PAPER

An Efficient Misalignment Method for Visual Tracking Based on Sparse Representation

Shan JIANG[†], ***Student Member***, **Cheng HAN**[†a)], ***and*** **Xiaoqiang DI**[†b)], ***Nonmembers***

SUMMARY Sparse representation has been widely applied to visual tracking for several years. In the sparse representation framework, tracking problem is transferred into solving an L1 minimization issue. However, during the tracking procedure, the appearance of target was affected by external environment. Therefore, we proposed a robust tracking algorithm based on the traditional sparse representation jointly particle filter framework. First, we obtained the observation image set from particle filter. Furthermore, we introduced a 2D transformation on the observation image set, which enables the tracking target candidates set more robust to handle misalignment problem in complex scene. Moreover, we adopt the occlusion detection mechanism before template updating, reducing the drift problem effectively. Experimental evaluations on five public challenging sequences, which exhibit occlusions, illuminating variations, scale changes, motion blur, and our tracker demonstrate accuracy and robustness in comparisons with the state-of-the-arts.

key words: *visual tracking, sparse representation, 2D transformation, template update*

1. Introduction

Computer vision system simulates human visual mechanism from the perspective of neurophysiology and psychology cognition. Relying on various imaging devices, computer can replace brain to understand visual image information. Visual target tracking, as an important branch of the field of computer vision, which is designed to simulate the ability of human eyes to estimate and track the motion of the target.

Object tracking is the core of computer vision, which combines the advanced achievements in different fields, such as image processing, pattern recognition, artificial intelligence, and automatic control [1], [2]. In the past decades, with the rapid development of multimedia technology and the continuous improvement of computer performance. Object tracking has important practical value and broad prospects in military vision guidance, robot vision navigation, industrial product inspection, medical diagnosis, traffic surveillance, virtual reality.

Recent years, sparse representation and compressed sensing techniques has been successfully applied to visual tracking [4], [7]–[10].The L1 tracker was first brought by [4], regarded tracking as finding a sparse approximation in template space, in addition, trivial templates were introduced to represent the occlusions and noise in the target during tracking. However, the method used in [4] for solving ℓ_1 minimization cost too much time to being real time. Then L1 tracker with minimum error bound was proposed by [8], particles were selected by the minimum error bound and accelerate the resampling procedure. Moreover, instead of updating template each frame, occlusion detection was added up before template update. However, these improvements are not enough to make the algorithm to be real time. Cheng [9] proposed the APG method to the original L1 tracker which reduced the computational cost and make the tracking algorithm to be real time. Although these trackers demonstrate good performance by using additional trivial templates, the tracking procedure can be generalized with better understanding.

In this paper, we present a robust object tracking algorithm. The contributions of this work are as follows. After sampling from the first frame of video sequence, we introduce a 2D spatial transformation to the observation set after sampling from the first frame, which effectively prevented error data from being introduced into the template, improved the robustness and tracking accuracy of the algorithm. In addition, at the template update stage, in order to balance the effective and performance, we detected the occlusion degree of the tracking target before the update scheme, updating schemes was determined based on the percentage of occlusion region. Consequently, we can obtain a more robust tracker than the L1 APG tracker. Our experimental evaluations on challenging video sequences validate the superior performance of our tracker to state-of-the-art trackers in terms of accuracy and robustness.

The rest of this paper is organized as follows. Section 2 briefly review the related work. Section 3 introduces the tracking framework, namely particle filter and sparse representation theory we used in this paper. Section 4 presents our tracking algorithm of the framework. Section 5 conducts experiments, Sect. 6 discusses a failure case of our tracker and Sect. 7 concludes the paper.

2. Related Work

Signal sparse representation has the advantages of fast computing speed, low storage, which has been widely used in various research fields. The application of sparse representation in signal processing domain mainly includes face recognition, image classification, image segmentation, image denoising and so on. Researches show that sparse

Manuscript received February 5, 2018.
Manuscript revised April 24, 2018.
Manuscript publicized May 14, 2018.

[†]The authors are with Changchun University of Science and Technology (CUST), Changchun, 130022, China.
a) E-mail: hancheng@cust.edu.cn (Corresponding author1)
b) E-mail: dixiaoqiang@cust.edu.cn (Corresponding author2)

DOI: 10.1587/transinf.2018EDP7052

representation have the advantage on solving the challenging problems in computer vision, such as large area occlusion and illumination change.

Resent years, many scholars put forward their own research and made their effort to improve the efficiency of tracking algorithms. According to the existing methods, visual tracking methods can be generally divided into two categories: discriminative methods and generative methods.

Discriminative methods formulated object tracking by the concept of classification. Different from generative methods, discriminative methods normally redefined the tracking problem as a binary classification which to distinguish whether a candidate target belongs to the background. Avidan [15] trained a SVM classifier for tracking, the SVM classifier is used to judge whether the candidate results are true, where obtained in the detection mechanism. But this method can't handle the situation where occlusion occurs. Then he proposed another tracking algorithm, Ensemble Tracker [16], which used data to learn multiple weak classifier and combined with Adaboost algorithm to accomplish object tracking. Babenko [17] studies MIL (Multiple Instance Learning) and applied it to object tracking, with which can solve the problem of uncertainty of the sample of list by placing the possible positive and negative samples in the positive and negative package respectively via multiple instance learning. Recent years, due to the powerful automatic feature extraction capability, deep learning has made great breakthrough on visual tracking. As a typical representative of discriminative methods, Wang [25] trained the tracker to learn image features from massive pictures offline, then tracked the object online. In order to solve the drift problem caused by inaccurate fine-tune in the model updating stage, the author designed two Convolution Neural Networks, CNNs and CNNI [26]. The former updated frequently so that it responds to the appearance change of the target, while CNNI updated less which is robust to error. Li [27] classified the state-of-art trackers based on deep learning from three aspects: network structure, network function and network training. Their research indicated that the CNN model has a high efficiency in template matching owing to its excellent performance on distinguishing the target from background.

Generative methods defined tracking as a similarity search problem. In this kind of method, object tracking algorithm based on sparse representation gained popularity, for this framework combined the sparse representation theory and particle filter tracking framework which achieves high tracking accuracy. Particle filter, which is also called the Sequential Monte Carlo methods [19], due to the dense sampling of particles for tracking result in high computation load, quantitative methods was brought out to improve the sampling efficiency. Rao-Blackwell [20] utilized subspace representation for tracking. Zhang [21] adopt a multi-task correlation filter to shepherd particles that reduce the number of particles as a result improving the real-time performance. John [3] proposed an algorithm that different expression of one man could be represented by a linear combination of the image of his face, and the sparsity coefficients also were sparse. Xue [4] utilized this sparse representation theory to solve object tracking problem, brought the idea of ℓ_1 -norm minimization for sparse tracking. Their work defined object tracking as the problem of searching for particles with minimum target reconstruction error in multiple candidate particles. The algorithm also assumed that the target candidate can be sparsely expressed by template dictionary constructed according to the target and trivial templates. By comparing the reconstruction error of each candidate particle and selecting the particle with the minimum error as the tracking result of the current time, then updated the template dictionary of the target to complete the continuous tracking.

Li et al. [8] constructed an object representation model based on the RIP (restricted isometry property) of compression perception theory, and utilized Orthogonal Matching Pursuit to tackle L1 minimization problem, Mei et al. [9] put forward BPR-L1 tracker which selected particles by minimum error bound, to a large extent reduced the number of particles during the ℓ_1 minimization. Bai [14] modeled the appearance of an object as a sparse linear combination of structured union, to solve the sparse representation issue he adopted the BOMP (Block Orthogonal Matching Pursuit) algorithm. Zhuang [11] proposed a tracking algorithm based on DSS (discriminative sparse similarity) map which illustrated the relationship between candidates and templates. Zhang [6] proposed a new algorithm for identifying spatial tracking with new graph embedding algorithm as core model. By combining the linear classifier and sparse representation theory, a sparse apparent model with discriminant properties is constructed [7], the tracking result was determined by reconstruction score of the apparent model. Wang [7] regards sparse representation for classification, sampling positive and negative samples, the sparse coefficients obtained in a complete dictionary are used to construct a linear classifier is used to estimate the target candidate's confidence value under two-step particle filtering. Hong [12] treated tracking as a multitask, multi-view sparse learning problem, which utilized multiple views to include various types of visual characteristics, such as intensity, color, and the edges. Each feature can be sparse represented as a linear combination of atom. A structure sparse tracking algorithm was constructed in [13], which not only used the intrinsic relationship between the target candidates to learn their sparse representation, but also preserved the spatial layout structure of local image blocks in each target candidate. Zhang [31] categorized the trackers based on sparse coding into appearance modeling based on sparse coding (AMSC) and target searching based on sparse representation (TSSR), he pointed that AMSR methods significantly outperform methods, an accurate description of the target appearance model affects the final tracking result. Considering the influence caused by noise, Bo [30] embedded a noise separation tracking mechanism into the LK tracking frame work, different from other methods, two noise items, Gaussian dense noise and sparse outlier noise

when constructing the representation model. Wang [29] presented a Least Soft-threshold Squares (LSS) method, handling outliers well via LSS distance. To obtain higher computational efficiency and more accurate results, the author utilized a cosine similarity function to measure the similarities between the target template and candidates. Lately, Wang [28] proposed a Least Soft-Threshold Square Tracking (LSST) algorithm, which formulate the error by a linear regression of Gaussian noise and Laplacian noise, alleviated the drifting problem under the condition of local occlusion and complex background, effectively improved the robustness of the algorithm.

3. The Tracking Framework of Particle Filter Joint Sparse Representation

3.1 Particle Filter

Filtering provides effective approach to estimate the current value of unknown on the assumption of the observations have already been known. In 1960, R.E. Kalman first brought the idea of state space into filter theory, Kalman Filter, which applied in temporal domain.

The Bayesian theory is an approach that estimate posterior probability distribution of state variables characterizing a dynamic system. Particle filter is a motion estimation algorithm in the Bayesian framework, which propagate massive random samples to describe probability distribution, and the random samples called particles.

Tracking a moving target from video can be considered as a process of estimating the posterior distribution of state variables. In the Bayesian framework, we need two variables:x_t, representing the observation x_t at time t of the tracking target which include the location, speed, x_t is continuously transferred from original state x_0, in order to decrease error caused by single variable, the introduction of observation z_t can deal with this situation, which utilized the camera to recorded the state of the target.

While tracking, in most cases we assume that the process of object state transition obey the Markov first order distribution. For all available observations $z_{1:t} = \{z_1, \ldots, z_{t-1}\}$, they are independent of each other.

Typically, there are two essentially steps to finish the tracking procedure: predict and update. In the predict stage, we obtained $p(x_t|z_{1:t-1})$ from $p(x_{t-1}|z_{1:t-1})$ as:

$$p(x_t \mid z_{1:t-1}) = \int p(x_{t-1} \mid z_{t-1}) p(x_{t-1} \mid z_{1:t-1}) dx_{t-1} \tag{1}$$

In the update stage, as $p(x_t \mid z_{1:t-1})$ had already known from previous work, by this time, we need to calculate $p(x_t \mid z_{1:t})$ as following:

$$p(x_t \mid z_{1:t}) = \frac{p(z_t \mid x_t) p(x_{t-1} \mid z_{1:t-1})}{p(z_t \mid z_{1:t-1})} \tag{2}$$

The denominator can be calculated by:

$$p(z_t \mid z_{1:t}) = \int p(z_t \mid x_t) p(x_t \mid z_{1:t-1}) dx_t \tag{3}$$

where $p(z_t \mid x_t)$ denotes the observation likelihood, which represents the appearance possibility of unknown state whose premises observations had already known, and this is also the logic connection between actual observation data and unknown variable. The particles are sampled according to the weight which updated by:

$$w_t^i = w_{t-1}^i \frac{p(z_t \mid x)}{q(x_t^i \mid x_{0:t-1}^i, z_{1:t})} \tag{4}$$

3.2 Sparse Representation of Tracking Target

The purpose of the sparse representation model is to calculating the observation likelihood of sample state x_t, when solving tracking problem, we cropped a patch from frame z_t to model $p(z_t \mid x_t)$. The patch needs to be normalized and stored in a 1D vector which act as target candidate that we denote as

$$T = [t_1, t_2, \ldots, t_n] \in \mathbb{R}^{d \times n} \ (d \gg n) \tag{5}$$

It is obvious that n target templates formed the target space where $a_i \in \mathbb{R}^d$. We utilized y to denote the tracking result, and $y \in \mathbb{R}^d$. According to Xue's work in [4], it could be approximately in this form:

$$y_t^i \approx TA = a_1 t_1 + a_2 t_2 + \ldots + a_n t_n \tag{6}$$

where $a_t^i = (a_1, a_2, \ldots, a_n)^T \in \mathbb{R}^n$ is referred to the target coefficient vector. When the target was corrupted by noise or partially occluded during tracking procedure may lead to unpredictable errors. To incorporate the corruption of occlusion and noise, (6) is rewritten as:

$$y_t^i = TA + e \tag{7}$$

e indicates error vector which could be caused by image impairment or inclement background. The nonzero data in e indicates the corresponding pixel which was corrupted by noise or occlusion. The locations of corruption can be obtained by trivial template $I = [i_1, i_2, \cdots, i_d] \in \mathbb{R}^{d \times d}$, and $i_i \in \mathbb{R}^d$ represent the $i - th$ entry of vector is 1, the rest is 0. We obtained the overcomplete dictionary $B = [T \quad I] \in \mathbb{R}^{d \times (n+d)}$ by combined the target candidate template space T and trivial template I, each observation y_t^i could be linearized by the overcomplete dictionary as

$$y_t^i = [T \quad I] \begin{bmatrix} a_t^i \\ e_t^i \end{bmatrix} = BC \tag{8}$$

Due to (8) is still underdetermined, it is difficult to acquire the unique solution of linear representation coefficient c_t^i. Considering for most of tracking sequences, only partial objective region was corrupted, therefore, there are finite nonzero entries in the trivial template coefficient vector e_t^i. The sparse coefficient vector obtained by minimizing the l_1 norm:

$$\min_{a,e} \|a\|_1 + \|e\|_1, \quad s.t. \quad y = TA + e \tag{9}$$

The observation model $p(z_t \mid x_t)$ reflects the likelihood

Table 1 Description of transformation algorithm

Algorithm 1

1: A_i represents target candidate set, contains K subjects, y stands for observation image and T is a deformation group

2: for each subject i

3: $\tau^{(0)} \leftarrow I$

4: while not converged $(j = 1,2,...)$ do

5: $\tilde{y}(\tau) \leftarrow \frac{y \circ \tau}{\| y \circ \tau \|_2}$; $J \leftarrow \frac{\partial}{\partial \tau} \tilde{y}(\tau)|_{\tau(j)}$;

6: $\Delta\tau = \arg\min \|e\|_1$ subj to $\tilde{y} + J\Delta\tau = TA_i + e$

7: $\tau^{(j+1)} \leftarrow \tau^{(j)} + \Delta\tau$;

8: end

9: end

10: select the top S candidates $k_1,...,k_S$ with the smallest residuals $\|e\|_1$

11: calculate an average transformation $\bar{\tau}$ from $\tau_{k_1}, \tau_{k_2}, \cdots, \tau_{k_S}$

12: for $i = k_1,...,k_S$

13: update $y \leftarrow y \circ \bar{\tau}$ and $\tau_i \leftarrow \tau_i \cdot \bar{\tau}^{-1}$

14: construct new set A with $\left[A_{k_1} \circ \tau_{k_1}^{-1} \mid A_{k_2} \circ \tau_{k_2}^{-1} \mid ... \mid A_{k_S} \circ \tau_{k_S}^{-1} \right]$

15: compute sparse vector according to [9]

estimation of the error caused by the approximation of the target through ℓ_1 minimization. The particle state transformation probability $p(x_t \mid x_{t-1})$ obey Gaussian distribution:

$$p(z_t \mid x_t) = \frac{1}{\Gamma} exp\left\{-\alpha \|y_t^i - TA\|_2^2\right\} \tag{10}$$

where Γ represents a normalized parameter, α is referred to a constant which is capable of controlling the shape of Gaussian kernel. The tracking result x_t^* at frame t is selected by:

$$x_t^* = argmax\, p(z_t \mid x_t^i) \tag{11}$$

4. Our Method

4.1 Transformation for Solving Misalignment

Wagner solve the misalignment of face recognition in [18], in fact, even images of the same object or scene can vary if the camera's position or pose change moderately. Motivated by his work, considering that the pose change of the object during moving will affect the tracker, we adjust the tracker before solving the optimal sparse solution so that it can change with the tracking candidate. We denote y as the observation image which is transformed by τ in the form:

$$y = y_0 \circ \tau^{-1} \tag{12}$$

where $\tau \in P$, and P represent a finite set of dimensional transformations with unknown parameters in the image domain, such as similarity transform, homograph, translation. And it is obvious that y no longer satisfied the sparse representation condition, therefore other feasible solution is needed. We estimate the transformation by linearizing the current estimate τ:

$$y \circ \tau + J\Delta\tau = TA + e \tag{13}$$

where $J = \frac{\partial}{\partial \tau} y \circ \tau$, stands for Jacobian of $z \circ \tau$, which is capable of estimating τ by repeatedly linearizing. This equation is still underdetermined. Now suppose there were only few corrupted pixels in the images between the well alignment image and ideally state, then we fixed error ϵ, and turn to seek the best solution for deformation step $\Delta\tau$

$$\Delta\hat{\tau}_1 = arg \min_{a,e,\Delta\tau \in T} \|e\|_1 \tag{14}$$
$$subject\ to \quad y \circ \tau + J\Delta\tau = TA + e$$

To prevent our algorithm from degenerating, we normalized $z \circ \tau$ in the way of

$$\tilde{y}(\tau) = \frac{y \circ \tau}{\|y \circ \tau\|_2} \tag{15}$$

With the best transformation parameter τ_i achieved by (15), the target candidate set A can be aligned to y. Detail procedure of our alignment algorithm is shown in Algorithm 1.

To obtain more reliable transformation, we used an average result $\bar{\tau}$, and while tracking, we need to align y with τ which is a parameterized set, consisted by $\tau = (\tau^1, \tau^2, \tau^3, \tau^4)$, each parameter has its own representation, where the first two represent the translations in x and y axis, the third represents the rotation angle and the last one denotes the scale. The average transformation τ is obtained by (16):

$$\bar{\tau}^i = \left(\tau_{k_1}^i + \tau_{k_2}^i + \cdots + \tau_{ks}^i\right) \Big/ S,\ i = 1, 2, 3, 4 \tag{16}$$

As we obtained the optimal $\bar{\tau}^i$, we utilized the method in [10] to get the sparsity solution.

4.2 Occlusion Detection

The dynamic template updating strategy enables the tracking algorithm to adapt to the various appearance changes during the tracking process. In the actual application scenario, the target is often corrupted by occlusion. When the template is updated, the shadowed trace results cannot be used as the updated target template. Therefore, in order to avoid incorrect template update, we need to detect the occlusion in the target area.

In sparse representations, regions where are contaminated or obscured by noise can be represented by trivial templates. The trivial template corresponding to its position is motivated when the feature value of the pixel in the target region cannot be approximated by the target template. First, the 1D trivial template is reconstructed into 2D trivial template coefficient image. Trivial template coefficients are relative to each pixel on the image. Then, the image of the block is obtained by binary processing of trivial template coefficients. On the binary occlusion image, the white pixel represents the occluded area, and the black pixel indicates the region that is not obscured. As the size of the occlusion is larger than the random noise, occlusion can be expressed as a large area of the white union region in the

Table 2 Description of update mechanism

Algorithm 2
1: y is the newly chosen tracking target, a is the solution to (9), w is current weights, θ, tr_1, tr_2 is predefined threshold
2: t_0 is the template which has the smallest coefficient, t_m has the largest coefficient
3: if ($sim(t_m, y) \geq \theta$)
4: calculate the corrupted rate tr
5: if($tr < tr_1$)
6: $t_0 \leftarrow y$
7: else if($tr_1 \leq tr \leq tr_2$)
8: replace the corrupted pixels with the median template, the specific locations could get from corruption map, then $t_0 \leftarrow rectified\ y$
9: end

occlusion map. Morphological operation was performed to remove the small area of white area and to fill the gap in the white connected area. If the largest white union region in the occlusion map is more than 30% of the entire image, the tracking result of the current frame is considered to have a significant occlusion and cannot be used as a template candidate.

Typically, a clear occlusion is retained in the target area for a period of time, so once a significant occlusion is detected, template will not be updated in the next few tracking sequences. This can effectively avoid drift problems caused by frequent template updates.

4.3 Template Update

The tracking algorithm based on particle filter and sparse representation generates the target template space in the first frame, then selects the region most similar to the target template space as the tracking result in the next tracking sequence. Because the target is often affected by various factors in the process of tracking, a fixed template space can't adapt to the appearance change of target area, and frequent template updating will bring drifting problem.

In fact, the appearance of the target will remain unchanged for a period of time, result in the final target template space will no longer be an accurate model representation of the target. If the template is not updated, the fixed template space will not be able to adapt to the changes caused by various illumination conditions or pose changes. But if the template is updated too frequently, error will be introduced when the template is updated each time, leads to offset the target area.

In order to dynamic update the target template space, the weight ω_i is introduced for each target template t. The larger weight value, the greater importance of the corresponding target template. The corresponding weights are evenly distributed because the differences between the templates in the initialized target template space are small. In the following tracking sequence, a indicates that the newly acquired tracking results correspond to the sparse representation coefficients of the target template t. Then, the weight of the template ω can be updated by the following formula.

$$\omega_i = \omega_i \exp(a_i),\ i = 1, 2, \ldots, n \tag{17}$$

In sparse representation, sparse representation coefficient indicates the similarity between the corresponding template and the tracking results. Therefore, we compare the largest coefficient t_m with the tracking result of the current frame. If the similarity is smaller than the preset threshold, which means that the current tracking result is similar to the target template space, the template does not need to be updated. If the similarity is larger than the preset threshold, it shows that the tracking result of the current frame is different from the target template space, so updating is needed. Choose the template which has the least weight as the tracking result, then averaging the weight of the rest target templates, and the averaging weight was regarded as the new template weight. Finally, the modified template space is normalized to the weight value according to (18). Detail for template update is shown in Algorithm 2.

$$\sum_{i=1}^{n} w_i = 1 \tag{18}$$

5. Experiments

In this section, firstly, we conduct several experiments to analyze and evaluate the method proposed in this paper. Secondly, we pose some experimental settings. Then we compared and analyzed five methods of tracking results on five challenging sequences. Finally, we make a comparison of speed among the five trackers.

5.1 Experimental Settings

Our algorithms is implemented with Matlab 2016a. On a computer with Quad-Core 3.30GHz Xeon processors and 8G memory, in order to evaluate the performance of our algorithm, we compile running on 5 challenging frame sequences (3 color sequences, 2 gray sequences), and compared the tracking algorithm with 4 state-of-art trackers named Multiple Instance Learning (MIL), Incremental Visual Tracking (IVT), Compressive Tracking (CT), L1 Tracker using Accelerated Proximal Gradient (L1APG). All of the sequences which have been used in this paper are obtained from http://cvlab.hanyang.ac.kr/tracker_benchmark/benchmark.html.

5.2 Qualitative Evaluation

The sequence Car4 shows a moving car and presents illuminating change. Figure 1 (a) shows the tracking results using 5 trackers. At the beginning of the experimental sequence, each algorithm can track the target accurately. During the subsequent tracking process, we can see that the IVT and MIL methods are less effective in this video sequence, our tracker shows good performance on illumination change, the

Fig. 1 A visualization of the tracking results of our tracker, MIL [17], CT [22], L1APG [10], IVT [1], on 5 challenging sequences (from top to down: Car4, Singer1, Clifbar, Walking2, Blurbody, respectively. Our tracker performs well against the state-of-the-art trackers)

rest trackers also have a good perform on this sequence.

The sequence Singer1 test the tracking algorithm mainly involves illumination change, scale change. As is shown in Fig. 1 (b). At the beginning, most algorithms can track the target well, as illumination increases gradually, the part of the target area was obscured by the lights, losing a lot of detail information, the scale of target also changing. Three trackers drift from the target, the rest are able to track the target, but have errors in scale. With the light gets dark, MIL, L1APG can track the target again, only IVT and our tracker can track the target throughout this sequence.

Figure 1 (c) illustrates the partial tracking results of the Clifbar sequence. The task is to track the man's face under the challenge of scale changing, occlusion. In the earlier sequences, all tracking algorithms can track targets accurately. When it comes to the 220th frame, the target area began to appear occlusion, and the 223th frame the target was completely blocked by a book, in this situation our algorithm and L1APG algorithm can still be successfully tracked. MIL, CT, IVT have occurred to varying degrees of drifting. In the later sequence, the scale of the target begins to change. Only the L1APG and our tracker achieve stable tracking results of the sequence.

Figure 1 (d) demonstrates partial tracking results of the Walking2 sequence. This series of sequences mainly test the robustness of each tracker under scale change, background interference, and occlusion. From the 187th frame, the man appears in the hall and similar to the target. As two objects moving in the scene, and occlusion happens, poses great challenge for tracking. It can been seen that L1APG, MIL, and IVT are gradually failing to track the target, our tracker can adapt to background clutter and handle occlusion situation, and able to tracking.

Figure 1 (e) is a part of the tracking result of the Blurbody sequence. During the video sequence, there is a significant motion blur caused by the camera shaking severely. L1APG fails to track the target in the earlier sequence, but relocates the target later, however, it drift from the target eventually. IVT, CT drift to the background, although MIL can locate the target, but has error in scale. This sequence shows that the proposed method with 2D transformation based sparse representation is effective in dealing the challenging situation.

5.3 Quantitative Evaluation

In the quantitative analysis, this paper adopts two criterion to measure the performance of the algorithms, one is precision, which is defined as the average Euclidean distance between the center of actual position of the target and the position of manual marking. Another evaluation metric is success rate which defined as $S_0 = |r_t \cap r_g|/|r_t \cup r_g|$. This criteria reflects the success ratio of bounding box overlap while the threshold ranging from 0 to 1. Results is shown in Fig. 2. We can infer from the curve that our tracker has the advantage over motion blur, occlusion, and scale variation. But our method has insufficient ability to deal with the problem of illumination transformation.

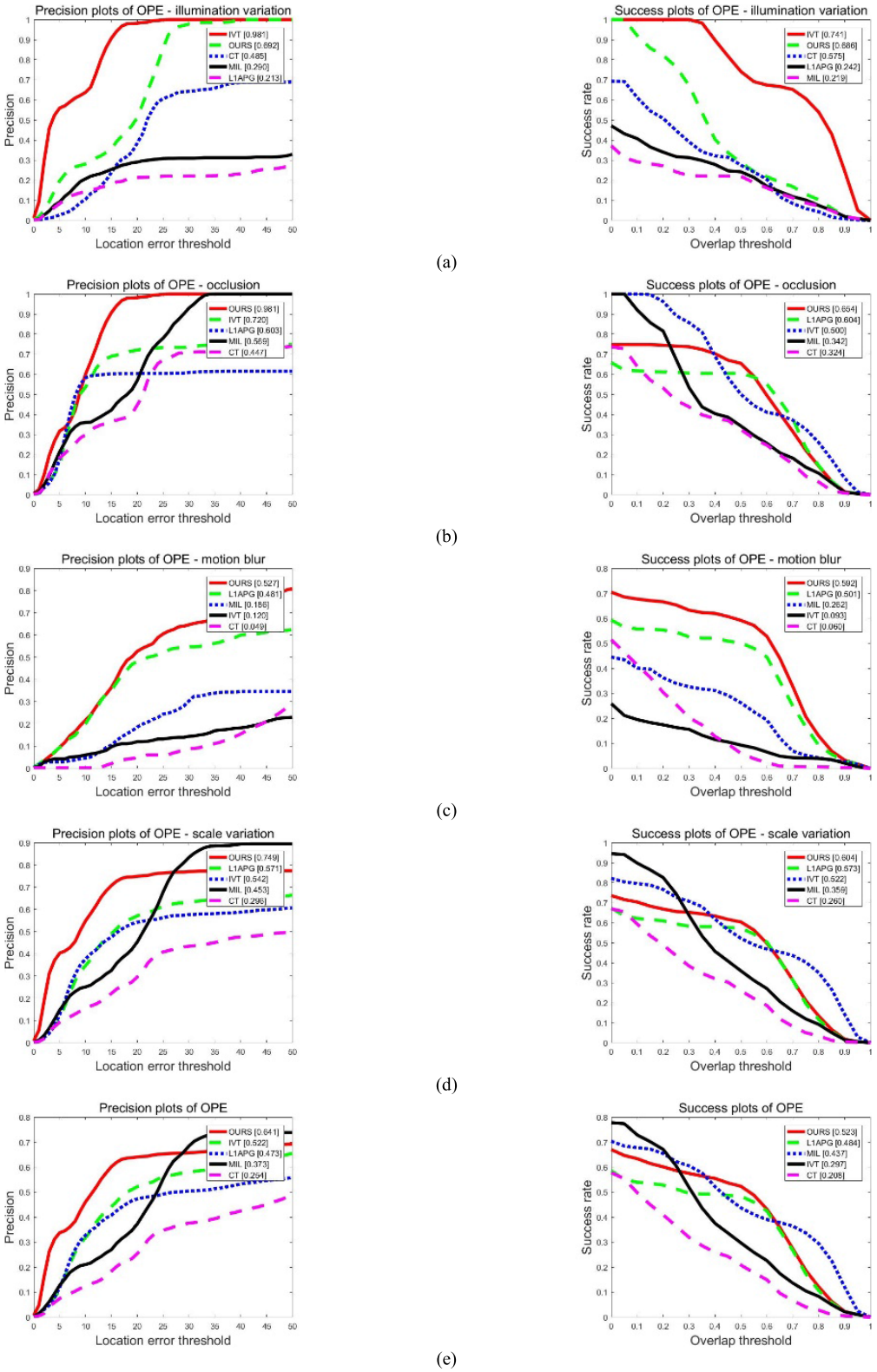

Fig. 2 Quantitative comparison of different algorithm: The success plots over four tracking challenges, including illumination variation (a), occlusion (b), motion blur (c), and scale variation. The last row is the precision and success plots over five sequences using one-pass evaluation, the average distance precision score at 20 pixels for each tracker.

Table 3 Speed of different trackers. The first and second best values are marked with bold and underline.

Squences	MIL	CT	L1APG	IVT	Ours
Car4	45.2	**130.4**	35.1	40.5	<u>46.3</u>
Singer1	38.7	**109.5**	18	35.6	<u>42.4</u>
Clifbar	<u>49</u>	**125.5**	28.9	40.8	47.4
Walking2	37.9	**113.7**	16.1	43.3	<u>44.8</u>
Blurbldy	25.6	**76.4**	23.5	<u>34.4</u>	33.6
AverageFps	39.3	**111.1**	24.32	38.9	<u>42.9</u>

Fig. 3 A failure tracking case of our tracker

5.4 Speed of Trackers

We further analyze the speed of our trackers with the other state-of-art trackers, results is shown in Table 3. The CT tracker is the most efficient among all evaluated methods, because the CT tracker utilize random sense matrix to reduce the dimension of multi-scale image features. Our tracker shows less advantage efficient than other trackers such as MIL, IVT, but the accuracy is superior to others, which is shown in Fig. 2.

6. Discussion

Although our tracker obtains good performance on the five sequences, however, there still exist limitation. It is necessary and important to analyze the reason making further improvement.

As is shown in Fig. 3, the car moving fast and it undergoes occlusion for a long time, our tracker drifted, the reason could be either appearance representation model could not be well described or the imperfection of template updating scheme. We will investigate this issue and more advanced appearance representation model and updating strategy will be studied in our future work.

7. Conclusion

In this paper, we employ a 2D transformation on the observation set based on sparse representation and L1 tracker, which effectively reduced the influence caused by the misalignment issue. The proposed algorithm can well handle occlusion, illumination changes, scale changes, motion blur. We analyze the performance of our tracker by comparing with 4 competing state-of-the-art methods on 5 challenging sequences. The results of qualitative and quantitative experiments demonstrated our tracker outperforms other algorithms in terms of efficiency, accuracy and robustness.

Acknowledgments

This work is supported by the National Science Foundation for Young Scientists of China (Grant No.61602058).

References

[1] D.A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental Learning for Robust Visual Tracking," International Journal of Computer Vision, vol.77, no.1-3, pp.125–141, 2008.
[2] J. Gao, T. Zhang, X. Yang, et al., "Deep Relative Tracking," IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 2017.
[3] J. Wright, A.Y. Yang, A. Ganesh, et al., "Robust Face Recognition via Sparse Representation," IEEE Trans. Pattern Anal. Mach. Intell., vol.31, no.2, pp.210–227, 2008.
[4] X. Mei and H. Ling, "Robust Visual Tracking using L1 Minimization," IEEE, International Conference on Computer Vision, DBLP, pp.1436–1443, 2009.
[5] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, and Y. Ma, "Towards a practical face recognition system: Robust registration and illumination by sparse representation," IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, IEEE, pp.597–604, 2009.
[6] X. Zhang, W. Hu, S. Maybank, and X. Li, "Graph Based Discriminative Learning for Robust and Efficient Object Tracking," IEEE International Conference on Computer Vision, pp.1–8, 2007.
[7] T. Bai, Y.F. Li, and X. Zhou, "Discriminative sparse representation for online visual object tracking," IEEE International Conference on Robotics and Biomimetic, pp.79–84, 2012.
[8] H. Li, C. Shen, and Q. Shi, "Real-time Visual Tracking Using Sparse Representation," Computer Science, 2010.
[9] X. Mei, H. Ling, Y. Wu, et al., "Minimum error bounded efficient L1 tracker with occlusion detection," pp.1257–1264, 2011.
[10] C. Bao, Y. Wu, H. Ling, and H. Ji, "Real time robust L1 tracker using accelerated proximal gradient approach," IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, pp.1830–1837, 2012.
[11] B. Zhuang, H. Lu, Z. Xiao, et al., "Visual tracking via discriminative sparse similarity map," IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, vol.23, no.4, pp.1872–1881, 2014.
[12] Z. Hong, X. Mei, D. Prokhorov, and D. Tao, "Tracking via Robust Multi-task Multi-view Joint Sparse Representation," IEEE International Conference on Computer Vision, pp.649–656, 2013.
[13] T. Zhang, S. Liu, C. Xu, et al., "Structural Sparse Tracking," pp.150–158, 2015.
[14] T. Bai, Y.F. Li, Robust visual tracking with structured sparse representation appearance model, Elsevier Science, 2012.
[15] S. Avidan, "Support vector tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no.8, pp.1064–1072, 2004.
[16] S. Avidan, "Ensemble Tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol.29, no.2, pp.261–271, 2007.
[17] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online Multiple Instance Learning," IEEE Conference on Computer Vision and Pattern Recognition, 2009, CVPR 2009. IEEE, pp.983–990, 2009.
[18] A. Wagner, J. Wright, A. Ganesh, et al., "Toward a practical face recognition system: Robust alignment and illumination by sparse representation," IEEE Trans. Pattern Anal. Mach. Intell., vol.34, no.2, pp.372–386, 2011.
[19] M. Isard and A. Blake, "CONDENSATION—Conditional Density Propagation for Visual Tracking," International Journal of Computer Vision, vol.29, no.1, pp.5–28, 1998.
[20] Z. Khan, T. Balch, and F. Dellaert, "A Rao-Blackwellized Particle

Filter for EigenTracking," Proc. 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, CVPR 2004, IEEE, Vol.2, pp.II-980–II-986, 2004.
[21] T. Zhang, C. Xu, and M.-H. Yang, "Multi-task Correlation Particle Filter for Robust Object Tracking," IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, pp.4819–4827, 2017.
[22] K. Zhang, L. Zhang, and M.H. Yang, "Real-time compressive tracking," European Conference on Computer Vision, pp.864–877, 2012.
[23] W. Guo, L. Cao, T.X. Han, S. Yan, and C. Xu, "Max-confidence boosting with uncertainty for visual tracking," IEEE Trans. Image Process., vol.24, no.5, pp.1650–1659, 2015.
[24] Y. Wu, J. Lim, and M.-H. Yang, "Object Tracking Benchmark," IEEE Trans. Pattern Anal. Mach. Intell., vol.37, no.9, pp.1834–1848, 2015.
[25] N. Wang and D.Y. Yeung, "Learning a deep compact image representation for visual tracking," International Conference on Neural Information Processing Systems, pp.809–817, 2013.
[26] N. Wang, S. Li, A. Gupta, et al., "Transferring rich feature hierarchies for robust visual tracking," Computer Science, 2015.
[27] P. Li, D. Wang, L. Wang, et al., "Deep visual tracking: Review and experimental comparison," Pattern Recognition, 2017.
[28] D. Wang, H. Lu, and M.-H. Yang, "Robust Visual Tracking via Least Soft-Threshold Squares," IEEE Trans. Circuits Syst. Video Technol., vol.26, no.9, pp.1709–1721, 2016.
[29] D. Wang, H. Lu, and M.-H. Yang, "Least Soft-Threshold Squares Tracking," IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, pp.2371–2378, 2013.
[30] C. Bo and D. Wang, "A registration-based tracking algorithm based on noise separation," Optik - International Journal for Light and Electron Optics, vol.126, no.24, pp.5806–5811, 2015.
[31] S. Zhang, H. Yao, X. Sun, and X. Lu, "Sparse coding based visual tracking: Review and experimental comparison," Pattern Recognition, vol.46, no.7, pp.1772–1788, 2013.
[32] K. Zhang, L. Zhang, and M.H. Yang, "Real-time compressive tracking," European Conference on Computer Vision, pp.864–877, 2012.

Shan Jiang received the B.S. and M.S. degrees from Changchun University of Science and Technology in 2012 and 2016, respectively. She is a Ph.D. candidate in Changchun University of Science and Technology. Her research interests included object tracking, binocular vision etc.

Cheng Han received the M.S. degree in computer science technology and received Ph.D. degree in measuring and testing technologies from Changchun University of Science and Technology in 2006, 2010, respectively. He is currently an associate professor in Changchun University of Science and Technology. His major research interests included pattern recognition, computer simulation, and virtual reality.

Xiaoqiang Di received B.S. degree in computer science and technology from Changchun University of Science and Technology in 2002, M.S. and Ph.D. degree in communication and information systems from Changchun University of Science and Technology in 2007 and 2014, respectively. He was a visiting scholar at Norwegian University of Science and Technology, Norway, from Aug. 2012 to Aug. 2013. He is currently an associate professor and supervisor of Ph.D. in Changchun University of Science and Technology. His major research interests include network information security, integrated network and visual tracking.