Tracking Completion

Yao Sui[1(✉)], Guanghui Wang[1,4], Yafei Tang[2], and Li Zhang[3]

[1] Department of EECS, University of Kansas, Lawrence, KS 66045, USA
suiyao@gmail.com, ghwang@ku.edu
[2] China Unicom Research Institute, Beijing 100032, China
tangyf24@chinaunicom.cn
[3] Department of EE, Tsinghua University, Beijing 100084, China
chinazhangli@tsinghua.edu.cn
[4] National Laboratory of Pattern Recognition,
Institute of Automation, CAS, Beijing, China

Abstract. A fundamental component of modern trackers is an online learned tracking model, which is typically modeled either globally or locally. The two kinds of models perform differently in terms of effectiveness and robustness under different challenging situations. This work exploits the advantages of both models. A subspace model, from a global perspective, is learned from previously obtained targets via rank-minimization to address the tracking, and a pixel-level local observation is leveraged simultaneously, from a local point of view, to augment the subspace model. A matrix completion method is employed to integrate the two models. Unlike previous tracking methods, which locate the target among all fully observed target candidates, the proposed approach first estimates an expected target via the matrix completion through partially observed target candidates, and then, identifies the target according to the estimation accuracy with respect to the target candidates. Specifically, the tracking is formulated as a problem of target appearance estimation. Extensive experiments on various challenging video sequences verify the effectiveness of the proposed approach and demonstrate that the proposed tracker outperforms other popular state-of-the-art trackers.

Keywords: Matrix completion · Object tracking · Subspace model · Local observation · Appearance estimation

1 Introduction

Visual tracking is an important topic in computer vision for its various applications, such as video analysis, robotics, and visual surveillance. In general, tracking models can be mainly classified into two categories: global and local. Global model exploits the overall information that varies in the entire target region. Local model treats the target as a series of small image patches to focus on the changes in each small region. It has been demonstrated that the global model is robust to some holistic appearance changes, like illumination variations and

B. Leibe et al. (Eds.): ECCV 2016, Part VIII, LNCS 9912, pp. 194–209, 2016.
DOI: 10.1007/978-3-319-46484-8_12

pose changes [1–4]. The local model, on the other hand, is intrinsically effective to the challenges, such as partial occlusions and local deformations [5–8]. This is because only some of the local patches are influenced by the distractive objects (noise contaminated regions), while the rest are considered to be noise-free. To effectively deal with various appearance changes, a robust tracker is desired to be able to exploit the advantages of both global and local tracking models.

In this work, we propose to leverage the effectiveness of the global method in capturing the overall information, and augment it with a local model to promote the accuracy and robustness of the tracker. The proposed tracking model integrates both the global and the local methods. Two efficient while effective methods, *i.e.*, subspace learning and pixel-level local observations, are designed, and a matrix completion approach [9] is employed to integrate the two models. The fundamental idea of our approach is to estimate the appearance of the target over the global subspace model and a number of local observations. As a result, the target is accurately located by means of the similarity between the estimation and the target candidates (regions of interest in the frame). Substantially different from previous tracking methods, which test each target candidate and then determine the best one as the target, the proposed approach works in a reverse way, *i.e.*, predicts the expected target and then verifies it against each target candidate. To this end, the following two issues need to be addressed.

1.1 Subspace Method

Subspace method is a classical algorithm in visual tracking [1,10–12]. Under this paradigm, the temporally obtained targets are assumed to reside in a low-dimensional subspace. For this reason, the current target can be accurately represented by the subspace learned from the previously obtained targets. It has been demonstrated that subspace method is effective to some challenges, such as pose changes and illumination variations [13,14]. However, this method is unstable in the presence of partial occlusions. The underlying assumption of subspace method, from a stochastic perspective, is that the representation errors obey the independent and identically distributed (*i.i.d.*) Gaussian with small variances. In the case of partial occlusion, however, the representation errors actually follow the *i.i.d.* Laplace or other heavy tailed distributions, because these errors may be extremely large but sparse. Consequently, a sparse (Laplace prior) additive error term is often used to compensate the instability of subspace model [15–17].

Inspired by the previous success, we exploit the subspace structure among the previously obtained targets by using a *rank*-minimization method, instead of computing orthogonal basis vectors as used in previous methods. We stack these targets into column vectors respectively and then combine these columns into a sample matrix. Since these targets are assumed to reside in a low-dimensional subspace, this sample matrix tends to be of low-rank. Thus, we minimize the rank of this sample matrix to exploit the subspace structure. Although several tracking methods also involve low-rank matrix estimation [17,18], their subspace assumptions are quite different from ours. Zhang *et al.* [17] assume that the target candidates in each frame reside in a low-dimensional subspace and construct the

subspace in the representation (transform) domain. Sui *et al.* [18] consider the obtained targets and the surrounding background regions reside in a mixture of several subspaces, and exploit these subspaces using a low-rank graph.

1.2 Integration with Local Method

Local tracking model is intrinsically robust to partial occlusions. Thus, it is reasonable to combine the subspace based tracking model with a local method to compensate the sensitivity of the subspace method in the case of occlusions. Previous local methods often transform the target region into a series of local image patches of small sizes with or without overlap. Different from those, our approach forces the local patches to shrink to the size of 1×1, leading to the *local observations*, *i.e.*, directly use a number of pixels[1]. Note that the goals of our approach and previous local methods are essentially the same: intending to sufficiently leverage the noise-free pixels (patches) and avoid the corrupted pixels (patches). In contrast to patch based methods, our approach considers the corrupted pixels as the unobserved values and intends to estimate them over the exploited target subspace. Intuitively, the pixel-level method may lose the relationship among the neighboring pixels, *i.e.*, correlations, which is well exploited in the patch-level strategy. Compared to the patches, however, the observed pixels are more flexible and much easier to be manipulated.

Matrix completion approach [9], by its nature, can be used to integrate the global target subspace model and the local observation method. The subspace model provides a prerequisite to ensure the success of the matrix completion, while the local observation method leads to a number of observed pixels to promote the accuracy of the matrix completion in the estimation of the unobserved (missing) pixels. In return, the matrix completion also implicitly maintains the subspace structure during the estimation. As demonstrated in our experiments, the estimation accuracy with respect to the target candidates, under the subspace assumption, is consistent with the similarity to the previously obtained targets, which is responsible to the target localization.

1.3 Contributions

The subspace model, from a global perspective, is learned via *rank*-minimization to address tracking, and the local observation approach, from a local point of view, is simultaneously leveraged to augment the subspace. The matrix completion is employed to integrate the two methods.

– Unlike previous methods, which emphasize on analyzing all fully observed target candidates for target localization, the proposed approach leverages each partially observed target candidate to estimate the target with the learned subspace model via the matrix completion.

[1] We only use *a number of* pixels from the target region; otherwise, it is, to some extent, equivalent to global method.

– The target is located according to the estimation accuracy of the matrix completion. It is shown that, under the subspace constraint, the estimation accuracy with respect to the target candidates is consistent with the similarity to the previously obtained targets. As a result, the proposed tracker performs much better than its counterparts.

2 Related Work on Tracking

Subspace learning is a conventional but effective method in visual tracking. Ross *et al.* [1] utilized incremental subspace learning method to represent the target and locate the target in terms of representation accuracy. Wang and Lu [11] proposed to use 2D principal component analysis method to construct a target subspace in original image domain. Sui *et al.* [12] proposed a group sparse subspace learning method to alleviate the influence of the distractive objects. Wang and Lu [19] employed a segmentation-like method to improve the robustness of subspace learning against occlusions. Zhang *et al.* [17] developed a low-rank and sparse representation to exploit the subspace structure among the candidates. Wang *et al.* [15] assumed the targets follow a Gaussian distribution (subspace prior) and the occlusions followed a Laplace distribution (sparsity prior).

There are extensive literatures on local tracking. Adam *et al.* [5] represented the target as histograms over a series local image patches. Liu *et al.* [6] developed a local sparse representation to describe the target. Jia *et al.* [7] designed an assignment pooling feature based on local sparse representation to improve the target description. Zhong *et al.* [20] utilized the local method to develop a collaborative target model. Kalal *et al.* [21] leveraged a local method to achieve a discriminative learning method for tracking. Sui and Zhang [22] constructed a locally low-rank and sparse representation to address tracking.

Many impressive tracking results are also achieved by various approaches beyond subspace and local methods. Hare *et al.* [4] employed the structured output support vector machine to address tracking. Gao *et al.* [23] analyzed the likelihood of a candidate to be the target by using Gaussian process regression. Henriques *et al.* [24] proposed a robust tracker via correlation filters from kernelized ridge regression point of view, achieving impressive performance.

3 *Rank*-Minimization and Matrix Completion

Recently, there has been a significant interest in *rank*-minimization. Some typical applications include matrix completion [9], robust principal component analysis [25], and low-rank representation [26]. *rank*-minimization focuses on the problem

$$\min_{\mathbf{X}} rank\left(\mathbf{X}\right), s.t.\ \mathbf{Y} = f\left(\mathbf{X}\right), \tag{1}$$

where $\mathbf{Y}$ denotes the observation matrix, $f\left(\mathbf{X}\right)$ is a restrict function with respect to the variable $\mathbf{X}$, and the $rank\left(\mathbf{X}\right)$ returns the rank of the matrix $\mathbf{X}$. The above minimization has been demonstrated to be a NP-hard problem. In practical

applications, the convex conjugate $\|\cdot\|_*$, named as the *trace*-norm, is often used to approximate the $rank(\cdot)$ function. The *trace*-norm is defined as a sum of the singular values of the input matrix. Note that the significance of *rank*-minimization is partially attributed to its close relation to the subspace method. Specifically, Eq. (1) is equivalent to principal component analysis if the restrict function $f(\cdot)$ is an identical function, *i.e.*,

$$\min_{\mathbf{X}} rank(\mathbf{X}), s.t. \|\mathbf{Y} - \mathbf{X}\|_F^2 \leq \varepsilon, \tag{2}$$

where $\mathbf{X}$ is the reconstructed version of $\mathbf{Y}$ over the subspace, $\varepsilon > 0$ is a very small number, and $\|\cdot\|_F$ denotes the *Frobenius*-norm. In fact, the matrix variable can be decomposed into $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ via singular value decomposition (SVD), where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices, and $\mathbf{\Sigma}$ is a diagonal matrix composed by the singular values of $\mathbf{X}$. It is clear that the columns of $\mathbf{U}$ form the basis vectors of the learned subspace, and the columns of $\mathbf{\Sigma}\mathbf{V}^T$ are the subspace representations (*i.e.*, the principal components). It is also evident that minimizing the rank of $\mathbf{X}$ is equivalent to making the diagonal elements of $\mathbf{\Sigma}$ as sparse as possible. In practice, $\mathbf{X}$ is reconstructed only from a few columns of $\mathbf{U}$, which correspond to the locations of the non-zeros of $\mathbf{\Sigma}$'s diagonal elements, so that the rank of $\mathbf{X}$ is minimized, leading to a subspace reconstruction. In this case, *rank*-minimization is directly related to the subspace method.

Matrix completion [9] is one of the most popular applications of *rank*-minimization. It can accurately recover a matrix with missing entries, even if some entries are corrupted by noise. It is mathematically formulated as

$$\min_{\mathbf{X}} \|\mathbf{X}\|_*, s.t. \mathcal{P}_{\Omega}(\mathbf{X}) = \mathcal{P}_{\Omega}(\mathbf{Y}), \tag{3}$$

where $\mathbf{X}$ is the recovered matrix, $\mathbf{Y}$ is the observation matrix, of which only the entries indexed by the set Ω can be observed, and $\mathcal{P}_{\Omega}(\mathbf{X})$ is a projection function such that $[\mathcal{P}_{\Omega}(\mathbf{X})]_{ij} = \mathbf{X}_{ij}$ for $(i,j) \in \Omega$ and zero otherwise. The goal of Eq. (3) is to estimate the missing entries (outside of Ω) in terms of the observed entries (indexed by Ω) of $\mathbf{Y}$. By minimizing Eq. (3), the missing entries can be recovered. The theoretical analysis and recovering conditions can be found in [9] and the references therein. Many algorithms have been developed to solve matrix completion, such as inexact augmented Lagrange multiplier (IALM) [27], and variational Bayesian inference [28,29].

4 The Proposed Approach

4.1 Problem Statement

We describe the target region in each frame by using a motion state variable defined as

$$\mathbf{z} = \{x, y, s\}, \tag{4}$$

where x and y denote the 2D position of the target, and s denotes the scale coefficient. According to the motion state variable, we can crop out the corresponding

region from the frame image. The cropped region is resized to a predefined value and stacked into a column vector, which is named as the *appearance observation*.

Our goal is to construct an estimator that can reliably predict an expected target appearance in each frame. Specifically, given the appearance observations $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{k-1}$ of previously obtained targets in the k-th frame, we can estimate the target appearance in the current frame as

$$\hat{\mathbf{y}}_k = \varphi\left(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{k-1}\right) \tag{5}$$

by using the estimator $\varphi(\cdot)$. To make the estimator as accurate as possible, some prior knowledge about the target appearance, is encouraged. Thus, the estimated appearance of the current target is reformulated as

$$\hat{\mathbf{y}}_k = \varphi\left(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{k-1} | \Phi\right) \tag{6}$$

by incorporating the prior information Φ. Then, we find a region, of which the corresponding appearance is most similar to $\hat{\mathbf{y}}_k$, as the target region in the current frame. Mathematically, given a set of the appearance observations $\mathcal{C}$ of all target candidates, the current target is located by

$$\mathbf{y}_k = \arg\min_{\mathbf{c} \in \mathcal{C}} \left\| \hat{\mathbf{y}}_k - \mathbf{c} \right\|. \tag{7}$$

From a global perspective, the previously obtained targets are considered to reside in a low-dimensional subspace due to their high similarity in appearances. From a local point of view, local observations (partial target information) can be obtained to help the estimator make a more accurate prediction. As a result, the problem is solved by integrating both the global and the local information. We exploit the global correlation to handle the previously obtained targets, and leverage the local information to deal with the target priors.

4.2 Estimator Design

As presented above, the estimator is built on the two kinds of information: the appearance observations of previously obtained targets and the prior knowledge of the target. The designed estimator will be discussed below.

Target summarization. In order to increase the computational efficiency, the tracking model employs a compact form, instead of using all appearance observations, to represent the previously obtained targets. Meanwhile, such a compact representation is also explored to maintain the subspace assumption. To this end, only a limited number of previously obtained targets, which can best describe the appearance changes of all the obtained targets, are employed as the estimation evidence of the estimator. We refer to the target template method [2] to implement the target summarization, which is called *target templates* hereafter.

Target priors. In the proposed model, the prior knowledge is extracted directly from a number of pixels in the target region, because such direct partial observations are the best and the strongest prior information for the target. There is, however, an obvious paradox, *i.e.*, we intend to estimate the target

appearance, while the estimator needs to partially observe the target appearance first. For this reason, we observe a number of pixels from each target candidate, and the target candidate is employed to eliminate the paradox. Under the low-dimensional subspace assumption, the target is expected to be estimated accurately by the estimator among all target candidates. The underlying reason is straightforward: since the previously obtained targets span a low-dimensional subspace, while the current target can be well represented by this subspace.

Based on the preceding analysis, the matrix completion approach is a desirable estimator for our tracking model. On one hand, matrix completion is a reliable estimator to predict unobserved entries. On the other hand, it can implicitly maintain the subspace constraint through the *rank*-minimization.

Given an appearance observation, denoted by $\mathbf{c}$, of a target candidate in each frame[2], we use a set Ω to index the observed pixels, and consider the rest as missing values. We first generate an *observed candidate* $\mathbf{c}'$ by setting the pixels of $\mathbf{c}$ outside Ω to zeros and leaving the rest unchanged. Let a matrix $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n]$ denote the n target templates, which are summarized from $\{\mathbf{y}_1, \ldots, \mathbf{y}_{k-1}\}$. We construct a new matrix $\mathbf{Y} = [\mathbf{T}, \mathbf{c}']$ and estimate the pixels outside Ω using matrix completion over $\mathbf{Y}$. For convenience, we use an equivalent form of Eq. (3) to address the matrix completion by introducing a slack variable $\mathbf{E}$.

$$\min_{\mathbf{X}} \|\mathbf{X}\|_*, s.t. \mathbf{Y} = \mathbf{X} + \mathbf{E}, \mathcal{P}_{\Omega}(\mathbf{E}) = 0. \tag{8}$$

The above minimization problem (8) can be solved by the IALM approach [27]. Let $\mathbf{X}^* = [\mathbf{T}^*, \mathbf{x}]$ denote the solution of Eq. (8), where $\mathbf{x}$ is the estimated candidate over the observed candidate $\mathbf{c}'$.

4.3 Target Localization

Within the Bayesian sequential inference framework [30,31], given all the obtained targets $\mathbf{y}_{1:k-1}$ in the k-th frame, the motion state of the k-th target, denoted by $\mathbf{z}_k$, is predicted by maximizing the posterior

$$p(\mathbf{z}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{z}_{k-1}) p(\mathbf{z}_{k-1}|\mathbf{y}_{1:k-1}) d\mathbf{z}_{k-1}, \tag{9}$$

where $p(\mathbf{z}_k|\mathbf{z}_{k-1})$ denotes the *motion model*. Then, a target candidate is generated according to its motion state $\mathbf{z}_k$. Thus, the corresponding appearance observation, denoted by $\mathbf{c}$, is obtained and the posterior is updated by

$$p(\mathbf{z}_k|\mathbf{c}, \mathbf{y}_{1:k-1}) \propto p(\mathbf{c}|\mathbf{z}_k) p(\mathbf{z}_k|\mathbf{y}_{1:k-1}), \tag{10}$$

where $p(\mathbf{c}|\mathbf{z}_k)$ denotes the *observation model*. The target on the k-th frame, denoted by $\mathbf{y}_k$, is found by

$$\mathbf{y}_k = \arg\max_{\mathbf{c} \in \mathcal{C}} p(\mathbf{z}_k|\mathbf{c}, \mathbf{y}_{1:k-1}), \tag{11}$$

[2] For the presentation simplicity, we use the term *candidate* to stand for the appearance observation of the target candidate hereafter.

Algorithm 1. Tracking Algorithm

Input: index set Ω and target templates $\mathbf{T}$.
Output: the target located in the k-th frame.
for *each candidate* $\mathbf{c} \in \mathcal{C}$ **do**
  Generate the observed candidate $\mathbf{c}'$ by setting $\mathbf{c}$'s entries outside Ω to zeros and leaving the rest unchanged.
  Construct the matrix $\mathbf{Y} = [\mathbf{T}, \mathbf{c}']$.
  Obtain the estimated candidate $\mathbf{x}$ from Eq. (8).
  Compute the observation model from Eq. (12).
end
Locate the target from Eq. (11).
Update the index set Ω and the target templates $\mathbf{T}$.

where $\mathcal{C}$ denotes the set of all the candidates that correspond to a series of regions sampled randomly in the frame according to the possibility $p(\mathbf{c}|\mathbf{z}_k)$.

The motion model in our work is defined as a Gaussian distribution $p(\mathbf{z}_k|\mathbf{z}_{k-1}) \sim \mathcal{N}(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{\Sigma})$, where the covariance $\mathbf{\Sigma}$ is a diagonal matrix, denoting the variances of 2D translation and scaling, respectively. We set $\mathbf{\Sigma} = diag\{3, 3, 0.005\}$ in our experiments. The observation model $p(\mathbf{c}|\mathbf{z}_k)$ reflects the likelihood of the candidate $\mathbf{c}$ to be the target. As discussed above, a good candidate can be estimated accurately by the matrix completion under our subspace assumption. The accuracy is measure by means of the estimation errors. Let us define the observation model for a candidate $\mathbf{c}$ with the motion state $\mathbf{z}_k$ as

$$p(\mathbf{c}|\mathbf{z}_k) \propto \exp(-\|\mathbf{c} - \mathbf{x}\|). \tag{12}$$

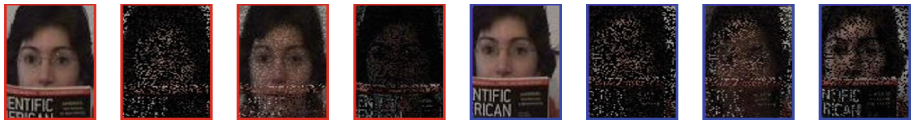
For all the candidates and their corresponding motion states, the target in the k-th frame can be located using Eq. (11). Note that under the definition of the observation model, Eq. (11) is equivalent to Eq. (7), and yields the same result in the target location. The implementation details of the tracking algorithm is outlined in Algorithm 1.

Below is a demonstration of the proposed approach. As shown in Fig. 1(a), two candidates are marked in red and blue, respectively. The representative target templates are shown in Fig. 1(b). We crop out the two target candidates from the image and resize them to the same size as the target templates, as shown in Fig. 2(a) and (e). Then, we sample a number of pixels of the two candidates at the same locations and use these pixels as the observed values, while the rest are treated as missing values, as shown in Fig. 2(b) and (f), where the missing values are set to zeros. Next, we estimate the missing values of each candidate from Eq. (8). Figure 2(c) and (g) show the two estimated candidates, respectively. Their estimation errors are shown in Fig. 2(d) and (h), respectively.

From the above results, it is evident that the good candidate (in red) is estimated much more accurately than the bad one (in blue). As shown in Fig. 2(c), the estimated good candidate is rarely influenced by the distractive object (the magazine), however, the estimated bad candidate, as shown in Fig. 2(g), is quite different from its original version shown in Fig. 2(e). Similar results can also be observed from their estimation errors. Most errors of the good candidate are small, and large errors only appear at the location of the distractive object,

(a) (b)

Fig. 1. (a) One frame image, where a good and a bad target candidate are marked in red and blue, respectively. (b) The representative target templates. (Color figure online)



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 2. (a)–(d) The good candidate, its observed pixels, estimated result, and estimation error. (e)–(h) The corresponding results as (a)–(d) with respect to the bad candidate. (Color figure online)

as shown in Fig. 2(d). In contrast, most errors of the bad candidate are large, and scatter all over the entire image, as shown in Fig. 2(h). Quantitatively, we also plot the distributions of the absolute estimation errors at the missing entries of the two candidates, as shown in Fig. 3. It can be seen that for most missing entries, the errors of the good candidates are much smaller than those of the bad ones. In addition, the residual errors of the good candidates normally converge faster than those of the bad ones because the good candidates better match the implicitly learned subspace via the *rank*-minimization. Typically, the matrix completion runs less than 30 iterations for good candidates, while about 40 iterations are required for bad candidates.

The good performance of the matrix completion in this case is attributed to two aspects: the low-dimensional subspace assumption on the previously obtained targets, and the local observations from the candidates. From a global point of view, the previously obtained targets span a low-dimensional subspace, which makes better representations of the good candidates, such that they can be estimated more accurately than the bad ones. From a local perspective, the local observations work as strong priors and promote the accuracy of the estimation.

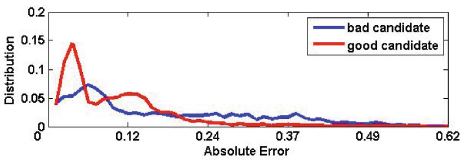


Fig. 3. Distributions of the absolute estimation errors at the missing entries of the two candidates.

Since the index set Ω is determined according to the previously obtained targets, some pixels observed from the bad candidate may be located on the distractive object, leading to a more inaccurate estimation.

4.4 Online Update

During tracking, the appearance of the target varies on successive frames. Thus, we need to update the tracker automatically to accommodate these appearance changes. In each frame, a number of pixels of the candidates are sampled so as to alleviate the influence of the distractive objects. Therefore, the set Ω is updated for every frame to exclude those unexpected pixels. Meanwhile, the target templates $\mathbf{T}$ are updated accordingly, in order to accurately reflect these appearance changes and satisfy the constraint of low-dimensional subspace.

In our work, each pixel of an obtained target is associated to a weight that reflects the possibility of this pixel to be observed in the next frame. Initially, we set all these weights equally. As analyzed in the above demonstration shown in Figs. 1, 2 and 3, the estimation errors are normally large in the regions of the distractive objects (see Fig. 2(d)). Thus, we adjust the weights in each frame to be inversely proportional to the corresponding estimation errors. To avoid that the observed pixels (they always have zero estimation errors) dominate the update, their weights are constrained during the computation. Finally, we draw the same number of entries randomly according to their weights and use these entries as the new index set Ω.

Specifically, in the k-th frame, the weight of the the j-th pixel, denoted by w_j^k, is updated by

$$w_j^k \propto \begin{cases} \frac{1}{e_j^k}, & j \notin \Omega \\ \frac{1}{e_a + e_j^{k-1}(e_b - e_a)}, & j \in \Omega, \end{cases} \tag{13}$$

where e_j^k denotes the estimation error of the k-th target in the j-th pixel, and e_a and e_b are determined by

$$e_{i_1} < e_{i_2} < \cdots < e_a < e_m < e_b < \cdots < \mathbf{e}_{i_N}, \tag{14}$$

where e_m denotes the median value of the N estimation errors, and $i_k \in \{1, 2, \ldots, N\}$. In the above equations, we divide the pixels into two categories and update their associated weights respectively. One category contains the pixels outside the index set Ω, *i.e.*, in the case of $j \notin \Omega$ for the j-th pixel of the k-th target. Among these pixels, the pixels with large estimation errors are unexpected to be observed in the next frame, since they have high possibilities to be located on the distractive objects. Thus, we directly set their associated weights inversely proportional to their estimation error e_j^k. The other category contains the pixels indexed by Ω. Because these pixels are the observed ones in the current frame, *i.e.*, they have zero estimation errors, they are expected to be observed in the next frame. In addition, in order to avoid that these pixels dominate the update, we deliberately decrease their possibilities to be observed to

(a) (b) (c)

Fig. 4. Illustration of the online update of Ω between two consecutive frames. (a) The observed pixels of the current target. (b) The possibilities of the pixels to be indexed by new Ω. The cooler pixel indicates the larger value. (c) The observed pixels of the next target, which are obtained according to the possibilities in (b).

some extent. For this reason, we constraint the possibilities of these pixels within an appropriate range, or equivalently assign them certain errors within an range $[e_a, e_b]$. In practice, the median of the target estimation errors in last frame, *i.e.*, the $(k-1)$-th target, is a reasonable reference in setting e_a and e_b, such that their values are not being set too low or too high. In our experiments, e_a and e_b are set to the errors just below and above the median error, respectively.

Figure 4 illustrates the online update strategy of Ω between two consecutive frames. It can be seen from Fig. 4(b) that the pixels from the distractive object (the magazine) have higher possibilities to be excluded (*i.e.*, not indexed in Ω) in the next frame. From Fig. 4(c), it is evident that the pixels belonging to the distractive object are reduced in the local observations of the target in the next frame. In our experiments, similar to the work [32], we use ten target templates.

5 Experimental Evaluations

Our tracker is implemented in MATLAB on a PC with an Intel Core 2.8 GHz processor. The average running speed is one frame per second. The colorful pixels in each frame are converted to gray scale and normalized to $[0, 1]$. The corresponding regions of the candidates and the target templates are normalized to the size of 20×20 pixels, and 70 % pixels are observed for the candidates.

We compared our tracker with respect to 14 popular state-of-the-art trackers, including subspace methods, local methods, and other state-of-the-art methods. Their parameters were set to the values recommended by respective authors. In order to demonstrate the effectiveness of different algorithms in various challenging situations, we collect the most popular 20 video sequences for the comparisons, which demonstrate various challenges, such as heavy occlusions, illumination variations, and background clutters, as shown in Fig. 5. In each frame, the target region is manually labeled using a bounding box as ground truth. Both the source codes and the video sequences can be publicly downloaded from the respective websites of the authors.

5.1 Qualitative Evaluations

Figure 5 shows the tracking results obtained by our tracker and the 14 competing trackers on the representative frames of the 20 video sequences.

Fig. 5. Tracking results on representative frames of the 20 video sequences.

In the case of illumination changes, *e.g.*, on the video sequences *car4*, *david* and *singer1*, our tracker achieves better results, owing to the assumption of low-dimensional subspace on the previously obtained targets, which makes matrix completion work successfully. As a demonstration of the effectiveness of the subspace assumption, a very good tracking result is obtained by the proposed tracker in the case of pose changes, *e.g.*, on the video sequences *bicycle*, *polarbear*, *surfing* and *walking*. In the presence of occlusions, *e.g.*, on the video sequences *bicycle*, *caviar3*, *faceocc*, *oneleaveshop*, *thusl* and *thusy*, our tracker also achieves better or competitive tracking performance. This is attributed to: (1) the local observations and the online update strategy leverage the pixels that are not located on the distractive objects; and (2) the matrix completion leads to a high estimation accuracy.

5.2 Quantitative Evaluations

Four criteria are used to quantitatively evaluate the performance of different trackers: tracking location error (TLE), precision, overlap rates (OR), and success rate (SR). The TLE is computed from the difference between the centers of the tracking and the ground truth bounding boxes. The precision in defined as the percentage of frames where the TLE are less than a threshold δ. The OR is computed by $\frac{A_T \cap A_G}{A_T \cup A_G}$, where A_T and A_G denote the areas of the bounding boxes of the tracking result and the ground truth, respectively. The SR is defined as the percentage of frames where the OR are greater than a threshold ρ. Table 1 shows the average TLE, precision (for $\delta = 20$), OR, and SR (for $rho = 0.5$)

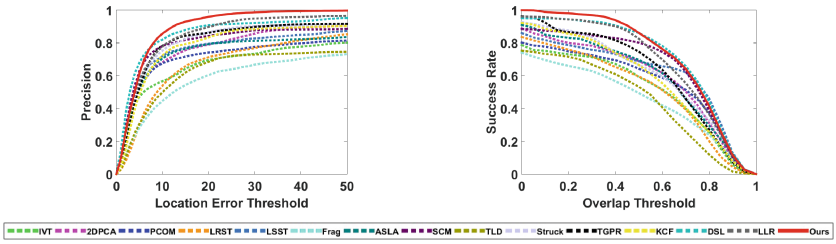


Fig. 6. Tracking performance of the proposed and the 14 competing trackers on the 20 video sequences in terms of precision (left) and success rate (right).

Table 1. Tracking performance of the proposed and the competing trackers on the 20 video sequences. The best results are shown in bold-face font.

Tracker	Ours	IVT [1]	2DPCA [11]	PCOM [19]	LRST [17]	LSST [15]	Frag [5]	ASLA [7]	SCM [20]	TLD [21]	Struck [4]	TGPR [23]	KCF [24]	DSL [33]	LLR [22]
Tracking error	5.8	46.4	23.8	48.6	25.9	28.6	34.7	30.9	29.9	30.1	20.7	23.2	22.7	15.2	9.9
Precision	0.96	0.70	0.79	0.74	0.71	0.79	0.60	0.79	0.84	0.68	0.87	0.86	0.83	0.91	0.90
Success rate	0.86	0.59	0.69	0.64	0.59	0.69	0.49	0.70	0.80	0.53	0.67	0.75	0.67	0.85	0.83
Overlap rate	0.72	0.51	0.58	0.56	0.51	0.50	0.47	0.59	0.65	0.50	0.60	0.61	0.57	0.70	0.68

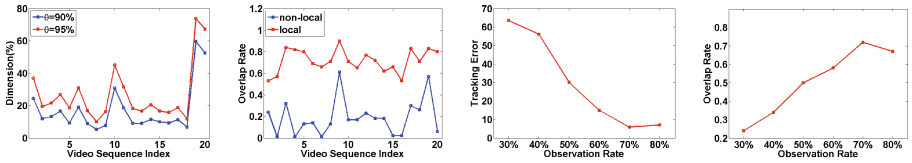
of ours and the 14 competing trackers on the 20 video sequences, respectively. Figure 6 plots the precision and the success rate of our tracker and its 14 counterparts on the 20 video sequences. From the quantitative evaluations on the 20 video sequences, it is evident that the proposed tracker outperforms its 14 counterparts in terms of all four criteria.

5.3 Analysis of the Tracking Model

The impressive performance of the proposed tracker is attributed to the integration of the global subspace assumption and the local observations. This is further demonstrated below from an experimental perspective.

First, we verify the assumption that the targets reside in a low-dimensional subspace. We collect all the targets from each of the 20 experimental video sequences according to their ground truths. Then, we analyze the corresponding target subspace by using singular value decomposition. If the target subspace is of low-dimensional, the metric of *low dimension degree*, defined as the number of the non-zero singular values of the target matrix, whose sum is more than θ times the sum of all singular values, should be small for a large θ. Figure 7(a) shows the results on the 20 video sequences with respect to $\theta = 90\,\%$ and $\theta = 95\,\%$, respectively. It can be seen that most of the low dimension degrees are located within the range of $[10\,\%, 40\,\%]$, which indicates that the targets truly reside in a low-dimensional subspace.

Next, we demonstrate the effectiveness of the integration of the local observations with the global subspace assumption. We compare the tracking results of the two trackers with and without local observations on the 20 video sequences.



(a) (b) (c) (d)

Fig. 7. Low dimension degrees (a) and overlap rates of the trackers with and without local observations (b) on the 20 experimental sequences. The order of the video sequences is identical to that showed in Fig. 5. Tracking location errors (c) and overlap rates (d) on the 20 video sequences with respect to different local observation rates.

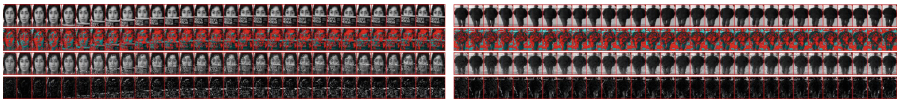


Fig. 8. Visualization of the local observation. The 1st row show the temporally obtained targets. The red pixels in the 2nd row indicate the observed pixels. The estimated targets and the residual errors are shown in the 3rd and the 4th rows, respectively. (Color figure online)

The results are shown in Fig. 7(b), from which we can see that the tracking performance is significantly improved by using the local observations. We also visualize the local observation in Fig. 8. It can be seen that, in the case of occlusion, the book tends to be observed with a small possibility, and in the case of deformation, the body of the person and the static surrounding background are encouraged to be observed, while the deformations (on shoulders and legs) are rarely observed. These experiments indicate that the local observations provide strong priors for the estimator (matrix completion), leading to more accurate estimations.

Furthermore, we show how the local observations influence the tracking results. Since the number of the observed pixels is a critical factor for the tracker, we investigate the TLEs and ORs with respect to different numbers of the observed pixels, as shown in Fig. 7(d). It is evident that our tracker yields the best performance when about 70 % pixels are observed. Observing too few pixels may lead to an inaccuracy of the matrix completion, while observing too many pixels may result in heavy influence from distractive objects. As a result, we observe 70 % pixels in our experiments for each frame.

6 Conclusion

We have formulated tracking as a problem of target appearance estimation by exploiting the advantages of both the global and local tracking models. Extensive experiments have been conducted and the results have demonstrated that: (1) our tracking model, by integrating the global and the local methods, effectively

alleviates tracking failures in various challenging situations; and (2) under the subspace assumption, the matrix completion provides an accurate estimation in target appearance for the target location. Both the qualitative and the quantitative evaluations have demonstrated that the proposed tracker outperforms most popular state-of-the-art trackers.

Acknowledgments. The work is partly supported by the National Natural Science Foundation of China (NSFC) under grants 61273282, 61573351 and 61132007, and the joint fund of Civil Aviation Research by the National Natural Science Foundation of China (NSFC) and Civil Aviation Administration under grant U1533132.

References

1. Ross, D.A., Lim, J., Lin, R.S., Yang, M.H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. (IJCV) **77**(1–3), 125–141 (2007)
2. Mei, X., Ling, H.: Robust visual tracking using L1 minimization. In: IEEE International Conference on Computer Vision (ICCV), pp. 1436–1443 (2009)
3. Babenko, B., Member, S., Yang, M.H., Member, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **33**(8), 1619–1632 (2011)
4. Hare, S., Saffari, A., Torr, P.: Struck: structured output tracking with kernels. In: IEEE International Conference on Computer Vision (ICCV), pp. 263–270 (2011)
5. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 798–805 (2006)
6. Liu, B., Huang, J., Yang, L., Kulikowsk, C.: Robust tracking using local sparse appearance model and K-selection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1313–1320, June 2011
7. Jia, X., Lu, H., Yang, M.H.: Visual tracking via adaptive structural local sparse appearance model. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1822–1829 (2012)
8. Liu, T., Wnag, G., Yang, Q.: Real-time part-based visual tracking via adaptive correlation filters. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4902–4912 (2015)
9. Candes, E., Plan, Y.: Matrix completion with noise. Proc. IEEE **98**(6), 925–936 (2010)
10. Kwon, J., Lee, K.: Visual tracking decomposition. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1269–1276 (2010)
11. Wang, D., Lu, H.: Object tracking via 2DPCA and L1-regularization. IEEE Sig. Process. Lett. **19**(11), 711–714 (2012)
12. Sui, Y., Zhang, S., Zhang, L.: Robust visual tracking via sparsity-induced subspace learning. IEEE Trans. Image Process. (TIP) **24**(12), 4686–4700 (2015)
13. Hager, G.D., Belhumeur, P.N.: Real-time tracking of image regions with changes in geometry and illumination. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 403–410 (1996)
14. Belhumeur, P.N., Kriegmant, D.J.: What is the set of images of an object under all possible lighting conditions? In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 270–277 (1996)

15. Wang, D., Lu, H., Yang, M.H.: Least soft-thresold squares tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2371–2378 (2013)
16. Wang, D., Lu, H., Yang, M.H.: Online object tracking with sparse prototypes. IEEE Trans. Image Process. (TIP) **22**(1), 314–325 (2013)
17. Zhang, T., Ghanem, B., Liu, S., Ahuja, N.: Low-rank sparse learning for robust visual tracking. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 470–484. Springer, Heidelberg (2012)
18. Sui, Y., Zhao, X., Zhang, S., Yu, X., Zhao, S., Zhang, L.: Self-expressive tracking. Pattern Recogn. (PR) **48**(9), 2872–2884 (2015)
19. Wang, D., Lu, H.: Visual tracking via probability continuous outlier model. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
20. Zhong, W., Lu, H., Yang, M.H.: Robust object tracking via sparsity-based collaborative model. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1838–1845 (2012)
21. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **34**(7), 1409–1422 (2012)
22. Sui, Y., Zhang, L.: Robust tracking via locally structured representation. Int. J. Comput. Vis. (IJCV) **119**(2), 110–144 (2016)
23. Gao, J., Ling, H., Hu, W., Xing, J.: Transfer learning based visual tracking with gaussian processes regression. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part III. LNCS, vol. 8691, pp. 188–203. Springer, Heidelberg (2014)
24. Henriques, J., Caseiro, R., Martins, P., Batista, J.: High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **37**(3), 583–596 (2015)
25. Candes, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? J. ACM **58**(3), 1–37 (2011)
26. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: International Conference on Machine Learning (ICML) (2010)
27. Lin, Z., Chen, M., Ma, Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, pp. 1–23. UIUC Technical report (2010)
28. Babacan, S., Luessi, M.: Low-rank matrix completion by variational sparse Bayesian learning. In: International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (2011)
29. Babacan, S.D., Luessi, M., Molina, R., Katsaggelos, A.: Sparse Bayesian methods for low-rank matrix estimation. IEEE Trans. Sig. Process. (TSP) **60**, 3964–3977 (2011)
30. Isard, M.: CONDENSATION - conditional density propagation for visual tracking. Int. J. Comput. Vis. (IJCV) **29**(1), 5–28 (1998)
31. Arulampalam, M., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans. Sig. Process. (TSP) **50**(2), 174–188 (2002)
32. Mei, X., Ling, H.: Robust visual tracking and vehicle classification via sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **33**(11), 2259–2272 (2011)
33. Sui, Y., Tang, Y., Zhang, L.: Discriminative low-rank tracking. In: IEEE International Conference on Computer Vision (ICCV), pp. 3002–3010 (2015)