

Robust Object Tracking with a Hierarchical Ensemble Framework

Mengmeng Wang¹, Yong Liu² and Rong Xiong²

Abstract—Autonomous robots enjoy a wide popularity nowadays and have been applied in many applications, such as home security, entertainment, delivery, navigation and guidance. It is vital for robots to track objects accurately in real time in these applications, so it is necessary to focus on tracking algorithms to improve the robustness, speed and accuracy. In this paper, we propose a real-time robust object tracking algorithm based on a hierarchical ensemble framework which incorporates information including individual pixel features, local patches and holistic target models. The framework combines multiple ensemble models simultaneously instead of using a single ensemble model individually. A discriminative model which accounts for the matching degree of local patches is adopted via a bottom ensemble layer, and a generative model which exploits holistic templates is used to search for the object based on the middle ensemble layer as well as an adaptive Kalman filter. We test the proposed tracker on challenging benchmark image sequences. The experimental results demonstrate that the proposed tracker performs superiorly against several state-of-the-art algorithms, especially when the appearance changes dramatically and the occlusions occur.

I. INTRODUCTION

Visual tracking is a well-studied problem in computer vision with a variety of applications such as surveillance, human motion analysis, robot guidance, human-computer interaction and so on. Recent attention has been focused to visual tracking in the robotic domains [1], [2]. However, due to the diverse environment and the complex motion of the robots, several tracking conditions such as occlusions, deformations, fast motion and background clutters remain difficult.

There are three fundamental tracking components that are essential [3] for improving performance of tracking: (1) the background information; (2) local appearance models; (3) motion models. This paper presents a hierarchical tracking framework which takes the above components into account. We model the object as an ensemble three-layer structure which can incorporate information including individual pixel features, the local patches and the target bounding box. The first component, i.e. the background information, is essential to overcome the background clutters due to the complexity of the environment. In our proposed method, we

*This work was supported in part by the National Natural Science Foundation Project of China under Project 61173123, in part by the Natural Science Foundation Project of Zhejiang Province under Project LR13F030003, and in part by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China, under Project ICT1502.

¹Mengmeng Wang is with the Institute of Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027, China.

²Yong Liu and Rong Xiong are with the State Key Laboratory of Industrial Control Technology and Institute of Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027, China (Yong Liu is the corresponding author of this paper, email: yongliu@ipc.zju.edu.cn).

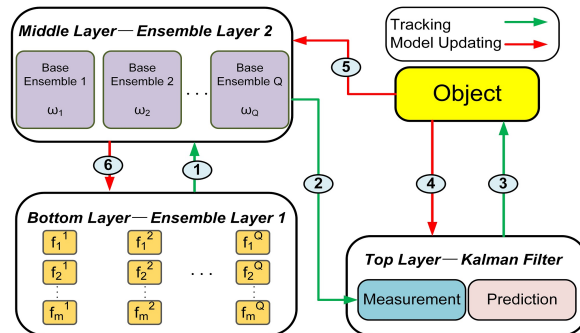


Fig. 1: An overview of the architecture of the layers. 1-combine the weak classifiers of each sub-patch to obtain the corresponding base ensembles and weights in the bottom layer; 2-combine the base ensembles to generate the measurement of the object in the middle layer; 3-employ an adaptive Kalman filter to increase the time consistency in the top layer; 4-update the top layer; 5-re-extract the sub-patches and update their weights in the middle layer; 6-update the parameters of weak classifiers in the bottom layer.

incorporate both the object and the background information into classifiers. For the second component, most of existing approaches [4], [5], which represent the target with a limited number of non-overlapping or regular local regions. So they may not cope well with the large deformations of the target. While our hierarchical tracker models the target with a series of overlapping and randomly sampled regions. We introduce the compressive sensing theory [6], [7] which significantly reduces the dimension of the pixel features in local regions. An overall schematic for the tracker is shown in Fig.1. For each sub-patch, we build a bottom ensemble layer which combines a collection of weak classifiers on the compressive features for the sub-patch into a strong classifier as a base ensemble. In the middle ensemble layer, we aggregate these base ensembles to generate the measurement of the target. As the robots move almost all the time when tracking an object, our approach needs to consider the third component and introduce an adaptive Kalman filter [8] in the top layer to consider the motion models and the temporal consistency in the target bounding box level. Above all, the contributions of our method are summarized as follows:

- 1) We legitimately organize compressive features, overlapping sub-patches and holistic target models to capture the detailed appearance of the object;
- 2) We propose a hierarchical ensemble framework that combines multiple ensemble models simultaneously

instead of using a single ensemble model individually;

- 3) We employ compressive sensing method to significantly reduce the feature dimensions so that our approach can handle colorful images without suffering from exponential memory explosion;
- 4) We take the motion model into consideration to overcome the temporary occlusions, missing and false detections with an adaptive Kalman filter.

In the experiment, we compare the proposed method against state-of-the-art tracking approaches which are feasible for robotic applications in terms of computational complexity and hardware requirements using an online object tracking benchmark [3]. Our method obtains superior results compared with the state-of-the-art tracking approaches. The results also show that our method performs much better in the moving human tracking than other approaches for the conditions with occlusions, deformations, background clutters and scale variations.

II. RELATED WORK

Recent tracking algorithms are developed in terms of three primary components: target representation, matching mechanism, and model update mechanism.

Target representation plays a pivotal role in visual tracking, and numerous representation schemes have been proposed. Several factors need to be considered for an effective appearance model in target representation. First, the features to represent the objects have many choices such as color histogram [9], superpixels [10], Haar-like features [11]–[13], etc. Second, the templates to represent the objects can be global or local. Global templates [1], [12] are easy to construct the object representation that contains information of the whole object. However, for the tracking problem of robots, holistic templates will have difficulty in handling significant appearance changes and deformations of the targets. While local templates [4], [14], [15] are more robust and flexible to these conditions. But the geometrical relationships for local patches remain tough since the environmental clutter, occlusions and partially similar objects can often distract such local patches and lead to drift.

Matching mechanism is used to classify candidate regions which are most similar to the target from background. There are two main streams of research on this: One is generative model which typically searches for the most similar candidate to the target within a neighborhood [16]–[18]. Another is discriminative model which poses the tracking problem as a binary classification task that determines the decision boundary for separating the target from the background [12], [13], [19], [20].

Online model update mechanism is quite essential for robust visual tracking to deal with appearance variations. Addressing on this problem, Kalal et al. [15] develop a bootstrapping classifier to select positive and negative samples for model update. Grabner et al. [21] formulate the update problem as a semi-supervised task where the classifier is updated with both labeled and unlabeled data. However, online boosting requires that the data should be independent and

identically distributed. This is not always satisfied in visual tracking because the data are often temporally correlated.

In the proposed method, we adopt the compressive sensing theory to reduce the dimension of Haar-like features and this process is operated similarly to [12]. We employ a joint representation which considers both global and local models of the target to better handle significant appearance changes, deformations, similar object identification and occlusions. Our local models are efficiently constructed with a number of overlapping and randomly sampled local patches and we re-extract the sub-patches at each time step to avoid the drifting caused by arbitrary sub-patch. We adopt a discriminative model via the bottom ensemble layer to account for the matching degree of local patches, and a generative model is used to seek for the object through the middle ensemble layer as well as an adaptive Kalman filter. For model update, we employ ensemble learning to update the patches and classifiers to capture appearance variations and reduce tracking drifts.

III. ROBUST OBJECT TRACKING WITH A HIERARCHICAL ENSEMBLE FRAMEWORK

In this section, we give a detailed description of the proposed hierarchical ensemble tracking (HET) framework. It is composed of two ensemble layers and a Kalman filter layer. At each time step, we start with detecting several samples around each local patch and try to formulate the corresponding base ensemble for each sub-patch with several weak classifiers in the bottom ensemble layer. Second, we recover the target location in the middle ensemble layer by incorporating these base ensembles, and regard this location as the measurement to an adaptive Kalman filter. Third, we ascertain the ultimate object location at the current frame with a motion model and the measurement via the adaptive Kalman filter in the top layer. Finally, we update the model by re-extracting the local overlapping image sub-patches efficiently in the final target region with a random spatial layout and updating the parameters of weak classifiers for tracking in the next frame.

A. Local Compressive Appearance Model

The compressive sensing theory shows that if the dimension of the feature space is sufficiently high, these features can be projected to a randomly chosen low dimensional space which contains enough information to preserve most of the salient information of the original high-dimensional features through a random projection matrix [22]. The signal can be recovered as long as the projection matrix \mathbf{R} follows the Restricted Isometry Property (RIP) [7]. Representing the object appearance by regions allows the proposed tracker to better handle occlusions and large appearance changes. The compressive appearance model also allows us to process a large number of regions in real-time.

In this paper, we build compressively sensed versions of sub-patches. Randomly extracted sub-patches are used and the relative location between sub-patches and the target bounding box are established when the tracking window is

given by a detector or manual label at the first frame. Every sub-patch is represented by four components: a compressive feature vector \mathbf{g}^q , a classification score c_q , a relative location $\Delta\mathbf{p}_q$, where $\Delta\mathbf{p}_q = [\Delta x_q, \Delta y_q]^T$ denotes the relative upper-left corner coordinate to upper-left corner of the target window, and the location of the sub-patch itself in the image space $\mathbf{p}_q = [x_q, y_q]^T$. Denoted q -th sub-patch λ_q as:

$$\lambda_q = \langle \mathbf{g}^q, c_q, \Delta\mathbf{p}_q, \mathbf{p}_q \rangle. \quad (1)$$

It is notable that the width and the height of each sub-patch are identical, denoted as w and h , which are determined at beginning. After extracting these Q local overlapping image sub-patches $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_Q\}$, where Q denotes the number of sub-patches, for the q -th sub-patch, we sample N sub-patches with the same size as the q -th sub-patch, whose Euclidean distances to the sub-patch is smaller than a threshold β that is fixed through the sequence. These samples can form a matrix $\mathbf{S}^q = [\mathbf{S}_1^q, \mathbf{S}_2^q, \dots, \mathbf{S}_N^q] \in \mathbb{R}^{w \times hN}$. Then we present all samples as $\mathbf{S} = [\mathbf{S}^1, \mathbf{S}^2, \dots, \mathbf{S}^Q] \in \mathbb{R}^{w \times hNQ}$.

In order to find a kind of feature that is invariant to scale, we adopt a multiscale image representation that is often formed by convolving the input image with a Gaussian filter of different spatial variances and speed up the process via integral image method. We replace the Gaussian filter with rectangle filters for computation consideration [12]. For N samples of the q -th sub-patch \mathbf{S}^q , we obtain the feature matrix $\mathbf{H}^q = [\mathbf{h}_1^q, \mathbf{h}_2^q, \dots, \mathbf{h}_N^q] \in \mathbb{R}^{n \times N}$, the k -th column $\mathbf{h}_k^q \in \mathbb{R}^n$, where $n \gg w \times h$ denotes the large multiscale feature vector of the k -th sample that is filtered with rectangle filters and concatenated as such a high-dimensional feature vector. Features of the total $N \times Q$ samples can denote as $\mathbf{H} = [\mathbf{H}^1, \mathbf{H}^2, \dots, \mathbf{H}^Q] \in \mathbb{R}^{n \times NQ}$.

We adopt a sparse random matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$, $m \ll n$ to reduce the original feature space n into a lower-dimensional space m such as $\mathbf{L}^q = [\mathbf{l}_1^q, \mathbf{l}_2^q, \dots, \mathbf{l}_N^q] \in \mathbb{R}^{m \times N}$ for q -th sub-patch. Concatenating Q local patches together, we obtain $\mathbf{L} = [\mathbf{L}^1, \mathbf{L}^2, \dots, \mathbf{L}^Q] \in \mathbb{R}^{m \times NQ}$, computed by

$$\mathbf{L} = \mathbf{R}\mathbf{H} \quad (2)$$

A typical choice of such a measurement matrix is the random Gaussian matrix $\mathbf{R}_{ij} \sim \mathcal{N}(0, 1)$. But when n is huge, the computational loads are still heavy because the random Gaussian matrix is dense. Thus it is common to employ a very sparse random measurement matrix that satisfies a weaker property than RIP but almost as accurate as the conventional random Gaussian matrix [23], as (3), where \mathbf{R}_{ij} denotes the element in the i -th row and j -th column of \mathbf{R} . This random matrix is fixed at the beginning and easy to compute for real-time tracking by fixing the maximum number Z of nonzero elements to be a lower number. The scheme to produce the random matrix in this work is similar to [12]. We illustrate the dimension reduction process in

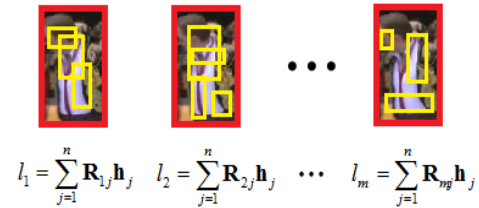


Fig. 2: An illustration for compressive representation for an arbitrary sample. Denote its high-dimensional feature vector as $\mathbf{h} \in \mathbb{R}^n$. After the dimension reduction from n to m , we get its m -dimensional feature vector $\mathbf{l} = [l_1, l_2, \dots, l_m]^T \in \mathbb{R}^m$. Each element in \mathbf{l} is linearly combined by the feature values of less than Z rectangles (yellow) inside the sample region (red) and the coefficient of the combination is in the rows of \mathbf{R} . The feature values of each rectangle is actually the convolution from the corresponding rectangle filter that is the same size as the rectangle itself, i.e., the sum of gray values of all pixels inside it which can be computed very fast using the integral map.

Fig. 2.

$$\mathbf{R}_{ij} = \begin{cases} \sqrt{p}, & \text{with probability } \frac{1}{2p} \\ 0, & \text{with probability } 1 - \frac{1}{p} \\ -\sqrt{p}, & \text{with probability } \frac{1}{2p} \end{cases} \quad (3)$$

B. Classification via Ensemble Layers

To link up the individual pixels with the local patches, we employ the naive Bayesian classifier to construct the pool of weak classifiers corresponding to each individual compressive feature in the bottom layer. We assume the compressive m -dimensional features of each sub-patch are independently distributed and build m weak classifiers corresponding to these features by considering both the object and the background information. Since \mathbf{R} is fixed during the tracking process, the way to compress the high dimensional features of samples stays consistent for all sub-patches. Let $\mathbf{l} = [l_1, l_2, \dots, l_m]^T \in \mathbb{R}^m$ denote an arbitrary compressive sample, for the i -th compressive feature, the i -th classifier is constructed as follows:

$$f(l_i) = \log \left(\frac{p(y=1|l_i)}{p(y=0|l_i)} \right) = \log \left(\frac{p(l_i|y=1)p(y=1)}{p(l_i|y=0)p(y=0)} \right), \quad (4)$$

where $y \in \{0, 1\}$ is a binary variable which represents the sample label. We assume $p(y=1) = p(y=0)$ by sampling the same quantity of positive and negative samples at update step. The conditional distributions $p(l_i|y)$ are almost Gaussian due to the random projections of the high dimension features [24]. Thus we have:

$$p(l_i|y=1) \sim \mathcal{N}(\mu_i^1, \sigma_i^1), p(l_i|y=0) \sim \mathcal{N}(\mu_i^0, \sigma_i^0), \quad (5)$$

where μ_i^1 (μ_i^0), σ_i^1 (σ_i^0) are the mean and standard deviation of the positive (negative) class.

Then we introduce an ensemble strategy which combines the output of weak classifiers to create a strong classifier as

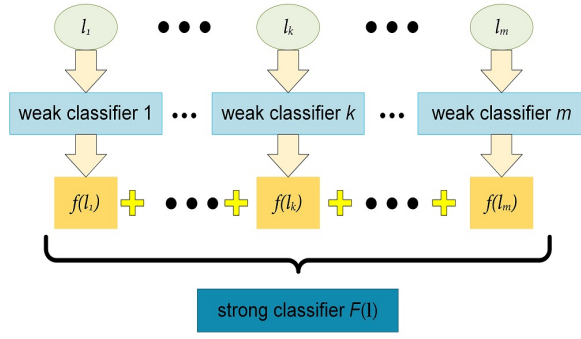


Fig. 3: The ensemble process in the bottom layer for an arbitrary sample $\mathbf{l} = [l_1, l_2, \dots, l_m]^T \in \mathbb{R}^m$.

a base ensemble to detect the sub-patches as shown in Fig.3, denoted as

$$F(\mathbf{l}) = \sum_{i=1}^m f(l_i), \quad (6)$$

For the q -th sub-patch, we seek its N samples for matching and its matching score c_q like:

$$\begin{aligned} \mathbf{g}^q &= \arg \max_k F(\mathbf{l}_k^q), \quad k = 1, \dots, N, \\ c_q &= F(\mathbf{g}^q), \end{aligned} \quad (7)$$

We match all Q sub-patches in the same way in the bottom layer and obtain the compressive feature of their optimal matching $\mathbf{G} = [\mathbf{g}^1, \mathbf{g}^2, \dots, \mathbf{g}^Q] \in \mathbb{R}^{m \times Q}$ and their scores $\mathbf{c} = [c_1, c_2, \dots, c_Q]^T$. In the ensemble learning field, it is often found that improved performance can be obtained by combining multiple models simultaneously like (6), instead of just using a single model individually [25].

In the middle layer, we propose a novel ensemble strategy to acquire the observed location of the object from the base ensembles like Fig.4 via these Q detected local patches.

Suppose the actual location of the object we are trying to predict is given by $H(\lambda)$, and $y_i(\lambda) = \Delta \mathbf{p}_i + \mathbf{p}_i$ denotes the i -th hypothesis of object location obtained by the i -th detected sub-patch. The output of each sub-patch model can be written as the true value plus an error in this form:

$$y_i(\lambda) = H(\lambda) + \varepsilon_i(\lambda) \quad (8)$$

To be convenient for comparison, we adapt the scores of sub-patches $\mathbf{c} = [c_1, c_2, \dots, c_Q]^T$ by the zero-mean normalization, then rescale them to $\omega = [\omega_1, \omega_2, \dots, \omega_Q]^T$, $\omega_i \in [0.1, 0.9]$. ω is regarded as the weights of candidates that obtained by the corresponding sub-patches. We update these weights adaptively for each new frame. The combined prediction is given by

$$y_{COM} = \frac{1}{W} \sum_{i=1}^Q \omega_i y_i(\lambda), \quad W = \omega_1 + \dots + \omega_Q \quad (9)$$

The average sum-of-squares error then takes the form as follows:

$$E_\lambda \left[(y_i(\lambda) - H(\lambda))^2 \right] = E_\lambda \left[\varepsilon_i(\lambda)^2 \right] \quad (10)$$

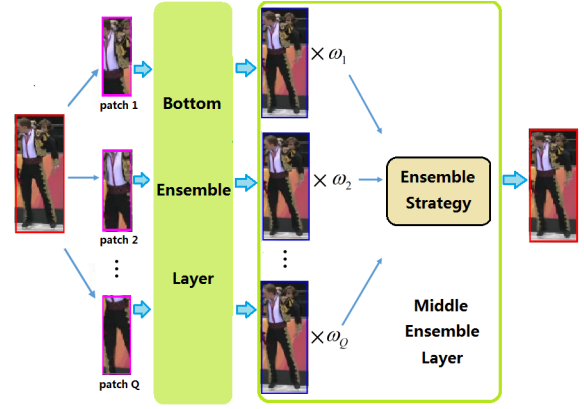


Fig. 4: The first column is the object at previous frame, second column denotes the randomly extracted sub-patches. Then transfer the sub-patches to the bottom ensemble layer to gain the base ensembles. The fourth column shows the scores and corresponding target candidates of the local patches which are the output of the base ensembles. Finally, we employ the proposed ensemble strategy to obtain the observation of object in the middle layer.

where $E_\lambda[\bullet]$ denotes a frequentist expectation. The average error made by the sub-patch models acting individually is

$$E_{AV} = \frac{1}{W} \sum_{i=1}^Q \omega_i E_\lambda \left[\varepsilon_i(\lambda)^2 \right] \quad (11)$$

We assume that the errors have zero mean and uncorrelated due to the sub-patches are randomly extracted. So we have:

$$E_\lambda \left[\varepsilon_i(\lambda) \right] = 0, \quad E_\lambda \left[\varepsilon_i(\lambda) \varepsilon_j(\lambda) \right] = 0, \quad i \neq j \quad (12)$$

The expected error from the combined prediction is computed by

$$\begin{aligned} E_{COM} &= E_\lambda \left[\left(\frac{1}{W} \sum_{i=1}^Q \omega_i y_i(\lambda) - H(\lambda) \right)^2 \right] \\ &= E_\lambda \left[\frac{1}{W^2} \left(\sum_{i=1}^Q \omega_i \varepsilon_i(\lambda) \right)^2 \right] \\ &= \frac{1}{W} E_\lambda \left[\frac{1}{W} \sum_{i=1}^Q \omega_i \varepsilon_i(\lambda)^2 \right] \\ &\leq \frac{1}{W} E_\lambda \left[\frac{1}{W} \sum_{i=1}^Q \omega_i \varepsilon_i(\lambda)^2 \right] \\ &= \frac{1}{W} E_{AV} \end{aligned} \quad (13)$$

We extract more than 10 sub-patches to ensure $W \geq 1$. The result suggests that the average error of a object model can be reduced weighted combining all the sub-patch models using (9) on the key assumption (12) that the errors of each model are uncorrelated by randomly choose the sub-patches.

C. Adaptive Kalman Filter

The top layer builds an adaptive Kalman filter based on the two ensemble layers to estimate the optimal system state and target image velocity so that the proposed tracker can overcome the temporary occlusion, missing and false

detections. We regard the observation result (9) from the bottom and middle ensemble layers as the measurement. The discrete time system state and measurement at time k are given by $\mathbf{x}(k) = [x(k), y(k), v_x(k), v_y(k)]^T$ and $\mathbf{z}(k) = y_{COM} = [x_o(k), y_o(k)]^T$, where $x(k), y(k), x_o(k), y_o(k)$ denote the center coordinate in the image space corresponding to system state and measurement at time k respectively, $v_x(k), v_y(k)$ denote velocities in both two axis of system state. The state and measurement in the next time step $k + 1$ is given by

$$\begin{aligned} \mathbf{x}(k+1|k) &= \mathbf{A}(k+1|k)\mathbf{x}(k|k) + \delta(k+1), \\ \mathbf{z}(k+1) &= \mathbf{B}\mathbf{x}(k+1|k) + \nu(k+1), \\ \mathbf{A}(k+1|k) &= \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \end{aligned} \quad (14)$$

where $\mathbf{A}(k+1|k)$ is modeled according to the Newton's equation of motion, Δt is the time between two frames, $\delta(k+1)$ and $\nu(k+1)$ are assumed to be white Gaussian noises with zero mean and covariance matrixes $\mathbf{Q}(k), \mathbf{R}(k)$ respectively. To achieve an adaptive Kalman filter, we take the mean of normalized scores in the middle layer to update these two covariance matrixes every frame like (15) and (16). We ascertain the ultimate object location at the current frame in the top layer with this adaptive Kalman filter.

$$\mathbf{Q}(k+1) = \left(\frac{1}{Q} \sum_{i=1}^Q \omega_i^{k+1} \right) \mathbf{Q}(0) \quad (15)$$

$$\mathbf{R}(k+1) = \left(1 - \frac{1}{Q} \sum_{i=1}^Q \omega_i^{k+1} \right) \mathbf{R}(0) \quad (16)$$

D. Model Update

It is important to update the target model continuously for robust tracking in the face of various difficult environment. The proposed method updates the hierarchical model via three mechanisms: re-extracting the sub-patches according to the object that we have found at the current frame, choosing the sub-patches that need to be updated and adjusting the parameters of the weak classifiers in the bottom layer. The update process is also shown in the Fig.1.

Once we find the object at the current frame, we need to correct the locations of all sub-patches in the middle layer due to the drift of the detection process. The way to re-extract the randomly overlapping sub-patches is fixed at the first frame. After that, we compress the features of these new sub-patches and put them into the weak classifiers in the bottom layer to obtain the updated scores $\mathbf{c}^{new} = [c_1^{new}, c_2^{new}, \dots, c_Q^{new}]^T$. We assume that c_j^{new} is Gaussian, and the sub-patches to be updated are those whose scores satisfy

$$c_j^{new} \in (\mu_c - \sigma_c, \mu_c + \sigma_c), j = 1, 2, \dots, Q, \quad (17)$$

where μ_c, σ_c are mean and standard deviation of the scores.

Then, for the j -th chosen sub-patch, we extract N positive samples whose Euclidean distances to the sub-patch are

smaller than a threshold value β and N negative samples whose Euclidean distances to the sub-patch are bigger than a threshold value π that is fixed at beginning. We update the parameters of its i -th weak classifier in (5) like

$$\begin{aligned} \mu_i^1 &= \lambda \mu_i^0 + (1 - \lambda) \mu^1 \\ \sigma_i^1 &= \sqrt{\lambda (\sigma_i^0)^2 + (1 - \lambda) (\sigma^1)^2 + \lambda (1 - \lambda) (\mu_i^0 - \mu^1)^2} \end{aligned} \quad (18)$$

where μ^1, σ^1 denote mean and standard deviation of the N positive samples. And μ_k^0, σ_k^0 are updated in a similar way.

IV. EXPERIMENTS

In this section, we show the experimental results of our method. Firstly, we present the implement details of the proposed tracker and the evaluation criteria to quantitatively assess the performance. Secondly, we validate the joint representation of our hierarchical ensemble framework with the base method. Thirdly, we compare our tracker to three most similar methods which are famous in the visual tracking field. Fourthly, we compare our method with 8 state-of-the-art methods which are feasible for robotic applications in terms of computational complexity and hardware requirements. Finally, we demonstrate that our tracker performs excellently for moving human tracking which is crucial for the tracking applications of robots.

A. Implementation Details

The proposed algorithm is implemented in Matlab(R2013a) and runs at 30 frames per second on an Intel i7-4790 machine with 3.6GHz CPU and 8GB RAM. For each sequence, the location of the target object is manually labeled at the first frame. For all reported experiments, we employ 150 weak classifiers in the bottom ensemble layer and randomly generate 11 sub-patches that are located inside the object and whose width and height are three quarters of the size of the object. We set learning rate $\lambda=0.85$, maximum number of nonzero elements $Z=4$ in random matrix \mathbf{R} and thresholds $\beta=20, \pi=2\beta$ in all experiments.

In the experiment, we employ two evaluation criteria to quantitatively assess the performance of the trackers including the average overlap rate and the center location error. Given the tracked bounding box ROI_T and the ground truth bounding box ROI_G , we use the detection criterion in the PASCAL VOC challenge [26], $score = \frac{area(ROI_T \cap ROI_G)}{area(ROI_T \cup ROI_G)}$ to evaluate the success rate.

B. Comparison with the base method

Compressive tracking(CT) [12] employs the compressive sensing theory to compress the appearance models. It is reasonable to consider CT as our base method since the way to compress a image sub-patch is almost the same. In the bottom layer of our method, we build compressively sensed versions of sub-patches, while CT presents objects by the compressive appearance models globally.

However, it's insufficient to present the holistic object by a single appearance model just like CT especially in the case of tracking non-rigid objects. So we adopt the joint

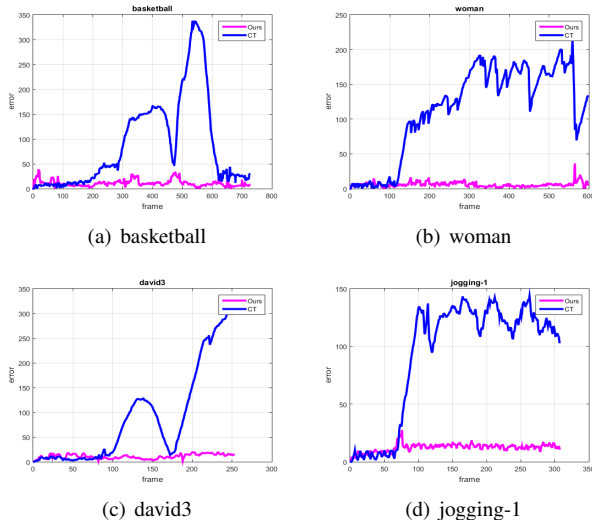


Fig. 5: Pixel center location error of our method and the base method CT at each frame on four video sequences. Our method tracks the objects more accurately than CT on the four videos.

representation which considers both global and local models of the targets to better handle significant appearance changes, deformations and occlusions. As shown in Fig.5 and Fig.7, our method obtains more accurate tracking performances than the base method and it outperforms CT by 24% for the success plots and by 38.3% for the precision plots.

C. Comparison with similar methods

There are three methods LSK [14], OAB [20] and MIL [19] that are most similar to our tracker in recent years. The proposed method outperforms them as shown in Fig.6 and Fig.7.

LSK proposes a robust tracking algorithm with a local sparse appearance model which combines a static sparse dictionary with a sparse coding histogram. This method outperforms several sparse representation methods according to [3]. However, LSK neglects the temporal consistency in the target bounding box level while we take this into consideration by employing an adaptive Kalman filter. Therefore our method is more robust to occlusions than LSK, as shown in Fig.9.

OAB and MIL are both boosting-based algorithms similar with ours. Our ensemble technique is much easier than the boosting of the two methods. However, they characterize the objects by global templates while we adopt both local representations and holistic templates. Thus we can better handle the deformations and occlusions, as shown in Fig.9.

D. Comparison with State-of-the-arts

For comparison, we run 11 state-of-the-art algorithms with the same initial positions of targets. These algorithms are CT [12], CN [27], Struck [11], TLD [15], ASLA [16], CSK [18], OAB [20], MIL [19], LSK [14], SCM [28] and VTD [17]. For fair evaluation, we evaluate the proposed

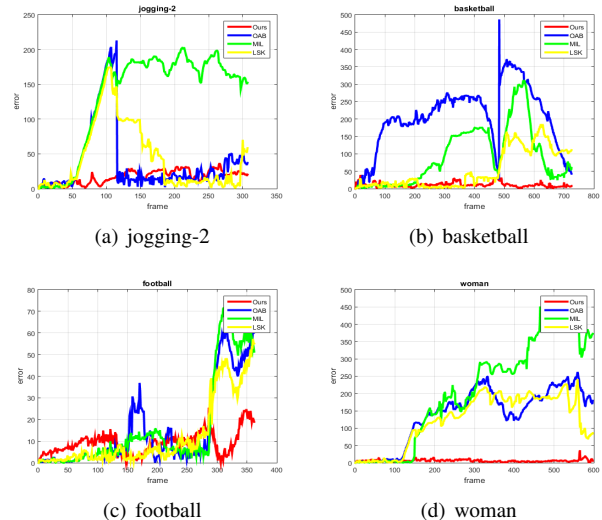


Fig. 6: Pixel center location error of our method and the three similar methods at each frame on four video sequences. Our method tracks the objects more robustly than the three methods on these videos.

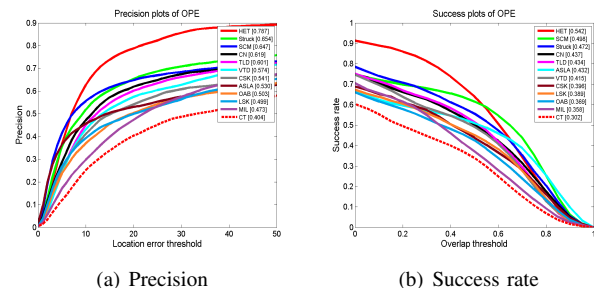


Fig. 7: Precision and success rate plots of overall performance comparison for the 50 videos in the benchmark.

HET against those methods using the source codes provided by the authors with adjusted parameters. We examine the effectiveness of the proposed approach on an online object tracking benchmark [3] tested with 50 sequences that cover most challenging tracking scenarios such as illumination variations, scale variations, occlusions, deformations, etc.

Note that we don't compare with some excellent methods such as MEEM [29], TGPR [30], MUSTer [31], etc, because they are not fast enough in robotic applications. For instance, the average time of MUSTer cost on the benchmark [3] is 0.287s/frame on a cluster node (3.4GHz, 8 cores, 32GB RAM). We also not consider some convolutional neural network based methods like FCNT [32] and HCF [33] because they require powerful GPUs to run the algorithms while still in a low frame rate. Although the performances of these trackers may be a little better than ours, their computational costs and hardware requirements are impracticable for robots.

The comparison results of precision plots and success plots of OPE on benchmark are shown on Fig.7. As for the other top-ranking trackers in the benchmark, the results show that



Fig. 8: Tracking results of 12 trackers (denoted in different colors and lines) on 6 image sequences. Frame indexes are shown in the top left of each figure in yellow color. Results are best viewed on high-resolution displays.

the proposed method achieves the best average performance. The performance of our approach can be attributed to the efficient ensemble methods on sub-patches with a spatial layout combining the adaptive Kalman filter.

E. Human body tracking for robots

In particular, we find that the proposed HET performs excellently for moving human tracking which is crucial for the tracking applications of robots. We choose 20 videos whose target objects are moving human bodies and includes the challenging conditions like occlusions, deformations, fast motions, background clutters, etc. Due to space limitations, we only show some shots of 6 videos among the 20 videos in Fig.8. We can see when the human body objects undergoing large deformations or occlusions, other methods almost lose the objects except the proposed HET.

Due to the suppleness structure of human limbs and the flexibility of human movements, the most challenging factors in moving body tracking are deformations and occlusions. It is obvious that splitting up a human object into local parts can make the appearance models more flexible. The local compressive appearance models of the proposed HET actually do these things. We build compressively sensed versions of local patches in the bottom layers which allows the proposed tracker to better handle occlusion and large appearance change. The tracking results on these moving human videos with occlusions(OCC), deformation(DEF), background clutter(BC), scale variations(SV), fast motion(FM) and illumination variation(IV) attributes based on the precision and success rate metrics are persuasive, as shown in Fig.9. Our method almost ranks the first on these attributes according to the two criteria.

The robustness and realtime performance to the human body tracking makes HET suitable for many robotic applications such as human-computer interaction, home service robots, robot teaching systems and unmanned vehicles.

V. CONCLUSION

In this paper, we propose a novel hierarchical ensemble framework, where the representations of the target candidates are localized and compressed. We incorporate information including individual pixel features, local patches and holistic target models. The multiple ensemble layers exploit the intrinsic relationship not only between the individual pixel features and local patches, but also between the patches and the target candidates. In the bottom layer, the base ensembles are created using linear combinations of outputs from the base weak classifiers. A diverse collection of base ensembles are systematically combined in order to generate a more strong ensemble classifier in the middle layer and the scores of the local patches are normalized to produce a vector of weights of the base ensembles. Experimental results with evaluations against several state-of-the-art methods on challenging image sequences demonstrate the robustness of the proposed HET tracking algorithm. Since our method is real-time, general and robust, we plan to apply it to the tracking tasks of robots. In particular, the proposed HET is very efficient for moving human tracking, we can apply this to many applications such as unmanned vehicles, robot teaching system, etc.

REFERENCES

- [1] A. Kolarow, M. Brauckmann, M. Eisenbach, K. Schenk, E. Einhorn, K. Debes, and H.-M. Gross, "Vision-based hyper-real-time object tracker for robotic applications," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 2108–2115, IEEE, 2012.
- [2] D. A. Klein, D. Schulz, S. Frintrop, and A. B. Cremers, "Adaptive real-time video-tracking for arbitrary objects," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 772–777, IEEE, 2010.
- [3] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 37, no. 9, pp. 1834–1848, 2015.

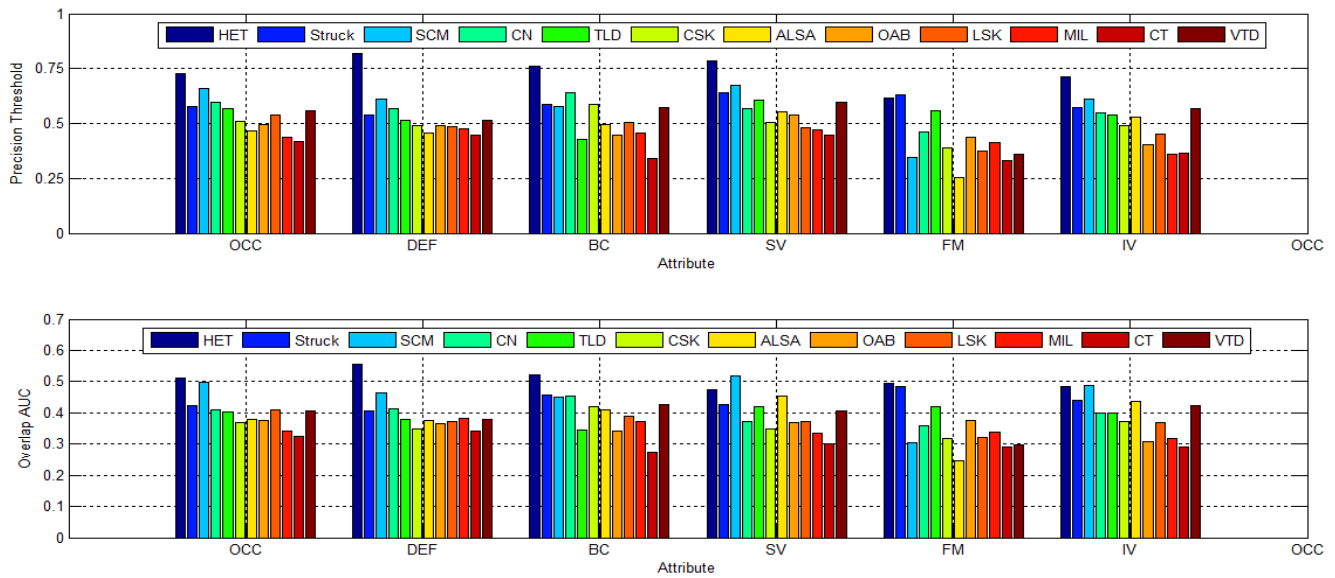


Fig. 9: Precision threshold and overlap AUC of 6 attributes on the 20 benchmark videos.

- [4] Y. Li, J. Zhu, and S. C. Hoi, "Reliable patch trackers: Robust visual tracking by exploiting reliable patches," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 353–361, 2015.
- [5] T. Zhang, S. Liu, C. Xu, S. Yan, B. Ghanem, N. Ahuja, and M.-H. Yang, "Structural sparse tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 150–158, 2015.
- [6] D. L. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [7] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [8] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [9] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 2427–2434, IEEE, 2013.
- [10] J. Xiao, R. Stolkin, and A. Leonardis, "Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4978–4987, 2015.
- [11] S. Hare, A. Saffari, and P. H. Torr, "Struck: Structured output tracking with kernels," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 263–270, IEEE, 2011.
- [12] K. Zhang, L. Zhang, and M.-H. Yang, "Fast compressive tracking," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 10, pp. 2002–2015, 2014.
- [13] S. Stalder, H. Grabner, and L. Van Gool, "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pp. 1409–1416, IEEE, 2009.
- [14] B. Liu, J. Huang, L. Yang, and C. Kulikowski, "Robust tracking using local sparse appearance model and k-selection," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1313–1320, IEEE, 2011.
- [15] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1409–1422, 2012.
- [16] X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on*, pp. 1822–1829, IEEE, 2012.
- [17] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1269–1276, IEEE, 2010.
- [18] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Computer Vision–ECCV 2012*, pp. 702–715, Springer, 2012.
- [19] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1619–1632, 2011.
- [20] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via online boosting," in *BMVC*, vol. 1, p. 6, 2006.
- [21] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised online boosting for robust tracking," in *Computer Vision–ECCV 2008*, pp. 234–247, Springer, 2008.
- [22] R. G. Baraniuk, "Compressive sensing," *IEEE signal processing magazine*, vol. 24, no. 4, 2007.
- [23] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 287–296, ACM, 2006.
- [24] P. Diaconis and D. Freedman, "Asymptotics of graphical projection pursuit," *The annals of statistics*, pp. 793–815, 1984.
- [25] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.
- [26] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [27] M. Danelljan, F. Shahbaz Khan, M. Felsberg, and J. Van de Weijer, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1090–1097, 2014.
- [28] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *Image Processing, IEEE Transactions on*, vol. 23, no. 5, pp. 2356–2368, 2014.
- [29] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *Computer Vision–ECCV 2014*, pp. 188–203, Springer, 2014.
- [30] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *Computer Vision–ECCV 2014*, pp. 188–203, Springer, 2014.
- [31] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "Multi-store tracker (muster): a cognitive psychology inspired approach to object tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 749–758, 2015.
- [32] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3119–3127, 2015.

- [33] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3074–3082, 2015.