International Journal of Advanced Robotic Systems

OPEN ACCESS
ARTICLE

Robust Tracking with Discriminative Ranking Middle-level Patches

Regular Paper

Hong Liu[1], Zilin Liang[1] and Qianru Sun[1,*]

1 Engineering Lab on Intelligent Perception for Internet of Things (ELIP), Shenzhen Graduate School, Peking University, Shenzhen, China
* Corresponding author E-mail: qianrusun@sz.pku.edu.cn

Received 24 Jan 2014; Accepted 28 Feb 2014

DOI: 10.5772/58430

Abstract The appearance model has been shown to be essential for robust visual tracking since it is the basic criterion to locating targets in video sequences. Though existing tracking-by-detection algorithms have shown to be greatly promising, they still suffer from the drift problem, which is caused by updating appearance models. In this paper, we propose a new appearance model composed of ranking middle-level patches to capture more object distinctiveness than traditional tracking-by-detection models. Targets and backgrounds are represented by both low-level bottom-up features and high-level top-down patches, which can compensate each other. Bottom-up features are defined at the pixel level, and each feature gets its discrimination score through selective feature attention mechanism. In top-down feature extraction, rectangular patches are ranked according to their bottom-up discrimination scores, by which all of them are clustered into irregular patches, named ranking middle-level patches. In addition, at the stage of classifier training, the online random forests algorithm is specially refined to reduce drifting problems. Experiments on challenging public datasets and our test videos demonstrate that our approach can effectively prevent the tracker drifting problem and obtain competitive performance in visual tracking.

Keywords Middle-level Patches, Selective Feature Attention, Random Forests, Tracking-by-detection

1. Introduction

Visual tracking plays a key role in a variety of practical fields, such as autonomous robot systems, human-robot interaction, driver assistance, security surveillance and so on. Despite tracking having drawn much attention in research, serious problems still exist in realistic applications. Scale, pose and illumination changes confuse the tracker, while background clutters and occlusion from other objects distract the tracker. Most state-of-the-art models of object tracking mainly focus on two aspects [1], object representation (i.e., object model) and mode seeking. Object representation describes the basic criteria of mode seeking, hence helps to locate candidate targets in videos. Recently, tracking-by-detection models, which formulate tracking as a binary classification between targets and background, have shown great promise in state-of-the-art frameworks [2,3,4]. Such methods involve the continuous detection in individual frames and the association of detections across frames. In contrast to background modelling-based trackers, they are generally robust to changing background and moving cameras. However, the existing challenge of such models when applied to real-world scenarios is the unavoidable drifting problem. That is, when learning a new model to adapt to appearance change and to maintain model plasticity, the model stability would be reduced. Hence, the distinctiveness of an appearance model is very important for improving tracking efficiency against the drifting problem.

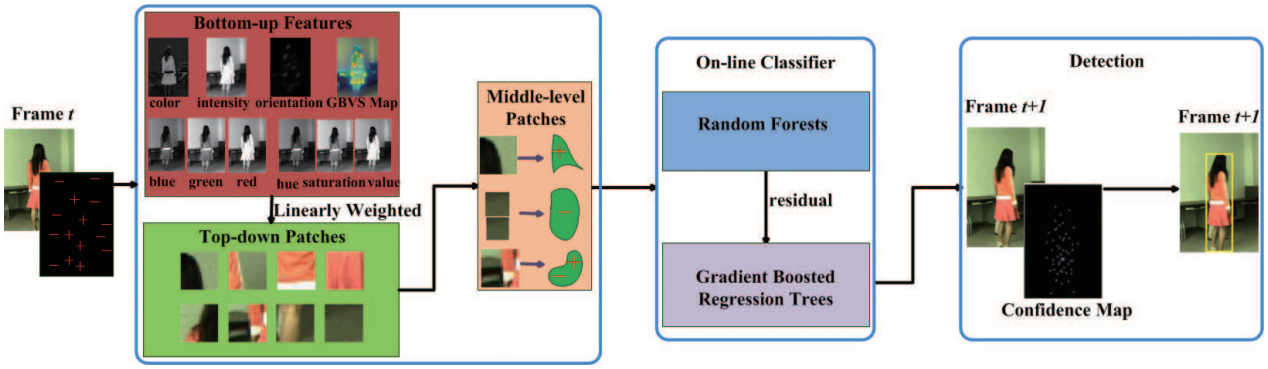

Figure 1. Framework of our tracking-by-detection algorithm

Grabner proposed an online boosting algorithm to generate strong classifiers as a tracker, in which bottom-up features (e.g., colour, orientation, intensity, etc.) were not considered in the appearance model [2]. Another work, called Ensemble Tracking [3], proposed to embed diverse feature vectors in an Adaboost framework to formulate an object/background appearance model. Its drawback was that several heterogeneous cues were merged in the same subspace. Penne *et al.* improved a modular version of Ensemble Tracking combined with a Markov Chain Monte Carlo particle filter [5]. This indicated the necessity to find the optimal combination of feature configuration that improves the tracking process. Grabner *et al.* [6,7] used semi-supervised online boosting to alleviate the drifting problem. Babenko *et al.* introduced multiple instance learning to describe and locate targets in videos where a bounding box including the object was considered as a positive bag and others as negative ones [8]. Supancic *et al.* [9] used self-paced learning to select reliable frames from which to extract additional training data. In this way, they obtained a good appearance model. They also proved that an appearance model is more effective than a strong motion model. Tang *et al.* proposed a discriminative ranking list tracker which constructed a pair of different scale models, and alleviated the distraction from backgrounds[10]. Their ranking lists were gained according to K-NN with Euclidean distance metrics. For comparison, our ranking middle-level patches are based on a totally different mechanism where our ranking metric is composed of discrimination scores.

In summary, aforesaid models mainly used top-down cognitive attention (e.g., faces, humans, etc.) to represent a target. They ignored the bottom-up features (e.g., colour, orientation, intensity, etc.) which are highly discriminative for individual objects. Therefore, their models are not discriminative enough and models drift easily. In contrast, some works provided tracking models based on salient bottom-up features [11-14]. Although they obtained compelling results, they lagged behind primates' performance in real scenes. For example, primates can recognize a person and track it when the person isn't salient but is meaningful from backgrounds, while the above methods cannot realize that using only bottom-up features. The reason for the gap largely lies in the role of top-down features [15]. Hence, it will be beneficial to combine top-down and bottom-up features for constructing models and reach primates' performance on object tracking.

In this paper, we propose a novel framework by integrating high-level top-down and low-level bottom-up features to boost the distinctiveness of the appearance model. For high-level top-down features, large-scale patch-wise features are extracted because they contain more appearance information than pixel-wise features. However, their sparser distribution and lower sensitivity than pixel-wise features, result in their disadvantage of lower location accuracy [16]. Therefore, low-level pixel-wise features are extracted to compensate for these drawbacks. More importantly, selective feature attention is utilized to determine feature scores in bottom-up spaces according to the discrimination ability that best separates the target from backgrounds. These scores are used to linearly weight corresponding top-down patches. As a result, rectangular top-down patches are clustered into irregular ranking middle-level patches. Our discriminative appearance model is described by these ranking middle-level patches, and treats tracking as a three-class classification problem, as shown in Figure 1.

In tracking-by-detection methods, another key part is how to train and maintain the classifier. Random forests (RFs) [17] and their melioration [6] have attracted considerable attention in computer vision for their excellent characteristics, such as they are more robust to noise than Adaboost [18], paralleled easily, not prone to overfitting and so on [19]. Hence, online random forests (ORFs) are adopted as our basic classifier for their inherent multi-class property. Additionally, online gradient boosted regression trees (GBRT) is novelly utilized to reduce the current residuals in gradient direction, in order to refine the training errors of RFs classifier.

The remainder of this paper is organized as follows, section 2 describes the proposed appearance model with top-down and bottom-up features. The online refined random forests classifier is presented in section 3. Section 4 details extensive experiments on challenging public datasets and test videos recorded by us on a mobile robot. Finally, we come to conclusions and discuss possible extensions in section 5.

2. Appearance Model with Top-down and Bottom-up Features

Our goal is to obtain a discriminative target appearance model. The whole framework of our method is shown in Figure 1. Firstly, target tracking is determined by hand in frame t, and bottom-up features and top-down patches are based on this. Selective feature attention calculates the scores of bottom-up feature spaces according to discrimination ability of the features. Then, three kinds of ranking middle-level patches are derived to be used to train our refined random forests classifier as samples. To adapt to appearance changes, an online learning algorithm is adopted to update the refined random forests classifier. The trained classification model recognizes the target from backgrounds on frame t+1, updating the samples of the target and its backgrounds at the same time. Finally, we come to run mean-shift [20] to identify the peek of the classification result (confidence map), i.e., location of the target. The tracking procedure repeats frame-by-frame. The collaboration of three-class ranking middle-level patches and online refined random forests is the key technique of our proposed method.

2.1 Visual Feature Spaces

Bottom-up features

Traditionally, intensity, orientation and colour can be used for saliency estimation [21]. In this paper, our bottom-up features also include other features which have shown correlation with bottom-up visual attention and have underlying biological plausibility.

- Three channels of colour spaces (such as RGB, HSV, no use for grey images)
- intensity, texture, orientation and colour (Red/Green and Blue/Yellow) contrast channels as calculated by Itti and Koch's saliency method [21].
- Graph-based visual saliency (GBVS) [22] bottom-up saliency models

Top-down features

For human-robot interaction and visual surveillance, faces (human or animal) can easily draw primates' attention. Hence, two excellent goal-driven, top-down cognitive features are adopted into our feature spaces.

- Haar-like features proposed by Viola and Jones [23].
- HOG (histogram of oriented gradients) features proposed by Dalal [24].

Dalal, who proposed HOG, believes that an object can be represented by the statistics of the local edge directions [24]. In our method, edges are divided into eight directions. Each patch is represented by a 40-dimension vector composed of five 8-bin histograms. Haar-like features are boosted by classifiers, as proposed in [2].

2.2 Selective Feature Attention Mechanism

The efficiency of tracking depends on the discriminative ratio between the target and its backgrounds [13]. Tracking a person in red cloth under the sun is very easy due to its salient colour. However, it is very difficult to keep tracking the person when he/she walks into a shadow.

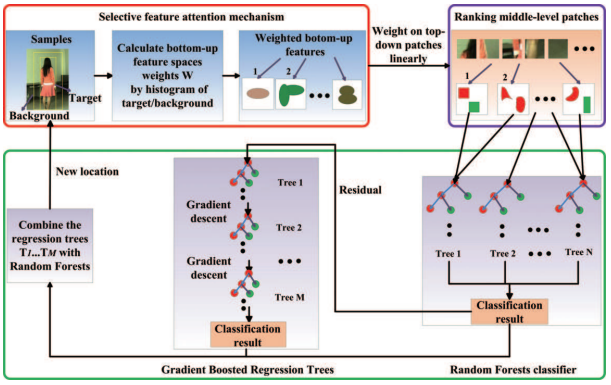

Figure 2. Overview of training classifier with ranking middle-level patches

At this time, red is no longer salient and we should shift our attention from the red colour to the distinctive shape in the feature spaces. Therefore, the attention mechanism of feature selection [11,25] plays an important role in the learning procedure to find the most discriminative feature space.

In this paper, the contribution of each selected feature is calculated by the variance ratio of the *log* likelihood function [11] which has been proved effective in [13]. $p(i)$ denotes the discrete probability distributions of one stimulus feature in target and $q(i)$ denotes the discrete probability distributions of the feature in backgrounds. They are separately estimated by normalizing their feature histograms $H_T(i)$ and $H_B(i)$ over pixel numbers n_T and n_B in them,

$$p(i) = \frac{H_T(i)}{n_T}, q(i) = \frac{H_B(i)}{n_B} \tag{1}$$

where $H_T(i)$ and $H_B(i)$ are obtained from target and background windows. Index i ranges from 1 to 2^b indicating patches, and b is the number of histogram buckets.

The *log* likelihood of the feature i is then given as,

$$L(i) = log\frac{max\{p(i),\sigma\}}{max\{q(i),\sigma\}} \tag{2}$$

where σ is a small value like 0.001 that prevents dividing by zero or taking the *log* of zero.

The variance ratio $VR(L;p,q)$ of $L(i)$ is calculated to quantify the feature's contribution and to distinguish the target from backgrounds. Given a discrete probability density function $d(i)$, the variance of $L(i)$ with respect to d is calculated as follows,

$$var(L;d) = \sum_i (d(i)L^2(i)) - [\sum_i (d(i)L(i))]^2 \tag{3}$$

The variance ratio of the *log* likelihood function $L(i)$ can now be defined as,

$$VR(L;p,q) = \frac{var(L;(p+q)/2)}{var(L;p) + var(L,q)} \tag{4}$$

A feature receives a high score if it renders the target more salient than distracters in backgrounds, and vice versa. Updating this score is shifting attention in bottom-up feature spaces, while the appearances of target and backgrounds are constantly changing. In our method, the variance ratio of the *log* likelihood function for each feature is calculated and normalized to determine the score s_i,

$$s_i = \frac{VR_i}{\sum_{i=n}(VR_i)} \tag{5}$$

2.3 Irregular Ranking Middle-level Patches

After discrimination scores of bottom-up features are obtained, they are used to linearly weight corresponding top-down patches. On one hand, if one patch is weighted with different scores, it will be segmented into different irregular patches. On the other hand, if different adjacent patches are weighted with approximate scores, they will be clustered into one irregular patch. Hence, the rectangular top-down patches are clustered into irregular ranking patches, which are in the form of a middle level between high-level patches and low-level features. These patches are called discriminative *ranking middle-level patches* in this paper. Considering the fact that one patch may contain both target and backgrounds since their appearances are similar with nearby scores, this paper models object tracking as a three-class classification issue. Three classes correspond to three kinds of patches, patches only containing the target, patches only containing backgrounds and patches containing both target and backgrounds. Target and backgrounds patches will be adaptively tuned in each round of training. An overview of the training classifier with irregular ranking middle-level patches is depicted in Figure 2.

3. Online Refined Random Forests Classifier

In a tracking-by-detection algorithm, another key part is how to train and maintain the classifier besides object representation [26]. RFs [17] have piqued researchers' interest as they demonstrate better or at least comparable performance to other state-of-the-art methods in classification, keypoint recognition and clustering applications [18,19,27,28]. More importantly, (1) inherent multi-class classifiers RFs are adapted to our three-class classification problem. (2) RFs can tell the importance of features in training, thus our proposed ranking patches with high weights can be judged by tree nodes preferentially. Hence, online random forests classifiers are used as basic classifiers. Taking the importance of precision into consideration, the classification results of RFs are refined by GBRT. The derived residual of the loss function by training RFs is used to initialize GBRT. Then GBRT corrects the residual in gradient directions. In this way, learning of wrong information can be effectively reduced, so that drifting problem can be alleviated. To cope with continuous model changes in tracking, we adopt online growing trees to constitute RFs and GBRT. In the following subsections, RFs and GBRT will be briefly introduced.

Algorithm 1 Online Refined Random Forests

INPUT: Sequential training sample $< x_i, y_i >$, the minimum number α, the minimum gain β, learning rate γ, the size of RFs N, the size of GBRT M
OUTPUT: T
BEGIN:
// Random Forests
for $t = 1 \rightarrow N$ **do**
  $f_t(x_i, \theta_t) = DecisionTree((x_i, y_i), \alpha, \beta)$
  update $OOBE_t$
  if $OOBE_t > rand()$ **then**
    // Discard the tree
    $f(x_i, \theta_t) = newDecisionTree((x_i, y_i), \alpha, \beta)$
  end if
end for
return $\mathcal{F} = \{f_1, ..., f_N\}$
// Initialization GBRT
$r_i = y_i - \mathcal{F}(x_i)$
for $t = 1 \rightarrow M$ **do**
  $T_t(x_i) = DecisionTree((x_i, r_i), \alpha, \beta)$
  // Update the residual of sample x_i when the number of samples is larger than α
  $r_i \leftarrow r_i - \gamma T_t(x_i)$
end for
// Combine the regression trees $T_1, ..., T_M$ with RF $\mathcal{F}$
$T = \mathcal{F} + \gamma \sum_{t=1}^{M} T_t$
return T
END

3.1 Random Forests

RFs are an ensemble of decision trees [17]. For each sample, its classification result is the weighted sum of all trees. Trees in RFs gain random samples with bagging for training, and select random features to evaluate finding the best spitting point at each node. RFs are an inherently parallel algorithm in that every single tree is independent from earlier trees. Another advantage of RFs is that they can provide extra information about the training dataset. Out-Of-Bag (*OOB*) samples of a tree which are not included during the training can be used to estimate the generalization error, called Out-Of-Bag-Error (*OOBE*) [17].

3.2 Gradient Boosted Regression Trees

Similar to RFs, GBRT is a machine learning technique which is based on tree averaging. GBRT sequentially adds a new tree in each iteration. The new tree focuses on samples that are responsible for the current remaining residual. In each iteration, GBRT uses boosting to reduce the current residual and improve the last results in the gradient direction [29], as illustrated in the gradient boosted regression trees of Figure 2.

Let $T(x_i)$ denote the current classification result of sample (x_i, y_i), where x_i denotes the values of the i^{th} feature vector, and y_i is the label of the sample. Furthermore, assume that $\mathcal{L} = (T(x_1), ..., T(x_n))$ denotes a continuous, convex and differentiable loss function which reaches its minimum when $T(x_i) = y_i$. In this paper, the loss function is equal to square loss function, just as follows,

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{n}(T(x_i) - y_i)^2 \quad (6)$$

GBRT performs a gradient descent in the sample space, that is, during each iteration the classification $T(x_i)$ is updated with a gradient step, as follows,

$$T(x_i) \leftarrow T(x_i) - \gamma\frac{\mathcal{L}}{T(x_i)} \quad (7)$$

where γ is the learning rate. In the case where $\mathcal{L}$ is the squared loss, the gradient becomes the residual, i.e.,

$$r_i = y_i - T(x_i) \quad (8)$$

3.3 Online Refined Random Forests

Since RFs and GBRT are based on decision trees, we adopt online tree growing strategy in [19] to train the online RFs and online GBRT classifiers. For each tree, the split principle of each node is that tests at node satisfy $g(x) > \theta$, where $g(x)$ is a test function, and θ is a threshold meaning quality measurement (e.g., information gain or Gini index). In online mode, the test function is randomly generated and the threshold θ is selected randomly, which are also adopted in an extremely randomized forest. Then the best tests and θ are determined by a quality measurement. Moreover, a node splits according to the statistics of samples falling in it. In online mode, statistics are gathered over time. Therefore, a node decides when to split depending on, (1) if there has been enough samples in a node to give a robust statistics and (2) if the splits are good enough for the classification purpose. In order to get a more robust estimation of statistics, two hyper-parameters are introduced which must be met, (1) α, the minimum number of samples a node has to see before splitting, (2) β, the minimum gain a node has to achieve when splits. The grown tree based on these tree-growing strategies is denoted as $DecisionTree((x,y),\alpha,\beta,)$.

The proposed online refined random forests algorithm is shown in Algorithm 1, $F_t(x_i)$ and $T_t(x_i)$ denote the t^{th} tree's classification results of sample (x_i, y_i) in RFs and GBRT respectively. Trees in RFs and GBRT are derived by online $DecisionTree((x,y),\alpha,\beta,)$ novelly instead of the off-line classification and regression trees. Moreover, the online classifier learns new information for model updating, and forgets old information by discarding the entire tree whose $OOBE$ is larger than the threshold in RFs.

Traditionally, GBRT is initialized with the all-zero function $T_0(x_i)$, then the residual is $r_i = y_i$, leading to the true convergence of $\mathcal{L}$ not holding in practice. This is because 1) in each iteration, the gradient is only approximated, 2) for true convergence, the learning-rate γ should be small, requiring an unrealistically large number of iterations $M >> 0$ [30]. In the online refined random forests algorithm, the initial residual r_i is set with the residual of the RFs classification result, i.e., $r_i = y_i - \mathcal{F}(x_i)$, the 15th line in Algorithm 1. In this way, errors derived from training samples by RFs can be refined. At the same time, the gradient descent procedure is conducted as the 19th line in Algorithm 1, thus GBRT can converge to its global minimum. The final boosted classifier is added with the initial results of RFs. The whole procedure can be seen in Figure 2.

4. Experiment and Analysis

To demonstrate performance of the proposed tracking method, extensive experiments are conducted. In order to show the effect of different pieces of our method, self-comparison experiments are conducted. In experiments without ranking middle-level patches, only Haar-like features are used.

- Set 1, only RFs without GBRT and without ranking middle-level patches.
- Set 2, RFs with GBRT and without ranking middle-level patches.
- Set 3, only RFs with ranking middle-level patches, without GBRT.
- Set 4, our tracker, RFs with GBRT and ranking middle-level patches.

We also compare our method with five state-of-the-art methods, online AdaBoost (OAB) [2], tracking-learning-detection (TLD) [4], Parallet Robust Online Simple Tracking (Prost) tracker [18], online random forests (ORF) [19] and Hough-based tracking (HT) [27].

For our tracker, 100 trees are used in RFs as in [18] and 10 trees in GBRT as the common rule. For $DecisionTree((x,y),\alpha,\beta,)$, the maximum tree-depth is set to five, each node with 10 random features, α = 100, β = 0.1, γ = 0.1. All experiments are implemented with fixed parameters. For compared trackers, we use tuned parameters from their source codes for the best results. Because the source code of the Prost tracker [18] is not available, the results of the Prost tracker are gathered from what is reported in [18]. Since all algorithms depend on some randomness, we run them 10 times and average the results for each sequence. For all trackers, Haar-like features are extracted. But the difference is that features of our tracker are selected by weights, while others are selected by boosting. The performance of trackers is measured by *Recall* - number of true positives divided by the length of the sequence (true positive is considered if the overlap with ground truth is larger than 50%) [4]. All experiments are carried out on an Intel Dual-Core 3.00 GHz CPU with 2GB memory. Our software relies on Microsoft Visual Studio 2008. Taking into consideration of real-time capability of the algorithm, we also compute the *frame per second* (FPS). In order to guarantee robustness, our method is about five FPS, which is also real-time.

4.1 Experiment Sequences

For quantitative analysis, we use the publicly available tracking sequences and videos recorded with a mobile robot in real scenes. David, Sylvster, Girl, Face Occ1, Face Occ2 and Tiger 2 are basic test sequences. Box and Lemming are from [18], while Moun bike and Cli-dive 1 are from [27]. Sequence 11 and 12 are recorded by us in an indoor environment and an outdoor environment

sequence	frames	challenges and difficulties
S1 David	462	scale and illumination changes
S2 Syl	1344	lighting and pose changes
S3 Girl	501	out-of-plane rotation
S4 Face Occ1	887	object occlusion
S5 Face Occ2	819	in-plane rotation and object occlusion
S6 Tiger 2	364	occluding clutter
S7 Box	1161	clutter and distraction
S8 Lemming	1161	motion blur and clutter
S9 Moun bike	228	non-rigid object deformation
S10 Cli-dive 1	75	large deformation
S11 In-door	187	lighting changes
S12 Out-door	120	dramatic distracters

Table 1. Challenges of experiment sequences

respectively. The challenges of all sequences are shown in Table 1.

4.2 Experimental Results and Analysis of Self-comparison

Table 2 shows the results of the self-comparison experiments. From the table, it can be found that our tracker gains the best average performance in all sequences. Figure 3 depicts the illustrative pixel error plots for David sequences. Pixel error represents the mean centre location error in pixels [18]. It can be seen that our tracker is the most stable and the drifts smallest since its mean errors are the smallest and it barely fluctuates. Near frame 250, Set 1 fluctuates strongly, and at the last part of sequences, Set 2 fluctuates frequently. Set 3 and Set 4 keep relatively stable for all sequences, and the mean errors of Set 4 are averagely smaller than for Set 3, which confirms that the drifting problem is alleviated at different levels with different pieces of our method.

The comparative results between Set 1 and Set 3 demonstrate the validity of ranking middle-level patches. Set 3 achieves the second best performance in self-comparison experiments, which shows the effectiveness of ranking middle-level patches. Through weighting top-down features by discrimination scores, three-class irregular patches are obtained to describe the target and its backgrounds, which avoids the distraction of clutter or similar backgrounds. Therefore, Set 3 gains much better results than Set 1, especially for Tiger 2, Face Occ 2 and Lemming sequences. The comparative results between Set 1 and Set 2 prove the validity of GBRT. Set 1 is actually the ORF tracker, so its results are the same as the ORF tracker. Set 2 is better than Set 1 in all sequences since Set 2 uses GBRT to refine the errors of the RFs classifier.

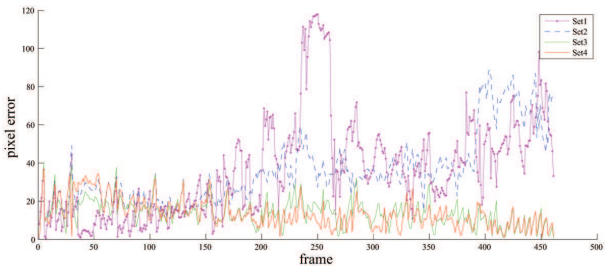

Figure 3. Pixel error plots for sequence David

sequence	Set 4	Set 1	Set 2	Set 3
David	**100**	95	**100**	**100**
Syl	**78**	73	75	73
Girl	**100**	99	**100**	**100**
Face Occ1	**100**	98	99	**100**
Face Occ2	84	72	75	82
Tiger 2	**80**	45	51	75
Box	45	38	42	77
Lemming	**94**	45	57	89
Moun bike	**100**	98	**100**	**100**
Cli-dive 1	**100**	**100**	**100**	**100**
In-door	98	87	93	92
Out-door	**100**	91	95	**100**
Average	**93**	78	82	91

Table 2. Recall(%) of Self-comparison

From Figure 3, it can be found that pixel errors of Set 2 are smaller than for Set 1 for most frames, which illustrates that Set 2 is more stable than Set 1. Set 4 shows the results of our tracker, which stems from integrating top-down and bottom-up features and the online refined random forests classifier. Its superiority will be introduced in the following experiment analysis.

4.3 Experimental Results and Analysis with State-of-the-art methods

Table 3 shows that our method delivers competitive results with other state-of-the-art trackers. From the table, it can be found that our method outperforms all other trackers in eight sequences. The recall of our tracker has been improved by 11% compared with the second best tracker HT [27]. Illustrative tracking results are shown in Figure 5, which depicts that our method can locate the tracking target with higher recall. Taking the Lemming sequences as examples, our method can track the target until the end, while all the others drift away and lose the target. For our tracker, the drifting problem is alleviated with the collaboration of the discriminative appearance model and online refined random frosts classifier, which avoids incorrectly updating the appearance models. An analysis of the comparative experiments will be shown on the basic of challenges in tracking.

sequence	Ours	ORF	OAB	TLD	Prost	HT
David	**100**	95	23	99	80	**100**
Syl	**78**	73	51	94	73	**78**
Girl	**100**	99	24	58	89	85
Face Occ1	**100**	98	35	52	**100**	**100**
Face Occ2	84	72	45	47	82	**89**
Tiger 2	**80**	45	19	39	-	69
Box	79	38	14	90	**91**	45
Lemming	**94**	45	35	88	70	51
Moun bike	**100**	98	74	31	-	**100**
Cli-dive 1	**100**	**100**	78	28	-	**100**
In-door	98	87	69	93	-	**99**
Out-door	**100**	91	65	90	-	**100**
Average	**93**	78	44	68	-	85

Table 3. Recall(%) of our method in comparison with ORF [19], OAB [2], TLD [4], Prost [18] and HT [27]

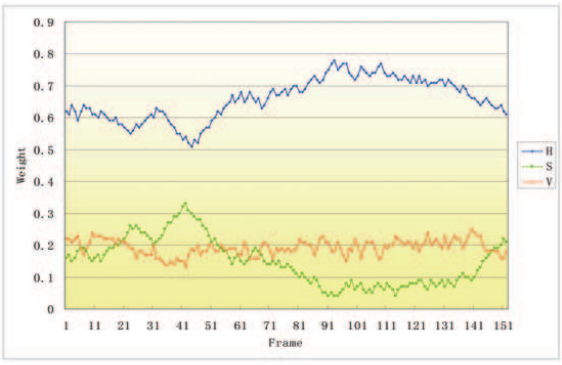

Figure 4. Normalized weights variation (hue, saturation, value for HSV)

Scale, illumination and pose changes. For the David and Syl sequences, our tracker gains the best result because our object model is highly robust to appearance changes. The selective bottom-up features mechanism utilized can maintain the discrimination between the target and its backgrounds. Moreover, Haar-like features are robust to scale and orientation changes, and the online learning algorithm can update the classifier to adapt to scale, illumination and pose changes. Although colour features cannot be used as sequences are grey images, other bottom-up features, such as intensity and orientation are still discriminative enough. TLD and the Hough tracker also perform well on these sequences. TLD adopts P-N learning to make up the drawback of tracking-by-detection and adopts a three layers cascaded classifier to improve detection. Hence, it is effective in dealing with appearance changes. The Hough tracker uses target contours instead of bounding boxes to represent targets. However, it is based on back-projection with GrabCut, which is sophisticated image processing.

For indoor sequences, despite the light changing, ranking middle-level patches can render the target more salient from its backgrounds, so our tracker can deal with small fluctuations in illumination. Our method outperforms ORF because that our method revised RFs' classification result. When the target is almost out of sight, OAB and ORF lose the target, while our tracker can maintain long-term tracking. Figure 4 shows the feature weight variation of HSV colour tuned by selective feature attention for indoor sequences. When the target moves into the region with the highest intensity, the importance of hue and value for separating target and backgrounds declines, while the weight of saturation accordingly increases. When the target moves out of this region, their weights will be restored. Hence, in this sequence, the influences of hue and value in separating target and backgrounds are the same, and they are contrary to the influence of saturation.

Occlusion and motion blur. Face Occ 1 and 2 sequences show that our tracker can handle occlusion more preferably than OAB, ORF and TLD. Since our appearance model is part-based, it is robust to occlusions. Moreover, selective features attention can strengthen the scores of target features while weakening backgrounds' scores when heavy occlusion occurs. Hence, our tracker can tackle heavy occlusion better than others. Our tracker is also more robust to outliers. It can be seen that our method outperforms other methods in handling the heavy occlusion and fast motion blur delivered from Tiger 2 and Lemming sequences (frame 361 and 553). This is because our salient appearance model derived from ranking patches can locate the target well according to the contrast sensitivity between target and backgrounds. Certainly, Prost performs well in dealing with occlusion, since it combines the template model and other trackers to compensate for the model drifting.

Background clutter and distracters. Moving across dramatic distracters with similar appearance is a great challenge for tracking. From Lemming sequences, we can see that OAB and ORF lose the target in early tracking and other methods track the wrong target in the later period. Only a few trackers can handle this problem well due to poor object model maintaining procedures and lack of consideration of the relationship between target and backgrounds. However, our tracker gains higher recall owing to the effect of selective feature attention on the tuning discrimination scores of features. Moreover, the collaboration of three-class target/backgrounds model and our multi-class refined random forests classifier achieves a more accurate target/backgrounds model. However, the box sequences show that TLD and Prost perform better than our method in grey pictures when background clutters. One reason is that TLD adopts P-N experts learning and Prost uses a optical-flow-based mean-shift tracker, in addition, colour features which are discriminative cannot be used in grey images.

Large deformation. Furthermore, our method performs much better than OAB and TLD for non-rigid objects with large shape deformation, which can be seen from the Moun bike and Cli-dive sequences. The tracking results of the Hough tracker are appealing too. The reason for this good performance is that random forests is tolerant to noise, which is superior than boosting [27]. Moreover, GBRT is applied to boost the random forests' classification results for higher precision, so our tracker can gain the best performance in dealing with large deformation. TLD gains poor results due to its poor involvement of only an optical-flow tracker, which is not suitable for non-rigid objects.

5. Conclusions

In this paper, a new appearance model composed of *ranking middle-level patches* is proposed for robust object tracking. This novel approach integrates top-down and bottom-up features through linearly weighting low-level feature scores and ranking high-level patches. As a result, high-level top-down patches are clustered into irregular ranking middle-level patches, which makes the tracking procedure a three-class appearance model. The collaboration of a three-class appearance model and a multi-class refined random forests classifier enables us to achieve more accurate target representation and to avoid incorrect appearance model updates. Extensive experiments demonstrate the superior performance of

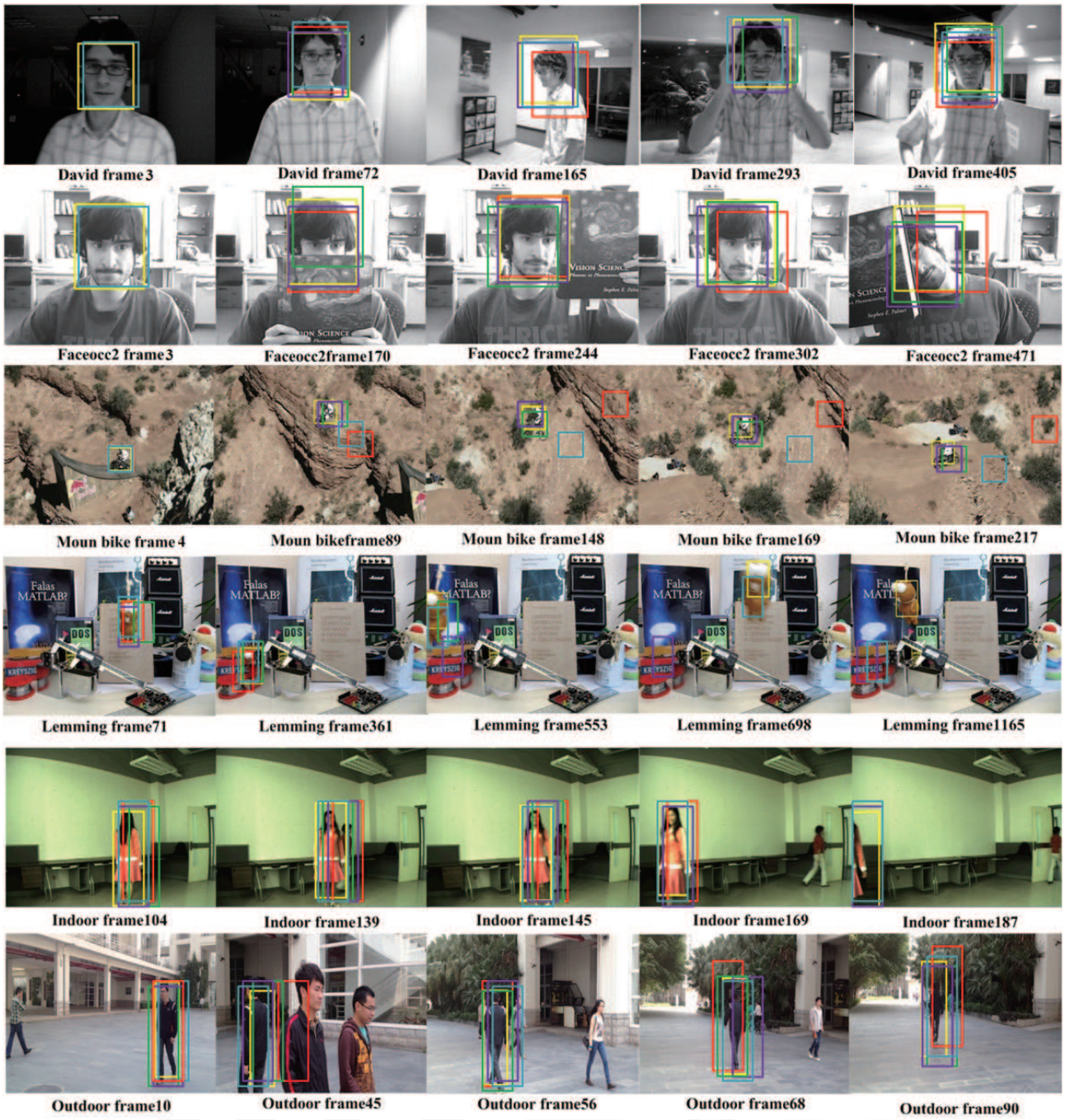

Figure 5. Illustration of tracking results. (Yellow-Ours, Red-OAB, Green-ORF, Purple-HT, Blue-PROST)

our tracker over several state-of-the-art tracking methods. They also verify that the proposed method can alleviate the drift problem well. Moreover, our method has been applied to a smart surveillance system for schoolyard, which realizes robust tracking in real scenes.

6. Acknowledgement

This work is supported by the National Natural Science Foundation of China (NSFC, nos. 61340046, 60875050, 60675025), the National High Technology Research and Development Programme of China (863 Programme, no. 2006AA04Z247), the Scientific and Technical Innovation Commission of Shenzhen Municipality (nos. JCYJ20120614152234873, CXC201104210010A, JCYJ20130331144631730, JCYJ20130331144716089), and the Specialized Research Fund for the Doctoral Programme of Higher Education (SRFDP, no. 20130001110011).

7. References

[1] A. Yilmaz, O. Javed and M.Shah, “Object Tracking: A Survey", in *ACM Computering Survey*, vol. 38, no. 4, pp. 1-45, 2006.

[2] H. Grabner and H. Bischof, “On-line Boosting and Vision", in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 260-267, 2006.

[3] S. Avidan, “Ensemble Tracking", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.29, no. 2, pp. 261-271, 2007.

[4] Z. Kalal, K. Mikolajczyk and J. Matas, “Tracking-Learning-Detection", in *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 1-14, 2010.

[5] T. Penne, C. Tilmant, T. Chateau and V. Barra, “Markov Chain Monte Carlo Modular Ensemble Tracking", in *Image and Vision Computing*, vol. 31, no. 6-7, pp. 434-444, 2013

[6] H. Grabner, C. Leistner and H. Bischof, “Semi-supervised On-Line Boosting for Robust Tracking", in *European Conference on Computer Vision*, pp. 234-247, 2008.

[7] S. Stalder, H. Grabner and L. Van Gool, “Tracking Should be as Simple as Detection, but not Simpler than Recognition", in *IEEE International Conference on Computer Vision*, pp. 1409-1416, 2009.

[8] B. Babenko, M. Yang and S. Belongie, “Visual Tracking with Online Multiple Instance Learning", in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 983-990, 2009.

[9] J. S. Supancic III and D. Ramanan, “Self-paced learning for long-term tracking", in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[10] M. Tang and X. Peng, “Robust tracking with discriminative ranking lists", in *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3273-3281, 2012.

[11] R. T. Collins and Y. Liu, “On-line selection of discriminative tracking features", in *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1631-1643, 2005.

[12] Y. Shi, H. Liu, Y. Liu and H. Zha, “Adaptive feature-spatial representation for Mean-Shift tracker", in *IEEE International Conference on Image Processing*, pp. 2012-2015, 2008.

[13] H. Liu, W. Wan and Y. Shi, “Collaboration of Spatial and Feature Attention for Visual Tracking", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2985-2992, 2009.

[14] H. Liu, H. He, “A Salient Feature and Scene Semantics-based Attention Model for Human Tracking on Mobile Robots", in *IEEE International Conference on Robotics and Automation*, pp. 4545-4552, 2010.

[15] A. Borji, “Boosting Bottom-up and Top-down Visual Features for Saliency Estimation", in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 438-445, 2012.

[16] Q. Sun and H. Liu, “Inferring Ongoing Human Activities Based on Recurrent Self-Organizing Map Trajectory", in *British Machine Vision Conference*, 2013.

[17] L. Breiman and E. Schapire, “Random forests", in *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.

[18] J. Santner, C. Leisner, A. Saffari, T. Pock and H. Bischof, “PROST: Parallel Robust Online Simple Tracking", in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 720-730, 2010.

[19] A. Saffari, C. Leistner, J. Santner, M. Godec and H. Bischof, “On-line Random Forests", in *IEEE International Conference on Computer Vision Workshops*, pp. 1393-1400, 2009.

[20] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, 2002.

[21] L. Itti, C. Koch and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis", in *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254-1259, 1998.

[22] J. Harel, C. Koch and P. Perona, “Graph-based visual saliency", in *Neural Information Processing Systems*, pp. 545-552, MIT Press, 2006.

[23] P. Viola and M. Jones, “Robust real-time face detection", in *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154, 2004.

[24] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection", in *Computer Vision and Pattern Recognition*, pp. 886-893, 2005.

[25] A. Fernĺćndez-Caballer, MT. Lĺőpez and S. Saiz-Valverde, “Dynamic stereoscopic selective visual attention (DSSVA): Integrating motion and shape with depth in video segmentation", in *Expert Systems with Applications* , vol. 34, no. 2, pp. 1394-1402, 2008.

[26] R. C. Luo, C. C. Kao and Y. C. Wu, “Hybrid Discriminative Visual Object Tracking with Confidence Fusion for Robotics Applications", in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2965-2970, 2011.

[27] M. Godec, P. Roth and H. Bischof, “Hough-based Tracking of Non-Rigid Objects", in *IEEE International Conference on Computer Vision*, pp. 81-88, 2011.

[28] J. Gall, A. Yao, N. Razavi, L.Van and V. Lempitsky, “Hough Forests for Object Detection, Tracking, and Action Reconition", in *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188-2202, 2011.

[29] T. Hastie, R. Tibshirani and J. Friedman, “The Elements of Statistical Learning", 2nd ed, New York: Springer, 2009.

[30] A. Mohan, Z. Chen and K. Weinberger, “Web-Search Ranking with Initialized Gradient Boosted Regression Trees", in *Journal of Machine Learning Research Workshops*, pp. 77-89, 2011.