SUMMARY  Aiming to alleviate the visual tracking problem of drift which reduces the abilities of almost all online visual trackers, a robust visual tracker (called CCMM tracker) is proposed with a coupled-classifier based on multiple representative appearance models. The coupled-classifier consists of root and head classifiers based on local sparse representation. The two classifiers collaborate to fulfill a tracking task within the Bayesian-based tracking framework, also to update their templates with a novel mechanism which tries to guarantee an update operation along the “right” orientation. Consequently, the tracker is more powerful in anti-interference. Meanwhile the multiple representative appearance models maintain features of the different submanifolds of the target appearance, which the target exhibited previously. The multiple models cooperatively support the coupled-classifier to recognize the target in challenging cases (such as persistent disturbance, vast change of appearance, and recovery from occlusion) with an effective strategy. The novel tracker proposed in this paper, by explicit inference, can reduce drift and handle frequent and drastic appearance variation of the target with cluttered background, which is demonstrated by the extensive experiments.

key words: coupled-classifier, representative appearance model, multiple appearance models, online visual tracking

1. Introduction

Visual tracking is aiming to estimate the states of the target in a video. As an important topic in the field of computer vision, it plays an elementary and critical role in many research and application fields such as intelligent surveillance, vehicle navigation, and human activity recognition. A visual tracker consists of three main components. The first one is an appearance model which matches the most similar patch candidate. The second one is a dynamic model which describes the motion mode of the target. The last one is a search strategy on how to find the most similar candidate in the current video frame. In this paper, we propose a robust tracker with a coupled-classifier based on a feature pool using multiple representative appearance models. Thus only the first component, appearance models, will be discussed in detail.

Although many good results have been published in online visual tracking, a challenging problem continues to be how to develop a robust tracker by maintaining an effective appearance model, which has been attracting much attention recently [2]–[8]. Naturally the existing tracking algorithms try to update the appearance model as similar as possible with the target’s real state by employing the last tracking result or the mean of the most recent several states in updating their visual templates, enabling the tracker to adapt to the appearance variation of the target as quickly as possible. Unfortunately, they often suffer from drifting problem due to the self-updating with the inappreciative templates.

Based on the analysis of failures mentioned above, two primary reasons can be found.

Firstly, the appearance model is updated straightforwardly using the last tracking result(s) aiming to get the most discriminative template, but there is no mechanism to guarantee the quality of the result(s) for updating the template, which is critical to maintaining the robustness of a tracker. Even though some results may be the best candidate, it certainly includes slight reconstruction error due to the nature of generative tracking approaches, and the sampling error depending on slide windows or other dynamic models (also called as a motion model). Furthermore, it may get worse if the result is not the best candidate when the tracker not configured with non-optimal arguments. Making things worse, the bias probably can be accumulated while the tracking algorithm runs repeatedly, degrading the models gradually, and then incurring drift. So, it is very important to judge the quality of the result and choose only the qualified ones for update. Several works, like P-N learning [2] or the two-stage algorithm [11], have tried various strategies but they are indirect methods that attempt to roughly constrain the sampling scope.

Secondly, the tracker does not remain the helpful learned templates, which are representative templates and may be useful in the following frames. Especially, for the challenging cases that the target appearance is quite different from the previous one(s), for an instance recovering from occlusion, it is helpful to recognize the target by looking for similar templates which occurred and recorded in the different past life-span. Some papers try to solve such an issue by incorporating two models, i.e., a static one from the ground truth in the initial frame and an adaptive one [11]. However, all these methods are on a premise that the observation of the target in the following frames is likely to be somewhat similar to the ground truth, i.e., a static model can contribute to determine the final results. Unfortunately, such an assumption does not always hold in real-world scenarios. A magic cube, as an extreme example, may exhibit several very different submanifolds during a tracking process.

Hence we propose a novel, intuitive and robust tracker
Visual tracking strategies can be roughly categorized into generative and discriminative approaches. Generative approaches follow the target by searching for the most similar patches to the models in each frame of a video. Recent generative trackers represent target appearance as models, and learn, maintain the online models to minimize the reconstruction error. Discriminative methods take a tracking task as a binary classification problem. The classifier distinguishes the target patch from background in each frame.

Most recently, several similar works attempt to reduce the drift problem by adopting different strategies on appearance models, and success has been demonstrated to a certain degree. Some algorithms try to combine all aspects of target appearance, but they sometimes failed to guarantee the quality of the templates and features may be ambiguous. The algorithm of “bag of features” visual tracking [12] combines features over time by building two codebooks with codewords of RGB and LBP local features directly abstracted from the last result. The incremental visual tracking (IVT) algorithm [8] learns an incremental subspace model to adapt to changes in appearance. But it may fail due to degraded templates when objects undergo heavy occlusion or abrupt variation in appearance, and low quality tracking results are still used to update the model. Visual tracking decomposition (VTD) [4] strives to maintain a complete space of known features, with multiple feature types and all the past states. A subset of templates is selected from the complete feature space, and the coming state is judged by each template in the subset. Consequently the best candidate is selected by the smallest diffusion distance among all elements in the subset. However its computational complexity is rather high, since the algorithm also has to maintain and update the complete space at each iteration; furthermore representative templates for detection need to be selected by extra cost. As well, the updating scheme is not free from noise samples (e.g. if the object is occluded). Hough-based algorithm [16] extends Hough Forests to the online domain, as well couples the voting-based detection and back-projection with a rough segmentation. Thus, it significantly reduces the possibility of drift for non-rigid objects by reducing the amount of noisy training samples. Another tracker using the algorithm of multiple instance learning (MIL) [2] tries to reduce visual drift by learning a discriminative model which is trained with the bags of ambiguous positive and negative samples. These approaches mentioned above try to learn a pool of appearance features to represent the target as adaptive as possible. Nevertheless, when the trackers undergo some challenging scenarios, they still suffer from drift. Ambiguity of features is one reason, and absence of a mechanism that guarantees the quality is the other.

Most recently, some trackers employ specific strategies to validate the quality of the results by utilizing history results. The TLD tracker with P-N learning algorithm [3] exploits the underlying structure with trajectory which is the statistic of the history results. An effective classifier is trained by a learning process which is guided by helpful constraints of the structure. Nevertheless, the result is not accurate enough because the algorithm can only judge whether the candidates are qualified or not. However it is the first attempt to control the quality of results by excluding obviously false ones.

Several similar works are proposed recently. One is the two-stage online algorithm [11] based on local sparse representation, which try to enhance tracking ability of a single appearance model using two classifiers. It exploits both a static observation model and an adaptive observation model. The algorithm depends on a strong assumption that all the following states are similar to the ground truth in the first frame to some extent, which is not the case in real scenarios. Obviously, more models are required to support the adaptive observation model. Despite this, it provides an idea to validate the quality of tracking results. Another algorithm [15] is equipped with multiple trackers, and employs strategies of tracker selection and interaction. It focuses on integrating different feature descriptors.

The sparse representation of an object is adopted by $\ell_1$ tracker [7] as a holistic appearance model. By constructing a template matrix by raw image patches and solving the $\ell_1$ minimization problem, a coefficient vector related with a candidate patch is obtained. Then, the quality of the results is estimated by the minimum reconstruction error. Two years later, the more efficient method proposed with minimum error bound [6]. In this paper, we simply adopt minimum reconstruction error as measure criterion.

Comparing to the state-of-the-art algorithms, our model has the following improvements. First, compared with the methods indirectly measuring the quality of results candidates, our estimation scheme of coupled-classifier is direct from the final results rather than immediate templates like $\ell_1$ tracker [7]; as well the final results are validated by the overlap ratio of results and affected by the confidence values of classifiers, rather than roughly constraint like TLD tracker with P-N learning algorithm [3]. Meanwhile, the update operation is guided by the overlap ratio and the confidence values for the alternative results; thereby our tracker is prevented from updating with successive degraded samples. Second, compared with the existing trackers equipped with multiple appearance models, our algorithm maintains multiple models to explicitly describe the
representative submanifolds of the target appearance, instead of ambiguous features in algorithms such as “bag of features” [12], IVT [8] and MIL [2], also different from only two models in two-stage online algorithm [11].

2.2 Bayesian Tracking Framework

Visual tracking can be formulated as an inference task in a Markov model with hidden state variables, and it is often addressed in the Bayesian framework. Following the traditional definitions, let the target’s hidden state vector $h_{st} = (x, y, \theta, s_c, \alpha, \phi)$, where $x$, $y$, $\theta$, $s_c$, $\alpha$, $\phi$ denote $x$ and $y$ translations, rotation angle, scale, aspect ratio, and skew direction at time $t$, respectively. Given a series of observations $z_{1:t} = \{z_1, z_2, \cdots, z_t\}$, the aim is to estimate the hidden state variable $h_{st}$. According to the Bayesian rule, the filtering distribution $p(h_{st}|z_{1:t})$ can be recursively estimated by the following formula:

$$p(h_{st}|z_{1:t}) \propto p(z_t|h_{st}) \int p(h_s|z_{t-1})p(h_{s,t-1}|z_{1:t-1})dh_{s,t-1},$$

(1)

where $p(z_t|h_{st})$ represents the observation model which scores how much the state and the observation of the proposed state coincide, which is implemented in next subsection. As well $p(h_{st}|h_{s,t-1})$ denotes the state transition model of the target between two consecutive frames with affine transform. Each parameter in $h_{st}$ is modeled independently with a Gaussian distribution around its counterpart. Therefore, $p(h_{st}|h_{s,t-1})$ can be formulated as $p(h_{st}|h_{s,t-1}) = N(h_{st}; h_{st-1}, \Sigma)$, where $\Sigma$ is a diagonal covariance matrix whose elements are the corresponding variances of target state parameters, i.e. $\sigma_x^2, \sigma_y^2, \sigma_\theta, \sigma_s^2, \sigma_\alpha, \sigma_\phi$.

Then, normally the best configuration of the target $\hat{s}_t$ can be obtained by the Maximum a Posteriori (MAP) over $N$ samples at time $t$, which is formulated as follows:

$$h_{s,t} = \arg \max_{s_t} p(h_s|z_{1:t}), \quad n = 1, \ldots, N,$$

(2)

where $h_{s,n}$ denotes the $n$-th sample of the state $h_{s,t}$; $N$ is the number of the candidate samples. Under the challenging cases like occlusion, the tracking result is estimated with the collaborative combination of multiple observers which represent appearance submanifolds of the target.

2.3 Classifier Based on Local Sparse Representation

As mentioned above, the feature vector of the object appearance is coded by local sparse representation, and the tracking task is formulated as a binary classification problem.

2.3.1 Local Sparse Representation of the Object Appearance Based on $\ell_1$ Minimization

Methods of object representation with local features have received considerable attention [10] due to the more robust performance than global approaches in recent years. As well, sparse coding has demonstrated much success in tracking tasks [11], [9]. Therefore, we code the feature of object appearance by local sparse representation. First, local patches inside the target region are cropped and encoded by an over-complete dictionary. And then the corresponding sparse codes are concatenated to form a sparse code of the target appearance.

In order to construct an over-complete dictionary, $r$ overlapped patches are cropped from the target object region in a frame. Then we obtain a template set $T = \{t_1, t_2, \cdots, t_r\} \in \mathbb{R}^{d \times r}$ by stacking $t_i$, each of which denotes one template of an object. Thus, each column vector of $T$ is a target template generated by rearranging pixels of the ground truth in the initial frame or from subsequent tracking results. Similar to [7], the over-complete dictionary is constructed as $D = \{T, I, -I\}$, where $I \in \mathbb{R}^{d \times d}$ is an identity matrix. By using $I$ and $-I$, the sparse code representing an image patch based on the dictionary $D$ can maintain non-negativity and sparsity constraints.

Then, for the $p$-th candidate patch $t_{i}^{(p)}$ at time $t$, $M$ image patches are cropped inside $s_i^{(p)}$ and are vectorized as $[s_i^{(p)}, \cdots, s_i^{(pM)}]$. For each patch $s_i^{(p)}$, $i = 1, 2, \ldots, M$, the corresponding sparse code $a_i \in \mathbb{R}^{r \times 2d}$ can be formulated as a minimum error reconstruction through a regularized $\ell_1$ minimization [7] function:

$$\min_{a_i} \frac{1}{2} || s_i^{(p)} - Da_i ||_2^2 + \lambda || a_i ||_1, \quad \text{s.t.} \quad a_i \geq 0,$$

$$i = 1, 2, \ldots, M$$

(3)

where $\lambda$ is a regularization parameter.

Finally, the sparse codes $[a_1, \cdots, a_M]$ of all the image patches from a target region are concatenated to form an object representation for visual tracking. In a simple way, the candidate patch of the object $s_i^{(n)}$ is represented by $s_i^{(n)} = [a_i^{T}, \cdots, a_M^{T}]^T$.

2.3.2 Classifier Learning with Sparse Representation

A similar to most discriminative tracking algorithms that treat visual tracking task as a classification problem, a linear classifier is employed here based on local sparse coding.

The linear classifier is able to achieve favorable tracking results because it is easy to separate target object from the background with the features encoded in the way of local sparse representation described in previous subsection, instead of raw image features. The classifier learns from local negative and positive patches that are extracted from the target and the background, respectively. With the dictionary $D$, the image patches from the region of target object are likely to be well reconstructed by only the target basis set $T$ only, but image patches from the background may need trivial templates for good reconstruction. Therefore, compared to the raw image features, features in sparse representation help the classifier identify the target object from the background more easily.

The training data for a classifier is extracted in the
following way. Suppose $I_t = (x_t, y_t)$ denotes the location of the center of the target object in the frame at time $t$. With Gaussian perturbation, positive samples are drawn from a circular region specified by $\|R_{pos} - I_t\| < \gamma$, and negative samples satisfy $\gamma < \|R_{neg} - I_t\| < \eta$, where $\gamma$ and $\eta$ are thresholds, $I_{pos}$ and $I_{neg}$ represent the locations of positive and negative candidates respectively. Then, a set of $M$ training samples are cropped from each of $I_{pos}$ and $I_{neg}$ respectively, and encoded further by sparse coding. Finally, we obtain a training data set $\{v^j, l^j\}_{j=1}^M$, where $v^j$ is the sparse code of the $j$-th local patch obtained from the $j$-th candidate or background, and $l^j \in \{-1, 1\}$ is the corresponding label.

Subsequently, the linear classifier similar to [11] is learned by minimizing the loss function
\begin{equation}
J(w) = \frac{1}{M} \sum_{i=1}^{M} L(v^i, w, l^i) + \frac{1}{2} \|w\|_2^2,
\end{equation}
where $w$ is the parameter of the classifier based on the corresponding dictionary $D$, $\lambda$ controls the strength of the regularization term, and the logistic regression loss function $L(l, w, v) = \log(1 + e^{-lwv})$, where $v = [v^1, \ldots, v^T]$ is an augmented vector. So far, the classifier based on local sparse representation is built and defined by the pair of $C = [D, w]$. Hence we use $C$ as the corresponding classifier from now on.

Therefore, for tracking tasks, the likelihood of the candidate state can be defined by the classification score $c$ of the learned classifier.
\begin{equation}
c = \frac{1}{1 + e^{-wv'^{T}v}},
\end{equation}

Then, the candidate with the highest score is regarded as the tracking result, and the score is considered as the confidence value.

3. Tracking with a Coupled-Classifier Based on Multiple Representative Appearance Models

In our CCMM tracker, visual tracking tasks are categorized into two cases, smooth and abrupt, according to the degree of appearance change. In normal cases of smooth appearance variation, the coupled-classifier can get more accurate result and control the update orientation by the two classifiers working collaboratively. For the challenging cases of abrupt appearance variation, multiple models cooperatively support the coupled-classifier by the representative basic models recorded in the past life-span.

As described in CCMM Tracking Algorithm with pseudo code, the proposed tracker integrates the two strategies well. First of all, the classifier $\{D_i, w_i\}$ is constructed by initializing the dictionary and its arguments are trained by the ground truth in the first frame where the target object is labelled either manually or automatically. Subsequently, the candidates $s^i_1, \ldots, s^i_p$ ($p$ is set as 400 in the experiment) are predicted with affine transformation in every subsequent frame, and each classifier seeks the result in $s^i_1, \ldots, s^i_p$. Normally the coupled-classifier succeeds to track the target in most cases; however a template pool of multiple models is triggered when the challenging scenarios are encountered. For the next frame, the root and head classifiers each find the target, say $s^r_t$ and $s^h_t$, respectively. One of them is chosen as final result if the overlap ratio is higher than the threshold $\theta_r$ and $c_t \geq \theta_{bad}$, where $c_t = \max(c^r_t, c^h_t)$ is the related confidence value, and $\theta_{bad}$ is the threshold of basic quality; otherwise the multiple models select the candidate $s^m_t$ to compare with the candidates $s^r_t$ and $s^h_t$. Meanwhile, the templates of the head and root classifiers are updated to maintain the adaptive and discriminative capacity. Also, the most similar model in multiple models is updated with the template of the root or head classifier to keep its representative ability.

### CCMM Tracking Algorithm

1. At $t = 1$, initialize the models with the ground truth.
2. for $t = 2$ to the number of frames do
   - Predict the candidates $s^r_t, s^h_t$ with affine transformation;
   - Track the target with the coupled-classifier and get results $s^r_t$ and $s^h_t$, respectively.
3. if $\frac{\text{Ratio}}{\theta_r}$ and $c_t \geq \theta_{bad}$ then
   - Either the candidate $s^r_t$ or $s^h_t$ is accepted as result.
4. else
   - Resort to the multiple representative models, and seek the best candidate $s^m_t$;
   - one of the candidate $s^r_t$, $s^h_t$ or $s^m_t$ is accepted as result depending on the corresponding confidence value;
   - Update the most similar model or create new model with the template of the root classifier.
5. end if
6. Update the templates of the root and head classifier.
7. end for

3.1 Collaborative Tracking with the Coupled-Classifier

In order to track the object in smooth appearance variation, our coupled-classifier attempts to solve the core problem of gradual degradation of templates. Before introducing our coupled-classifier, the essential problem with online update and the underlying factors causing drift need to be discussed. At first, an intrinsic problem of online update is analyzed by explaining an ideal update strategy described in Fig. 1. In order to construct an ideal update strategy of online tracking problem, we assume the ground truth $(s_0, s_1, \ldots, s_n)$ are known, which are corresponding to an input image sequences $(X_0, X_1, \ldots, X_n)$. When a new tracking result $s'_t$ is obtained, its quality is estimated by comparing $s'_t$ with $s_t$. As long as the quality is good enough, $s'_t$ may

**Input:** $X_0, X_1, X_2, \ldots, X_n$

**Classifier:**

- $X_0$ -> $X_1$ -> $X_2$ -> ...
- $s_t$'s are predicted
- $s_t$'s are compared
- $s_t$'s are accepted

**Tracking result:**

- $s'_t$ obtained
- $s'_t$ is compared
- $s'_t$ is accepted

**Ground truth:** $s_0, s_1, s_2, s_3, s_4, \ldots, s_n$

**Fig. 1** The ideal online tracking based on hypothesis.
be accepted as a new template for updating the classifier. Otherwise, the classifier keeps the same. Figure 1 shows an example when $s_1'$ and $s_2'$ are accepted to update the classifier. Thus the belief of $s_0$ is inherited entirely through $s_1'$, $s_2'$, but cross $s_2'$. However in reality, since only $s_0$ (but not the others) is known, we need to find something that does a similar role of the unknown ground truths.

Subsequently, the underlying reasons of the drift in visual tracking can be found by analyzing the update process. An online tracker often chases the most adaptive appearance model by updating its template with the last result. Even though the result can be the best candidate, it naturally contains a slight reconstruction error compared to the ground truth due to the nature of holistic representation of the object appearance with a rectangle or ellipse silhouette, and the sampling error with slide windows or other dynamic models (also called a motion model). Furthermore, it becomes worse when the result is not the best candidate if the tracker is not configured with non-optimal arguments. The worst thing is that the bias probably can be accumulated while the tracker algorithm runs repeatedly, i.e. the model is degraded gradually and a drift happens eventually. In a word, an error exists naturally between the tracking result and the ground truth, and the drift inevitably happens unless the accumulation of the error can be controlled.

According to the above analysis, there are two effective ways to reduce the drift by controlling the error accumulation. One is by improving the quality of the tracking result, i.e. by reducing the error volume. In this way, the time it takes for the error to accumulate to the allowable limit can be extended, that is the probability of drift may decrease. Another way is by controlling the orientation of the error variation (also called update orientation). As is known, the worst is the case when the error is accumulated successively by similar variation, which can incur an immediate drift. Furthermore, the error is often accumulated fast with a single adaptive or static model, as is demonstrated in Fig. 2, which is the comparison of X coordinate of the center points among the experiment results of the pure static, adaptive models and the ground truth. Apparently, the static model is too stable to adapt the variation of the target appearance; as well the adaptive model is too agile to the disturbance. Therefore, we propose a coupled-classifier which attempts to get accurate results, and to neutralize the accumulation of the error by guiding the update orientation.

As shown in the structure of coupled-classifier in Fig. 3, the head classifier is updated with the previous tracking result $s_{i-1}$ ($s_{1-i}$ or $s'_{i-1}$) which is the result for the previous frame $X_{i-1}$, i.e. the head classifier is a descendant of the coupled-classifier. Thus, the template of the root classifier is nearer the known state (the extreme is the ground truth in the initial frame for online tracking) compared to the template of the head classifier which is adaptive to the last appearance of the target.

For the next frame $X_i$, the tracking process is described in the algorithm Tracking With a Coupled-classifier. First of all, predicting candidates of the target in the frame $X_i$ is implemented with affine transformation, which is mentioned in Sect. 2.2. In the following process, the classifiers seek the best result in these candidates. Then, both the classifiers work independently to produce two pairs of a result and a corresponding confidence value $(s_i^h, c_i^h), (s_i^c, c_i^c)$ separately. Then, one of them will be chosen as the result according to two factors: the overlap ratio of the candidates $s_i^h, s_i^c$ and the related confidence value $c_i^h, c_i^c$. The higher overlap ratio, the more credible results, since the ratio of an ideal case is 1 (i.e. if $s_i^h$ is equal with $s_i^c$, the two classifiers would choose the same candidate). If the ratio is high enough, the result can be accepted; otherwise the coupled-classifier resorts the template pool of multiple representative models. Further, the suggestion $s_i^h$ or $s_i^c$ is accepted depending on the related confidence value $c_i^h, c_i^c$. It is obvious that the higher the confidence value the better the quality of result under the same overlap ratio. Meanwhile, there are two schemes to update the two classifiers, i.e., one of the candidate $s_i^h$ or $s_i^c$. If $s_i^h$ is selected, that means the head classifier is more suitable for the current state of the target. Then, the update scheme is that the root classifier is updated with the head classifier, and then the head classifier is updated with its result $s_i^h$ (also it is the last final result). The other is the case when $s_i^c$ is selected as final result because the root classifier is more suitable for the current state, as well the root classifier re-

\[ \text{ratio} = \frac{\text{area}(R_T \cap R_G)}{\text{area}(R_T \cup R_G)} \]

\[ \text{The overlap ratio is defined by the PASCAL VOC2010[5]} \text{ criterion.} \]
main its template and the head classifier is updated with the result of the root classifier $s'_r$.

**Tracking with Coupled-classifier**

At $t = i$, the head classifier was trained by the result of the coupled-classifier in the frame $X_{i-1}$:

1. For the coming frame $X_i$;
2. The root and head classifiers track the object separately and get the results, say the candidates $s'_r$ and $s'_h$, respectively.
3. if $\text{Ratio} \geq \theta_i$:
   - The candidate $s'_r$ is accepted as the final result;
   - The root is updated with the template of head classifier;
   - The head is updated with the candidate $s'_h$.
4. else
   - The candidate $s'_h$ is accepted as the final result;
   - The template of the root classifier is kept;
   - The head classifier is updated with the candidate $s'_h$,
5. end if

- Trigger the multiple representative models.

Reviewing the collaborative tracking process, twofold nature of the coupled-classifier can be observed. First, the root classifier plays a similar role as the ground truth, as described in Fig. 1. It supervises and guides the variation of the head classifier. In other words, the root classifier constrains the update orientation of the head classifier by validated results, meanwhile the quality of the results can be guaranteed. Second, the template of the head classifier, as a good validated suggestion, is employed to update the root classifier if it is good enough. With a good suggestion, the root classifier can be more adaptive to the current state lowering the risk of update with a candidate of bad quality. That is, the root classifier can be updated with the validated candidate of good quality and adaptive to the current which is suggested by the head classifier. Thereby, the capacity of “root” can be inherited to a new root classifier sustainably.

Therefore, the proposed coupled-classifier controls the error accumulation to reduce the drifting problem by the two effective ways. One is the result is validated and further being template of better quality to update the head classifier. The other is error accumulation is controlled by guiding the update orientation. For the head classifier, it is updated with the result of high confidence value, and validated by the overlap ratio. For the root classifier, it is only updated with the template of the head classifier whose result is already agreed by the root classifier. Consequently, the proposed coupled-classifier ensures the current tracking result faithful to the previous result, which can be back to the ground truth in the initial frame; meanwhile the adaptive capacity is also obtained by the trial of the head classifier.

3.2 Cooperative Tracking with Multiple Representative Classifiers

For an online tracker, it is required to continually update the template to adapt the change of the last state since the initial frame where the ground truth is given, yet only partial information of the target is provided. Consequently, they fail quite often in dynamic and complex scenes with drastic appearance variation such as persistent disturbance from background, occlusion, varying illumination, camera motion, and deformation.

We propose a template pool to account for such abrupt appearance variation of the target object. For the challenging cases, it is reasonable and effective to use the proposed strategy based on multiple representative appearance submanifolds. Naturally, 2D appearance information is just partial information of a 3D object; as well a reasonable and possible assumption is that the object may show multiple aspects of its appearance in different life-span. Explicitly, it is supported theoretically and intuitively that the appearance feature of the target can be approximately decomposed into multiple subspaces of the target appearance over time. Intuitively, if all the representative appearance submanifolds are maintained, the tracker can detect the target as accurately as most possible.

Therefore, for online visual tracking, it is a good choice to employ multiple representative submanifolds to collaboratively recognize the object accurately rather than a single one, especially for tracking in challenging scenarios like large deformation of shape and recovery from occlusion. Each model in the template pool may be considered as a memory of one representative feature of the target appearance from the past learning periods. When the coupled-classifier cannot tackle the challenging cases well, the best model in template pool is picked out by joint confidence values of the multiple models. Further, its tracking result is decided as the final result if the related confidence value is higher than that of the coupled-classifier. Moreover, the model is employed to update the template of the root classifier.

The best model is selected by a dynamic mechanism. For the known $R$ (normally the value $R$ is set to 6 in experiments) representative models, top $r$ ($r$ is set to 3 in experiments, which is up to half the number of total representative models commonly.) of them are selected according to their maximum confidence value. The reasonable reason is that, commonly there are a few and only a few models may be closely similar to the target due to the independence among the representative basic models. Then the best model is decided by a data fusion scheme, which is illustrated by an example as follows. Assume that three models $C_1$, $C_3$ and $C_4$ are selected. Further, $C_1$ suggests the a-th candidate $s^a_q$ as the best candidate with the confidence value of $c^a_{1q}$; similarly, $C_3$ recommend $s^a_q$ with $c^a_{3q}$, and $C_4$ regarded $s^a_q$ as $c^a_{4q}$. Then, the fused score with the selected classifiers is formulated as

$$c_{1q} = \max\{c^3_{1q} + c^4_{1q} + c^4_{1q}\}.$$  \hspace{1cm} (6)

If $q = b$, (6) is satisfied. It means that the best final result is $s^b_q$ with the likelihood $c^b_{1b}$ based on the model $C_1$.  

1831
The last but not the least is how to maintain the discriminative and representative capacity of multiple models. In order to track the target robustly even in challenging cases (such as vast change of appearance, occlusion) happen, it is helpful and necessary to maintain the representative features of the target appearance which occurred in the past temporal space, as well are learned and recorded by the tracker. Therefore, a dynamic scheme of online update is designed to guarantee the representativeness and the discriminative capacity of each basic model. Further, the essentiality of the update diagram concerns about how to maintain representativeness and adaptability of the basic models rather than the number.

The update strategy for multiple representative models includes three steps, as described in Update Algorithm for Multiple Representative Models, which is implemented after employing the mechanism of multiple models. The first step is to initialize all models with the ground truth. The second one is that every model is configured with the “seed” feature, except the last one which keeps the original template. The seed features are from the template of the root classifier when the multiple models are triggered and the coupled-classifier still have much higher confidence value $c_f^t$ for tracking the current frame than $c^m_f$, the confidence value of the best model in the multiple models pool, thus it can be considered as new feature. The third is to update the most similar model by the template of the classifier whose result is accepted as the final result and $c_f^t$ is higher than $c^m_f$, after the mechanism of multiple models is triggered. Additionally, the condition for triggering the multiple models is, whether the overlap ration of the two results of the root and head classifier is higher than threshold $\theta_t$ or not, but $c_f^t$ is smaller than the threshold of bad result quality $\theta_{bad}$, or the ratio cannot prove the good quality of the result of the head classifier and the value of the root classifier $c_f^t$ is also smaller than the threshold of good result quality $\theta_{good}$.

After judging by multiple models, our algorithm can recognize the case of occlusion with the two criterions. One is the biggest overlap ratio value is larger than $\theta_t$, as well all confidence values are less than $\theta_{bad}$. That means the current tracking result is much different with all known templates. Another one is the biggest overlap ratio value is less than $\theta_t$, as well all confidence values are less than $\theta_{good}$. It means the current tracking result is much different with the results of all known templates, as well its classifier is not very confident. For the case of occlusions, the tracking result is not considered as new features so that all the templates are not violated even the target is occluded a long time.

**Update Algorithm for Multiple Representative Models**

1: At $t = 1$, initialize $C_t = \{D_t, w_t\}$, $i = (1 \ldots R)$ with the ground truth
2: for $t = 2$ to the number of frames do
   Run tracking with the coupled-classifier and get result $\delta_t$ related with the confidence value $c_f^t$
3: if $(\text{Ratio} \geq \theta_t$ and $c_f^t < \theta_{bad})$ or $(\text{Ratio} < \theta_t$ and $c_f^t < \theta_{good})$ then
   Trigger the multiple models
4: if $(\# \text{of models} < R - 1)$ and $(c_f^t > c^m_f)$ then
   Configure a new model with the seed feature, the template of the root classifier
5: else if $c_f^t > c^m_f$
   Update the best model with the template of the root or head classifier, depending on the bigger one of $c_f^t$, $c^m_f$
6: end if
7: end if
8: end for

The benefits of the update diagram mainly depend on two aspects. First of all, the seed feature comes from the template of the coupled-classifier which is already validated, so it is credible. As well, the seed features ensure the models to represent different target appearance to their best. Secondly, the update strategy of updating the similar model is different from others, so that the models can keep their representativeness during the evolution process. In addition, the initial ground truth remains forever.

There are three main advantages of the proposed CCMM tracker: a) to keep the healthy state of the templates of the coupled-classifier by avoiding accumulation the reconstruction error; b) to get more accurate results by validating the tracking results; c) to guarantee the adaptability and discriminability of the appearance model by novel update strategies. As well, the effective fusing strategy is designed to combine multiple models.

### 4. Experimental Results

The proposed algorithm is implemented using MATLAB and configured with the following arguments: the number of the multiple models is set to 6; the numbers of positive and negative samples for $\ell_1$ classifier are both 30, and the size of the template for $\ell_1$ classifier is $24 \times 24$. In addition, only gray value of the patches is utilized as features. As well, the programs were run on the machine equipped with Intel(R) Core(TM) i5-2410 CPU @ 3.10 GHz, 4 G RAM.

In experiments, several challenging datasets employed to evaluate the performance of the proposed tracker, which are publicly available on websites (http: //ice.dlut.edu.cn/lu/publications.html; http: //groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1; http: //cv.snu.ac.kr/research/ -vtld/). As listed in Table 1, each video is with different challenging factors including pose variation, partial occlusion, background clutter, illumination variation, and scale change.

Aiming to demonstrate the advantage of the proposed

| Table 1 | Evaluated videos. |
|---------|-------------------|
| dataset | # frames | challenging factors |
| Caviar 1 [14] | 382 | partial occlusion, scale change |
| Caviar 2 [14] | 500 | partial occlusion, scale change |
| Car 4 [8] | 659 | illumination variation, scale change |
| Car 11 [8] | 393 | background clutter, illumination variation, scale change |
| Gir1 [13] | 501 | in-plane and out-of-plane rotations, scale change |
| Deer [4] | 71 | abrupt motion, large appearance variation, background clutter |
Fig. 4  Quantitative evaluation of the trackers in terms of position error (in pixels). This figure shows the overlap ratios for challenging datasets Caviar1, Caviar2, Girl, Car4, Car11 and Deer. Our algorithm is compared with six state-of-the-art trackers Bag [12], VTD [3], MIL [1], TLD [2], \( \ell_1 [7] \) and Two-stage [11].

Table 2  Average center error (in pixels). The best two are in red or blue.

|          | Bag  | VTD  | MIL  | TLD  | \( \ell_1 \) | Two stage | Ours   |
|----------|------|------|------|------|-------------|-----------|--------|
| Caviar 1 | 5.527| 3.998| 44.498| 5.593| 119.932| 47.758 | 1.748  |
| Caviar 2 | 5.191| 4.724| 70.269| 8.571| 3.243 | 3.889  | 2.079  |
| Girl     | 106.143| 21.443| 32.209| 23.358| 62.453 | 12.753 | 12.215 |
| Car 4    | 271.439| 12.299| 60.105| 18.797| 4.081 | 191.038| 4.825  |
| Car 11   | 74.975| 27.055| 45.465| 25.113| 33.252 | 8.098  | 1.721  |
| Deer     | 189.945| 11.920| 66.457| 25.653| 171.688 | 119.685| 8.579  |
| Average  | 108.870| 13.556| 55.501| 17.804| 65.735 | 63.8698| 5.196  |

On the other hand, the tracking overlap ratio represents the accuracy and stability of each algorithm. The overlap ratio is employed to evaluate a successful tracking rate by the detection criterion defined by the PASCAL VOC2010 [4], as mentioned before. Commonly, a tracking result is regarded correct if the overlap ratio is above 0.5. The overlap ratios of the trackers for all the evaluated videos are illustrated in Fig. 5 and the average overlap ratios are summarized in Table 3. In general, the proposed tracker achieves best performance in almost all the challenging sequences.

4.2 Qualitative Evaluation

For online visual trackers, it is the core issue to deal with the variation of target appearance such as occlusion, illumination variation, background clutter, in-plane and out-of-plane rotations and motion blur.

**Occlusion:** For the evaluated sequences Caviar1 and Caviar2 which have challenging factor of occlusion, algorithms Bag [12], VTD [3] and ours perform better, as illustrated in Fig. 6. In the video clip Caviar1, our method achieves the best performance and is followed by VTD, as shown in Fig. 6 (1). Our method gets accurate result before
the occlusion (the ID of image frame 95 is represented as #95, and the same as following) by the coupled-classifier, which provides a strong support to pass the heavy occlusion, then benefits from local sparse representation to perform well during the occlusion, recovering successfully from the occlusion (#126) and keeps accurate tracking to the end.

Also, the algorithms of VTD and Bag method get good results depending on multiple cues and the bag of multiple templates respectively. For the sequence Caviar2 in which the illumination change is larger than that in Caviar1, the methods $\ell_1$ [7], Two-stage [11] and ours perform better. Our algorithm achieves the best similar to the case in Caviar1, and which is followed by Two-stage and $\ell_1$ tracker, as shown in Fig. 6 (2). But, both Two-stage and $\ell_1$ tracker perform much worse than ours when occlusion happens from #154 to #222.

Illumination change and Background clutter: The results from two challenging sequences Car 4 and Car 11 with significant variation of illumination change and background clutter are shown in Fig. 7 (1) and Fig. 7 (2). In the sequence Car 4, there is a drastic lighting variation while the car runs in the shadow of the bridge or the trees. The methods $\ell_1$ [7], Two-stage [11] and ours give better results, and our tracker is the best one. Especially, our tracker works well under drastic variation of illumination such as #199, #237 and #306.

In the evaluated sequence Car 11, the small car with low contrast is a very cluttered background with drastic illumination change. Our algorithm performs the best and is followed by the method Two-stage [11]. However, the other trackers drift away after #282 due to vast lighting variation and the cluttered background. Only the Two-stage tracker and ours can catch the target after #300, but our results are more accurate.

Rotation: For the evaluated sequence Girl which is with in-plane and out-of-plane rotations, algorithms VTD [3] and ours perform better, as illustrated in Fig. 8 (1).

Motion blur: Fast motion of the target or the camera may blur the appearance of the target like #12 and #41 in evaluated sequence Deer whose results are shown in Fig. 8 (2). The trackers VTD [3], TLD [2] and ours perform better, and our algorithm achieves the top one. The blurred target object is almost indistinguishable, which results in failure of most tracking algorithms after the frames, especially frame #12. At #41, the TLD tracking method [12]...
is confused by a similar object, and then only the VTD and our trackers can work well.

5. Conclusions

Compared to existing algorithms, the proposed approach performs better with two contributions.

Firstly, a novel tracking strategy, a coupled-classifier, is proposed. The root and head classifiers work collaboratively to get more stable tracking results and update the templates with more credible candidates. Each classifier, being trained with sparse codes of local image patches, achieves more discriminative capacity than holistic methods to track the target separately. In the model of the coupled-classifier, the root classifier guides the head classifier along the “right” orientation by a verified tracking result; meanwhile its result of the head classifier may provide a qualified suggestion to update the root classifier.

Secondly, the mechanism of the multiple representative appearance models is designed. The multiple representative appearance models of the target are maintained to support the tracker to handle challenging scenarios. Also a scheme is designed to employ multiple models effectively. As well, an update strategy for the template pool is proposed to keep the representativeness of each template, enabling the pool to explicitly represent different aspects of the target appearance to the best of its ability. In addition, with the favor of sparse representation, the learned classifiers are able to generate a more accurate similarity measure to distinguish the target object from background.

For a further research work, it is an interesting art to utilize the learned multiple classifiers in an efficient and effective way.

Acknowledgments

This work was supported by GRRC program of Gyeonggi province ([GRRC SUWON2012-B1] Cooperative CCTV Image Based Context-Aware Process Technology).

References

[1] B. Babenko, M.-H. Yang, and S. Belongie, “Visual tracking with online multiple instance learning,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), pp.983–990, June 2009.
[2] Z. Kalal, J. Matas, and K. Mikolajczyk, “P-N learning: Bootstrapping binary classifiers by structural constraints,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.49–56, 2010.
[3] J. Kwon and K.M. Lee, “Visual tracking decomposition,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.1269–1276, 2010.
[4] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, “The PASCAL visual object classes challenge 2010 results,” 2010.
[5] I. Matthews, T. Ishikawa, and S. Baker, “The template update problem,” IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.6, pp.810–815, 2004.
[6] X. Mei, E. Blasch, and H. Ling, “Minimum error bounded efficient tracker with occlusion detection,” Computer Engineering, vol.3, no.1, pp.1703–1709, 2011.
[7] X. Mei and H. Ling, “Robust visual tracking using L1 minimization,” IEEE 12th International Conference on Computer Vision, pp.1436–1443, 2009.
[8] D.A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, “Incremental learning for robust visual tracking,” Int. J. Comput. Vis., vol.77, no.1-3, pp.125–141, 2007.
[9] S. Agarwal and D. Roth, “Learning a sparse representation for object detection,” ECCV, pp.113–130, 2002.
[10] R. Sukthankar, “PCA-SIFT: A more distinctive representation for local image descriptors,” CVPR, pp.506–513, 2004.
[11] Q. Wang, F. Chen, and W. Xu, “Online discriminative object tracking with local sparse representation,” 2012 IEEE Workshop on Applications of Computer, pp.425–432, 2012.
[12] F. Yang, H. Lu, and Y.-W. Chen, “Bag of features tracking,” 20th International Conference on Pattern Recognition, vol.153, no.1, pp.153–156, 2010.
[13] W. Zhong and H. Lu, “Robust object tracking via sparsity-based collaborative model,” International Conference on Computer Vision, 2012.
[14] CAVIAR. http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDA TAI
[15] J.H. Yoon, D.Y. Kim, and K.-J. Yoon, “Visual tracking via adaptive tracker selection with multiple features,” ECCV 2012, Part IV, LNCS 7575, pp.28–41, 2012.
[16] M. Godde, F.M. Roth, and H. Bischof, “Hough-based tracking of non-rigid objects,” 2011 ICCV, pp.81–88, 2011.

Deqian Fu is a PhD. candidate in the Department of Computer Science at the University of Suwon in Korea, and an associate professor in the School of Informatics at Linyi University in China. His research interests include computer vision, machine learning, networking, and security and surveillance technology. He received his M.S. (2005) degree in Computer Science from Shandong University of Science and Technology in China, and his B.S. (1996) degree in Mechanical and Electronic Engineering from Jiangsu University in China.

Seong Tae Jhang is a professor in the Department of Computer Science at the University of Suwon in Korea. His research interests include multi-processor system, cache architecture, memory model, security and surveillance technology. He received his B.S. (1986) degree in Computer Engineering from Seoul National University, his M.S. (1988) degree in Computer Engineering from Seoul National University, and his Ph.D. (1994) degree in Computer Engineering from Seoul National University.