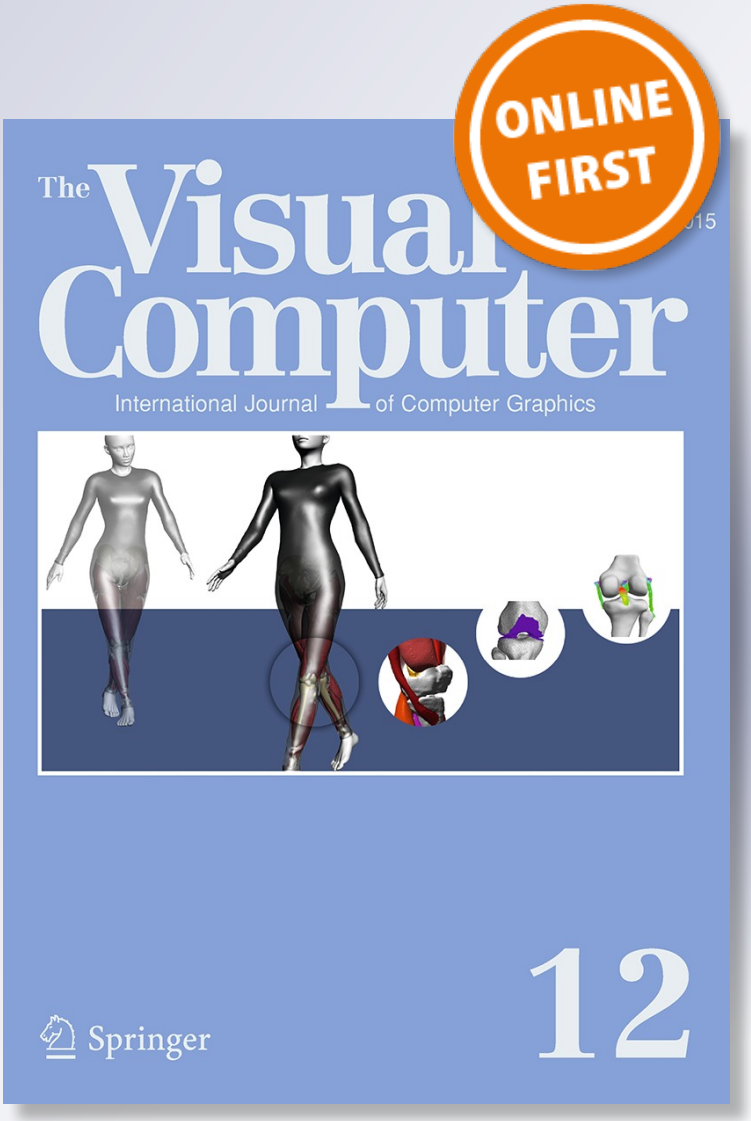
Robust tracking via monocular active vision for an intelligent teaching system

Rui Wang, Hao Dong, Tony X. Han & Lei Mei

The Visual Computer
International Journal of Computer Graphics

ISSN 0178-2789

Vis Comput
DOI 10.1007/s00371-015-1206-8



Springer

Vis Comput
DOI 10.1007/s00371-015-1206-8

ORIGINAL ARTICLE

Robust tracking via monocular active vision for an intelligent teaching system

Rui Wang[1] · **Hao Dong**[1] · **Tony X. Han**[2] · **Lei Mei**[3]

Abstract The research of this paper investigates a practical intelligent tracking teaching system, addressing the problem of teacher detection and tracking via monocular active vision in real time. The split lines and position-based visual servo rules are created to realize the robust and stable tracking, which is designed to keep the tracked teacher in the middle of image with a fixed size by automatically controlling a pan/tilt/zoom monocular camera in either rostrum region or other regions in the classroom. Face tracking in rostrum region is initiated by a face detector based on Adaboost followed by a novel long-term tracking algorithm named as informative random fern-tracking-learning-detection (IRF-TLD), which has advantages for its high accuracy and low memory requirement using real-valued feature and Gaussian random projection. Moreover, Gaussian mixture model can be automatically started to detect the teacher's movement when face tracking fails or stand-up students are detected. Experimental results on many benchmark sequences, which include various challenges for tracking, such as occlusion, illumination and pose variations, and scaling, have demonstrated the superior performance of the proposed IRF-TLD method when compared with several state-of-the-art tracking algorithms. Extensive experiments in a series of challenging real classroom scenarios also demonstrate the effectiveness of the complete system.

Keywords Long-term tracking · Informative random fern · Monocular active vision · Intelligent tracking teaching system

1 Introduction

With the recent advance in hardware and algorithms, computer vision systems, employed in traditional classroom, are changing the mode of modern education. Using cameras to automatically record the process of teaching as multimedia materials has led to a convenient and cost-effective way of learning and education. This approach visualizes the teaching state data, teaching operation process and teaching quality assessment. To develop the aforementioned intelligent tracking teaching system (ITTS), robust tracking of the teacher is essential; but the robust tracking problem remains as an open problem. To achieve an effective ITTS, the following requirements must be met:

(a) It should be able to track the teacher in real time, and robustly handle illumination changes and occlusions caused by other faces or objects with skin-colored surface.
(b) It should contain a smooth camera motion and give a close-up recording when the teacher emphasizes the content on the blackboard or when a student stands up.
(c) It should be able to automatically choose the highest confidence response and switch the field of view to the

✉ Hao Dong
qianxin_dh@163.com

Rui Wang
wangr@buaa.edu.cn

1 Laboratory of Precision Opto-Mechatronics Technology, School of Instrumentation Science and Opto-Electronics Engineering, Beihang University, No.37 Xueyuan Road, Haidian District, Beijing 100191, China

2 Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211, USA

3 Software System Research Department, China Electronics Technology Group Corporation No. 38 Research Institute, No. 199 Xiangzhang Avenue, Hi-Tech Zone, Hefei, Anhui 230088, China

Published online: 16 January 2016

region of interest to ensure no view obstruction in the tracking process.

(d) It should be able to achieve the aforementioned functions with a minimum number of cameras to reduce the cost of hardware.

To meet the above requirements, various approaches to a practical ITTS have been proposed in the last few years. Tsuruoka [1] describes a distance lecture support system using two cameras: one is a fixed camera and the other is an active camera. The image taken by a fixed camera is used to determine the parameters (the angles of pan/tilt and the zooming rate) of an active camera which is used for recording. Meanwhile, the fuzzy camera control method based on the behavior recognition of a lecturer which determined from the lecturer's silhouette is proposed. Another face tracking system for Multimedia Teaching is proposed in [2], which uses an active camera with pan–tilt function and combines the AdaBoost face detection with CamShift tracking algorithm. Wulff [3] develops an open-source framework for scene analysis in lecture recording scenarios, including a pan–tilt camera and a webcam that gets an overview of the room. Moreover, a scene segmentation technique using motion cues and background modeling has been implemented in its system. Wang [4] presents a monocular active vision module to track teachers' movement in real-time. Face tracking is initialized by robust face detection followed with the Expectation Maximization (EM) algorithm [5] based on HSV color space and prediction of face position.

For the design of our ITTS, we explore both the robustness and smoothness in face tracking and use pan/tilt/zoom (PTZ) camera heading to autonomously switching mechanism between two operational functions, namely, teacher tracking and stand-up student detection. Our scheme is related to the latest work described in [4], but with substantial algorithm advantages: the method in [4] relies on skin color of the face based on Hue channel from HSV color model, but it faces the challenge of locating only the skin-color area which is easily distracted by other faces and skin-like color areas in the scene. To address these issues, we present a novel tracking method named as informative random fern-tracking-learning-detection, which is inspired by tracking-learning-detection (TLD) [6] and the visual tracking with randomly projected ferns presented in [7].

The rest of this paper is organized as follows. In Sect. 2, we give an overview of the proposed ITTS. Sect. 3 describes the relevant research works. The proposed technical paradigm and method are detailed in Sect. 4. Section 5 shows the experimental results, followed by the conclusion in Sect. 6.

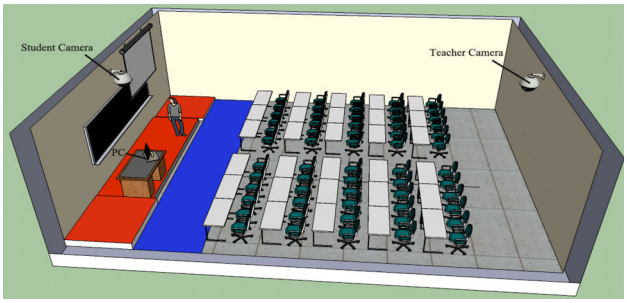


Fig. 1 Overview of our ITTS

2 System overview

Based on the aforementioned requirements for ITTS, we propose an effective and efficient way to build our ITTS with two PTZ cameras. One of them is denoted as Teacher Camera which is responsible for tracking the teacher during lecturing, while the other one is Student Camera which is applied to observe the activities of students. The configuration of our system in space is shown in Fig. 1. The Teacher Camera installed in the rear of the classroom supports multiple preset positions that help to track the teacher in different regions. The Student Camera placed in the front of the classroom covers students with a panoramic view as the initialization.

The architecture for the hardware and software of our system are illustrated in Fig. 2. Four modules, including image capture, camera control, user interface and active tracking, can run "concurrently" on a PC as multiple threaded tasks. An active tracking module with two threads: teacher tracking and stand-up student detection can be automatically triggered to execute the ITTS function. To achieve teacher tracking during the whole teaching process, we divide the classroom into three areas: Rostrum area (red color region in Fig. 1), Transition area (blue color region in Fig. 1) and Student area (gray color region in Fig. 1). Besides, when a student stands up to answer questions, the student camera will switch to the student with a close-up operation. In this paper, we mainly explore the tracking and detection of the teacher motion.

3 Related work

For active tracking based on the PTZ cameras, we review related literatures from the following two aspects.

3.1 Visual tracking

Although visual tracking can be formulated under different settings, we focus on the single target long-term tracking in this paper. In general, tracking algorithms can be categorized as either generative models or discriminative models.

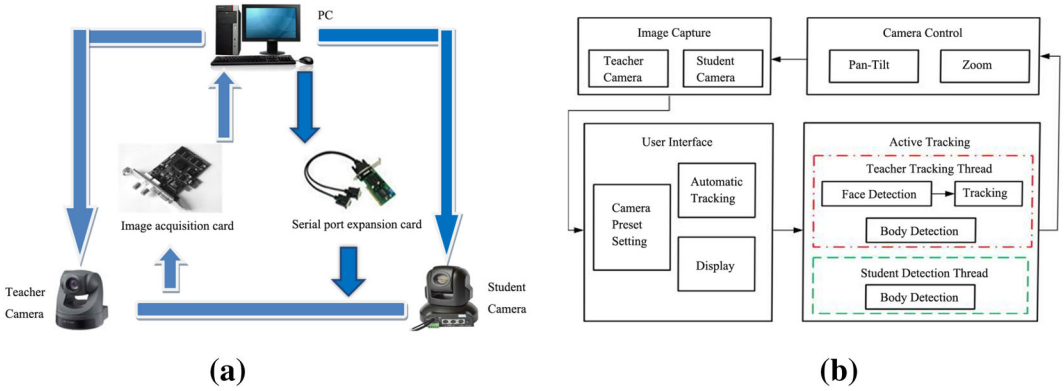


Fig. 2 Structure of our ITTS (**a** hardware structure, **b** software structure)

Generative models, which mainly consist of holistic templates and local templates, typically assume a generative process of the appearance of the target and search for the most probable candidate in the video. The holistic templates as one most straightforward approach, can be used to track the target by minimizing mismatch between the target template and the candidate patch [8,9]. To better account for appearance changes, subspace-based models have been proposed [10,11]. Recently, numerous tracking approach based on local sparse representations [12,13] have been proposed to handle occlusion and improve the run-time performance. Generally, the purely generative models do not take advantage of background information. Thus, they are easy to drift away from the target. For discriminative models, tracking is treated as a binary classification problem which aims to distinguish the target from the background. Numerous classifiers have been adopted for target tracking, such as structured output SVM [14], boosting [15], multiple instance learning [16] and deep learning [17]. Besides, some methods exploit correlation filter for the target or context [18,19]. Their primary advantage is that only fast Fourier transforms and several matrix operations are needed, making them very suitable for real-time applications.

Recent benchmark studies show that the top-performing trackers, especially some aiming to develop increasingly robust "longer" tracking, are usually discriminative models [6,14] or hybrid ones [13]. The Tracking Learning Detection (TLD) [6] method, in which tracking and detection are independent processes that exchange information via learning, was designed for long-term tracking of arbitrary objects with necessary drift resistance and redetection after full occlusions. Moreover, TLD developed a novel P–N learning method. Positive and negative examples are learned according to the disagreement of these two components, improving further detection performance. In addition, significant advances in long-term single-object tracking research have been made over the past few decades. In the context of faces, face detection which is essential in long-term tracking, has been extensively studied, and an off-the-shelf face tracker based on the detection-learning method is available [20]. For a much better systematic review and comparison of tracking literature as well as face detection and recognition, please refer to the recent benchmark [21] and review articles [22,23].

3.2 Camera control in active vision

An active vision system is one that can automatically interact with its environment by altering its viewpoint rather than passively observing it, and by operating on sequence of images rather than on a single frame. To investigate the environment more effectively, numerous studies have been carried to deal with the control problem in the active vision system.

Most PTZ camera control methods fall into two categories: PID or image position-based approach. PID methods typically use PID control scheme for minimizing the position error to keep the target in the center area of the camera's view field. Some related works have been published. Haj et al. [24] designed two PID controllers: one is used to control the pan–tilt operation, and the other for the zoom operation, which is used to reactively track the object at a constant image velocity while simultaneously maintaining a desirable target scale in the image. In [25], an increment digital PID controller with dead zone is applied in an active vision condition where camera and object may move simultaneously. However, the method is quite fragile in a complex environment. On the other hand, image position-based method usually assumes that the camera system can achieve stable tracking at each frame. For this approach, to make the target appear on the center of image, the displacement from the center of target to the image center is usually used to design a control scheme. Chen [26] proposed an active disturbance rejection control (ADRC) method to improve the control performance of the pan–tilt camera. This method uses two ADRC controllers working in parallel for a pan–tilt camera and only needs the deviation of target image centroid and image cen-

ter. Although this method has high accuracy, it is not able to make Pan/Tilt as fast as the tracked moving target. Bernardin [27] presented an automatic system for monitoring indoor environments using a PTZ camera. Meanwhile, the fuzzy controlling scheme, in which the input is the target image position displacement and the output are the required pan, tilt and zoom speeds for the camera, allows for smooth tracking of moving targets. Their approach does, however, need to continuously update the camera parameters using rotation and zoom information.

In summary, many approaches with the state-of-the-art algorithms have been proposed to follow objects with an active camera. However, some approaches need highly configured devices while our goal is tracking via a general commercial PTZ camera with limited support.

4 Component of Our ITTS

As was stated in Sect. 2, our system consists of four modules: image capture, camera control, user interface and active tracking. The image capture module is responsible for capturing video sequences from the teacher camera and student camera, respectively, while the camera control module is used to update the PTZ parameters of the teacher camera and student camera. The user interface is GUI based and enables users to observe the actual situation of teaching. The active tracking provides the location information of the target. Multiple threaded tasks are employed in this module which includes the teacher tracking thread and stand-up student detection thread. For the former, the algorithm is divided into two parts: (i) face detection and long-term tracking, (ii) body detection with Gaussian mixture models (GMM) [28]. For the latter, our system uses the same GMM algorithm to achieve the stand-up student detection.

In the following section, we discuss in detail the active tracking for teacher and camera control modules.

4.1 Teacher tracking initialization

To perform real-time teacher tracking, the teacher is asked to stand in a predetermined region facing to the teacher camera at the beginning; our system is designed to acquire the teacher face as the selected target in 2–3 s. The real-time face detection framework we adopt in this initialization stage is proposed in [29], which uses the Adaboost algorithm to generate an effective cascade of classifiers based on Haar-like features. The face detector is a filter that receives a 24×24 pixel region of the preprocessed image and generates an output of 1 or 0, signifying the positive and negative samples, respectively. It runs at 15fps on 320 by 240 pixel image and can detect faces that tilt up to about $\pm 15^\circ$ in plane and about $\pm 45^\circ$ out of plane.

4.2 Teacher tracking

When the teacher's face is detected, it is tracked by our tracking approach called IRF-TLD, which decomposes the long-term face tracking task into tracking, detection and learning. The target is followed by a tracker from frame to frame and simultaneously learned so as to build a detector that localize all appearances observed so far. We adopted the tracker and learning method from the original TLD [6], but different from the original detector which is consisted of three stages: (i) patch variance classifier, (ii) standard random fern classifier, and (iii) nearest neighbor classifier, two improvements have been made in this paper w.r.t. TLD: (i) An adaptive detection strategy with a belt region which is determined by the center ordinate of the teacher's face is developed to narrow the detection region. (ii) Instead of the binary comparison in the standard random fern, the more informative real value from the subtraction is used. Moreover, a random projection is utilized to map the value of each fern derived from feature value to a parametric distribution, specifically, Gaussian distribution, in which the classification is done. With these measures, we can achieve the benefits of both high accuracy and low memory requirement.

Our whole IRF-TLD tracking approach is summarized in Fig. 3. It works as follows:

4.2.1 Tracker

As described in TLD, the tracker estimates the target's motion between consecutive frames under the assumption that the frame-to-frame motion is limited and the target is visible. A grid of 10*10 points inside the target box from the previous frame is extracted, and its motion is estimated using the Lucas–Kanade [8] tracker extended with failure detection. The target location is computed based on the 50 % of the most reliable displacements using median.

4.2.2 Cascade detector

The detector which consisted of three stages in IRF-TLD is named as Cascade Detector, which treats every frame as independent and performs a novel adaptive belt scanning of the image to localize all appearances that observed and learned in the past. In the first frame, the all possible sub-windows based on the size of initial target box are generated with the following parameters: scales step = 1.2, horizontal step = 10 % of width, vertical step = 10 % of height, minimal box size = 20 pixels. The cascade detector is responsible for selecting the most possible target candidate in each frame. In consideration of the heavy computational work of calculating all the sub-windows, we design an adaptive detection strategy, namely 'belt scanning', for which only a belt region determined by the center of the teacher's face needs a scan.

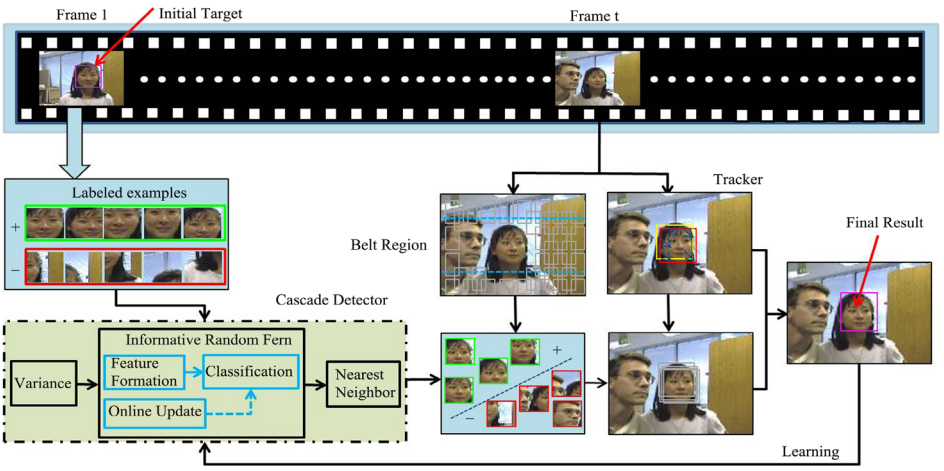


Fig. 3 Framework of IRF-TLD tracking algorithm

Furthermore, the belt region can be defined flexibly in terms of a trade-off between accuracy and real-time performance. In this paper, the height of the belt region is four times as much as that of the initial target box.

Compared with TLD, which regards the entire image as the detection scope, the belt scanning strategy improves the detection speed at the cost of losing the traversing detection capability in the whole image. However, we tend to assume that the belt region is enough to cover the teacher's moving range in our ITTS. Besides, by comparison with other local detection strategy that searches within a fixed radius, such as MIL [16] and CT [30], our search strategy has greater probability to recover from tracking failures.

Patch variance In the belt region, a large number of patches which include background need to be rejected in advance. This will be done by the first stage of our cascade detector, i.e., patch variance. It exploits the fact that gray-value variance of a patch$I$ can be expressed as $E(I^2) - E^2(I)$, and the expected value $E(I)$ can be measured using integral images. This stage restricts the maximal appearance change of the target and rejects those patches with gray-value variance smaller than 50 % of the variance of the target patch.

Informative random fern The inputs to the Informative Random Fern (IRF) classifier, which is the second stage of our cascade detector, are the image patches that were not rejected by the variance filter. In the original TLD, every image patch which may be available in different size consists of a number of ferns [31]. Each fern considered as a base classifier performs a quantity of pixel comparisons on the patch resulting in a binary code, which indexes to an array of posteriors. The posteriors of all base classifiers are averaged and the patch is classified as the tracking target if the average posterior is larger than 50 %. While the standard random fern used in TLD as the classifier showed excellent performance, lots of other information will be lost due to the only two possible outputs, 0 or 1 for comparison of each pixel pair. Furthermore, to compensate for the loss, more pixel pairs are usually used, thus leading to an enormous memory requirement growing exponentially with the number of pixel pairs in a fern. For the sake of accuracy improvement and memory saving, we promote the standard random fern classifier in TLD to an IRF classifier which produces the real value feature for a fern based on subtraction. In the following, the proposed IRF classifier will be introduced from three steps: feature formation, classification with probability and online update.

1. Features formation: We adapt the real value feature from [7], i.e., the real value feature $f_{i,j}$ described in Eq. (1) is extracted from pixel pair j of fern i:

$$f_{i,j} = I(d_1(i,j)) - I(d_2(i,j)),\quad i \in \{1,2,...,T\},\ j \in \{1,2,...,S\} \tag{1}$$

where T is the total number of ferns, S is the number of pixel pairs in each fern, and $I(d)$ represents the intensity of an image patch I at d. $d_1(i,j)$ and $d_2(i,j)$ denote the coordinates of the randomly generated pixel pair j of fern i. Here, we discrete the space of pixel locations within a patch and generate all possible horizontal and vertical pixel coordinates for better comparisons using normalization: $d(i,j) = (g_1 \times w, g_2 \times h)$. Where g_1 and g_2 are the randomly generated numbers of 0–1 intervals, while w and h are the width and height of current patch, respectively. As a result, the robustness of the relative position for every given pixel pair in different patch sizes is guaranteed.

Obviously, the real value feature can preserve more information about the intensity difference between two pixels because of $f_{i,j} \in \mathbb{R}$ instead of $f_{i,j} \in \{0,1\}$.

In TLD, the binary features in each fern are combined into a binary code that indexes to the posterior probability. However, in the proposed method, feature $f_{i,j}$ is a real value, it is necessary to "encode" S real values in each fern into

a single real value to simplify the subsequent comparison and classification. A theoretical basis for this idea has been stated by Johnson–Lindenstrauss (JL) lemma [32] that with high probability the distances between the points in the high-dimensional space $A \in \mathbb{R}^m$ are preserved if they are projected onto a randomly selected low-dimensional subspace $B \in \mathbb{R}^n$ according to a mapping function $g : \mathbb{R}^m \rightarrow \mathbb{R}^n$, where $n \ll m$. Besides, Baraniuk [33] also proved that for k-sparse data (e.g, image and audio signal), the random matrix such as Gaussian, Bernoulli and Fourier matrix satisfying the JL lemma holds true for the restricted isometry property in compressive sensing. Therefore, the typical random Gaussian matrix $R \in \mathbb{R}^{n\times m}$ with each entry an independent and identically distributed Gaussian random variable, as used in numerous works recently [34,35], can be selected to reconstruct A with minimum error from B with high probability. Formally:

$$B = RA \tag{2}$$

As the special case in this paper, if we defined $B \in \mathbb{R}^1$, $A \in \mathbb{R}^S$, then Eq. (2) can be represented as Eq. (3) to facilitate efficient projection from feature values of different pixel pairs into a single real value:

$$F_i = \sum_{j=1}^{S} r_j f_{i,j} \tag{3}$$

where $r_j \sim N(0, 1)$ is a real value generated randomly according to a Gaussian distribution. In this case, the projection result F_i keeps most of the information of the original features $f_{i,j}$, $j = 1, 2, ..., S$ of fern i. In this way, simpler classifiers by each informative random fern with a single real value could be built.

Besides, comparing the proposed IRF with the standard random ferns method, we can find that the IRF has the advantages of requiring a constant and much lower memory from the following analysis. Assuming that the number of classes is $\gamma = 2$ (foreground and background) and the real value feature is stored in a single precision type (e.g., *float* in C++) which occupies 4 bytes. Then the memory requirement is $\mathrm{MEM}_{\mathrm{Our}} = T \times \gamma \times 4$. While in the standard random ferns method used in TLD, a specific binary code is stored in an integral type (e.g., *int* in C++) and occupies 4 bytes. The memory requirement is $\mathrm{MEM}_{\mathrm{TLD}} = 2^S \times T \times \gamma \times 4$. It is clear that the standard random ferns method in TLD needs memory 2^S times more than the proposed IRF method.

2. Classification with probability: Every base classifier maintains a distribution of posterior probabilities. In TLD, the posterior probability can be calculated by counting the frequency of a specific binary code. In our algorithm, the output F_i is calculated as a single real value produced randomly on the basis of Gaussian distribution. For simplicity, we model the probability $p(F_i|c)$ as a Gaussian distribution with parameters $\left(\mu_i^c, \sigma_i^c\right)$ for fern i of class c. Whereupon, the discriminative function that distinguishes foreground from background is

$$\begin{aligned} H(F) &= \log\left(\frac{\prod_{i=1}^{T} p(F_i|c=1)p(c=1)}{\prod_{i=1}^{T} p(F_i|c=0)p(c=0)}\right) \\ &= \sum_{i=1}^{T} \log(p(F_i|c=1)) - \sum_{i=1}^{T} \log(p(F_i|c=0)) \end{aligned} \tag{4}$$

Where we assume uniform prior $p(c = 1) = p(c = 0)$, $c \in \{0, 1\}$ is a binary variable which represents the sample label and $F = \{F_1, F_2, ..., F_T\}$ is a set containing the value of all ferns for an image patch.

The IRF classifies the patch as the target if the corresponding value $H(F)$ is larger than zero.

3. Online update: In the real-time long-term visual tracking, it is necessary to update the classifier online to follow the target with appearance variations. To integrate our IRF feature that the value of each fern is modeled as a Gaussian distribution with parameter $\left(\mu_i^c, \sigma_i^c\right)$ to the target model, we simplify the update of the classifier as a weighted parameter update:

$$\begin{aligned} \mu_i^c &\leftarrow \lambda\mu_i^c + (1-\lambda)\mu_i^{c,\mathrm{new}} \\ \sigma_i^c &\leftarrow \sqrt{\lambda\left(\sigma_i^c\right)^2 + (1-\lambda)\left(\sigma_i^{c,\mathrm{new}}\right)^2 + \lambda(1-\lambda)\left(\mu_i^c - \mu_i^{c,\mathrm{new}}\right)^2} \end{aligned} \tag{5}$$

where λ is the learning rate, $\mu_i^{c,\mathrm{new}} = E[F_i|c]$ and $\sigma_i^{c,\mathrm{new}} = \sqrt{E[(F_i|c)^2] - (E[F_i|c])^2}$ are estimated from the training samples at current frame.

Nearest neighbor classifier After filtering the patches by IRF classifier, we use the online model [6] and classify the left patches using an NN classifier. Online model $O = \{p^+, p^-\}$ includes target patches $p^+ = \{p_1^+, p_2^+, ..., p_m^+\}$ and background patches $p^- = \{p_1^-, p_2^-, ..., p_n^-\}$. To classify the candidates that are not yet rejected, the normalized correlation coefficient (NCC) is used to measure the similarity between the image patch I and template patches p_i^+ or p_i^-, illustrated as $\mathrm{NCC}(I, p_i^+)$ or $\mathrm{NCC}(I, p_i^-)$. The similarity between I and O can be measured by

$$\begin{aligned} &\mathrm{Sim}(I, O) \\ &= \frac{\max_{p_i^+\in O} \mathrm{Sim}\left(I, p_i^+\right)}{\max_{p_i^+\in O} \mathrm{Sim}\left(I, p_i^+\right) + \max_{p_i^-\in O} \mathrm{Sim}\left(I, p_i^-\right)} \end{aligned} \tag{6}$$

An image patch can be classified as positive if Sim(I,O) > thr.

With this stage, the performance of the cascade detector is improved.

4.2.3 P–N learning

The task of learning is to initialize the cascade detector in the first frame and update it in run-time using the P-expert and the N-expert. According to [36], the online P–N learning performs the following steps: (1) P-expert—recognizes false negatives, and adds them to training set with positive label. (2) N-expert—recognizes false positive, and adds them to training set with negative label. The independence of the two experts enables mutual compensation of their error.

In every frame, the P-expert outputs a decision about the reliability of the current tracked result. If the result is reliable, the online model and the informative random fern are updated using new labeled samples. Our algorithm generates labeled samples based on the overlap of the sub-window and tracked target box. The overlap of two boxes is measured as a ratio between their intersection and union. By setting the thresholds thr_p and thr_n, we collect some patches as positive with overlap$>$ thr_p and negative samples with overlap$<$ thr_n. The labeled training samples are then used to update the online model by P–N experts and the informative random fern by Eq. (5).

4.3 Body detection

In the teaching scenario, the teacher movable space in the classroom is divided into three areas as illustrated in Fig. 1. The automatic switch technique from one area to another area is detailed in Sect 4.4.2. More specifically, the teacher face tracking algorithm IRF-TLD is employed in Rostrum area and Transition area. However, once the face tracking fails in Transition area or the teacher moves to Student area, GMM is adopted to detect the teacher's moving until the teacher's face can be found again. The inspiration for this body detection methodology comes from the fact that one of the powerful attributes of GMM is its ability to form smooth approximations to arbitrarily shape densities. As a result, when body detection is triggered by the automatic switch, each pixel is modeled as a mixture of Gaussian distribution and an online approximation is carried to update parameters of the model. The GMM contributes to a better moving target detection using a discrete set (3–5 in this paper) of Gaussian function, each with its own mean and covariance matrix.

4.4 Camera control

In our ITTS, camera control is the requisite module in automatically keeping more effective tracking of a selected target and making a close-up shot. Here, we present a practical solution, which controls the movement of the camera solely based on the information in the active camera images. The specifics of the camera control design are discussed below.

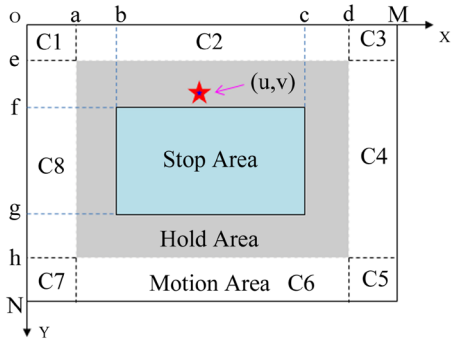


Fig. 4 Multi-rectangular block division for camera control (XOY denotes the image coordinate system, *a–h* are positive real numbers, *C1–C8* denote sub-area of motion area)

4.4.1 Pan–tilt control

As noted earlier, we can get the central coordinates (u, v) as well as the image size of the tracked target in every frame, and obtain the tracking error: the displacement from the center of target to the image center, if the target is not at center of image. In our approach, this error and (u, v) will be used to design a multi-rectangular block division (MRBD) camera Pan–tilt control strategy to keep the adaptability for variation of the target and the stability of tracking from a single PTZ camera on real-world scenarios with non-cooperating subjects. An overview of our MRBD scheme can be seen in Fig. 4. Suppose an image of size $M \times N$, we formulate Stop area, Hold area, Motion area in a number of blocks. Different camera control strategy patterns in terms of the target image location, i.e., (u, v), are listed as follows.

1. When (u, v) is detected in stop area, shown in Eq. (7), the camera stops Pan–tilt operation immediately and will keep waiting for the next P/T instruction;

$$\text{Stop area} = \{(u, v) | b \leq u \leq c,\ f \leq v \leq g\} \tag{7}$$

2. When (u, v) is detected in hold area, shown in Eq. (8), the camera maintains its previous pan–tilt action until the target moves to other areas;

$$\text{Hold area} = \{(u, v) | a \leq u \leq d, e \leq v \leq h, (u, v) \notin \text{stop area}\} \tag{8}$$

3. When (u, v) is detected in motion area, the pan–tilt behavior of the camera in different sub-area C_i is determined by a set of rules specified by Eq. (9).

$$\text{Motion area} = C_1 \cup C_2 \cup C_3 \cup C_4 \cup C_5 \cup C_6 \cup C_7 \cup C_8$$

$$\begin{cases} C_1 = \{(u,v)|0 \le u \le a, 0 \le v \le e\} & \text{LeftUp} \\ C_2 = \{(u,v)|a \le u \le d, 0 \le v \le e\} & \text{Up} \\ C_3 = \{(u,v)|d \le u \le M, 0 \le v \le e\} & \text{RightUp} \\ C_4 = \{(u,v)|d \le u \le M, e \le v \le h\} & \text{Right} \\ C_5 = \{(u,v)|d \le u \le M, h \le v \le N\} & \text{RightDown} \\ C_6 = \{(u,v)|a \le u \le d, h \le v \le N\} & \text{Down} \\ C_7 = \{(u,v)|0 \le u \le a, h \le v \le N\} & \text{LeftDown} \\ C_8 = \{(u,v)|0 \le u \le a, e \le v \le h\} & \text{Left} \end{cases} \tag{9}$$

In particularly, the parameters *a*–*h* can be adjusted to specify the size of each rectangular block division area according to the actual demand. In our implementation, the parameters *a*–*h* have been set manually, and the simple yet effective pan–tilt control strategy yields satisfactory results on a range of test scenarios.

4.4.2 Zoom control

Apart from the rules for adjusting camera pan/tilt parameters, zoom control is also an essential requirement in our system. It can adjust the camera focal length not only to be triggered to the region of interest with zoom-in mode when the teacher writes on the blackboard, but also to preserve the tracked teacher at a proper predefined image size wherever he(she) moves in the classroom. To achieve the smooth switch between different vision fields, we divide the classroom into three areas: rostrum area (P1), transition area (T2) and student area (S3), which is demonstrated in Fig. 1. Besides, the split lines L_1 (red), L_2 (blue), L_3 (green) are formulated in image as the border of every two of the three different areas P1, T2 and S3 (Fig. 5). Moreover, three preset positions Z1, Z2, Z3 correspond to these areas, respectively, i.e, $Z1 \Rightarrow P1, Z2 \Rightarrow P2, Z3 \Rightarrow P3$. As soon as the target is detected to enter one area, the camera will be zoomed in/out to a preset position corresponding to this area. Detailed rules are described below:

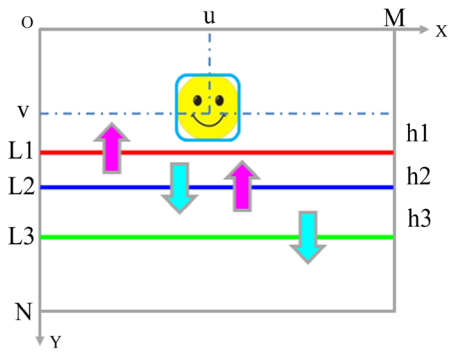


Fig. 5 Split lines (the *arrows* represent the direction of the teacher's movement)

Table 1 Camera zoom operation

Teacher moving direction	Teacher location	Regional and algorithm determination	Camera preset
From P1 To S3	$v < L_2$	P1 (IRF-TLD)	Z1
	$L_2 < v < L_3$	T2	Z2
	$L_3 < v$	S3 (GMM)	Z3
From S3 To P1	$L_2 < v'$	S3 (GMM)	Z3
	$L_1 < v' < L_2$	T2	Z2
	$v' < L_1$	P1 (IRF-TLD)	Z1

The positions of split lines are initialized in terms of the ordinate: $L_1 = y_0 + h_1$, $L_2 = y_0 + h_2$, $L_3 = y_0 + h_3$, where y_0 is the central ordinate of teacher's face in the initial frame, and h_i is the distance from y_0 to L_i. h_1, h_2, h_3 are specified by the size of classroom. In our paper, h_1 is equal to the height of the initial target box, $h_2 = h_1 + 10$ and $h_3 = h_1 + 20$. When the teacher walks around in the classroom, the IRF-TLD algorithm for teacher face tracking and GMM algorithm to determine the location of the teacher will be adopted alternatively. Suppose that (u, v) indicates the central coordinate of the teacher's face, while (u', v') denotes the top-left coordinate of the body detection result by GMM, the camera zoom operation can be summarized in Table 1.

In Transition area T2, whether the body detection algorithm or the face tracking algorithm is used may depend on the criterion listed in the Eq. (10).

$$\begin{cases} (A_T \in T2)\&(F_L \ge th) \Rightarrow & \text{Body detection (GMM)} \\ (A_T \in T2)\&(F_L < th) \Rightarrow & \text{Face tracking (IRF-TLD)} \end{cases} \tag{10}$$

where A_T denote the area where the teacher is located, F_L is the successive frames that the face is no longer tracked, and th is an integer.

In case that the teacher is writing on the blackboard, our tracking will usually fail due to the teacher's face becoming invisible to the camera. The decision criterion for zooming in, i.e., the camera operating for the close-up of blackboard writing can be defined as:

$$(A_T \in P1)\&(F_L \ge th) \Rightarrow \text{ zoom in} \tag{11}$$

In our case, th is chosen as 40 empirically.

Therefore, with the adoption of the simple but efficient control P/T/Z strategy to drive the movement of the monocular PTZ camera, we achieve to keep the position of the tracked target in the middle of the image and preserve its proper predefined image size in the long-term tracking.

5 Experiments

Our ITTS is implemented in C++, which runs at 25 frames per second (FPS) on an Intel Dual-Core 3.30GHz CPU with 4G RAM. Two Sony EVI-D70P cameras, tracking the teacher and shooting the students, respectively, serve to capture the video of the teaching process. Each camera offers a range of pan angle ($-170° \rightarrow +170°$), tilt angle ($-30° \rightarrow +90°$) and an 18X optical zoom feature. Besides, a dual image acquisition card and a serial port expansion card are employed to connect the cameras with PC as shown in Fig. 2a.

Two kinds of experiments and elaborate study are made in this section. Firstly, to validate our IRF-TLD tracking algorithm, we compare it with several state-of-the-art algorithms in terms of accuracy and running time (Sect 5.1). Next, the performance of our ITTS of monocular active vision is evaluated in real classroom scenes (Sect 5.2). In all the experiments, the total number of ferns is set to $T = 50$ and the number of pixel pairs in a fern is decided as $S = 4$. Experimentally, we find that in Eq. (5), the learning rate λ selected as 0.85 is more suitable with the capability to follow the target appearance variation and to overcome drifting. Meanwhile, the similarity threshold in Eq. (6) is set as thr$= 0.6$. The overlap thresholds for positive and negative training samples are set as $\text{thr}_p = 0.8$ and $\text{thr}_n = 0.2$, respectively (discussed in Sect. 4.2.3). This setting generates approximately 50 positive samples and a large number of negative samples. Among the large number of negative samples, 100 of the negative samples are selected randomly since the appearance of background is more diverse than that of foreground.

5.1 Algorithm accuracy

5.1.1 Datasets and evaluation metric

Nine state-of-the-art algorithms on 17 fully annotated video sequences (8882frames) included TLD [6], Struck [14], SCM [13], OAB [15], ASLA [12], IVT [10], LSK [37], CT [30] and DFT [38] are put to use in this comparison to validate our IRF-TLD tracking algorithm. The proposed IRF-TLD in our ITTS is used to track the teacher's face, hence the test sequences purposely chosen are face tracking scenarios based on different challenging factors: occlusion, illumination, plane rotation, scaling, etc. These sequences with the corresponding ground truth files and the compared code library are available on the website: http://visual-tracking.net. Note that we integrate our IRF-TLD algorithm in the code library with uniform input and output formats to achieve the objective comparison. In addition, all of these algorithms are evaluated in the one-pass evaluation (OPE) [21].

We use the precision plots based on the center location error and the success plots based on the overlap metric [21] to evaluate the robustness of tracking algorithms quantitatively. Center location error is defined as the Euclidean distance between the center location of the tracked target box and that of the ground truth box. The precision plot shows the percentage of frames whose estimated locations are within the given threshold distance of the ground truth. To compare the performances of different algorithms, the score for the threshold equal to 20 pixels is used to be the representative precision score. The overlap is defined as $OS = \text{Area}(b_t \cap b_a) / \text{Area}(b_t \cup b_a)$, where b_t is the tracked target box and b_a denotes the ground truth box. To evaluate the performance on a sequence of frames, we count successful frames whose overlap OS exceeds the given threshold. The success plot shows the ratios of successful frames at the thresholds varied from 0 to 1. Instead of using a specific threshold (e.g., 0.5) for evaluation, the area under curve (AUC) of each success plot is employed to rank these algorithms in our paper. For the reason that the success plot measured by AUC is more convincing than the precision plot calculated at one threshold, we compare the performance of different algorithms mainly based on success plot but use precision plots as auxiliary in the following.

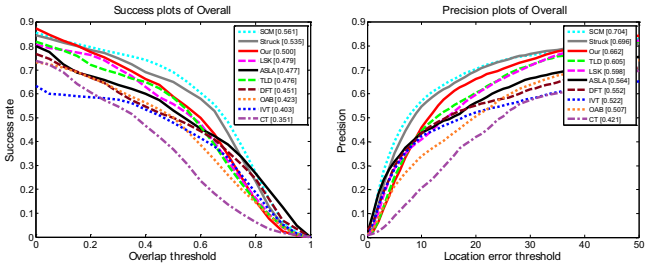


Fig. 6 The overall performance of the 10 tracking algorithms in terms of success plots and precision plots

5.1.2 Results and analysis

The overall performance of the 10 tracking algorithms based on success plots and precision plots are illustrated in Fig. 6. The corresponding ranked results are displayed in the legends of each drawing. According to the experimental results, our algorithm achieves outstanding performances in both the metric overlap and center location error: in the success plot, it achieves an AUC score of 0.500 and ranks 3rd following SCM (0.561) and Struck (0.535), but outperforms TLD by 2.4 %. Meanwhile, the overall precision of our IRF-TLD at 66.2 % is the highest among all algorithms except for SCM (70.4 %) and Struck (69.6 %), yet beating TLD by 5.7 %. In particular, although Struck and SCM perform better than our IRF-TLD in precision and overlap scoring, the fps achieved by our algorithm (25fps) is more practical than Struck (20.2fps) and SCM (0.51fps) [21].

To facilitate analyzing the strength and weakness of the proposed IRF-TLD algorithm, we further evaluate these tracking algorithms on sequences with 7 attributes. The suc-

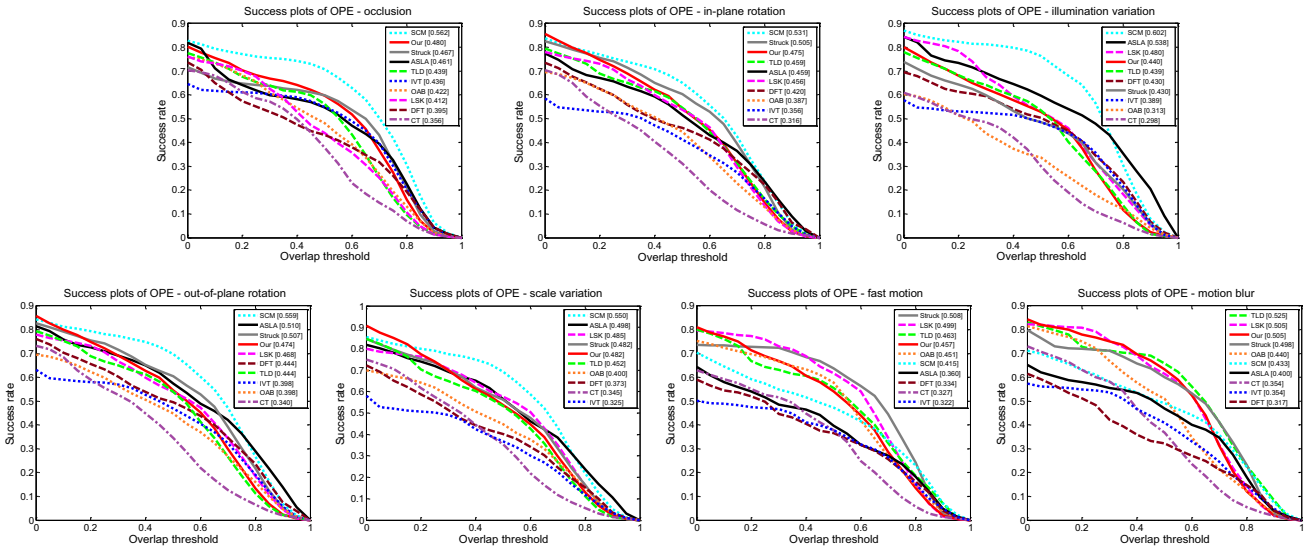


Fig. 7 Success and precision plots of sequences with different attributes

cess plots of different attributes are shown in Fig. 7. We note that the proposed IRF-TLD ranks within top 5 on all of attributes, and outperforms TLD on 5 attributes including occlusion, illumination variation, scaling and (in-)out-of plane rotation attribute.

For the sequences with attributes such as occlusion, in plane rotation, IRF-TLD correspondingly ranks 2nd and 3rd on all evaluated algorithms while SCM ranks 1st. Different from our method that only utilizes the target representation, SCM employs a sparsity-based generative model for target representation, and develops a novel histogram-based method using overlapped sliding windows that takes the spatial information of each patch into consideration with an occlusion handling scheme. These measures are helpful to locate the target from heavily occlusion and plane rotation. In our IRF-TLD model, we promote the original random fern classifier in TLD with two possible 0 or 1 outputs to the IRF classifier of the real value F_i via a Gaussian project matrix as described in Eq. (3). The update scheme facilitates the cascade detector combining with P–N learning tactics in IRF-TLD to account for more informative patterns of the target object than what is done by TLD. As such, IRF-TLD can achieve more discriminative detection result and handle occlusion robustly, performing even better than TLD by 4.1 % (Occlusion) and 1.6 % (In plane rotation) respectively.

On the sequences with the illumination variation, out of plane rotation and scale variation attributes, IRF-TLD bears some similarity to TLD in the use of an online update scheme that can adapt to the target appearance variations ranks in the top 5, while the top 2 (i.e., SCM and ASLA) all take advantage of spatial and local information of the target. This helps locate the target more accurately when the target appearance changes greatly due to out of plane rotation. Besides, the feature selection scheme of sparsity-based discriminative classifier in SCM can choose suitable number of discriminative feature, which can better separate the target from the background in spite of the illumination variation. To this end, ASLA generates the dictionary for local sparse coding by the dynamic template, which are updated online based on both incremental subspace learning and sparse representation. In particular, with affine motion model (e.g., SCM and ASLA), the trackers often handle better on the scale variation subset. As IRF-TLD produces the real value for a fern based on subtraction and Gaussian random projection, leading to a more informative and meticulous result than the binary feature used in the TLD. Hence, the maintaining of the diversity of real value features enables IRF-TLD to practice better than TLD does in the presence of significant drastic appearance changes. The results indicate that IRF-TLD outperforms TLD by 0.1 % (Illumination variation), 3 % (Out of plane rotation) and 3 % (Scale variation), respectively.

Finally, for the sequences with Fast Motion and Motion Blur attributes, IRF-TLD performs well and its corresponding ranking is 4th and 3rd, while TLD ranks 3rd and 1st, respectively. According to the [39], most sequences in each subset of motion blur fall into the subset of fast motion. Thus, we conclude that the fast motion attribute in each subset significantly affects the evaluation because IRF-TLD does not address fast motion well due to the quick belt scanning strategy and so do SCM and ASLA with the application of the simple dynamic model based on stochastic search. On the contrary, the trackers based on dense sampling (e.g., Struck, TLD) perform much better than others in the subset of fast motion due to their full-scale and large range search strategy.

To further analyze the performance of IRF-TLD, the AUC scores and precision scores for each sequence are also generated and shown in Table. 2. Some sampled results on video sequences are illustrated in Fig. 8. From Table 2, we

Table 2 The AUC/Precision scores on each sequence in OPE

Sequence	IRF-TLD	TLD	Struck	SCM	OAB	ASLA	IVT	LSK	CT	DFT
Faceocc1	***0.739***/0.802	0.581/0.203	0.718/0.575	**0.780**/0.933	0.652/0.253	0.325/0.180	0.716/0.645	0.484/0.122	0.630/0.330	0.678/0.622
dudek	***0.704***/0.781	0.640/0.597	0.723/0.897	**0.756**/0.883	0.649/0.685	0.725/0.755	*0.742*/0.886	0.721/0.820	0.640/0.418	0.682/0.662
david	0.693/1.000	0.707/1.000	0.249/0.329	*0.711*/1.000	0.395/0.384	**0.735**/1.000	0.637/1.000	0.558/0.669	0.497/0.815	0.301/0.314
Blurface	0.665/0.753	**0.856**/1.000	*0.764*/0.972	0.363/0.237	0.569/0.284	0.129/0.097	0.159/0.112	0.638/0.765	0.234/0.148	0.328/0.282
mhyang	***0.643***/0.956	0.627/0.978	0.803/1.000	0.794/1.000	0.737/0.944	**0.897**/1.000	0.783/1.000	*0.823*/1.000	0.596/0.819	0.699/0.765
boy	0.636/0.942	0.653/1.000	0.747/1.000	0.370/0.440	*0.777*/1.000	0.362/0.440	0.256/0.332	**0.786**/1.000	0.586/0.930	0.393/0.485
Faceocc2	***0.619***/0.792	0.611/0.856	**0.771**/1.000	0.716/0.860	0.593/0.708	0.637/0.792	0.717/0.993	0.623/0.663	0.602/0.681	*0.755*/1.000
girl	***0.605***/0.960	0.566/0.918	**0.734**/1.000	0.672/1.000	*0.711*/1.000	0.700/1.000	0.172/0.444	0.303/0.486	0.314/0.608	0.293/0.296
Football1	***0.497***/0.784	0.369/0.554	*0.661*/1.000	0.396/0.568	0.271/0.378	0.488/0.797	0.549/0.811	0.317/0.473	0.237/0.351	**0.838**/1.000
trellis	***0.494***/0.503	0.481/0.529	0.610/0.877	*0.665*/0.873	0.141/0.193	**0.788**/0.861	0.251/0.332	0.665/0.967	0.341/0.387	0.358/0.506
fleetface	***0.488***/0.358	0.486/0.506	**0.602**/0.639	*0.600*/0.529	0.521/0.444	0.563/0.301	0.457/0.264	0.503/.0314	0.517/0.438	0.480/0.358
Shaking	0.386/0.381	0.394/0.405	0.356/0.192	**0.680**/0.814	0.015/0.005	0.465/0.485	0.037/0.011	0.459/0.466	0.109/0.047	*0.627*/0.830
Freeman1	***0.381***/0.902	0.283/0.540	0.343/0.801	**0.609**/0.982	0.358/0.742	0.265/0.390	*0.427*/0.807	0.249/0.368	0.149/0.396	0.392/0.942
Biker	0.278/0.514	0.299/0.493	0.263/0.514	**0.385**/0.500	0.203/0.507	*0.367*/0.500	0.363/0.493	0.239/0.535	0.008/0.063	0.251/0.514
DragonBaby	***0.269***/0.265	0.147/0.195	**0.363**/0.504	0.172/0.159	0.177/0.230	0.244/0.274	0.249/0.327	*0.306*/0.416	0.283/0.248	0.147/0.124
KiteSurf	***0.248***/0.405	0.232/0.345	0.288/0.417	**0.603**/0.833	0.159/0.452	0.289/0.667	0.304/0.405	0.171/0.560	0.209/0.452	*0.388*/0.619
matrix	***0.162***/0.160	0.162/0.160	0.097/0.120	0.259/0.350	*0.263*/0.400	0.129/0.050	0.024/0.020	**0.302**/0.510	0.019/0.020	0.060/0.060

The bold fonts indicate the best performance, the underline indicate the second best performance, while the italic fonts indicates the sequences that IRF-TLD performs better than TLD

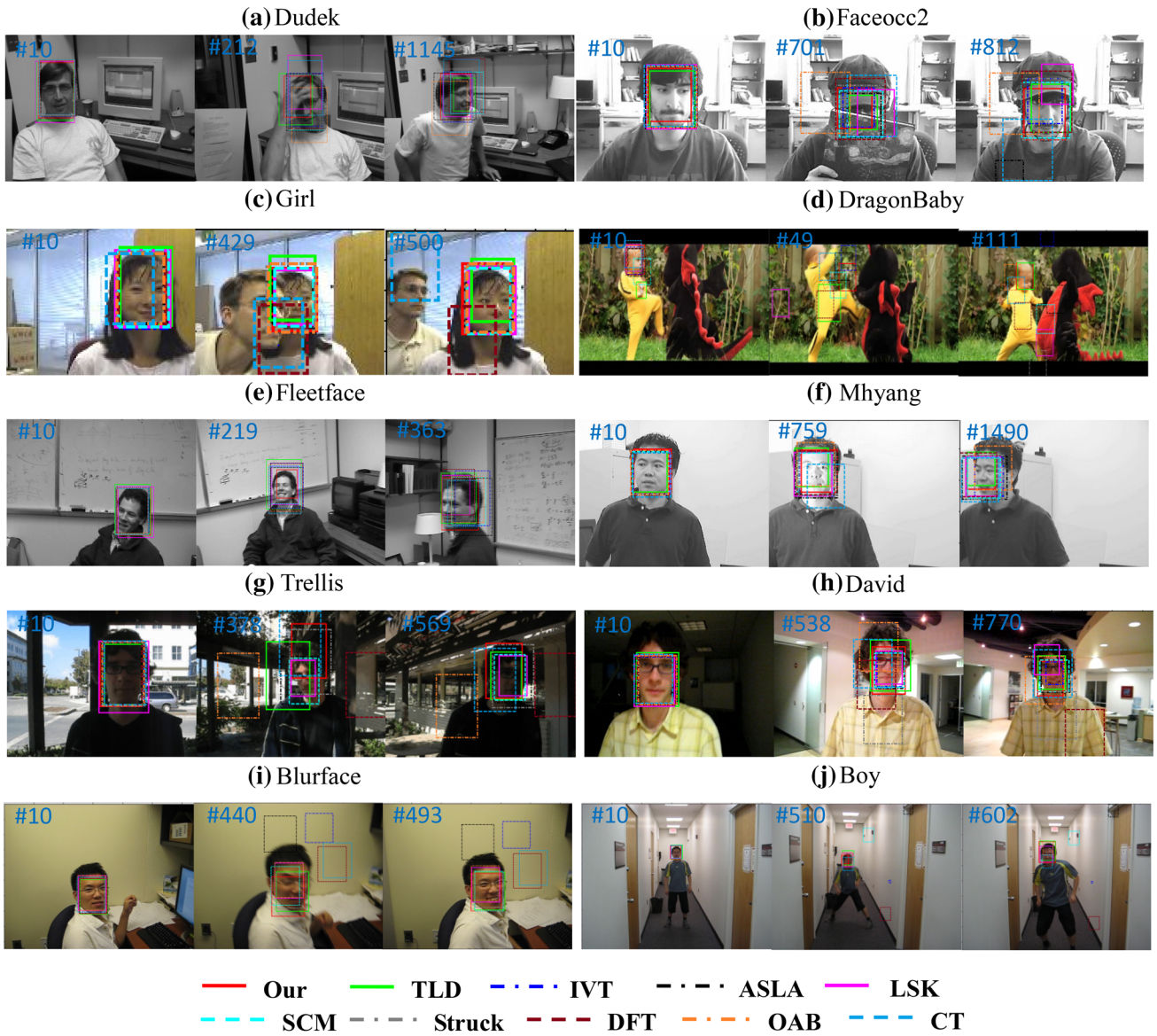


Fig. 8 Screenshots from some of some sampled tracking results

can observe that IRF-TLD performs better on 12 out of 17 sequences than TLD. Note that there exist many challenging factors in these videos that IRF-TLD achieves favorable results. For instance, the sequences *dudek*, *faceocc2*, *girl* and *dragonbaby* have the attributes of occlusion, in which *dudek*, *girl* and *dragonbaby* also have scale variation and (in-)out-of plane rotation attribute, thereby making them far more challenging. Notwithstanding, IRF-TLD performs persistently well from beginning to end. Meanwhile, the data from Table 2 show that IRF-TLD achieves higher AUC scores when compared with TLD on above sequences. Furthermore, the sequences *mhyang*, *trellis* and*david* have the attributes of illumination variation and out-of-plane rotation. The AUC scores of our IRF-TLD exceed that of TLD (except for sequence *david)*; in contrast, on other sequences with the fast motion, motion blur and in-plane rotation attribute, such as *blurface* (#440/#493) and *boy* (#510/#602), the tracking results with IRF-TLD may gradually drift when the target starts to move fast. This further verifies that the fast motion attribute significantly affects the performance of our IRF-TLD. However, when the target recovers the normal motion, such as the #493 frame in *blurface* and #602 frame in *boy*, IRF-TLD can accurately localize the target again.

In fact, no tracker can perform better than others on all video sequences. Particularly worth mentioning is that the memory requirements when comparing IRF-TLD with TLD. As mentioned in 4.2.2, the proposed IRF-TLD owns the advantage of low memory requirement. In this experiment, the parameter of standard random fern in TLD are set to $T = 10$, $S = 13$, then its memory requirement (MEM_{TLD})

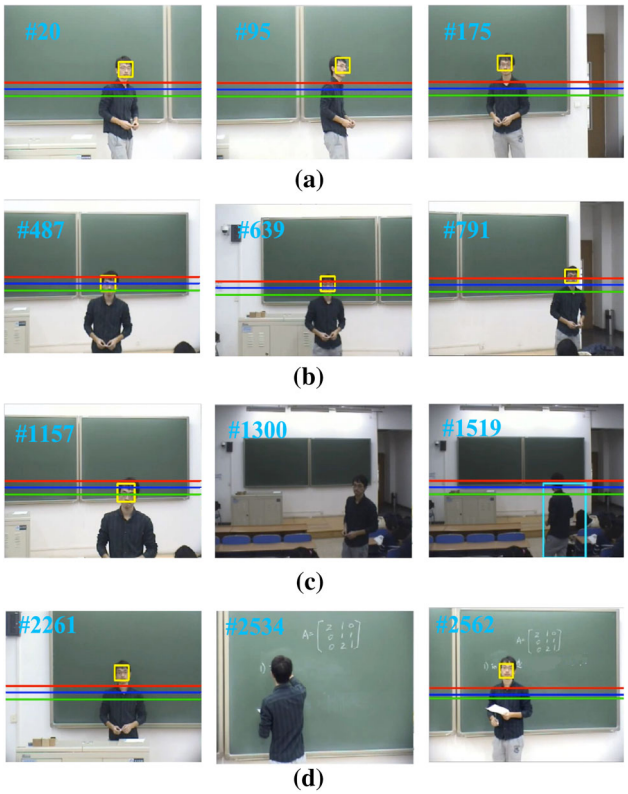


Fig. 9 Snapshots of our ITTS. **a** Rostrum area **b** Transition area, **c** student area, **d** blackboard writing. The *light blue rectangle* is used to represent the body detection result by GMM

is 6,55,360 bytes. However, in our IRF-TLD, $T = 50$, the memory requirement (MEM_{Our}) is 400 bytes, thus it can save more memory resources.

According to these experiments on the benchmark sequences, the IRF-TLD algorithm, with its excellent performance, is proved suitable for our system. Next, we will combine the camera control to verify the performance of our ITTS.

5.2 System performance

To evaluate the effectiveness of our ITTS, especially the teacher tracking in teaching process, a series of sample scenarios were tested in real classrooms.

5.2.1 Scenario 1: single person tracking

The purpose in this scenario is to verify the feasibility and effectiveness of camera control. The teacher can walk at normal pace, change direction, turn his back to the camera or write on the blackboard. The split lines L_1 (red), L_2 (blue), L_3 (green) in Fig. 9 were initialized in terms of the initial target box and the size of the classroom as described in Sect. 4.4.2. Figure 9a displays the case that the teacher camera can automatically swing to keep a smooth track of the target when the teacher walked around the Rostrum area. Once the image location of the teacher's face was detected having crossed the line L_2 (frame 487) or moved across the lines L_3 (frame 1157), following the rules stated in Sects. 4.4.1 and 4.4.2, the IRF-TLD face tracking and GMM body detection will be used alternatively to calculate the new location of the teacher in the transition area or student area. Moreover, the output data of the new position are used to control the PTZ teacher camera moving to the destination (preset Z2 or Z3) automatically. If the result showed that the teacher returned back to the platform, the IRF-TLD for teacher face tracking will start again (frame 2261).

The close-up operation will be trigged as showed in Fig. 9d. When the teacher was writing on the blackboard (frame 2534), it was known from Sect. 4.4.2 that this performance could be explained by Eq. (11). Besides, the zoom out operation was carried out when the teacher's face became visible again, and then our ITTS successfully recovered the teacher face tracking (frame 2562).

For the following Scenario 2 and 3, all of the scenes were arranged just in rostrum, therefore, those split lines will no longer be displayed.

5.2.2 Scenario 2: occlusion

This scenario is used to test the robustness of the system in case of occlusion. We design the scene where the teacher interacted with the student in the platform. For the long-term tracking, it is necessary to recover from failure and re-detect after full occlusion. Figure 10 shows the sample sequences. In frame 3516, the tracking failed due to the heavily occlusion. The cascade detector of our teacher tracking algorithm (IRF-TLD) is able to re-detect the target and correct the tracking failure when the target appears in the camera's field of view. As can be seen, the system recovered to track the teacher in frame 3528 when the teacher's face became visible again. In the case when the teacher stays occluded for a long time, our system would be mistaken for blackboard writing with

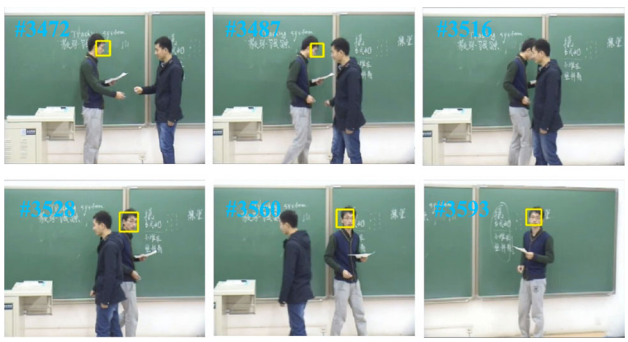


Fig. 10 Keeping track through occlusion by the student

Fig. 11 Keeping track through projection screen

the close-ups, but it is able to recover quickly as long as the teacher's face becomes visible in the view field eventually.

5.2.3 Scenario 3: illumination variation

The goal in this scenario was to test the robustness of the system in case of illumination changes (Fig. 11). As can be seen from the images (frame 1–299), our system is robust enough to keep stable track although the light intensity changes obviously when the teacher passed in front of the projection screen. Besides, the occlusion under the situation of lighting variation appears in frame 3049. Our system can still detect the correct target and recover quickly in frame 3054.

The sample scenarios above have suggested that our ITTS is able to keep tracking the teacher through strong lighting variations or occlusions. Meanwhile, the stability of recovering from track losses ensures the long-term tracking ability of our system. With the combination of a highly reliable automatic control strategy for monocular PTZ camera, our ITTS is guaranteed to have the capability of keeping smooth track as well as quickly obtaining close-ups of blackboard writing.

6 Conclusion

We proposed an effective and efficient way to build our ITTS for displaying a high-quality teaching scene with two generic commercial PTZ cameras. An active tracking module with two threads: Teacher tracking and stand-up student detection can be automatically triggered to fulfill our ITTS function. The Student Camera is applied to observe the activities of students, while the Teacher Camera is responsible for tracking the teacher during lecturing.

Two main contributions of our system have been detailed in Sect. 4. First, we designed a fully automatic face detection and tracking mechanism for long-term teacher face tracking with resistance to occlusion and illumination changes by complementary algorithms such as the face Adaboost detection algorithm, a novel long-term tracking algorithm IRF-TLD and GMM body detection algorithm. The proposed IRF-TLD tracking algorithm, whose superior performance has been demonstrated on benchmark challenging sequences when compared with some other state-of-the-art tracking algorithms, fits well to the control part to form a real-time ITTS. Besides, the IRF-TLD has advantages of high accuracy and low memory requirement, therefore, it is very appropriate for embedded systems. Second, based on image position, unique and effective close-loop feedback strategies for monocular active vision camera, such as the split lines and multi-rectangular block division (MRBD) camera Pan–tilt control strategy as well as the rules of camera zoom controlling, are created to make the real-time-specific target face tracking robust to view field changes.

There are several possible directions to extend this work. First, an accuracy and robust stand-up student detection is still an open issue necessitating further improvements. Another interesting line of further research is developing our framework on the embedded device such as DSP for broader applications.

Acknowledgments The authors thank the anonymous reviewers for helping to review this paper. This work was partially supported by National Natural Science Foundation of China (60974108).

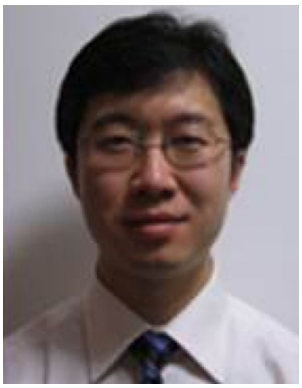
References

1. Tsuruoka, S., Yamaguchi, T., Kato, K., Yoshikawa, T., Shinogi, T.: A camera control based on fuzzy behavior recognition of lecturer for distance lecture. In: 10th IEEE International Conference on Fuzzy Systems, pp. 940–943 (2001)
2. Ruiguo, Y., Xinrong, Z.: The design and implementation of face tracking in real time multimedia recording system. In: 2nd International Congress on Image and Signal Processing (CISP), pp. 1–3 (2009)
3. Wulff, B., Rolf, R.: Opentrack-automated camera control for lecture recordings. In: IEEE International Symposium on Multimedia (ISM), pp. 549–552 (2011)
4. Rui, W., Lei, M.: Intelligent Tracking Teaching System based on monocular active vision. In: IEEE International Conference on Imaging Systems and Techniques (IST), pp. 431–436 (2013)
5. Wang, R., Wang, Y.Y., Wang, L., Chen, X.Q., Zhu, S.P.: Robust and automatic tracking method of infrared extended object based on EM-like algorithm. Infrared Laser Eng. **37**(4), 616–620 (2008)
6. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1409–1422 (2012)
7. Zhang, J., Liu, K., Cheng, F., Li, Y.: Visual tracking with randomly projected ferns. Signal Process. Image Commun. **29**(9), 987–997 (2014)
8. Bouguet, J.-Y.: Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. Intel Corp. Microprocess. Res. Labs Tech. Rep. (2000)
9. Ji, H.: Real time robust L1 tracker using accelerated proximal gradient approach. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1830–1837 (2012)
10. Ross, D.A., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental learning for robust visual tracking. Int. J. Comput. Vision. **77**(1–3), 125–141 (2008)

11. Xue, M., Ling, H.: Robust visual tracking and vehicle classification via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. **33**(11), 2259–2272 (2011)
12. Jia, X., Lu, H., Yang, M.-H.: Visual tracking via adaptive structural local sparse appearance model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1822–1829 (2012)
13. Zhong, W., Lu, H., Yang, M.-H.: Robust object tracking via sparsity-based collaborative model. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1838–1845 (2012)
14. Hare, S., Saffari, A., Torr, P.H.: Struck: Structured output tracking with kernels. In: IEEE International Conference on Computer Vision (ICCV), pp. 263–270 (2011)
15. Grabner, H., Grabner, M., Bischof, H.: Real-Time Tracking via Online Boosting. In: British Machine Vision Conference (BMVC), pp. 47–56 (2006)
16. Wang, Z., Yoon, S., Xie, S.J., Lu, Y., Park, D.S.: Visual tracking with semi-supervised online weighted multiple instance learning. Vis. Comput. pp. 1–14 (2015)
17. Wang, N., Li, S., Gupta, A., Yeung, D.Y.: Transferring Rich Feature Hierarchies for Robust Visual Tracking. Eprint Arxiv (2015)
18. Henriques, J.O.F., Caseiro, R., Martins, P., Batista, J.: High-Speed Tracking with Kernelized Correlation Filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 1–1 (2014)
19. Zhang, K., Zhang, L., Liu, Q., Zhang, D., Yang, M.H.: Fast Visual Tracking via Dense Spatio-temporal Context Learning. In: European Conference on Computer Vision (ECCV), pp. 127–141. Springer International Publishing (2014)
20. Kalal, Z., Mikolajczyk, K., Matas, J.: Face-tld: Tracking-learning-detection applied to faces. In: 17th IEEE International Conference on Image Processing (ICIP), pp. 3789–3792 (2010)
21. Wu, Y., Lim, J., Yang, M.-H.: Online object tracking: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2411–2418 (2013)
22. Zafeiriou, S., Zhang, C., Zhang, Z.: A survey on face detection in the wild: Past, present and future. Comput. Vision Image Underst. **138**, 1–24 (2015)
23. Zhao, W.Y., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face recognition: A literature survey. ACM Comput. Surv. **35**, 399–458 (2003)
24. Al Haj, M., Bagdanov, A.D., Gonzalez, J., Roca, F.X.: Reactive object tracking with a single PTZ camera. In: International Conference on Pattern Recognition (ICPR), pp. 1690–1693 (2010)
25. Feng, P., Xuanyin, W., Quanqi, W.: Moving object tracking research based on active vision. In: Fifth World Congress on Intelligent Control and Automation (WCICA), pp. 3846–3849 (2004)
26. Chen, H., Zhao, X., Tan, M.: A novel pan-tilt camera control approach for visual tracking. In: 11th World Congress on Intelligent Control and Automation (WCICA), pp. 2860–2865 (2014)
27. Bernardin, K., Van De Camp, F., Stiefelhagen, R.: Automatic person detection and tracking using fuzzy controlled active cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8 (2007)
28. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted Gaussian mixture models. Digital Signal Process. **10**(1), 19–41 (2000)
29. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 511–518 (2001)
30. Zhang, K., Zhang, L., Yang, M.-H.: Real-time compressive tracking. In: 12th European Conference on Computer Vision (ECCV), pp. 864–877 (2012)
31. Ozuysal, M., Fua, P., Lepetit, V.: Fast keypoint recognition in ten lines of code. In: IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pp. 1–8 (2007)
32. Achlioptas, D.: Database-friendly random projections: Johnson-Lindenstrauss with binary coins. J. Comput. Syst. Sci. **66**(4), 671–687 (2003)
33. Baraniuk, R., Davenport, M., DeVore, R., Wakin, M.: A simple proof of the restricted isometry property for random matrices. Construct. Approx. **28**(3), 253–263 (2008)
34. Wang, Z., Salah, M.B., Zhang, H.: Object joint detection and tracking using adaptive multiple motion models. Vis. Comput. **30**(2), 173–187 (2014)
35. Wu, Y., Jia, N., Sun, J.: Real-time multi-scale tracking based on compressive sensing. Vis. Comput. **31**(4), 471–484 (2014)
36. Kalal, Z., Matas, J., Mikolajczyk, K.: Pn learning: Bootstrapping binary classifiers by structural constraints. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 49–56 (2010)
37. Liu, B., Huang, J., Yang, L., Kulikowsk, C.: Robust tracking using local sparse appearance model and k-selection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1313–1320 (2011)
38. Sevilla-Lara, L., Learned-Miller, E.: Distribution fields for tracking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1910–1917 (2012)
39. Zhang, K., Liu, Q., Wu, Y., Yang, M.-H.: Robust Tracking via Convolutional Networks without Learning. arXiv preprint arXiv:1501.04505 (2015)

Rui Wang is an associate professor in the School of Instrumentation Science and Opto-electronics Engineering at Beihang University, China. Her current research interests include Computer Vision and Image Processing, Machine Learning and Visual Tracking.

Hao Dong is currently pursuing the MS degree in the School of Instrumentation Science and Opto-electronics Engineering at Beihang University, China. His research interests mainly include Visual Tracking and Image Processing.



Tony X. Han is an associate professor in Electrical and Computer Engineering Department, University of Missouri-Columbia, USA. His research interests mainly include Computer Vision, Machine Learning and Human Computer Interaction.

Lei Mei received the MS degree from the School of Instrumentation Science and Optoelectronics Engineering at Beihang University, China. His research interests mainly include Software System Development.