Clustering Ensemble Tracking

Guibo Zhu, Jinqiao Wang, Hanqing Lu

National Laboratory of Pattern Recognition,
Institute of Automation, Chinese Academy of Sciences,
Beijing, 100190, China.
{gbzhu, jqwang, luhq}@nlpr.ia.ac.cn

Abstract. A key problem in visual tracking is how to handle the ambiguity in decision to locate the object effectively using the target appearance model with online update. We address this problem by incorporating sequential clustering and ensemble methods into the tracking system. In this paper, clustering is used for mining the potential historical structure in the parameter space and feature space. Then we fuse multiple weak hypotheses to construct a strong ensemble learner for object tracking. Different from previous methods for updating classifier ensemble in a fixed weak classifier pool frame-to-frame, the proposed ensemble method is taking three weak hypotheses into consideration: spatial object-part view, parameter space view, and feature space view. Specially, spatial object-part view represents the object by a collection of part models that are spatially related (e.g. tree-structure). Meanwhile, analyzing the latent group structure in the parameter space and feature space is essential to take full advantage of the historical data in the tracking process. Therefore, we propose a novel ensemble algorithm that fuses object-part predictor, parameter clustered predictors and feature clustered predictors together. Furthermore, the weights of different views are updated by the relative consistency between weak predictors and final ensemble tracker. The formulation is tested in a tracking-by-detection implementation. Extensive comparing experiments on challenging video sequences demonstrate the robustness and effectiveness of the proposed method.

1 Introduction

Visual tracking has attracted significant attention due to its wide variety of applications such as terrorist detection, wearable computing and self-driving cars. Much progress has been made in the last two decades. However designing robust visual tracking methods is still an open issue. Challenges in visual tracking methods include no-rigid shape and appearance variations of the object, occlusions, illumination changes, cluttered scenes, etc [1], [2].

To solve the above problem, a popular approach is to learn a discriminative appearance model for coping with complicated appearance changes [3]. Typically, this assumes that the object/non-object discriminative information from different frames during long-term tracking is generated from a temporally homogeneous source. However this assumption may not hold in practice, as object

appearance and environmental conditions vary dynamically over time. In face of challenging factors, only fitting one updating discriminative model which can satisfy all cases is unlikely to optimally distinguish an object from its background through tracking-by-detection methods [4], [5], [6], [7], [8]. Tracking-by-detection requires training of a classifier for detecting the object in each frame. One common approach for detector training is to use a detector ensemble framework that linearly combines the weak classifiers with different associated weights, e.g., [4], [6]. A larger weight implies that the corresponding weak classifier is more discriminative and thus more useful.

Although most previous online ensemble methods originated from offline algorithms achieve many successes in online visual learning task, there are some limitations in visual tracking. As noted by Bai *et al.* [9], the common assumption was that the observed data (examples and their labels) had an unknown but stationary joint distribution. It may not apply in tracking scenarios where the appearance of an object can undergo significant changes. Due to the uncertainty in the appearance changes that may occur over time and the difficulty of estimating the non-stationary distribution of this observed data directly, they used Bayesian estimation theory to estimate a Dirichlet distribution of classifier weights. Different from their pre-defined non-stationary distribution and high computational complexity, we propose a simple and robust cumulative sum method to model how the different view predictor weights evolve so as to represent the non-stationary distribution which doesn't need to satisfy some specific distribution and is efficient.

At the same time, Grabner and Bischof [6] noted that updating the weights of online self-learning classifiers through the incoming data without annotation is difficult. Babenko *et al.* [5] treated tracking as multiple instance learning problem. Bai *et al.* [9] estimated the ensemble weights using Bayesian interpretation and ensures that the update of the ensemble weights is smooth. Yu *et al.* [10] proposed a co-training based approach to continuously label incoming data and online update a hybrid discriminative and generative model. We consider the three views of the object-part view, parameter space view and object feature space view at the same time. They are robust to different cases that object-part view covers the occlusion, discriminative parameter space view focuses the difference between the object and the background and the generative object feature space view handles the variants of the object appearance itself.

Moreover, the tracking problem has a temporal dimension which is not present in the classification methods [11] or subspace learning methods [12] by the previous works. We get temporal interval predictors through sequential clustering so as to better utilize the temporal learned structural information in parameter space and object appearance space directly.

Our method models three views of predictors whose weights ensemble with a non-stationary distribution, where their information geometry can be explored by sequential clustering methods. Our method focus on estimating the state of the object with three diverse view predictors in temporal dimension, not the

independent and identically distributed variable in a fixed weak classifier pool. In summary, our contributions are as follows:

1. We first propose a clustering ensemble tracker with three diverse views of weak predictors: object-part predictor, parameter space predictor, and feature space predictor. The different views have specific properties for tracking.

2. The sequential clustering is utilized to estimate the temporal non-stationary distributions of weak structure predictor in parameter space and appearance predictor in feature space. Based on sequential clustering theory, it provides a probabilistic interpretation of which interval structured predictor of the object are more discriminative.

3. We propose a simple weighting strategy to ensemble different weak predictors based on the prediction consistency between weak predictors and final ensemble tracker.

2 Related Work

A tracking-by-detection method usually has two major components: object representation and model update. Previous methods employ various object representations [13], [6], [5], [7], [14], [15], [8]. Our approach is most related to the methods that use structured prediction [7], [8].

From the perspective of that the tracked objects are treated as labeled positive samples and the other as training samples with some structure loss, the tracking problem can be considered as supervised learning problem in each frame. Supervised learning algorithms are commonly described as performing the task of searching through a hypothesis space to find a suitable hypothesis that makes good prediction for one particular problem. Even if the hypothesis space contains hypotheses that are very well-suited for object tracking, it may be very difficult to find a good one to locate the object precisely.

"Ensemble methods" is a machine learning paradigm where multiple (homogenous/heterogenous) individual learners are trained for the same problem, e.g., neural network ensemble [16], bootstrap aggregating (bagging) [17], boosting[18], Bayesian model averaging [19], [20], etc. Avidan [4], who was the first to explicitly apply ensemble methods to tracking-by-detection, extended the work of [21] by adopting the Adaboost algorithm [18] to combine a set of weak classifiers maintained with an online update strategy. Along this thread, Grabner *et al.* [6] inspired from the online boosting algorithm [22] by introducing feature selection from a pool of features for weak classifiers. Several other extensions to online boosting also existed, including the work by Banbenko *et al.* [5] who adopted Multiple Instance Learning in designing weak classifiers. In a different approach [23], Random Forests undergoed online update to grow or discard decision trees during tracking. Bai *et al.* [9] treated weight vector as a random variable and estimate a Dirichlet distribution for ensemble's weight vector. They all are a binary classifier realized by an ensemble method and don't exploit the structured data properties which can improve the tracking performance significantly, like as [7], [24]. At the same time, online boosting based trackers [6], [5] only considered

the parameter state in current time period. Different from them, we explore the structure of parameter state in parameter space over different time periods in tracking process.

Zhong *et al.*[25] considered visual tracking in a weakly supervised learning scenario where (possibly noisy) labels but no ground truth are provided by multiple imperfect oracles (i.e., trackers). Kwon and Lee [26] proposed visual tracker sampler to track a target by searching for the appropriate trackers in each frame. They are all ensemble methods applied in visual tracking. Unlike these methods, our method is not a heterogenous method which focuses on the tracker space but an homogenous approach which there is just one tracker. Due to the trained weak trackers in historical sequences, our method is more efficient than heterogenous methods.

Our online ensemble method is most related with online bagging scheme, in the sense that we adopt random combination of weak classifiers. However, we characterize the temporal ensemble weight vector as a clustering center and evolve its distribution with sequential clustering manner. As a result, the final strong classifier is an expectation of the ensemble with respect to the weight vector, which is approximated by an average of the ensemble clustering centers. To the best of our knowledge, in the context of tracking-by-detection, we are the first to present such an online learning scheme that adopt clustering in parameter space and object appearance space to characterizes the uncertainty of a self-learning algorithm.

3 Clustering Ensemble Tracking

In this section, we introduce our tracking algorithm, clustering ensemble tracking (CET), which is a clustering ensemble based appearance model. We begin with an overview of our tracking system which includes a description of structure learning-based part models predictor. We then briefly review the concepts of sequential clustering and ensemble with temporal weak structure predictors. Finally, we give our clustering ensemble based tracking algorithm.

3.1 Overview

We illustrate the framework of our tracking system (diagram shown in Fig. 1). At each frame, our method starts with a structure predictor $h(x)$, several clustering centers based on historical weight vectors $W = \{w_1, w_2, ..w_N, ...\}$ of $h(x)$ and input data x. Our method obtains the incremental parameter cluster centers $C_p = \{C_{p,1}, ..., C_{p,M}\}$ and object appearance cluster centers $C_o = \{C_{o,1}, ..., C_{o,M}\}$ through sequential clustering method, where there is only one cluster, and then the number of clusters increases as the change of the input parameter vectors W or object feature vectors $O = \{o_1, ...o_N...\}$. Every parameter cluster center $C_{p,i}$ and the latest parameter vector w_N are treated as the parameters of weak structure predictors $h(x)$. Meanwhile, each appearance cluster center $C_{o,i}$ evaluates the object candidates through similarity measurement. Then the output of

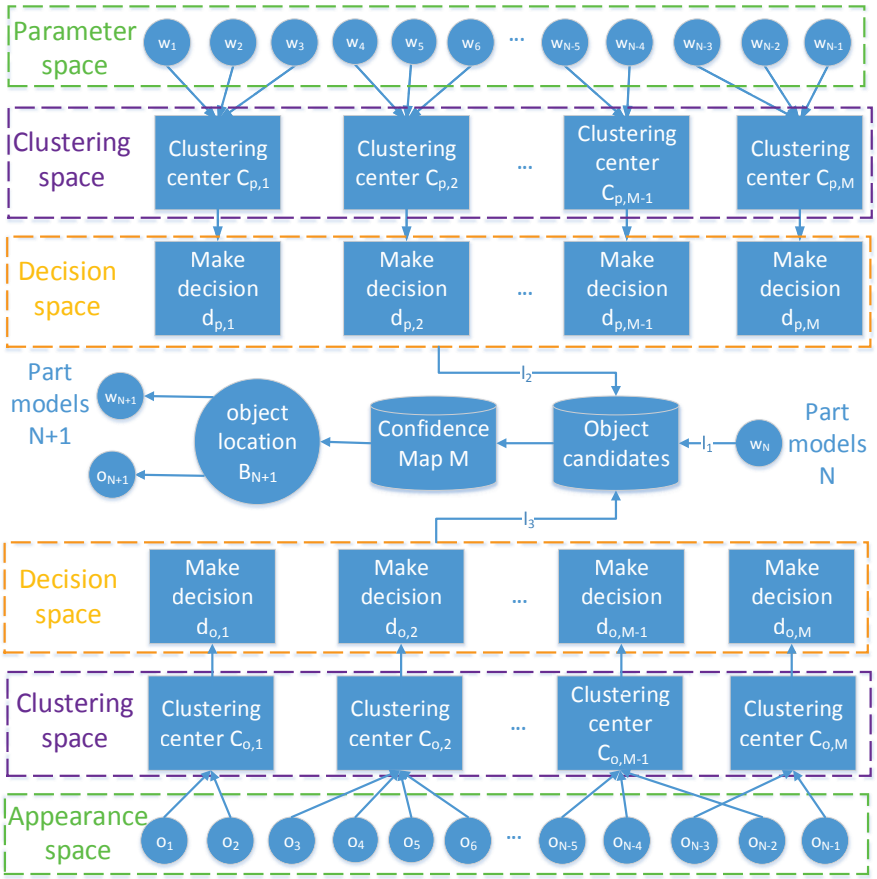

Fig. 1. Framework of the proposed clustering ensemble algorithm. W_i represents the parameter of part models in i^{th} frame. $C_{i,j}$ denotes clustering center. $d_{i,j}$ expresses the decisions related to $C_{i,j}$.

these weak structure predictors $h(x)$ and the degree of similarity with respective weights $l = \{l_1, l_2, l_3\}$ are combined to yield the final decision where the object is. For reducing the computing complexity, the cascade method are adopted in experiments. The cascade is that using the most stable weak classifier or the latest classifier rejects most of object candidates and retains a small number of object candidates which are difficult to predict precisely by one weak classifier so that multiple weak predictors give a combined solution of higher quality than any individual solution (empirically proved by [25], [26]).

3.2 Sequential Clustering

In online visual object tracking, the tracked object appearance usually changes gradually. While there are some various factors such as noise or occlusion or

fast and abrupt object motion or illumination changes or variations in pose and scale, the object appearance got from the object location will changes much. Meanwhile, the weight vector trained through the changed object training samples varies with the changes of object appearance. Through updating object appearance model, the classifier can adapt the variation of the object appearance. However, model update itself is not absolutely correct without effective supervised information. For alleviating the drift problem resulted by degraded classifier update which comes from incorrectly labeled training samples, we exploit the structure of the parameter space of the trained weak trackers and the predicted object appearance space in historical temporal dimension to guarantee the accuracy of current decision by the final ensemble tracker through sequential clustering. We will introduce the sequential clustering algorithm as follows.

In basic form, parameter or weight vectors $W = \{w_1, ..., w_n\}$ are presented only once and the number of clusters $C = \{C_1, ..., C_m\}$ is not known a priori. The common approach is to define the dissimilarity $d(x_i, C_j)$ and set the threshold of dissimilarity Θ and the number of maximum clusters allowed q. The idea is to assign every newly presented vector to an existing cluster or create a new cluster for this sample, depending on the distance to the already defined clusters. In the application of online tracking, the parameter vector changes gradually so that the threshold Θ and the number q are difficult to set. Here, to avoid the setting problem above, we create a new cluster using a simple heuristic. As pseudo, the algorithm works like the following:

Algorithm 1 Sequential clustering

```
Init the first sample as the first cluster C_m = {w_1}, m = 1;
for each w_i ∈ {w_2, ..., w_n} do
    find the cluster C_k such that min d(w_i, C_k);
    if i mod D == 0 then
        Create a new cluster C_m = {w_i}, m = m + 1;
        Using K-means clustering algorithm to re-clustering the space of samples w,
        K = m + 1
    else
        Add the sample w_i to the nearest cluster C_k = {C_k, w_i}, while the predicted
        object satisfied some update condition.
    end if
end for
```

As can be seen the algorithm is simple but still quite efficient. Different choices for the distance function $d(w_i, C_k)$ lead to different results. We define:

$$d(w_i, C_k) = 1 - \frac{< w_i, C_{k,c} >}{||w_i||||C_{k,c}||} \tag{1}$$

where $< A, B >= \sum_{i=1}^{n} A_i \times B_i$ is the dot product of two vectors, $||A|| = \sqrt{\sum_{i=1}^{n}(A_i)^2}$, and $C_{k,c}$ is the average of all vectors in the set C_k. Due to structured time series property of online tracking, our method creates one new cluster

after each D interval frames and uses K-means [27] to re-clustering. The sequential clustering is used in section 3.3.

3.3 Clustering Ensemble Tracker

We adopt the bagging-like method to get the final ensemble results. Bagging predictors is a method for generating multiple versions of a predictor and using these to get an aggregated predictor. The aggregation averages over the versions when predicting a numerical outcome and does a plurality vote when predicting a class. The multiple versions are formed by making bootstrap replicates of the learning set and using these as new learning sets. Here, we use the trained structure predictor in every frame as the basic version of a predictor.

Object-Part Predictor. In our paper, similar to [24], a structured part models predictor is trained by an online manner based on the tracked object locations in previous frames. We represent the object bounding box $B_i = \{\mathbf{x}_i, w_i, h_i\}$ with center location $\mathbf{x}_i = (x_i, y_i)$, width w_i and height h_i. The HOG features extracted from image $\mathbf{I}$ that correspond to locations inside the object bounding box B_i are extracted to obtain feature vector $\Phi(\mathbf{I}; B_i)$. The part indicators $i \in V$ where $V = \{V_0, V_1, ..., V_n\}$ represents the set of object and object parts. Here, V_0 denotes the object itself. Subsequently, we define a graph $G = (V, E)$ over all objects $m \in V$ that we want to track with edges $(m, n) \in E$ between the objects. The edges in the graph model can be viewed as springs that represent spatial constraints between the tracked objects. Next, we define the score of a configuration $S = \{P_1, ..., P_{|V|}\}$ of multiple tracked parts as the sum of two terms: (1) an appearance score that sums the similarities between the observed image features and the classifier weights for all objects and (2) a deformation score that measures how much a configuration compresses or stretches the springs between the tracked objects. Different from [8], the weak base predictor is not our focus, but just part of our method. Mathematically, the score of a configuration S_b is defined as:

$$S_b = \sum_{i \in V} \mathbf{w_i^T} \Phi(\mathbf{I}; B_i) + \lambda \sum_{(m,n) \in E} ||(\mathbf{x_m} - \mathbf{x_n}) - \mathbf{e_{mn}}||^2. \tag{2}$$

Where the parameters $\mathbf{w}_i$ represent linear weights on the HOG features for object i, $\mathbf{e}_{ij}$ is the vector that represents the length and direction of the spring between objects i and j, the set of all parameters is denoted by $\Theta = \{\mathbf{w_1}, ..., \mathbf{w}_{|\mathbf{V}|}, \mathbf{e_1}, ..., \mathbf{e}_{|\mathbf{E}|}\}$. We treat the parameter λ as a hyper-parameter that determines the trade-off between the appearance and deformation scores. For reducing the computing complexity, we set $m = 0$, which means only to compute the distance between the parts V_i and the root V_0 in $D(x)$. We use a passive-aggressive algorithm to perform the parameter update[28],[24].

Parameter and Feature Clustered Predictor. In this paper, we redefine the goal of tracking problem as to find the best state that not only using the current trained classifier in the case where the object is easy to identify (see Fig. 2(a)), but also exploiting the historical trained classifier through clustering

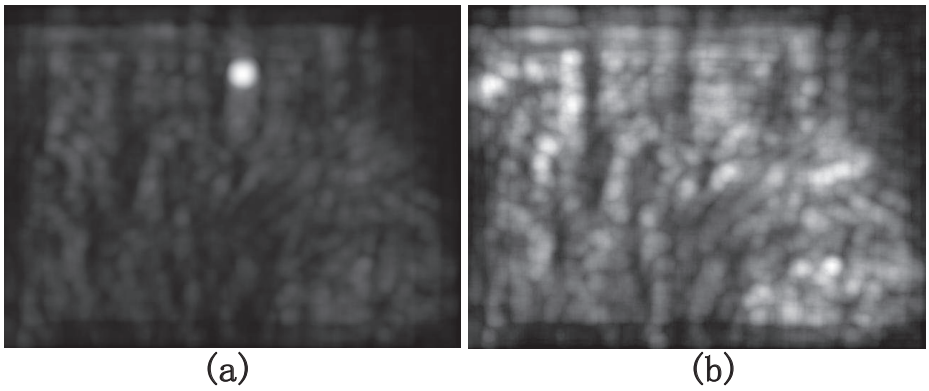

Fig. 2. Two confidence maps to decide where object is. The lighter, the more likely the object is.

ensemble methods in the case where the object is difficult to identify (see Fig. 2(b)). In Fig. 2(a)), the object is easy to decide because other regions' confidences are much lower than the lighter region so that the object is discriminated easier from the background. In Fig. 2(b), the background has many regions in which there are similar confidences as the object so that if the current trained classifier's decision is wrong, the tracker will drift to the background. After drift, the classifier's update will be wrong. For reducing the decision ambiguities of the object, we adopt the clustering ensemble method (see sec. 3.2) in the historical parameter space and object appearance space and use the clustering centers to make a decision where the object is. To improve the computational efficiency and robustness, we get the extremal points in the confidence map as the object candidates. After getting the object candidates, we use the clustering centers as weak classifiers to vote the best state.

Each cluster center is treated as a sub-weak clustered predictor in ether discriminative parameter space or generative object appearance space. The score of one object candidate B_c based on the predictor in parameter space can be computed:

$$S_p(B_c) = \sum_{i=1}^{N_p} C_{p,i}^T \Phi(\mathbf{I}; B_c) \tag{3}$$

where N_p is the total number of clusters in parameter space by the end of the current frame, and $C_{p,i}$ is the representation of the i^{th} cluster center in parameter space. The score of one object candidate using the j^{th} predictor in object parameter space can be mathematically expressed:

$$S_o^j(B_c) = \rho(Q(B_C), C_{o,j})), \tag{4}$$

where ρ is euclidean metric function, $Q(B_C)$ is object representation directly extracted from the object candidate bounding box, $C_{o,j}$ is the j^{th} clustering

center in object feature space. In our experiment, $Q(B_C)$ is a vectorization after resizing the B_C to its quarter. The same is to extract feature in object space.

According to the Eq. (2-4), then the final object candidate's score is:

$$S = \lambda_1 S_b + \lambda_2 S_p + \lambda_3 S_o, \lambda_1 + \lambda_2 + \lambda_3 = 1, \tag{5}$$

where $\{S_b, S_p, S_o\}$ are the scores of weak part models predictor, weak parameter predictor and weak object appearance predictor and $\{\lambda_1, \lambda_2, \lambda_3\}$ are their weights respectively. How to learn the λ is introduced in next section. The final object location is inferred based on Eq. (5):

$$B^* = \arg\max_{B_c} S(B_c), \tag{6}$$

where the B_c is object bounding box candidates sampled from the search region near the previous object location.

3.4 Weight Update

Our model updates the weights of three different predictors after the decision stage in each step, not each frame which doesn't satisfy the update condition (e.g. heavily occluded), so that the model can evolve. For each step, after performing the decision, our method obtains the labels of data predicted by our strong predictor and the observation of performance of weak view predictors, that is, the prediction consistency of weak classifiers with respect to the strong classifier, likely to [9], [29].

The weight distribution is dependent on the accumulative normalized central-pixel error probability. The accumulative property reflects on the cumulative sum of observation of relative reliability of each predictor. The normalized central-pixel error probability is incarnated by normalized probability directly related to the distance between the object's center and weak predictor observations' centers. Mathematically, we have

$$p(o_i^t|x^t) = \frac{1}{Z_t} exp(-(o_i^t - x^t)^2/\sigma^2), \tag{7}$$

$$Z_t = \sum_i^n p(o_i^t|x^t) \tag{8}$$

where o_i^t is the observation state center location of the i^{th} weak predictor in step t, x^t is the predicted object's center location, Z_t is a normalization factor in each step t, and $\sigma = 25$. Each part weight is defined as:

$$\lambda_i = \frac{\sum_{t=2}^T p(o_i^t|x^t)}{\sum_{t=2}^T Z_t} \tag{9}$$

which computes relative reliability of each part predictor.

4 Experiments

For the experiments, publicly available video sequences obtained from [11], [5], [30], [31] were utilized. Using the sequences, the proposed method (CET) was analyzed and compared with 7 state-of-the-art tracking methods: Multiple Instance Learning (MIL) [5], Visual tracking decomposition [30], Struck [7], Tracking-Learning-Detection (TLD) [32], PartTracker(PT) [33], Structure preserving object tracking(SPOT) [24], Randomized Ensemble Tracking(RET) [9]. All algorithms are compared in terms of the same initial positions in first frame in [31].

4.1 Implement Details

In all of the experiments, the parameters of our trackers are fixed. The experimental results of MIL, VTD, Struck, TLD are dependent on the public dataset where the sequences' ground truth are re-annotated by Wu *et al.* [31] and some trackers' results through the third party appraisal are attached. For fairness, we adopt the other tracker codes provided by the respective authors in their homepages. The binary code of PT is public. We just need to prepare a config file and then can get their results. The source code of SPOT is published in the website of zhang and van der Maaten [24]. There is one limitation in SPOT is that the parts' initialization for single object tracking is missing in their source code because it is mainly designed for multiple object tracking. We want to use it as our base tracker so that it is necessary to initialize the parts. For handleability and robustness, we divide the object into four parts equally and then complete the part initialization. The source code of RET is also provided by its authors. MIL and TLD use the haar-like feature[34] or LBP-like feature which is sensitive to large illumination illumination, while Struck, VTD, PT, SPOT, RET and CET are based on edge information or HOG feature [35] that is robust to illumination and mirror misalignment. We use the given parameter in their code and get the sequences' results. In our method, one cluster is initialized newly in every $D = 100$ frames. The time complexity is mainly determined by the number of parts, the clustering computation complexity, feature extraction and the search region for deciding where object is.

4.2 Quantitative Analysis

The quantitative comparison results with several state-of-the-art trackers and our tracker (CET) are shown in Fig. 3 and Table 1. We follow the same evaluation protocol proposed in [31]. Overall, our method outperforms them consistently in the view of overall performance (see Fig. 3). In addition, Fig. 4 shows the comparison on different subsets such as occlusion and illumination subsets. The quantitative results are shown in Table 1. From the table, CET achieves the competitive performances well against the other state-of-the-art algorithms on all tested sequences. As summarized in Table 1, our method (CET) most accurately tracked the targets in terms of the center location error and the success rate, even though there are several types of appearance changes.

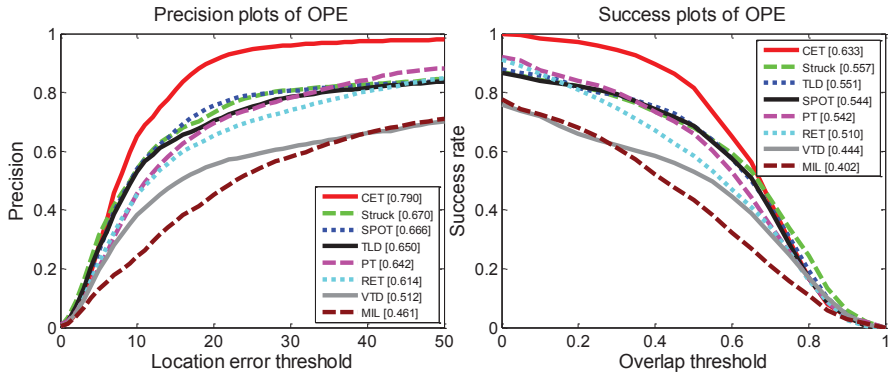

Fig. 3. Plots of overall performance comparison for the 22 videos in the benchmark [31]. The proposed method ("CET")obtain better performance in terms of precision (left) and success (right) plot

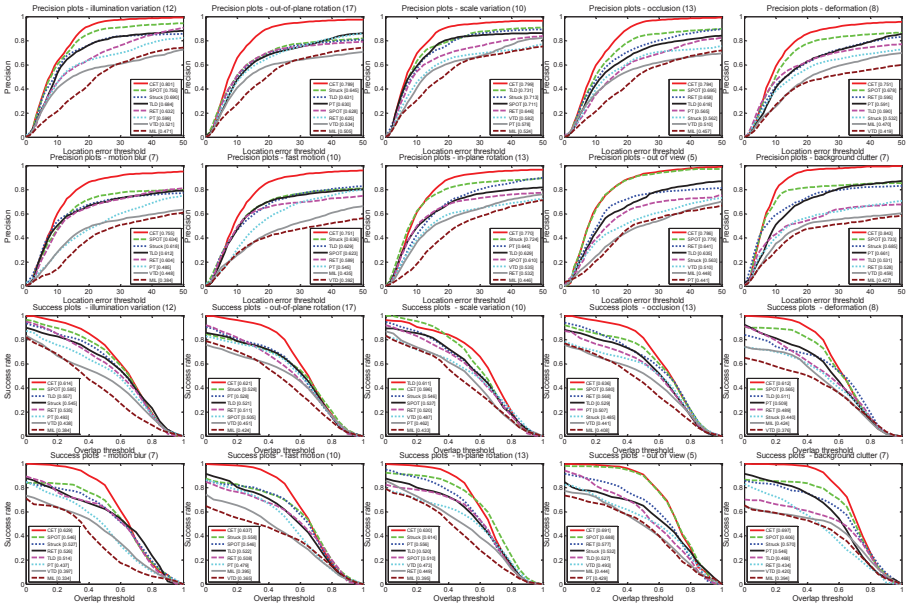

Fig. 4. Several comparisons in different subsets divided based on main variation of the object to be tracked. The details of the subsets refer to [31]. The proposed method ("CET") obtains better or comparable performance in all the subsets

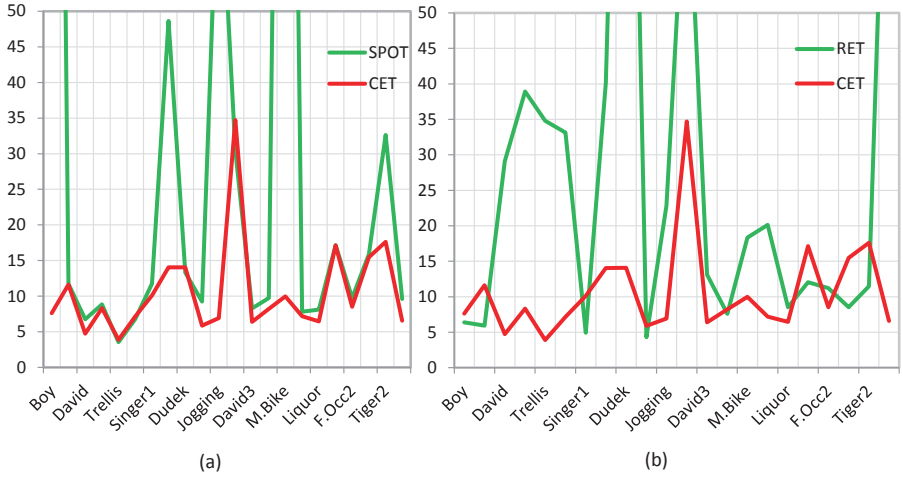

Fig. 5. Center location errors comparing CET with SPOT and RET. (a) represents the comparison between CET and its base tracker SPOT; (b) denotes the comparison between CET and the latest state-of-the-art ensemble tracker RET

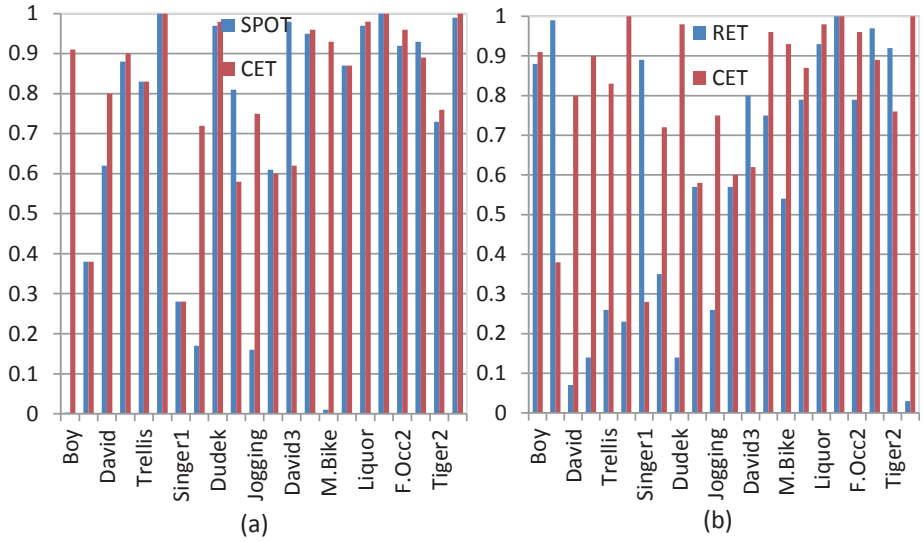

Fig. 6. Success rate based on overlap rate comparing CET with SPOT and RET. (a) represents the comparison between CET and its base tracker SPOT; (b) denotes the comparison between CET and the latest state-of-the-art ensemble tracker RET

Table 1. Comparison of tracking results. The numbers indicate the average center location errors in pixels. The bold, underlined, and italic represent the best, the second, and the third best, respectively. Other numbers in () indicate the percent of successfully tracked frames, where tracking is success when the overlap ratio between the predicted bounding box A_p and ground truth bounding box A_g is over than 0.5:$\frac{A_p \cap A_g}{A_p \cup A_g} > 0.5$.

	TLD[32]	MIL[5]	VTD[30]	Struck[7]	PT[33]	SPOT[24]	RET[9]	CET
Boy	5(94)	13(39)	8(79)	**4**(**98**)	8(78)	238(0.3)	*6*(*88*)	8(*91*)
Car4	13(79)	51(28)	37(35)	*9*(*40*)	8(*40*)	12(38)	**6**(**99**)	12(38)
David	**5**(**97**)	24(16)	12(68)	10(57)	47(71)	*7*(*62*)	29(7)	**5**(80)
Sylv.	7(93)	12(74)	20(80)	**6**(93)	**6**(**95**)	9(88)	39(14)	*8*(*90*)
Fish	31(47)	72(24)	32(50)	7(78)	*8*(80)	**4**(**83**)	35(26)	**4**(**83**)
Trellis	*7*(*96*)	27(23)	17(64)	**3**(**100**)	6(**100**)	*7*(**100**)	33(23)	*7*(**100**)
Singer1	*8*(**99**)	16(28)	**4**(*43*)	15(30)	31(22)	12(28)	5(89)	10(28)
Coke	25(29)	70(3)	69(14)	**12**(**94**)	*15*(*71*)	49(17)	40(35)	14(72)
Dudek	18(84)	18(86)	**10**(**100**)	12(98)	15(94)	*13*(*97*)	140(14)	14(98)
Couple	**3**(**100**)	35(67)	104(8)	11(54)	21(36)	9(81)	4(57)	*6*(*58*)
Jogging	**7**(**97**)	95(23)	83(22)	62(23)	**7**(88)	75(16)	23(26)	**7**(*75*)
F.Face	41(57)	63(54)	46(*71*)	23(67)	**22**(**86**)	*31*(61)	75(57)	35(60)
David3	208(10)	30(68)	67(48)	107(34)	7(89)	*8*(**98**)	13(*80*)	**6**(62)
Suv	13(*84*)	82(13)	57(55)	50(58)	35(53)	*10*(95)	**8**(7)	8(**96**)
M.Bike	216(26)	73(58)	*10*(**100**)	**9**(**86**)	**9**(**100**)	198(1)	18(54)	10(93)
Lem.	*16*(59)	171(17)	79(49)	38(*64*)	136(45)	8(**87**)	20(79)	**7**(**87**)
Liquor	38(58)	142(20)	60(58)	91(41)	95(34)	8(97)	*9*(*93*)	**7**(**98**)
F.Occ1	27(83)	37(62)	20(93)	*19*(100)	17(100)	17(100)	**12**(**100**)	17(**100**)
F.Occ2	12(83)	20(68)	8(99)	**6**(**100**)	**6**(**100**)	10(92)	11(79)	9(*96*)
Tiger1	50(46)	37(37)	107(12)	129(18)	33(49)	16(93)	**9**(**97**)	16(*89*)
Tiger2	37(18)	30(36)	41(17)	*22*(*65*)	48(29)	33(73)	**12**(**92**)	18(76)
Deer	31(*73*)	101(13)	135(4)	**5**(**100**)	24(38)	*10*(99)	97(3)	7(**100**)

Comparison of Competing Tracking Algorithms. Although SPOT is our base tracker, we can get better performance in most video sequences through introducing the hidden clustering information by sequential clustering method (see Fig. 5(a) and Fig. 6(a)). RET exploits the non-stationary distribution of weight vector in parameter space to ensemble and get good performance. Our tracker CET adopts the sequential clustering method to utilize the hidden non-stationary distribution of parameter and object appearance. Through Fig. 5(b) and Fig. 6(b), we also get the better performance comparing with RET. We compare the proposed tracking algorithm with nine state-of-the-art tracking algorithms, Table 1 summarises the average center location error performance and success rate of the compared tracking algorithms over the 22 sequences. From the experimental results, we see that our tracking algorithm obtains the best performance on ten sequences in the terms of the center location error or the success rate, seven sequences the second best, four sequences the third best. Fig. 4 shows that our method can handle occlusion, illumination and out-of-view well. The robustness of our CET tracker lies in the object-part structure which are

discriminatively trained online to account for the variations, the historical hidden structure information in parameter space of base tracker and in the object appearance space of the historical predicted object.

5 Conclusion

In this paper, we deal with the tracking problem about decision ambiguities by fusing object-part predictor, parameter clustered predictor and feature clustered predictor together. Object-part predictor exploits the structure between object and its parts which is effective to object deformative appearance changes. Parameter clustered predictor utilizes temporal hidden group structure in object parameter space in some extent. Feature clustered predictor guarantees the object from the distracters in parameter space and get the better performance. Then we propose a tracker, clustering ensemble tracking (CET), based on structure learning and sequential clustering framework to avoid the drifting problem. Extensive experiments show that our algorithm is robust to occlusion, illumination and out-of-view because different predictors have different properties. The accuracy of CET is superior or competitive to several state-of-the-art tracking algorithms in a more effective way.

Acknowledgement. This work was supported by 863 Program (2014AA015104), and National Natural Science Foundation of China (61273034 and 61332016).

References

1. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. CSUR **38** (2006) 13
2. Smeulders, A.W.M., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. IEEE-TPAMI **99** (2013) 1
3. Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., Hengel, A.: A survey of appearance models in visual object tracking. IEEE-TIST **4** (2013)
4. Avidan, S.: Ensemble tracking. IEEE-TPAMI **29** (2007) 261–271
5. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR, IEEE (2009) 983–990
6. Grabner, H., Bischof, H.: On-line boosting and vision. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Volume 1., IEEE (2006) 260–267
7. Hare, S., Saffari, A., Torr, P.: Struck: Structured output tracking with kernels. In: ICCV, IEEE (2011) 263–270
8. Zhang, L., van der Maaten, L.: Structure preserving object tracking. In: CVPR, IEEE (2013) 1838–1845
9. Bai, Q., Wu, Z., Sclaroff, S., Betke, M., Monnier, C.: Randomized ensemble tracking. In: ICCV, IEEE (2013)
10. Yu, Q., Dinh, T., Medioni, G.: Online tracking and reacquisition using co-trained generative and discriminative trackers. In: ECCV. Springer (2008) 678–691
11. Ross, D., Lim, J., Lin, R., Yang, M.H.: Incremental learning for robust visual tracking. IJCV **77** (2008) 125–141

12. Mei, X., Ling, H.: Robust visual tracking and vehicle classification via sparse representation. IEEE-TPAMI **33** (2011) 2259–2272
13. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: CVPR. Volume 1., IEEE (2006) 798–805
14. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Robust visual tracking via multi-task sparse learning. In: CVPR, IEEE (2012) 2042–2049
15. Zhang, K., Zhang, L., Yang, M.H.: Real-time compressive tracking. In: ECCV-Computer Vision. Springer (2012) 864–877
16. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE-TPAMI **12** (1990) 993–1001
17. Breiman, L.: Bagging predictors. Machine Learning **24** (1997) 123–140
18. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Jouranl of Computer and System Sciences **55** (1997) 119–139
19. Jennifer, A., David, M., Adrian, E., Chris, T.: Bayesian model averaging: A tutorial. Statistical Science **14** (1999) 382–417
20. Hong, S., Kwak, S., Han, B.: Orderless tracking through model-averaged posterior estimation. In: ICCV, IEEE (2013)
21. Avidan, S.: Support vector tracking. IEEE-TPAMI **26** (2004) 1064–1072
22. Oza, N.C.: Online bagging and boosting. In: Systems, man and cybernetics, 2005 IEEE international conference on. Volume 3., IEEE (2005) 2340–2345
23. Saffari, A., Leistner, C., Santner, J., Godec, M., Bischof, H.: On-line random forests. In: ICCVW, IEEE (2009) 1393–1400
24. Zhang, L., van der Maaten, L.: Preserving structure in model-free tracking. IEEE-TPAMI **36** (2014) 756–769
25. Zhong, B., Yao, H., Chen, S., Ji, R., Ji, X., Yuan, X., Liu, S., Gao, W.: Visual tracking via weakly supervised learning from multiple imperfect oracles. In: CVPR, IEEE (2010) 1323–1330
26. Kwon, J., Lee, K.: Tracking by sampling and integrating multiple trackers. IEEE-TPAMI **pp** (2013) 1
27. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Volume 1., California, USA (1967) 281–297
28. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. JMLR **7** (2006) 551–585
29. Wang, N., Yeung, D.: Ensemble-based tracking: Aggregating crowdsourced structured time series data. In: ICML, JMLR. org (2014)
30. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: CVPR, IEEE (2010) 1269–1276
31. Wu, Y., Lim, J., Yang, M.H.: Online object tracking: A benchmark. In: CVPR, IEEE (2013) 2411–2418
32. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE-TPAMI **34** (2012) 1409–1422
33. Yao, R., Shi, Q., Shen, C., Zhang, Y., van den Hengel, A.: Part-based visual tracking with online latent structural learning. In: CVPR. (2013)
34. Paul, V., Michael, J.: Rapid object detection using a boosted cascade of simple features. In: CVPR. Volume 1., IEEE (2001) I–511
35. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. Volume 1., IEEE (2005) 886–893