Robust human tracking via key face information

Weisheng Li[1], Xinyi Li[1], Lifang Zhou[1]
[1] Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[e-mail: liws@cqupt.edu.cn]
[e-mail: lixinyimayday@163.com]
[e-mail: zhoulf@cqupt.edu.cn]
*Corresponding author: Xinyi Li

Received March 22, 2016; revised July 10, 2016; revised August 18, 2016; accepted September 3, 2016; published October 31, 2016

Abstract

Tracking human body is an important problem in computer vision field. Tracking failures caused by occlusion can lead to wrong rectification of the target position. In this paper, a robust human tracking algorithm is proposed to address the problem of occlusion, rotation and improve the tracking accuracy. It is based on Tracking-Learning-Detection framework. The key auxiliary information is used in the framework which motivated by the fact that a tracking target is usually embedded in the context that provides useful information. First, face localization method is utilized to find key face location information. Second, the relative position relationship is established between the auxiliary information and the target location. With the relevant model, the key face information will get the current target position when a target has disappeared. Thus, the target can be stably tracked even when it is partially or fully occluded. Experiments are conducted in various challenging videos. In conjunction with online update, the results demonstrate that the proposed method outperforms the traditional TLD algorithm, and it has a relatively better tracking performance than other state-of-the-art methods.

Keywords: Visual tracking; Tracking-Learning-Detection (TLD); Key face information; Occlusion; Prediction

This work was supported in part by Natural Science Foundation of China (No. 61100114, 61272195, 61472055, U1401252), Chongqing Research Program of Application Foundation and Advanced Technology (cstc2014jcyjjq40001, cstc2015jcyjA40011), China Scholarship Council (201407845019) and Chongqing research and innovation project of graduate students (CYS15157). The authors would like to thank the anonymous referees for their valuable comments and suggestions.

http://dx.doi.org/10.3837/tiis.2016.10.026 **ISSN : 1976-7277**

1. Introduction

Visual tracking [1-5], as a significant pattern recognition topic, is vital for computer vision. Its real-world applications cover intelligent monitoring and transportation, robot navigation, video content analysis and management, etc. Although latest developments in hardware, e.g. RGB-D cameras can make cluttered background no longer relevant, significant changes in the appearance of an object caused by occlusion, illumination change, cluttered background, and also fast/abrupt object motion are still a challenge for visual tracking when the ordinary high definition cameras are used [2, 4]. The recent years have witnessed significant advances in visual tracking with the development of efficient algorithms and fruitful applications [1-9]. But how to completely solve the occlusion and get a robust tracker is still a highly difficult task.

There are several related tracking approaches to solve the problem of partial or full occlusions. As a classical occlusion target tracking algorithm, Mean-shift [5] is a non-parametric pattern matching algorithm based on kernel density estimation. It is utilized to describe the color distribution of the object by setting up a weighted histogram. The tracking algorithm based on trajectory prediction such as particle filters [6] uses the moving inertia to predict the position of the object. In OAB tracker [7], tracking is posed as a discriminative classifier in order to obtain more performance characteristics. And in Compressive Tracking (CT) [8] proposed by Zhang, tracking is formulated as searching for the discriminative feature using the compressed sensing to improve the accuracy of local features. Moreover, the multiple instance learning (MIL) tracker [9] extends multiple instance learning to online learning in order to obtain a more robust tracking. The IVT tracker [10] finds an adaptive appearance model that accounts for limited deformable motion or appearance variation. Although they are able to address the problem of tracker drift, these methods [6-10] do not handle the problem of large non-rigid shape deformation, especially when the target gets complete occlusion then re-appears in camera[11-12].

Above-mentioned methods realize tracking by an online learned detector. That detector discriminates the target from its background. In other words, it is just a single process representation for detection and tracking.

Compared with these methods, TLD [1] is a special algorithm which has an outstanding performance in long term single target tracking. Its tracking and detection are independent processes that exchange information through learning. By keeping the tracking and detection separated, TLD does not have to compromise in its tracking or detection capability. Therefore, TLD has a more outstanding tracking effect than those algorithms which mentioned above. It contains three core parts: In the process of tracking, L-K optical flow and backward-forward error method are used to track target human; In the learning part, the P-N learning method is utilized to correct mistakes made by the tracker and detector. It has a strong ability to learn and restore; In the process of detection, the random ferns method is used. However, frequent and straightforward updates of tracking results may gradually lead to tracking drift problems as a result of accumulated errors in the

original TLD tracker, especially when occlusion occurs. And it also does not handle the problem of clutter backgrounds. In the worst scenario, the tracker never recovers from the drifts in the subsequent frames.

As the targets are generally presented in context, it has a close link among the contextual objection or state itself at different times. Numerous algorithms have been proposed using the effective appearance model [4, 10]. Yang and Wu [13] utilized data mining to automatically find advantageous auxiliary objects. Grabner [14] propose an online algorithm within the Harris operator to extract relevant features and characteristics associated with the target model between objects. Cerman [15] proposed to utilize a single consistent with the objectives of the campaign "regional partners" (company region) to improve the tracking performance. When target disappeared because of occlusion or severe apparent changes, the context model can be used to predict the position of tracking target and it is possible to suppress the error message leading to an apparent shift at the same time.

Based on this advantage, to revise the problem of occlusion, a simple yet robust tracking algorithm is proposed in this paper that exploits both holistic templates and local representations. With Tracking-Learning-Detection framework [1] and context information [13-15], it combines the context information and auxiliary model to the TLD tracker for object detection, thereby enabling the tracker to deal with the drift problem. The contributions of this paper are threefold: (1) spatial information is utilized in the proposed as context information to get a more accurate tracking result; (2) a correlation model is put forward to get the relation between target objection and locally useful information. The purpose of them is to find the tracking position when the tracker has lost target; (3) new rectangle regions are picked as positive samples to re-initialize the detector and tracker when fully occlusion happens. Compared with other tracking algorithms, our method has better performances in long-time tracking and has obtained promising results.

2. Review of Tracking-Learning-Detection

Tracking-Learning-Detection is proposed by Zdenek Kalal, which is designed for long-term tracking of an unknown single object where tracking and detection run in parallel [1]. That decomposes the task into three subtasks: tracking, learning, and detection. Each section is shown in **Fig. 1**. The motion model of the target is established under the assumption that the frame-to-frame motion is limited [2, 3], and all objects are visible. However, the tracker tends to drift and never recover if the object moves out of the camera view. The detector treats every frame as independent and localizes all the observed appearances, yet detector may make two types of errors: false positive samples and false negative samples; the role of the learning part is to assess detector’s errors and updates it to avoid these errors in the future.

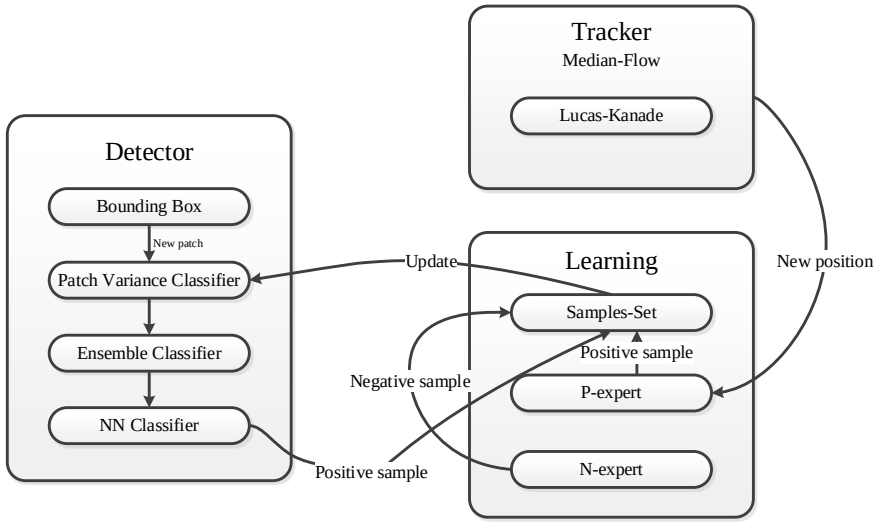

Fig. 1. the flow chart of TLD

2.1 Learning

Learning is a semi-supervised learning. As the term suggests, two types of "experts" which analysis sample classified as a negative or positive label can identify the errors of detector. Let x indicate an example of a feature space X and Y indicate the label $Y = \{-1,1\}$. The combination of both of them $L = \{(x, y)\}$ is called a labeled set. The input of P-N learning is a label set $L_l = \{(x_l, y_l)\}$ and an unlabeled set X_u. By iterative learning, the last iteration is utilized to obtain the classifier to classify all the unlabeled samples.

(x_u, y_u^k) indicates the sample set (x_u, y_u) getting through the classifier after k iterations. The output sample set (x_c^k, y_c^k) is obtained via the structural constraints. P-expert analyzes negative examples, estimates false negatives, corrects them and adds them to the training set with positive label 1. In the same vein, N-expert analyzes positive examples, evaluates false positives, and adds them with negative label -1 to the training set.

2.2 Detector

Target detection module is built on the basis of search box and random forest classifier. Random forest classifier is composed by 10 decision trees. Each tree can be established by a training sample $s = (x, l)$. The gray features of the sample can be represented by x and a sample category labels can be shown by $l \in \{0,1\}$. Each leaf node of each tree saves the posterior probability $P_i(y \mid x)$. It indicates the probability of positive samples in the leaf node.

$$P_i(y \mid x) = \frac{\# p}{\# p + \# n} \tag{1}$$

where $\#p$ indicates the number of positive samples, and $\#n$ indicates the number of negative samples. In the sample training, each internal node needs the dichotomous tests like this:

$$t_{pa,pb,\theta}(x) = \begin{cases} 1 & x(pa) - x(pb) > \theta \\ 0 & otherwise \end{cases} \tag{2}$$

where $x(pa)$ indicates the grayscale value of the position pa, $x(pb)$ indicates the grayscale value of position pb. Θ means a random threshold.

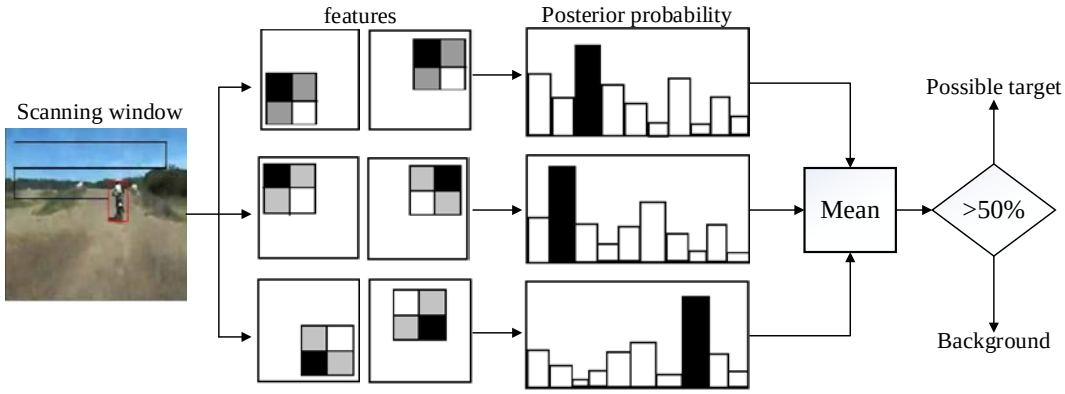

Fig. 2. the detect process of TLD

The steps of detecting module include:

Step1: Training a random forest classifier based on the first frame of the target. Each random forest classifier includes 10 decision trees.

Step2: In subsequent frames, the sliding window is used to get the grayscale value of an image. And according to the **Eq. (2)**, the classification of image blocks in the window is obtained by the random forest classifier.

Step3: When the reliability of the test results is higher, the detection results are utilized as the sample information and then train each decision tree in a random forest classifier.

2.3 Tracker

The tracking part of the TLD is based on forward-backward trajectory optical flow. Traditional optical flow [16, 17] assumes that the brightness change of two adjacent targets is not obvious. In TLD, each frame is divided into many uniform grids. Then top left vertex of every grid is chosen as the feature point. Depending on the features' position in the last frame, successively, the forward-backward optical flow is used to find the corresponding position in the next frame. The basic principle of forward-backward trajectory optical flow is shown in **Fig. 3**.

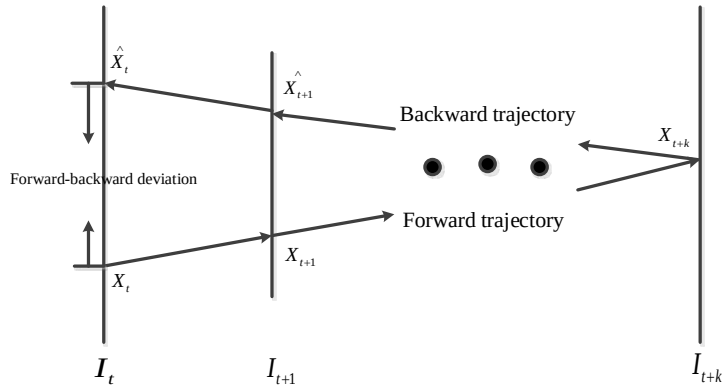

Fig. 3. forward-backward trajectory optical flow

Assuming that the video sequences are $S=(I_t, I_{t+1},\ldots,I_{t+k})$, X_t is a point of I_t. At first, LK optical flow is used to obtain forward trajectory $T_f^k=(x_t, x_{t+1},\ldots,x_{t+k})$ where x_{t+1} indicate the point in I_{t+i}. Then supposing the x_{t+k} of I_{t+k} as start point, use LK tracker to do reverse forecast until the first frame I_t and get backward trajectory $T_b^k=(\hat{x}_t, \hat{x}_{t+1},\ldots,\hat{x}_{t+k})$, $\hat{x}_{t+k}=x_{t+k}$. The error between them is calculated by **Eq. (3)** and **Eq. (4)**:

$$FB(T_f^k|S)=dis(T_f^k,T_b^k) \tag{3}$$

$$dis(T_f^k,T_b^k)=\|x_t-\hat{x}_t\| \tag{4}$$

A displacement deviation could be got via the method of forward-backward trajectory optical flow. If the $FB(T_f^k|S_{t+1})$ of I_{t+1} is less than $FB(T_f^k|S_t)$ of I_t, it shows that this point is more reliable. Otherwise, the trajectory of forward tracking is incorrect. This point will be removed. Therefore, in each frame, few feature points will be removed. Tracking an updatable feature set makes tracker more robust.

Although TLD model is a simple and efficient tracking that can online learning the most apparent characteristics, it is unable to retrieve track results when the tracking targets disappear and reappear in the video. The problems mainly come out during the learning process. With the errors of the target model are accumulated, the classification performance of classifier is declining. In addition, these may eventually lead to tracking drift. That is, when prolonged occlusion happens or the appearance of a target has serious change, TLD algorithm is prone to track failure.

3. Improved TLD method

In response to these problems, a robust TLD framework method is proposed in the paper which adds contextual information to improve tracking accuracy and reduce tracking failures. Since the use of the auxiliary information to help locate the track target. We are no longer continuing the traditional binary classification for an image, but redraw the image into three parts. That is, tracking body as the target, face information as the auxiliary objects and background.

Firstly, face detector is used to get face information as the tracking auxiliary information. Then, we establish the model of a human face and tracking targets. Thirdly, the update scheme is presented to refine the relative variation in the position of human face and tracking body.

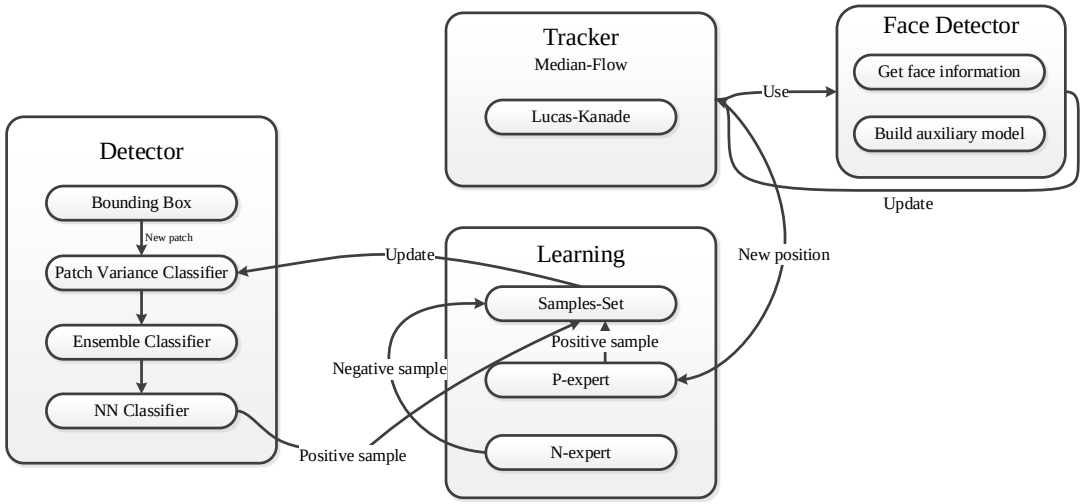

Fig. 4. the flow chart of the proposed algorithm

3.1 Optimized Part Mixtures and Cascaded Deformable Shape Model

The robust method named as Optimized Part Mixtures and Cascaded Deformable Shape Model (OPCD) [18] is used to capture face information accurately in the improved TLD method. OPCD utilizes a two-stage cascaded deformable shape model to effectively and efficiently localize facial landmarks with large head pose variations.

This optimized mixtures framework [18] what we use as a face detector can simultaneously handle face detection, pose-free landmark localization and tracking in real time. In order to promote the performance, this method introduces a pictorial structure [19] to organize the landmarks. For the purpose of achieving real-time performance for tracking, it proposes a group sparse learning based method. The framework can automatically select the landmarks and re-organize them into a new tree structure part mixture, which dramatically decreases the number of landmarks and still preserves the detection

effectiveness.

First, the optimized mixtures method is utilized to locate the position of the tracked face as the spatial information. Then the corresponding model would be obtained. By the first step, a set of point coordinates can be obtained. Knowing calibration points set, the following formula is used to obtain the center coordinates:

$$\begin{cases} x_{center} = \{\max(x) + \min(x)\} / 2 \\ y_{center} = \{\max(y) + \min(y)\} / 2 \end{cases} \tag{5}$$

where (x_{center}, y_{center}) indicates the center of facial calibration points. With the center position, face box's width and height can be obtained. By setting the face information as the auxiliary objects, the auxiliary model between the human face and tracking body can be established.

Fig. 5. face center position determination

3.2 The auxiliary model between human face and tracking body

The vital face information is obtained from Optimized Part Mixtures and Cascaded Deformable Shape Model which proposed by Yu [18]. Through the above method, the center of the face information is known as $F = (x_f, y_f)$. In order to establish the relation between the human face and tracking body, we use the auxiliary model within the TLD framework.

The basic idea of the algorithm is to continuously update the trajectories between the face and tracking body, which are based on association model. According to the motion relationship, the relative position relationship between them can be obtained. In particular, the target object can be viewed as a relative position vector (0, 0) of the auxiliary object.

Specifically, the algorithm was proposed in this paper works as follows: In the first frame, the position of the target object is determined by the value of groundtruth. We can get the position information (x, y, x_w, y_h). The first two values show the coordinates of the upper left corner point of the bounding box. The latter two values show the width and height of the bounding box. As known from the tracking human body and human face's position, the relative position vector *R* between them would be obtained as follows:

$$R = (x + x_w / 2 - x_f, y + y_h / 2 - y_f) \tag{6}$$

When the tracking target is occluded in frame *t*, we can use the face detector to get face auxiliary information in frame *t*, and then use the latest auxiliary model to estimate the

approximate location of the tracking target $P_t = F_t + R_{t-1}$. Through the above formula, new tracking boxes can be obtained. Then the new tracking box is picked as positive samples to re-initialize the detector and tracker.

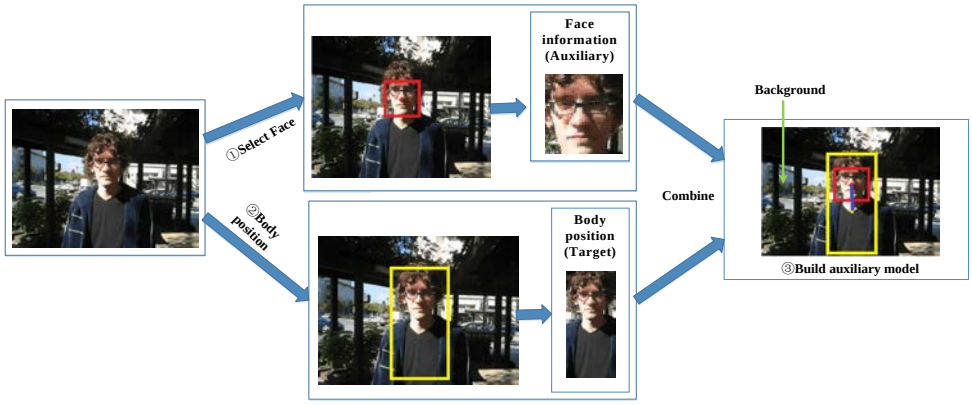

Fig. 6. the flow chart of auxiliary model

3.3 Update scheme

Since the appearance of a target and the auxiliary model often changes significantly during the tracking process, the update scheme is extremely important and necessary.

For the face detector and the auxiliary model, we update them every several frames. In our paper, the system is set to update in every 5 frames. Through a lot of experiments show that updates every 5 frames can balance the accuracy rate and time efficiency. Comparing with no updating or update in more than 5 frames, the former's track results is not accurate enough and the latter is more time-consuming with no growth of accuracy.

The algorithm steps are shown as follows:

The Proposed Algorithm:

Input: Video sequences
Output: Image sequences which marked tracking results
Step 1: Initialization
Mark a target object bounding box and use face detector to get the auxiliary information F. Initialize the relative positional relationship between two of them R.
Step 2: Use the TLD framework to track. Tracking the target object and face information.
If the target is occluded
Depending on the latest relative position vector R_{t-1}, the current frame face position F_t is used to get target position $P_t = F_t + R_{t-1}$.
Else
Keep tracking
Step 3: Update scheme
Find the change in the relative position. Update it in every several frames (5 in our experiments)

4. Experiments

In this section, we test our tracker on publicly available video sequences about human tracking to evaluate the performance of our proposed tracker. The sequences come from LEAR dataset, which is a joint team of Inria, Laboratoire Jean Kuntzmann, the CNRS, and Univ. Grenoble Alpes. SPEVI Datasets and visual tracker benchmark [20, 21] are used to test our tracking method with other 29 trackers. All Datasets have ground-truth annotations. The initial position of the image sequence is determined by the ground-truth annotations.

These sequences cover challenges in human tracking: heavy occlusion, motion blur, large illumination change, fast motion, complex background and so on. **Table 1** lists all the evaluated image sequences indoor and outdoor.

Table 1. Video sequence introduction

Video sequence	Total frame	Challenging factor
Trelli	569	Illumination Variation，Scale Variation，Background Clutters, Rotation
Jumping	313	Motion Blur, Fast Motion
Freeman1	326	Scale Variation, Rotation
Girl	501	Occlusion, scale and pose change
FaceOcc1	892	Occlusion
FaceOcc2	812	Occlusion，Illumination Variation，Rotation
Dudek	1145	Occlusion，Fast Motion, Rotation, scale and pose change
Mhyang	1490	Illumination Variation，Background Clutters，Deformation
David	770	Illumination Variation, Scale, Motion Blur, Occlusion, Rotation, Deformation

Specifically, the proposed tracker is evaluated against 11 state-of-the-art visual tracking algorithms, including the TLD [1], CT [8], Frag [26], Struck [23], IVT [10], SCM [24], VTD [22], L1APG [25], MIL [9], DFT [27], OAB [7] trackers. For fair evaluation, the code that comes from visual tracker benchmark [20, 21] is used to evaluate our algorithm where the parameters of each method are tuned for best performance. Compared with these leading methods, we demonstrate the effectiveness and robustness of our method. The proposed algorithm is implemented in Matlab2011b on a PC with an Intel(R) Core(TM) i5-4590u processor and 8G RAM. For fairness, we use the publicly available benchmark source code to get the evaluation value. For each sequence, the location of the target object is manually labeled in the first frame with ground truth.

4.1 Quantitative comparison

To quantitatively evaluate the performance of each tracker, we use three widely accepted evaluation metrics including the successful tracking rate (STR) [20], the average center location error (ACLE) [20] and the average overlap rate (AOR) [20]. The successful tracking rate is the ratio of the number of successful tracking frames and the number of the sequence. If the overlap between the predicted and ground truth bounding box is more than 0.5, we label the frame as the successful tracking frame. The center location error is the

Euclidean distance between the center of the tracking result and the ground truth for each frame. The overlap rate is based on the Pascal VOC criteria. Given the tracked bounding box ROI_T and the ground truth bounding box ROI_{GT} , the overlap rate is computed by $area(ROI_T \cap ROI_{GT}) \ / \ area(ROI_T \cup ROI_{GT})$. To compare the tracking performance, we compute the average center location error and the average overlap rate across all frames of each video sequence as done in most tracking literatures. For fairness, we evaluate our proposed algorithm by taking the average over five runs. Due to spatial limitations, we list the results of 11 leading algorithms. **Table 2, 3** and **Table 4** report the quantitative comparison results respectively.

Table 2. STR the best three results are shown in red and bold, blue and green. The results of some algorithm come from visual tracker benchmark [20, 21]

	CT	IVT	Frag	SCM	VTD	Struck	L1APG	MIL	DFT	OAB	**TLD**	**Proposed**
David	0.4268	0.7941	0.121	0.913	0.6773	0.2357	0.6921	0.2293	0.2335	0.3581	0.9703	**0.971**
Trellis	0.3497	0.3093	0.3585	**0.8541**	0.5009	0.7838	0.1529	0.2443	0.5185	0.2475	0.4728	0.6712
Mhyang	0.7302	**1**	0.7228	0.9973	0.9483	**1**	0.9705	0.3886	0.7752	0.7019	0.8933	0.9964
Dudek	0.8524	0.9677	0.5895	0.9755	**1**	0.9799	0.7939	0.8568	0.8009	0.4662	0.8419	0.9310
Girl	0.178	0.186	0.536	0.882	0.652	**0.98**	0.97	0.294	0.252	0.6311	0.764	0.83
Fleetface	0.6379	0.4639	0.454	0.7058	**0.7143**	0.6662	0.5474	0.5375	0.5559	0.4805	0.5672	0.6757
Freeman1	0.1012	0.3252	0.1994	**0.8067**	0.2147	0.2147	0.1411	0.1534	0.1779	0.4336	0.2117	0.6037
Jumping	0.0064	0.099	0.8466	0.1214	0.1118	0.7987	0.1182	0.476	0.1182	0.5985	0.8466	**0.8626**
Faceocc1	0.8543	0.9753	**1**	**1**	0.9249	**1**	**1**	0.7646	0.8027	0.7609	0.8341	0.9024
Faceocc2	0.7438	0.9138	0.7537	0.8744	0.9938	**1**	0.8017	0.936	0.9951	0.6432	0.8288	0.9483

The results of average overlap rate are come from visual tracker benchmark [20, 21]. These data have proven that the higher is the better.

Table 3. AOR the best three results are shown in red and bold, blue and green. The results of some algorithm come from visual tracker benchmark [20, 21]

	CT	IVT	Frag	SCM	VTD	Struck	L1APG	MIL	DFT	OAB	**TLD**	**Proposed**
David	0.5257	0.6506	0.4218	**0.7379**	0.6471	0.6096	0.5399	0.5475	0.5876	0.3581	0.6515	0.7252
Trellis	0.3008	0.3706	0.3196	**0.6912**	0.5133	0.688	0.483	0.2516	0.3727	0.2475	0.4767	0.5320
Mhyang	0.5837	0.8274	0.5999	**0.8302**	0.7702	0.7644	0.795	0.6139	0.6737	0.7019	0.6813	0.7329
Dudek	0.6462	0.6909	0.5518	0.6156	**0.73**	0.6353	0.6376	0.6472	0.5825	0.4662	0.552	0.6822
Girl	0.3456	0.4172	0.5128	**0.6813**	0.509	0.6725	0.6735	0.4125	0.4922	0.6311	0.5235	0.6179
Fleetface	0.5062	**0.5490**	0.5126	0.5474	0.525	0.544	0.473	0.5008	0.5137	0.4805	0.4592	0.4710
Freeman1	0.3870	0.4544	0.3803	0.4737	0.4391	0.406	0.317	0.45	0.4939	0.4336	0.3559	**0.6120**
Jumping	0.3489	0.3663	0.7409	0.2431	0.317	0.7265	0.4342	0.6171	0.2677	0.5985	0.6457	**0.7523**
Faceocc1	0.6572	0.77	0.8579	**0.8624**	0.677	0.7919	0.7602	0.6466	0.7402	0.7609	0.6409	0.8476
Faceocc2	0.6297	0.6723	0.5128	0.6959	0.6877	**0.7498**	0.6895	0.6606	0.722	0.6432	0.6226	0.7054

The results of **Table 4** are average overlap rate which comes from visual tracker benchmark [20, 21]. These data have proven that the lower is the better.

Table 4. ACLE the best three results are shown in red and bold, blue and green. The results of some algorithm come from visual tracker benchmark [20, 21]

	CT	IVT	Frag	SCM	VTD	Struck	L1APG	MIL	DFT	OAB	**TLD**	**Proposed**
David	20.24	5.49	31.9	**5.282**	11.01	12.328	19.2	14.485	19.187	35.726	13.606	8.256
Trellis	59.174	81.40	60.573	16.68	27.89	**10.076**	38.58	69.021	52.842	72.063	31.278	22.751
Mhyang	14.71	2.9843	19.639	4.044	5.520	4.037	4.962	13.506	10.756	9.3473	8.21	**4.023**
Dudek	21.478	19.01	52.18	38.065	19.595	23.49	33.49	19.654	44.628	71.678	24.93	**14.811**
Girl	16.995	16.437	12.597	5.591	12.39	**4.9244**	5.939	15.388	13.281	6.863	12.463	6.68
Fleetface	68.146	45.518	58.574	**40.66**	61.835	44.028	98.013	61.759	57.645	58.01	50.734	44.142
Freeman1	28.918	13.442	29.952	33.42	29.675	29.124	79.514	19.173	19.507	22.65	27.79	**7.435**
Jumping	22.358	32.58	4.25	45.98	25.897	4.439	23.999	8.9594	52.9	15.96	7.627	**4.2**
Faceocc1	25.212	13.27	6.545	**6.08**	17.052	12.075	14.58	26.32	18.686	15.8	22.01	19.263
Faceocc2	17.721	**8.791**	34.09	11.51	10.744	7.461	13.22	14.355	11.254	15.70	11.31	10.486

Table 2, **Table 3** and **Table 4** shows STR, AOR and ACLE obtained by the tracking algorithms with different similarity metrics on the video sequences. Compared with traditional TLD algorithm, we can draw the following conclusions that our proposed algorithm achieves the better overall performance at 10 sequences excluding Fleetface sequences and so on.

Table 5 FPS: The Frames per second. The table shows the result of traditional TLD and the proposed [20, 21]

	David	Trellis	Mhyang	Dudek	Girl	Fleetface	Freeman1	Jumping	Faceocc1	Faceocc2
TLD	26.8	24.5	15.1	6.52	9.05	10.53	26.4	13.82	11.05	23.6
proposed	7.12	5.9	5.58	2.04	2.48	3.48	8.06	4.25	3.18	6.5

Compared with the traditional TLD method, the proposed algorithm surpasses the previous algorithms in all video sequences while the time complexity of the proposed algorithm is increased. The proposed method resolves the situation that target reappears but the original method cannot recapture it. It is due to the model by combining the state of occlusion and appearance variation at each time step. Compared with other methods, our proposed algorithm achieves the relatively best overall performances in David, Trellis, Fleetface, Freeman1, Jumping on the evaluation of STR and on the evaluation of ACLE and AOR, it reveals that the proposed algorithm could be able to deal with such problems as occlusion, scale and pose change, background clutters, fast motion. It is almost against all the sequences and is more effective and robust than other state-of-the-art algorithms.

4.2 Qualitative Evaluation

The tracking results of 11 trackers are plotted for qualitative comparison as shown in **Fig. 7**. The results are discussed based on the main challenging such as occlusion, appearance variation, scale variation, illumination variation, rotation etc.

Occlusion. In the sequences of David, Faceocc1, Faceocc2, Girl, Dudek, occlusion occurs in all databases. From **Fig. 7** we can see that all algorithms fail to track the target at frame 979 and 1115 when the target was occluded except TLD and our algorithm. From **Table 5**, In Dudek, The Frames per second (fps) of traditional TLD is 6.52, while the fps of the proposed method is 2.04. However, the result of ACLE is best, and in AOR, the result of Dudek is relatively better than other state-of-the-art methods. In the faceocc2 sequence, the fps of traditional TLD is 23.6; however, the fps of the proposed algorithm is 6.5. When his hat occludes the target man, all other trackers fail to locate the object except SCM. However, our proposed method handles the problem well and tracks the target accurately. In the other sequences with occlusion, our tracker achieves at least the third best performance through the entire sequence. Once the target is partially occluded or heavily occluded, our method can be able to relocate the target with the model. Moreover, our method considers occlusion as well.

Scale and pose change. In the David, Dudek, Girl, Freeman1, Trellis sequences, the target human undergoes heavy shape variation, especially in the Trellis and girl. From **Table 2, Table 3, Table 4** and **Table 5**, In Freeman1, although the fps of traditional TLD is 26.4, the fps of the proposed method is 8.06. The results of the three evaluation indicators are better than other algorithms. From **Fig. 7**, in David, at frame 745 and in Trellis, at frame 112 all the other methods tend to drift, but our proposed approach locates the man well. At frame 390, only SCM and our tracker can track the target successfully.

Illumination Variation. In Trellis, David, Faceocc2, Mhyang, these sequences have very strong illumination changes. From **Table 4**, Obviously, our algorithm has better performance in these video sequences. From **Fig. 7**, at frame 390, many other algorithms have been drifting. However, the proposed algorithms can capture the tracking target successfully.

Rotation. Rotation is one of the challenge factors in the sequences of Trelli, Freeman1, Faceocc2, Dudek, David. From **Table 4**, the improved TLD have a better performance in all these sequences. Especially, in Freeman1 and Dudek, our algorithm has the best performance. From **Fig. 7**, at frame 1115 in Dudek, The tracking results of all algorithms have drifted more or less except our algorithms and TLD.

As mentioned in occlusion, scale variation and shape change, our tracker can handle the appearance variation better than the other methods.

Fig. 7. tracking results

5. Conclusions

Based on TLD, we propose a simple yet robust algorithm which adds the context to establish the model. Compared with traditional algorithm, the appearance auxiliary model is estimated to deal with the appearance variation, possible occlusion and target disappearance. Because of the importance of face information, firstly, we use optimized part mixtures and cascaded deformable shape model to define the face position. Then, according to the initial tracking target position, the relative position between the face and tracking target is established. Extensive experimental results and evaluations against several state-of-the-art methods on some challenging video sequences demonstrate the effectiveness and robustness of our proposed algorithm. Since the algorithm is conducted

in the visual tracker benchmark, considering the practical application of our algorithm, we will improve the performance in real time. Besides, various tracking algorithms that can track multiple objects have been proposed; we will achieve the function of tracking different people in the complex environment.

References

[1] Kalal Z, Mikolajczyk K, Matas J., "Tracking-Learning-Detection [J]," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 34(7): 1409-1422, 2011. Article (CrossRef Link).

[2] Yilmaz A, Javed O, Shah M., "Object tracking: A survey [J]," *Acm Computing Surveys*, 38(4): 81-93, 2006. Article (CrossRef Link).

[3] Ioffe S, Forsyth D., "Human Tracking with Mixtures of Trees[C]," in *Proc. of Computer Vision, ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 690-690, 2001. Article (CrossRef Link).

[4] Beleznai C, Frühstück B, Bischof H., "Human Tracking by Fast Mean Shift Mode Seeking [J]," *Journal of Multimedia*, 1(1): 1-8, 2006. Article (CrossRef Link).

[5] Tada K, Takemura H, Mizoguchi H., "Robust tracking method by MeanShift using Spatiograms[C]," in *Proc. of SICE Annual Conference 2010, Proceedings of IEEE*, 1985-1988, 2010. Article (CrossRef Link).

[6] Nummiaro K, Koller-Meier E, Gool L V., "An adaptive color-based particle filter [J]," *Image & Vision Computing*, 21(1): 99-110, 2003. Article (CrossRef Link).

[7] Grabner H, Grabner M, Bischof H., "Real-Time Tracking via On-line Boosting [C]," in *Proc. of the British Machine Conference*, 47-56, 2006. Article (CrossRef Link).

[8] Zhang K, Zhang L, Yang M H., "Real-time compressive tracking [C]," in *Proc. of the 12th European conference on Computer Vision - Volume Part III*, Springer-Verlag, 864-877, 2012. Article (CrossRef Link).

[9] Babenko B, Yang M H, Belongie S., "Visual tracking with online Multiple Instance Learning [C]," in *Proc. of Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Computer Society Conference on*, 983-990, 2009. Article (CrossRef Link).

[10] Ross, D. A., Lim, J., Lin, R.-S. & Yang, M.-H., "Incremental Learning for Robust Visual Tracking," *International Journal of Computer Vision 77 (1-3)*, 125-141, 2008. Article (CrossRef Link).

[11] Zhang, Tianzhu, et al., "Robust Visual Tracking Via Consistent Low-Rank Sparse Learning," *International Journal of Computer Vision 111.2*, 171-190, 2014. Article (CrossRef Link).

[12] Kristan M, Matas J, Leonardis A, et al., "The Visual Object Tracking VOT2015 Challenge Results[C]," *ICCV*, 564-586, 2015. Article (CrossRef Link).

[13] Yang M, Wu Y, Lao S., "Intelligent Collaborative Tracking by Mining Auxiliary Objects [C]," in *Proc. of Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Computer Society Conference on*, 17-22, 2006. Article (CrossRef Link).

[14] Grabner H, Matas J, Gool L.V, Cattin P., "Tracking the Invisible: Learning Where the Object Might Be [C]," in *Proc. of Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Computer Society Conference on*, 1285-1292, 2010. Article (CrossRef Link).

[15] Cerman L, Matas J, Hlaváč V., "Sputnik Tracker: Having a Companion Improves Robustness of the Tracker [J]," *Lecture Notes in Computer Science*, 5575(6): 291-300, 2009. Article (CrossRef Link).

[16] Otero J, Otero A, Muniz R, et al., "Robust optical flow estimation[C]," in *Proc. of Image Processing (ICIP), 1999 IEEE Computer Society Conference on*, 780-784, 1999. Article (CrossRef Link).
[17] Senst T, Eiselein V, Sikora T., "Robust Local Optical Flow for Feature Tracking [J]," *IEEE Transactions on Circuits & Systems for Video Technology*, 22(9): 1377-1387, 2012. Article (CrossRef Link).
[18] Yu X, Huang J, Zhang S, et al, "Pose-Free Facial Landmark Fitting via Optimized Part Mixtures and Cascaded Deformable Shape Model [C]," in *Proc. of Computer Vision (ICCV), 2013 IEEE Computer Society Conference on*, 1944-1951, 2013. Article (CrossRef Link).
[19] Felzenszwalb P F, Huttenlocher D P., "Pictorial Structures for Object Recognition [J]," *International Journal of Computer Vision*, 61(1): 55-79, 2005. Article (CrossRef Link).
[20] Wu Y, Lim J, Yang M H., "Online Object Tracking: A Benchmark[C]," in *Proc. of Computer Vision and Pattern Recognition (CVPR), Proceedings of IEEE*, 2411-2418, 2013. Article (CrossRef Link).
[21] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, "Visual Tracking: An Experimental Survey," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 36: 1442-68, 2014. Article (CrossRef Link).
[22] Kwon J, Lee K M., "Visual Tracking Decomposition [C]," in *Proc. of Computer Vision and Pattern Recognition (CVPR), Proceedings of IEEE*, 1269-1276, 2010. Article (CrossRef Link).
[23] Hare S, Saffari A, Torr P H S., "Struck: Structured output tracking with kernels[C]," *IEEE Trans Pattern Anal Mach Intell*, 263-270, 2011. Article (CrossRef Link).
[24] Zhong W, "Robust object tracking via sparsity-based collaborative model [C]," in *Proc. of Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Computer Society Conference on*, 1838-1845, 2012. Article (CrossRef Link).
[25] Bao C, Wu Y, Ling H, et al., "Real time robust L1 tracker using accelerated proximal gradient approach [C]," in *Proc. of Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Computer Society Conference on*, 1830-1837, 2012. Article (CrossRef Link).
[26] A. Adam, E. Rivlin, and I. Shimshoni. "Robust Fragments-based Tracking using the Integral Histogram [C]," in *Proc. of Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Computer Society Conference on*, 798-805, 2006. Article (CrossRef Link).
[27] Sevilla-Lara L., "Distribution fields for tracking [C]," in *Proc. of Computer Vision and Pattern Recognition (CVPR).2012 IEEE Computer Society Conference on*, 1910-1917, 2012. Article (CrossRef Link).

Weisheng Li graduated from the School of Electronics and Mechanical Engineering at Xidian University in July 1997. He received his M.S. Degree and Ph.D. from the School of Electronics and Mechanical Engineering and School of Computer Science and Technology at Xidian University in July 2000 and July 2004, respectively. Currently he is a Professor at Chongqing University of Posts and Telecommunications. His research focuses on intelligent information processing and pattern recognition.

Xinyi Li received her B.S. degree in Computer Science and Technology from Chongqing University of Posts and Telecommunications, Chongqing, China. She is currently a graduate student at the Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, China. Her research interests include pattern recognition and machine vision.

Lifang Zhou was born in Tianshui, Gansu Province, PR China. She received her M.S. degree and Ph.D. degree from the Chongqing University of Posts and Telecommunications in July 2007 and the Chongqing University in December 2013, respectively. Currently she is an Associate professor of Chongqing University of Posts and Telecommunications. Her research focuses on pattern recognition and machine vision, etc.