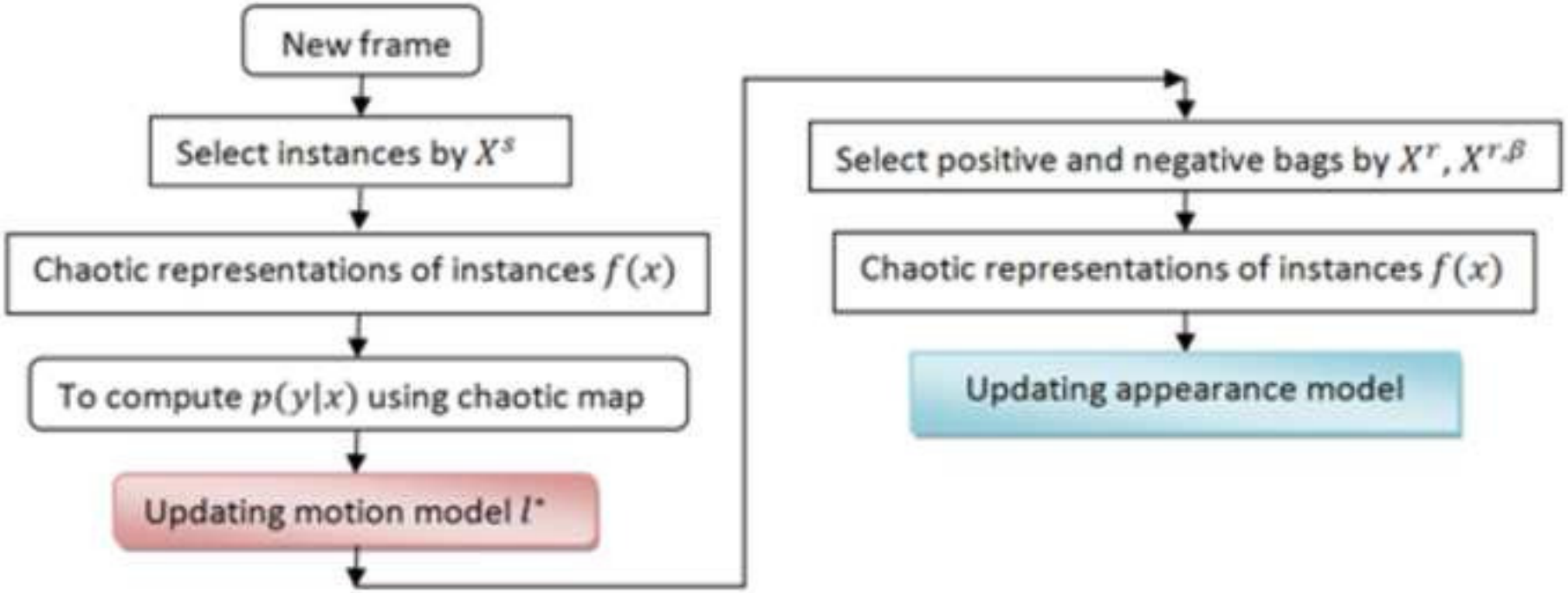

Frame t
Frame t+1
Training inputs
Testing inputs
Bags
Classifier
Bag probability
Object localization

Figure 1

Figure 2



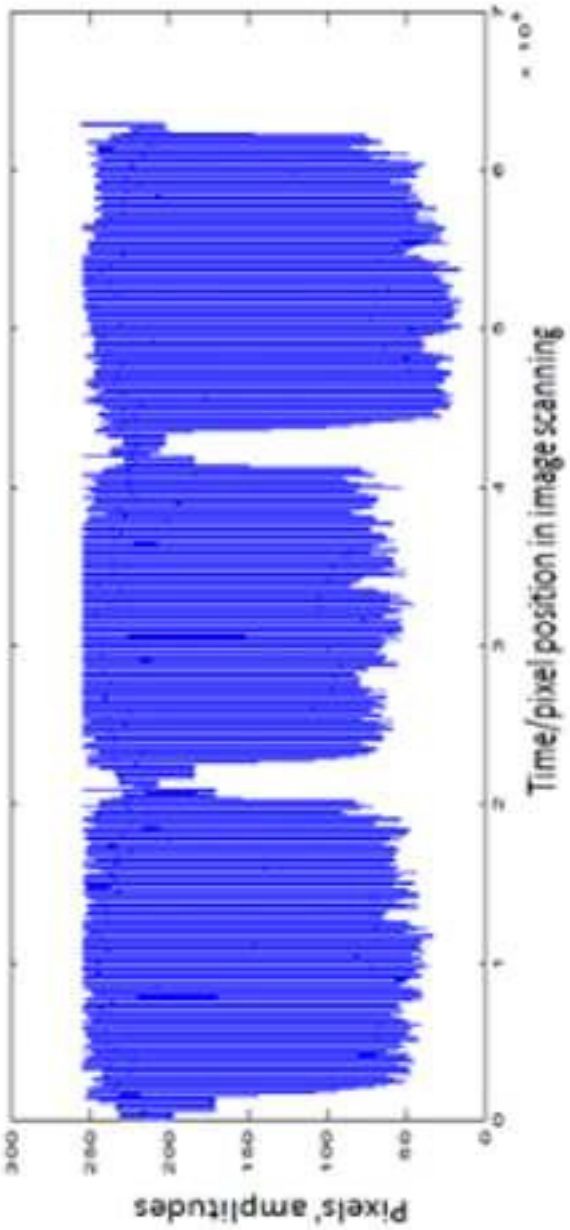

Pixels' amplitudes
Time/pixel position in image scanning
300
250
200
150
100
50
0
0
1
2
3
4
5
6
7
x 10^4

Figure 3

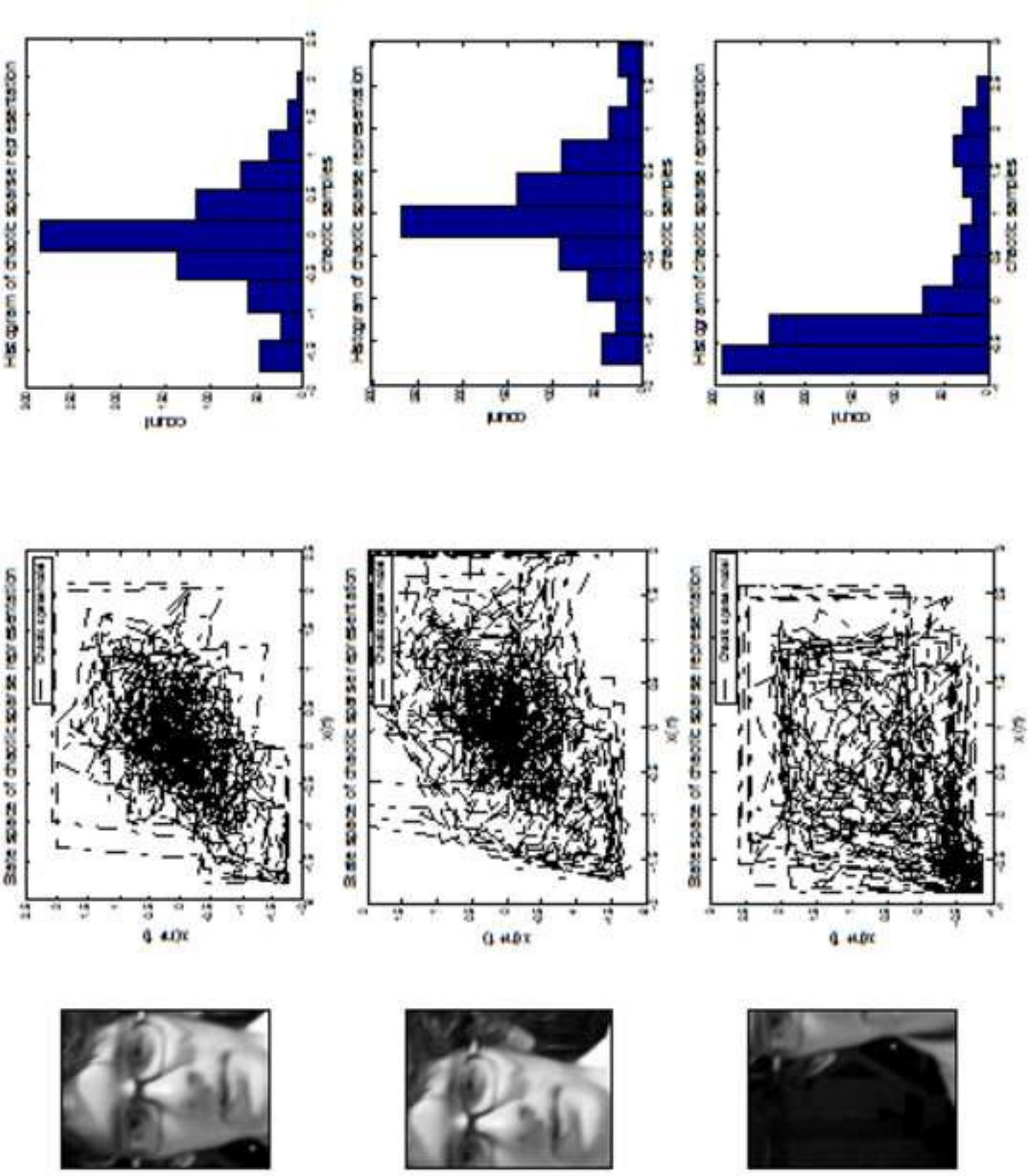

Histogram of chaotic source representation
chaotic samples
count
State space of chaotic source representation
Chaotic signal model
x(n)
x(n-1)

Figure 4

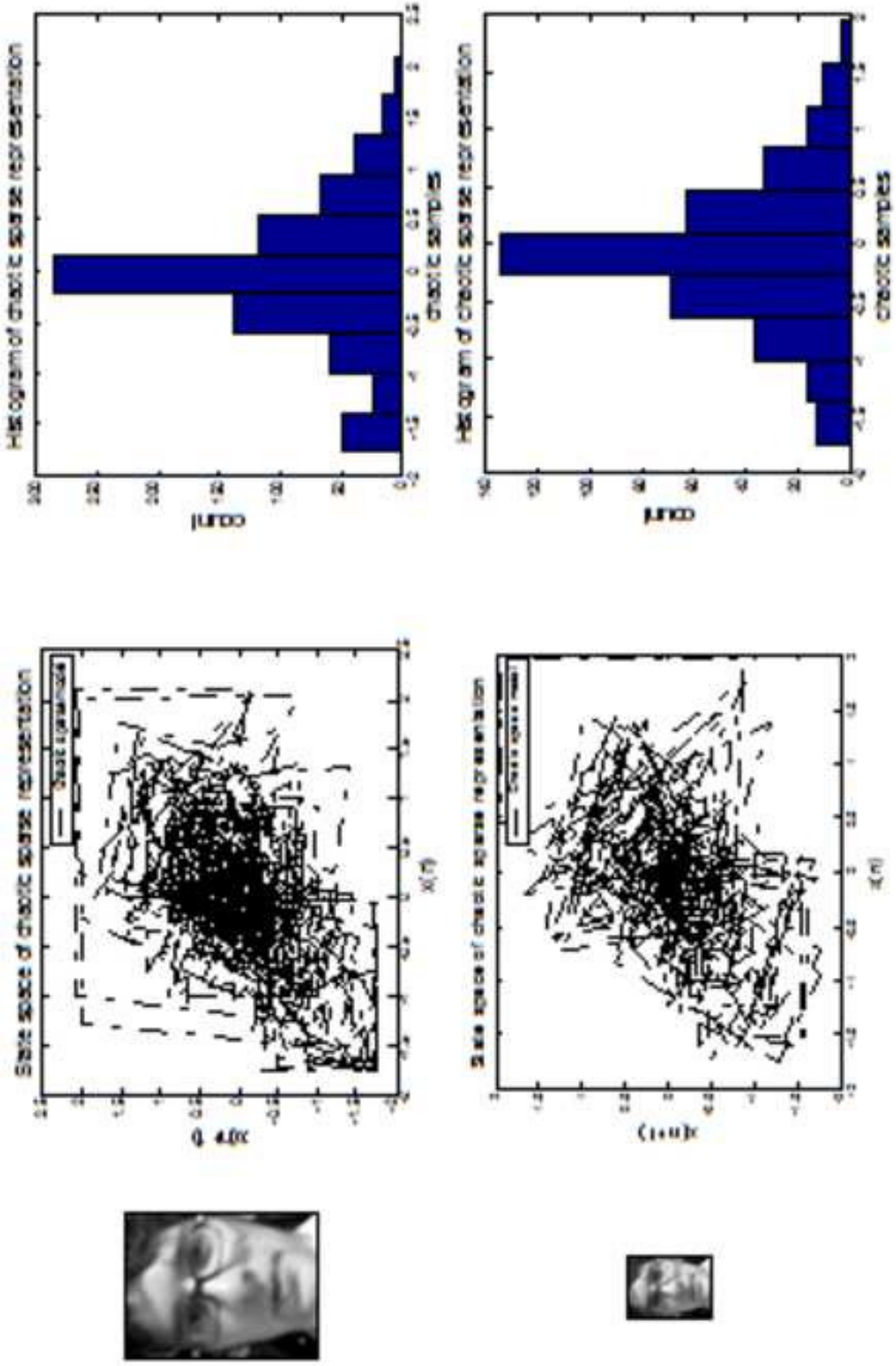

Histogram of chaotic sparse representation
chaotic samples
count
Histogram of chaotic sparse representation
chaotic samples
count
State space of chaotic sparse representation
x(n)
State space of chaotic sparse representation
x(n)

Figure 5

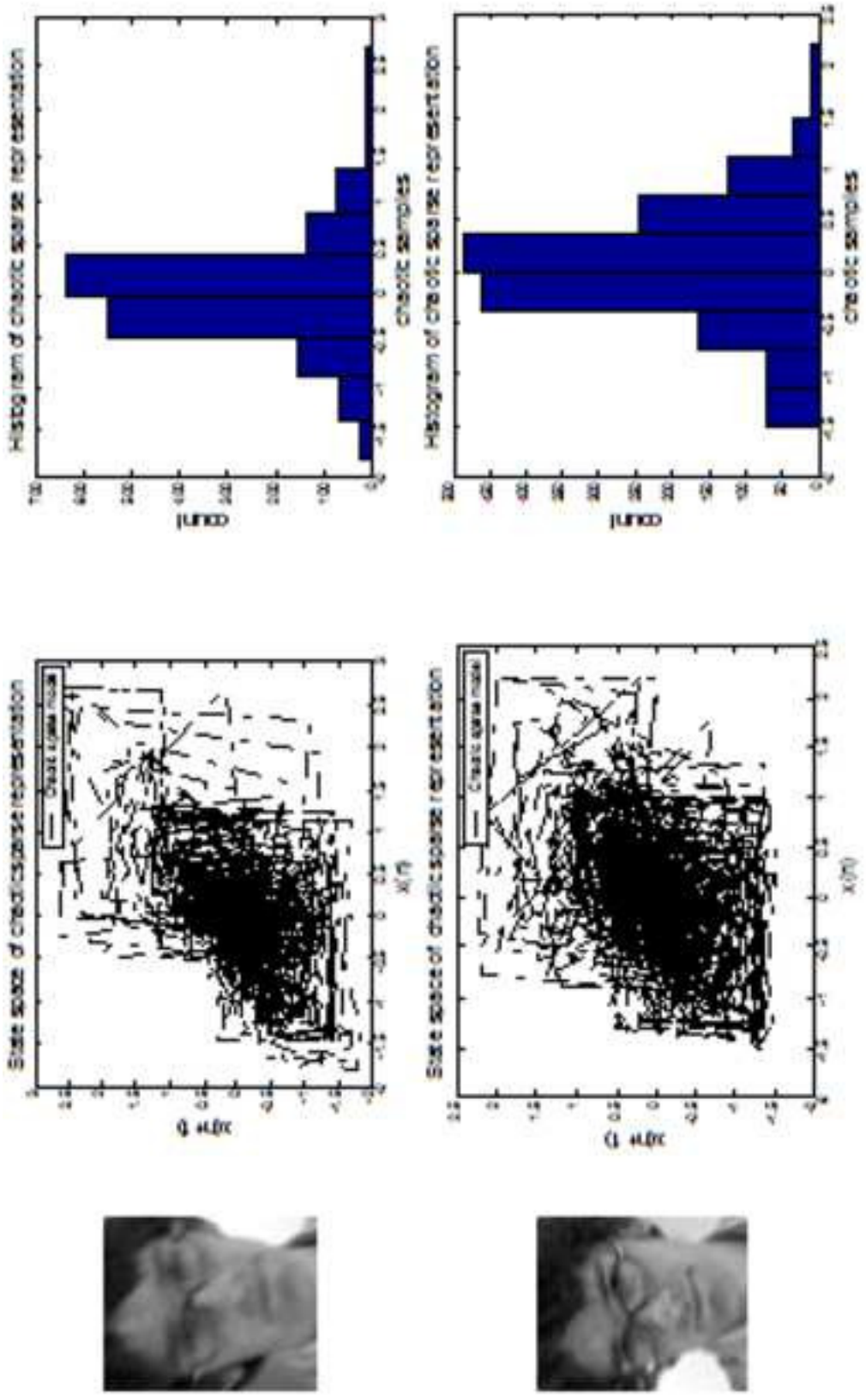

Histogram of chaotic source representation
chaotic samples
Histogram of chaotic source representation
chaotic samples
State space of chaotic source representation
State space of chaotic source representation

Figure 6

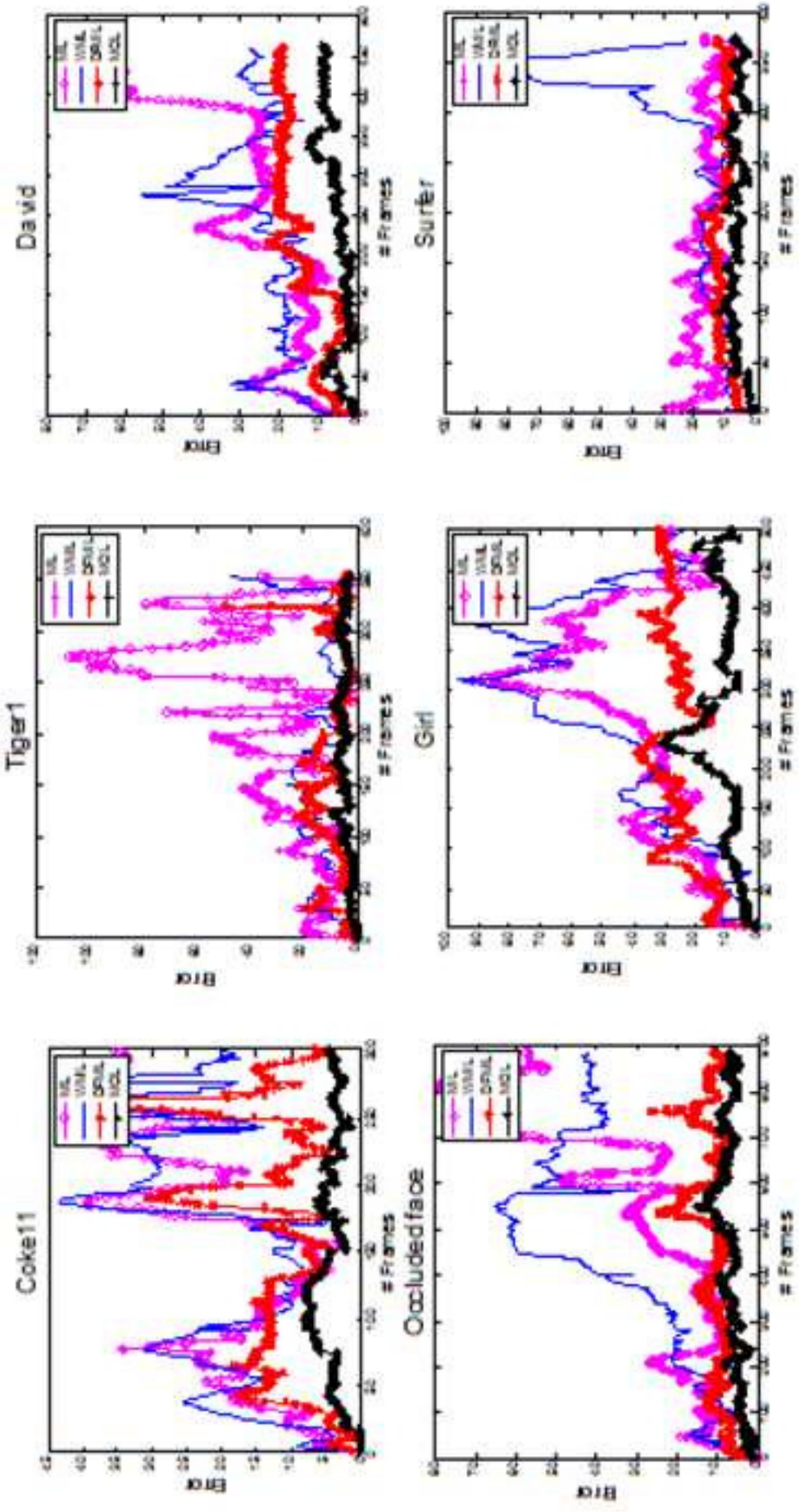
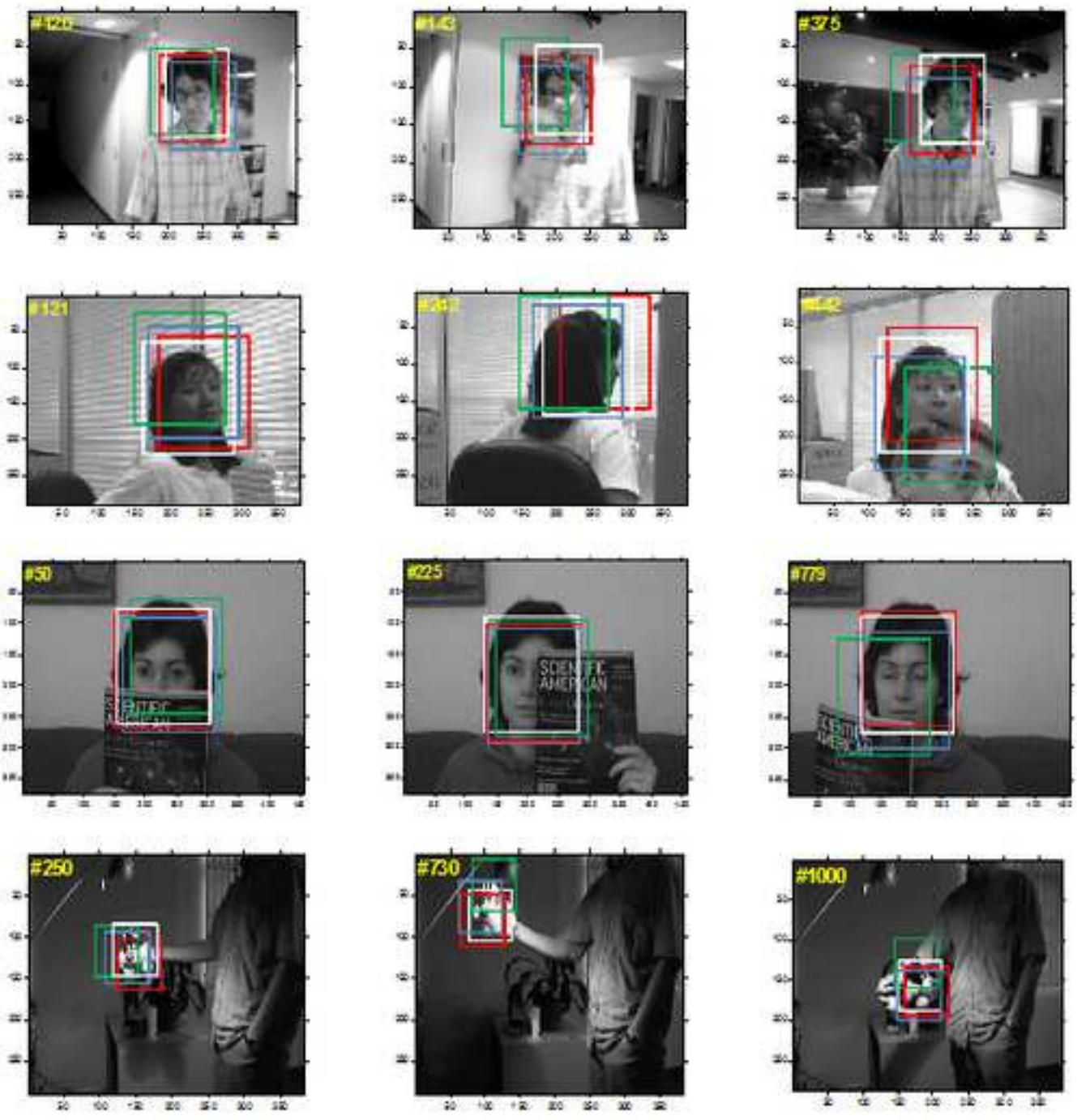

David
Surfer
Tiger1
Girl
Coke11
Occluded face
Error
Frames

Figure 7

Figure 8



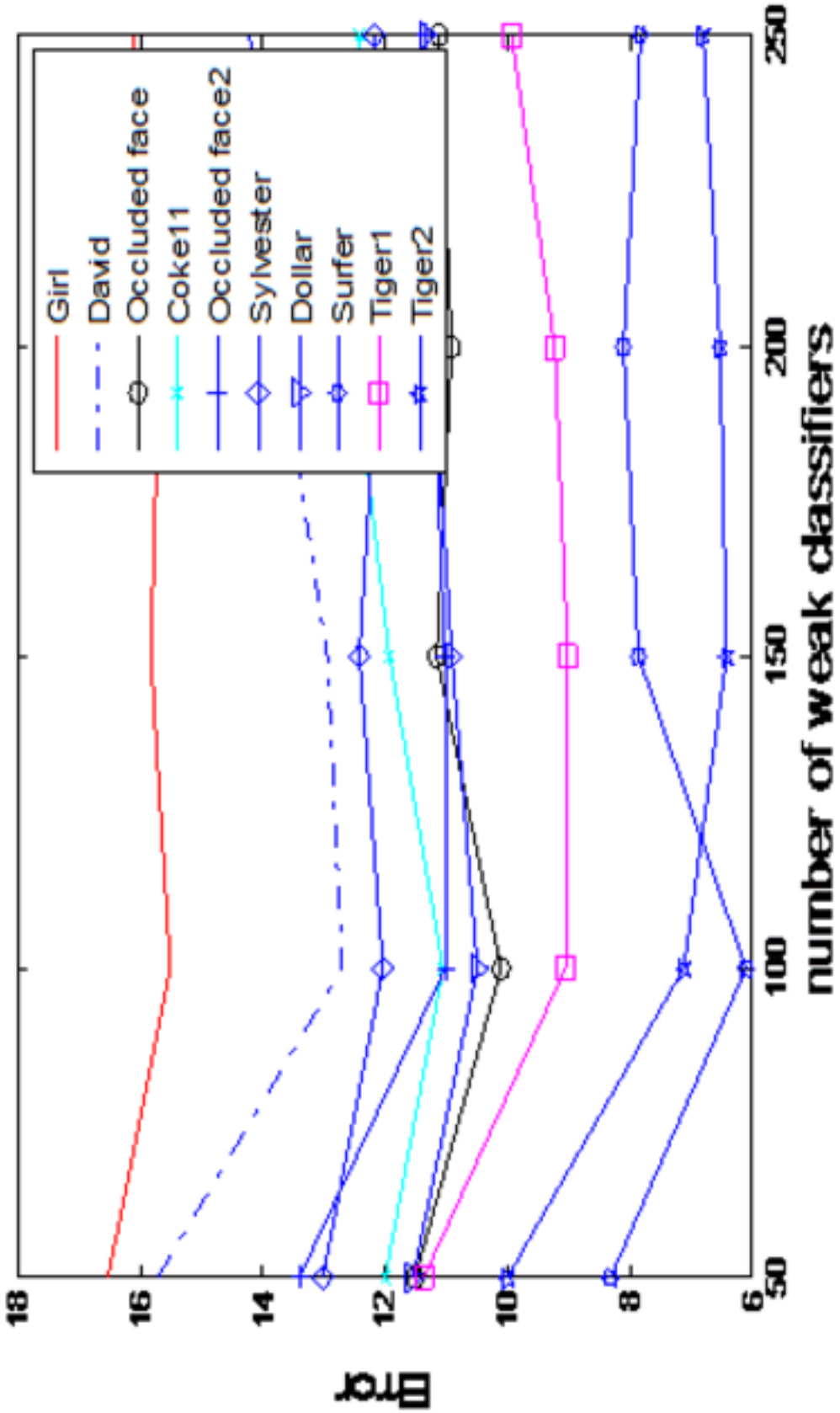

Girl
David
Occluded face
Coke11
Occluded face2
Sylvester
Dollar
Surfer
Tiger1
Tiger2
Error
number of weak classifiers
18
16
14
12
10
8
6
50
100
150
200
250

Figure 9

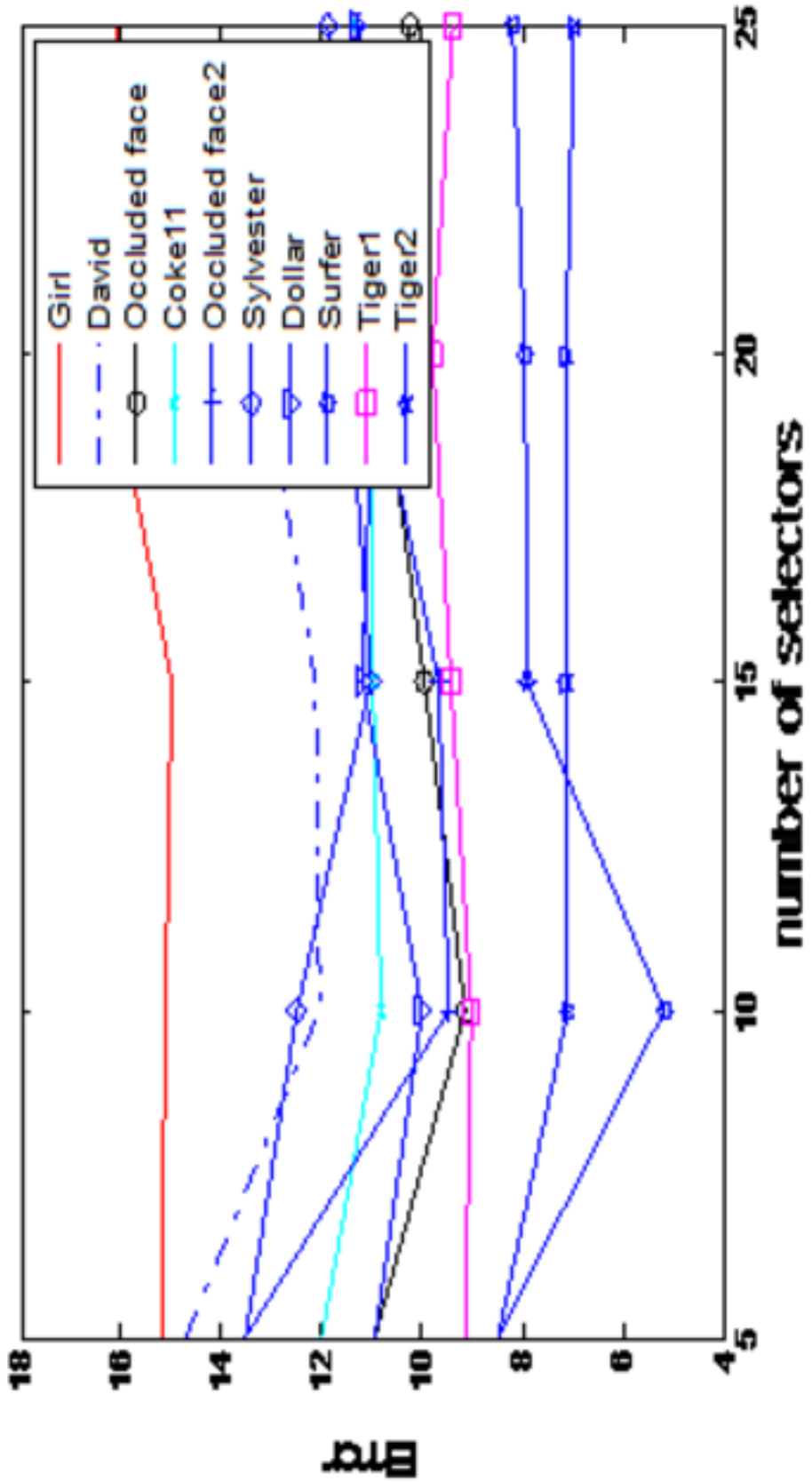

Girl
David
Occluded face
Coke11
Occluded face2
Sylvester
Dollar
Surfer
Tiger1
Tiger2
Err
number of selectors
18
16
14
12
10
8
6
4
5
10
15
20
25

Figure 10

Figure 11

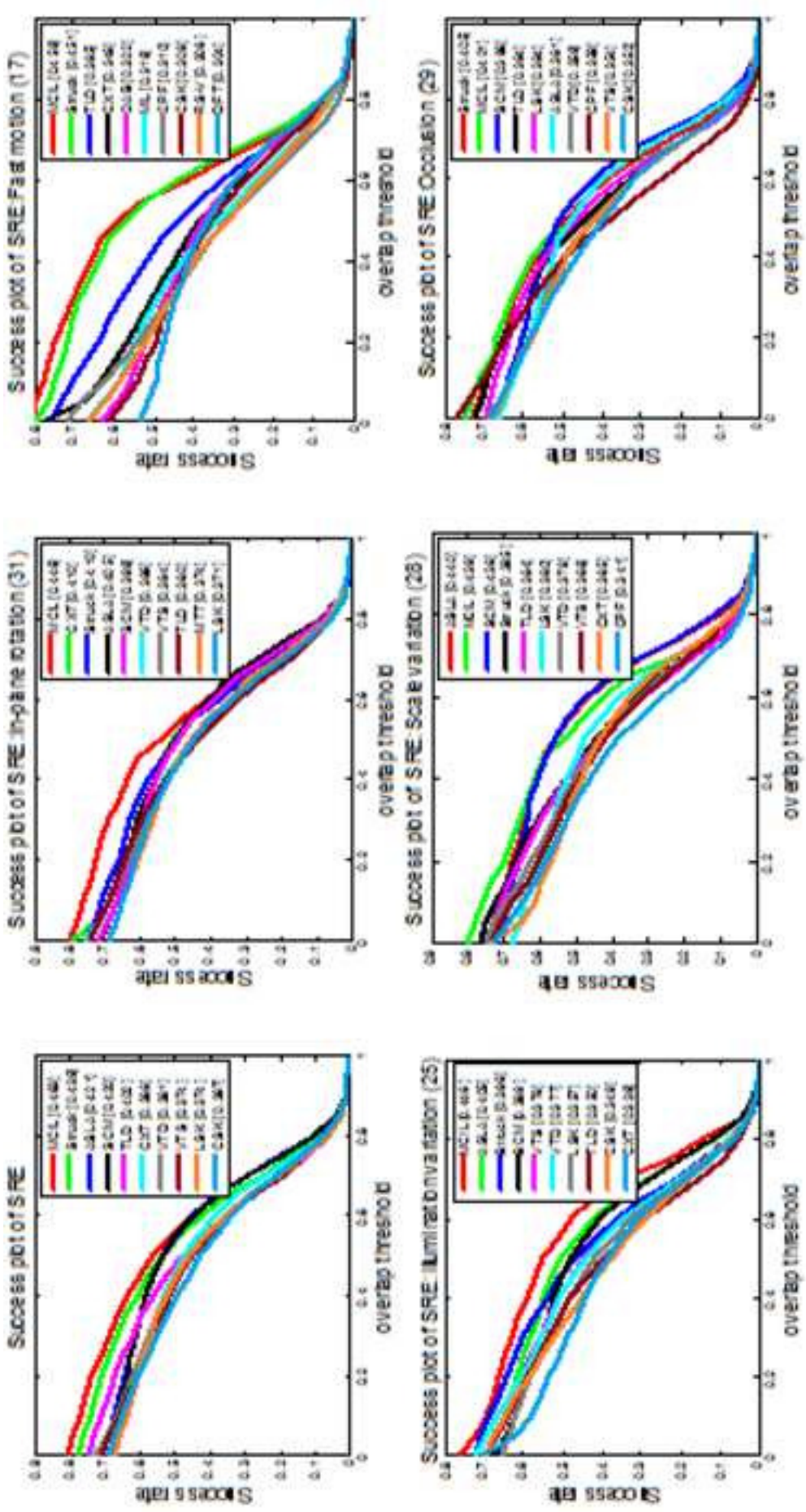

Success plot of SRE
Success plot of SRE: Illumination variation (25)
Success plot of SRE: Scale variation (28)
Success plot of SRE: In-plane rotation (31)
Success plot of SRE: Fast motion (17)
Success plot of SRE: Occlusion (29)
Success rate
overlap threshold

Figure 12

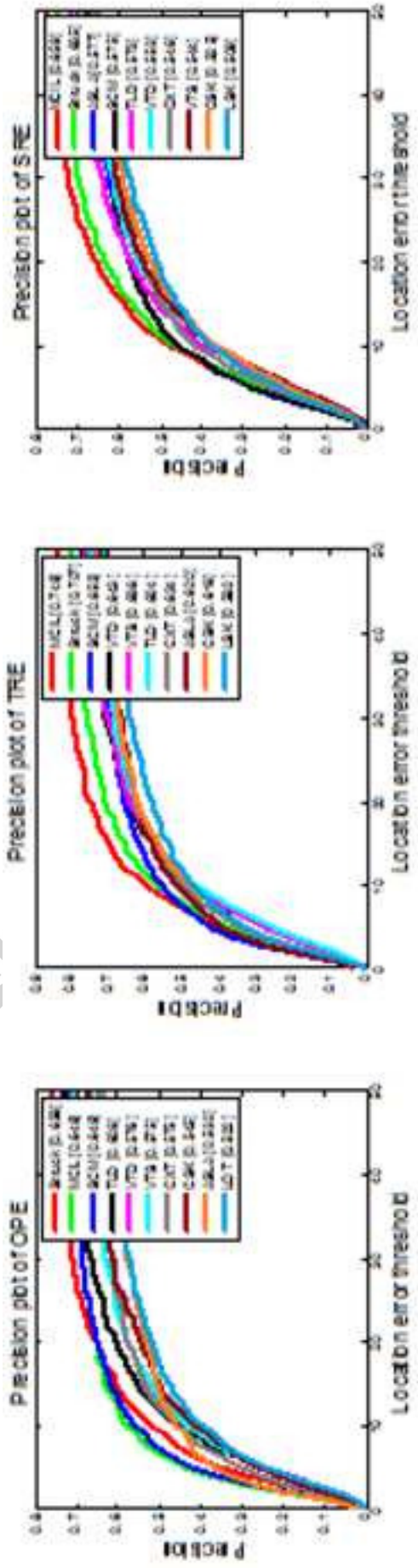

Precision plot of OPE
Precision plot of TRE
Precision plot of SRE
Location error threshold
Location error threshold
Location error threshold
Precision
Precision
Precision

Figure 13