Int J Comput Vis (2016) 118:337–363
DOI 10.1007/s11263-015-0879-9

Similarity Fusion for Visual Tracking

Yu Zhou[1,3] · **Xiang Bai**[1] · **Wenyu Liu**[1] · **Longin Jan Latecki**[2]

Received: 5 June 2014 / Accepted: 30 December 2015 / Published online: 25 January 2016

Abstract Multiple features' integration and context structure of unlabeled data have proven their effectiveness in enhancing similarity measures in many applications of computer vision. However, in similarity based object tracking, integration of multiple features has been rarely studied. In contrast to conventional tracking approaches that utilize pairwise similarity for template matching, our approach contributes in two different aspects. First, multiple features are integrated into a unified similarity to enhance the discriminative ability of similarity measurements. Second, the neighborhood context of the samples in forthcoming frame are employed to further improve the measurements. We utilize a diffusion process on a tensor product graph to achieve these goals. The obtained approach is validated on numerous challenging video sequences, and the experimental results demonstrate that it outperforms state-of-the-art t racking methods.

Keywords Visual tracking · Similarity measure · Fusion

Communicated by Yasushi Yagi.

✉ Xiang Bai
xiang.bai@gmail.com; xbai@hust.edu.cn

Yu Zhou
yuzhou@hust.edu.cn

Wenyu Liu
liuwy@hust.edu.cn

Longin Jan Latecki
latecki@temple.edu

[1] Huazhong University of Science and Technology, Wuhan, People's Republic of China

[2] Temple University, Philadelphia, PA, USA

[3] Beijing University of Posts and Telecommunications, Beijing, People's Republic of China

1 Introduction

Tracking an arbitrary target in long-term video stream is a challenging task in computer vision. It has a wide range of practical applications, e.g., surveillance, robot vision, and so on. The main challenges in designing a robust tracking algorithm are caused by shape deformation, pose variation, illumination changes, cluttered background, occlusion, etc. (Yilmaz et al. 2006). To overcome these challenges, tracking by matching approaches have attracted much attention in recent years (Fan et al. 2006; Yang et al. 2007; Wu et al. 2009). Roughly speaking, there are two essential tasks that should be addressed: *target modeling* and *visual matching*.

To explicitly capture complex appearance changes and reduce the influence of noises, abundant *target modeling* algorithms have been introduced, e.g., holistic target model (Comaniciu et al. 2003), compact subspace-based target models (Ross et al. 2008; Hu et al. 2010; Li et al. 2013), over-completed dictionary based sparse representations (Mei and Ling 2011; Mei et al. 2011), etc. These approaches achieve reasonable performance in limited cases, and may fail in more challenging real world settings, e.g., the abrupt motion or the presence of background noises. The reason is that these methods only focus on seeking a robust target model in a given *single* feature space, e.g., RGB or gray-value. They ignore the significant factor that single feature space is frequently insufficient for modeling target in more challenging conditions. In particular, different image features reflect different levels of robustness to certain changes in the object appearance, e.g., HOG features capture the shape information of the target (Dalal and Triggs 2005), which is insensitive to slight illumination variation, while LBP feature obtains texture information of the target (Ojala et al. 2002), which has been shown to be more effective in handling occlusion.

Hence, different features should be utilized for robust tracking.

Visual matching, i.e., matching visual appearance of the target in consecutive frames, also plays a critical role in tracking. An appropriate distance/similarity measure strategy is the key component for visual matching. Most existing tracking methods utilize pre-specified distance metrics, e.g., Euclidean distance, Bhattacharyya coefficient, etc. However, as observed in Wang et al. (2010), Jiang et al. (2011), fixed distance measures often cannot obtain desirable matching results. Hence, they introduce metric learning based tracking approaches. The appropriate metric can be learned from a set of training samples. The common issue of these metrics is that they work on single feature space. However, if the foreground target cannot be separated from the cluttered background in a given single feature space, robust matching results cannot be obtained even with online learning of the adaptive metrics, a single image feature is not always discriminative enough. Furthermore, existing algorithms for learning metrics are computationally expensive, and the collection of training samples is empirical. Hence, training samples that differ from test sequences may significantly reduce the quality of the learned metrics.

In contrast to conventional tracking approaches, our approach utilizes the relationship among the candidates in the next frame to enhance the similarity measurement in tracking. From the perspective of semi-supervised learning (Zhou et al. 2004; Sinha and Belkin 2009; Zhu 2006), there exists a hidden manifold structure in unlabeled data, e.g., the low dimensional geometry (Belkin and Niyogi 2004), or the neighborhood context (Bai et al. 2010a, b), which can be employed to improve the classification accuracy.

In summary, the proposed tracking approach has the following key properties:

1. It utilizes several visual cues, since a single image cue is often insufficient for visual tracking. Target models based on a single image feature cannot reflect the complex appearance variations in challenging environments, and metrics based on single image feature often cannot guarantee that the target can be separated from the background.
2. It utilizes the relationship among the samples in the forthcoming frame to enhance the similarity measure.

Our philosophy lies in designing a computationally efficient *fused similarity measure* that integrates both the context structure in the forthcoming frame and the complementary characteristics of multiple image features. The basic idea is illustrated in Fig. 1. Based on the tracked targets up to time $t-1$, the incremental target mean (Ross et al. 2008) is learned and treated as labeled positive sample. The target candidates sampled at time t are regarded as unlabeled samples (shown as blue rectangles in Fig. 1a). Then both the labeled positive sample and the unlabeled samples are mapped to nodes of a weighted consecutive frame single graph (CFSG). Its edges determine the pairwise similarities between pairs of samples (saffron lines in Fig. 1a). As aforementioned, single cue is insufficient in most cases, hence we further design a consecutive frame Tensor Product Graph (CFTPG) to integrate pairs of CFSGs, each with different similarity measure between nodes. As shown in Fig. 1b, both G_α and G_β are CFSGs. Hence CFTPG can reflect rich manifold structure information hidden not only in the original feature spaces but also in the space composed of integrated features.

In order to obtain more reliable joint similarity measure, a diffusion process is proposed to exploit the manifold structure hidden in the CFTPG. Specifically, the neighborhood structure of pairs of samples is utilized as context information to enhance the original pairwise similarities, where the neighborhood structure is determined by a set of nearest neighbors of a sample in CFTPG. Since CFTPG incorporates similarity measures based on two different features, the information within a feature space and between the pairs of feature spaces is integrated by the diffusion process. Furthermore, since CFTPG is constructed on consecutive frames, the diffusion process exploits the context structure of unlabeled samples in the current frame, which is very different from existing methods. The obtained new diffused similarity, is then interpreted as the label probability over the unlabeled samples, which indicates the confidence of each sample being the real target.

However, while pairs of CFSGs have n nodes, their CFTPG has n^2 nodes, which significantly increases the time complexity of the diffusion on CFTPG. To address this issue, we introduce an iterative algorithm that operates on the original pairs of CFSGs and prove that it is equivalent to the diffusion process on CFTPG. We call this process Fusion with Diffusion (FD). FD is a generalization of the approach in Yang et al. (2013), where only a single similarity measure is considered. Furthermore, since CFTPG only considers pairs of CFSGs, to incorporate multiple cues, a novel unsupervised weight estimation method is proposed for each learned similarity by FD, which reflects the measurement ability of the corresponding similarity. Consequently, the final similarity is the weighted sum of multiple learned similarities.

In summary, the main contributions of this paper include:

- A novel tracking by matching approach, whose key insight lies in designing a fused similarity measurement, it is simple but computationally efficient.
- The fused similarity exploits the complementary character among different image features and fuses them into a single similarity measure. In addition, the context structure of the candidates in the forthcoming frame is utilized

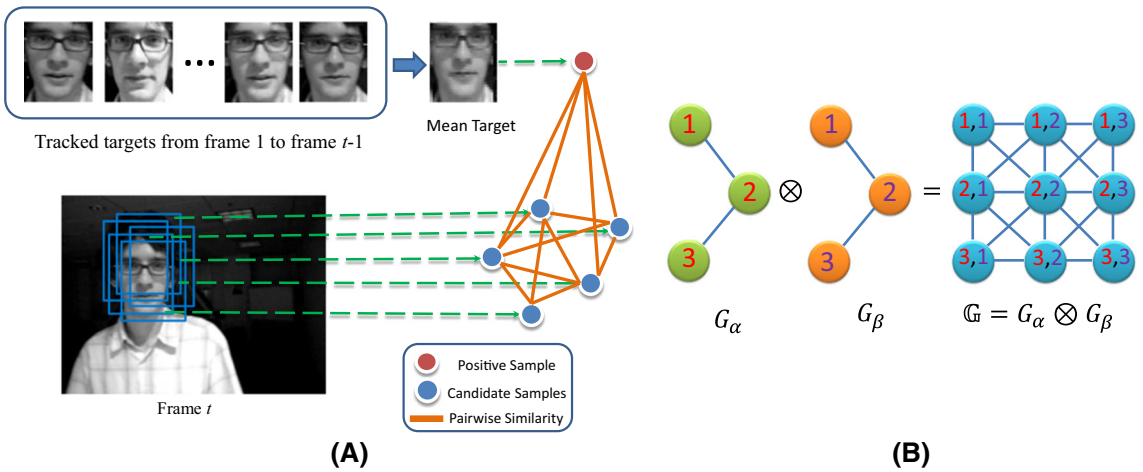

Fig. 1 **a** Construction of the consecutive frame single graph (CFSG); **b** construction of the consecutive frame tensor product graph (CFTPG)

by the diffusion on CFTPG to further enhance the measurement.

– The proposed framework is general in that any kind of image features can be utilized as input to our framework, and the output similarity can be integrated into any existing tracking by matching approach to further enhance the tracking performance.

The rest of the paper is organized as follows: Sect. 2 gives the description of the related work, Sect. 3 presents the problem formulation, Sect. 4 describes the appearance model and motion model that are utilized in our approach. Section 5 introduces the FD process, Sect. 6 describes the unsupervised weight estimation algorithm, Sect. 7 demonstrates the quantitative comparison results, and Sect. 9 concludes the whole paper.

2 Related Work

Visual tracking has a long research history, a detailed survey is presented in Yilmaz et al. (2006). In this section, we elaborate only on target modeling, similarity measures, and multiple cues fusion strategies which are most relevant to our method.

2.1 Target Modeling

Target modeling roughly includes discriminative modeling and generative modeling (Li et al. 2013). Discriminative modeling formulates the task as foreground and background binary classification (Avidan 2004, 2007; Grabner et al. 2006, 2008; Babenko et al. 2011; Kalal et al. 2012). Taking the famous MIL tracker as an example (Babenko et al. 2011), the objective is to learn a classifier to best separate the foreground and the background. Because only simple Haar-like feature is utilized in MIL tracker, it is very different from our approach. Generally, most of these discriminative approaches suffer from the issue that they cannot reflect complex variations, since binary classifiers cannot handle large state variations, and hence are limited to simple variations. e.g., only translation is considered in those approaches.

Generative modeling is closely relevant to our approach, and various generative object models have been introduced, e.g., holistic target models (Comaniciu et al. 2003; Collins et al. 2005), compact subspace models (Ross et al. 2008; Hu et al. 2010; Li et al. 2013), sparse representation models (Mei and Ling 2011; Liu et al. 2012; Zhang et al. 2013; Wang et al. 2013; Jia et al. 2012; Zhong et al. 2012), covariance-based target models (Porikli et al. 2006; Wu et al. 2012), part based models (Adam et al. 2006; Čehovin et al. 2013; Kwon and Lee 2013; Fan et al. 2010), and segmentation based approaches (Fan et al. 2012; Wang et al. 2011). Yang and Hua (2009) presents a context aware tracker, the auxiliary objects are learned by a data mining approach, and the random field is employed to measure the set-to-set similarity between the appearance model and the candidates. Only single target representation is considered in this approach. However, as aforementioned, these approaches only focus on seeking the robust target model in the specific single feature space, which is insufficient in more challenging environments. In contrast, our approach aims to fuse multiple features in a simple but effective way. Our approach can be easily integrated to those methods to further improve the tracking performance.

2.2 Distance Metric

The distance/similarity metrics are frequently utilized in tracking include Euclidean distance, reconstruct error (Ross et al. 2008; Li et al. 2013), Bhattacharyya coefficient

(Comaniciu et al. 2003), differential EMD (Zhao and Yang 2010), and cross-bin metric (Leichter 2012). There metrics are all pre-specified and fixed, and the mismatch removal is frequently employed to enhance the similarity (Ma et al. 2014, 2015). In addition, learning based adaptive metrics have attracted much attention recently, e.g., Wang et al. (2010), Jiang et al. (2011), Jiang et al. (2012), Tsagkatakis and Savakis (2011). Our method is relevant to learning based metrics. However, different from existing methods utilizing supervised learning, our approach can be interpreted as semi-supervised learning, since we utilize the context structure of the candidates in the forthcoming frame. Moreover, multiple cues can be integrated to a uniform similarity measure naturally in our approach. As we observed, those two characters play a key role to enhance the similarity measure.

2.3 Multiple Cues Fusion

Multiple cues fusion seems to be an effective way to improve the tracking performance. Kwon and Lee (2010, 2011) implement multiple cues integration by sampling the state space; Santner et al. (2010) formulates tracking task as the combination of different trackers, three different trackers are combined into a cascade; Li et al. (2013) incorporates Dempster-Shafer information fusion into the tracking approach. Hong et al. (2013) proposes a Multi-Task Multi-view sparse representation for tracking. The goal of this approach is to learn a fused target representation. Each view in each particle is treated as an individual task, and the underlying relationship between tasks is considered jointly. Yoon et al. (2012) presents an adaptive tracker selection approach. Multiple trackers with different features are employed using two step configurations, i.e., tracker selection and interaction, is employed to fuse these independent trackers. Badrinarayanan et al. (2007) presents probabilistic multi-cue tracking approaches, where the state of the object is estimated from distributions associated to the cues at each tracking step. Stenger et al. (2009) proposes an approach to integrate multiple observation models. Francesc Moreno-Noguer and Samaras (2008) presents a probabilistic framework to integrate multiple features based on the cue dependencies. Wang et al. (2011) introduces a pedestrian tracking approach based a new manifold subspace. The proposed subspace can well capture the variations of continuous pedestrian postures. Wang and Yagi (2008) presents a tracking approach based on the integration of color and shape-texture features. The candidate feature set includes seven color features, and a gradient orientation histogram feature. These features are ranked according to the discriminative ability by comparing with the background, and the top two discriminative features are utilized for tracking. Spengler and Schiele (2003) introduces a multi-cue integration framework for tracking. Aim to address dynamic environments, the tracking results are evaluated and fed back to make adaptation to the integration mechanism and visual cues.

In contrast to these approaches, our approach fuses multiple features in a novel way. A fused similarity measure is obtained by the presented diffusion process. In addition, the above methods simply integrate different representations without fully exploring the contextual information from the unlabeled frames, as we do.

Our approach is a generalization of the process introduced in Yang et al. (2013), which is applied to image retrieval and image segmentation. Our approach takes multiple features integration into consideration, while Yang et al. (2013) considers only a single feature. As we shown in the experimental results section, the complementary characteristics of different representations can improve the performance efficiently.

In a preliminary conference version of this work (Zhou et al. 2012), we exploit the validity of our approach when combined with simple low-level image features, i.e., HOG (Dalal and Triggs 2005), LBP (Ojala et al. 2002) and Haar-like feature (Babenko et al. 2011). In this paper, starting with these features, we first construct higher level features using compact subspace modeling, which takes the statistical information of the previously observed target into consideration, and hence it leads to more robust results. In addition, we introduce a new weight estimation strategy of the higher level features, which is derived from the reconstruction error of the compact subspace representation. The fact that our improved appearance model, based on higher level features, leads to improved performance demonstrates the flexibility of the proposed construction of the joint similarity measure.

3 Problem Formulation

The problem of matching based visual tracking boils down to the following simple formulation: given the observed real target set $\mathbb{I} = \{\hat{\mathcal{I}}_1, \ldots, \hat{\mathcal{I}}_{t-2}, \hat{\mathcal{I}}_{t-1}\}$ up to frame I_{t-1}, with each image patch $\hat{\mathcal{I}}_i$ containing the target in frame I_i, a target appearance model M_{t-1} is learned based on $\mathbb{I}$. There are also given a set of candidate target patches $\mathcal{C}_t = \{\mathcal{I}_t^n | \, n = 2, \ldots, N\}$ in frame I_t, and a similarity measure S defined on the set of $V = \{M_{t-1}\} \cup \mathcal{C}_t$, i.e., S is a function from $V \times V$ into positive real numbers. Then our tracking goal can be formally stated as

$$\hat{\mathcal{I}}_t = \arg \max_{\mathcal{I}_t^n \in \mathcal{C}_t} S(M_{t-1}, \mathcal{I}_t^n) \tag{1}$$

meaning that the patch in $\mathcal{C}_t$ with most similar appearance to target model M_{t-1} is selected as the real target in frame I_t, and $\mathcal{C}_t$ is determined by the motion model, which is defined in Sect. 4.

In conventional methods, patches in V are represented by single image feature, e.g., gray value of the image. However, due to the challenging issues discussed in Sect. 1, single pre-specified image feature is often insufficient to identify the target in the next frame. Therefore, we consider multiple image features. In our approach, each sample in V is represented by Q types of different features, i.e, gray-level feature, HOG feature, and LBP feature are utilized as the low-level image representation (hence $Q = 3$). For each of these features a higher level appearance feature is derived by online subspace learning of the target appearance. Based on the higher level image features, pairs of samples are compared using multiple similarities, i.e., $\mathcal{S} = \{S_1, \ldots S_Q\}$, each S_α defined on $V \times V$ for $\alpha = 1, \ldots, Q$.

Each node of the graph represents the sample in the subspace representation, and the reconstruction error similarity is employed to define the edge weight of the graph, i.e., the similarity of two samples. Therefore, we interpret each similarity measure S_α as the affinity matrix of a graph G_α, i.e., S_α is a $N \times N$ matrix with positive entries.

However, in order to solve Eq. (1), we need a single similarity measure S. Hence we face a question how to combine the measures in $\mathcal{S}$ into a single similarity measure. We propose a two stage approach to answer this question. First, we combine pairs of similarity measures S_α and S_β into a single measure $\mathrm{P}^*_{\alpha,\beta}$, which is a matrix of size $N \times N$, $\mathrm{P}^*_{\alpha,\beta}$ is defined in Sect. 5 and it is obtained with the proposed process called **Fusion with Diffusion**.

In the second stage we combine all $\mathrm{P}^*_{\alpha,\beta}$ for $\alpha, \beta \in \{1, \ldots, Q\}$ into a single similarity measure S defined as a weighted matrix sum

$$S = \sum \omega_{\alpha,\beta} \mathrm{P}^*_{\alpha,\beta} \tag{2}$$

where $\omega_{\alpha,\beta}$ are positive weights associated with measures $\mathrm{P}^*_{\alpha,\beta}$ defined in Sect. 6.

We also observe that in contrast to many tracking by matching approaches, the combined measure S is not only a function of similarities between M_{t-1} and the candidate patches in $\mathcal{C}_t$, but also of similarities of patches in $\mathcal{C}_t$ to each other.

4 Target Modeling and Motion Model

4.1 Target Modeling

In Zhou et al. (2012), simple low-level image features are utilized to express the observed target, which is insufficient in more challenging settings. In this paper, multiple subspace models are employed to obtain more robust target model. Based on Ross et al. (2008), besides IPCA model learned on gray-level image feature, the IPCA models are also learned on HOG feature and LBP feature, respectively. Figure 2 illustrates the three IPCA models. Compared with directly utilizing the HOG or LBP features, the learned HOG or LBP IPCA models are more robust, since the learned orthogonal basis not only captures the crucial information of the image, but also brings in statistical information from several previous frames. Like the original image features, different subspaces also reflect different level of robustness to certain changes in an object's appearance. Hence, they are complementary to each other, which is suitable for our fused similarity measure.

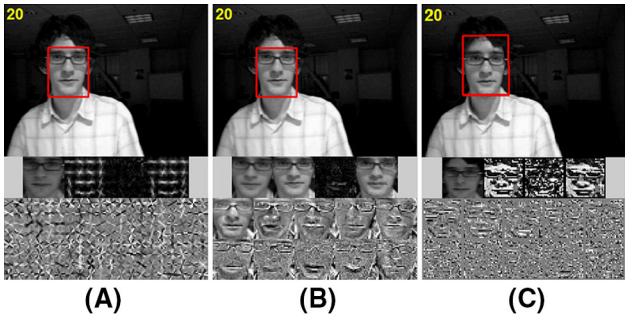

Fig. 2 Subspace model learned in our method. **a** HOG subspace model; **b** Gray-level feature subspace model; **c** LBP subspace model

Specifically, for each individual image feature type α, the target model $M_{(t-1,\alpha)}$ is determined by incremental sample mean, i.e.:

$$M_{(t-1,\alpha)} = \mu_{(t-1,\alpha)} = \frac{a}{a+b}\mu_{(a,\alpha)} + \frac{b}{a+b}\mu_{(b,\alpha)} \tag{3}$$

where $\mu_{(t-1,\alpha)}$ is sample mean of the target up to frame I_{t-1} based on feature type α, $\mu_{(a,\alpha)}$ is the sample mean up to frame I_a, $a = t - b - 1$, and $\mu_{(b,\alpha)} = \frac{1}{b}\sum_{i=a+1}^{a+b} \mathcal{I}_i, \mathcal{I}_i \in \mathbb{I}$, b is batch number used for update the sample mean. Based on Ross et al. (2008), a set of orthogonal eigenvectors $\{U^i_{(t-1,\alpha)} | i = 1, .., b\}$ and eigenvalues $\{\lambda^i_{(t-1,\alpha)} | i = 1, \ldots, b\}$ can be obtained from the target set we have observed in previous frames, each eigenvalue $\lambda^i_{(t-1,\alpha)}$ corresponding to the variance along the $U^i_{(t-1,\alpha)}$'s direction. Consequently, the projection coefficients of each $\mathcal{I}_i \in V$ in the b-dimensional subspace are

$$A_{(i,\alpha)} = U^T_\alpha (\mathcal{I}_i - \mu_{(t-1,\alpha)}) \tag{4}$$

where $A_{(i,\alpha)} = [a^1_i, \ldots, a^b_i]$ denotes the coefficient vector of $\mathcal{I}_i$. Hence, the reconstruction error of $\mathcal{I}_i \in V$ with respect to $(\mu; U)$ is

$$\begin{aligned} \mathbf{c}_{(i,\alpha)} &= (\mathcal{I}_i - \mu_{(t-1,\alpha)}) - U_\alpha A_{(i,\alpha)} \\ &= (\mathcal{I}_i - \mu_{(t-1,\alpha)}) - U_\alpha U^T_\alpha (\mathcal{I}_i - \mu_{(t-1,\alpha)}) \end{aligned} \tag{5}$$

We update the sample mean and eigenvectors incrementally following (Ross et al. 2008). Based on the new representation $\mathbf{c}_{(i,\alpha)}$ for each $\mathcal{I}_i \in V$, the similarity between $\mathcal{I}_i \in V$ and $\mathcal{I}_j \in V$ is determined by the reconstruction error between $\mathbf{c}_i$ and $\mathbf{c}_j$:

$$S_\alpha(\mathcal{I}_i, \mathcal{I}_j) = \exp\left(\frac{-d(\mathbf{c}_{(i,\alpha)}, \mathbf{c}_{(j,\alpha)})}{\gamma^2}\right) \tag{6}$$

where γ is the bandwidth parameter that controls how quickly the weight decreases, d is the Euclidean distance. In practice, the performance of tracking task is sensitive to the selection of γ. Following Zelnik-Manor and Perona (2004), local scaling is utilized to learn an adaptive γ, to be specific, we replace γ^2 with $\gamma_i\gamma_j$, where $\gamma_i = d(\mathcal{I}_i, \mathcal{I}_\mathrm{k})$, $\mathcal{I}_\mathrm{k}$ is the k'th neighbor of $\mathcal{I}_i$, k = 3 is utilized in all experiments. Equation (6) has the intuitive meaning in that two image patches have large similarity if their reconstruction error is small.

4.2 Motion Model

The candidate set $\mathcal{C}_t$ is determined by the motion model, which reflects potential target variations. To address more complex appearance variations, e.g., deformation, out-plane rotation and scale variation, affine image warping is utilized to model the object motion between consecutive frames in this paper, which includes six parameters: $(x, y, \theta, s, \alpha, \phi)$, where x, y indicate the 2-D center position of the image patch, θ denotes the rotation angle, s is the scale of image patch, α, ϕ is aspect ratio and skew direction, respectively. All the parameters are modeled independently by a Gaussian distribution around the previously observed target $\hat{\mathcal{I}}_{t-1}$ in frame I_{t-1}. Specifically, the motion model can be stated as:

$$p(\mathcal{I}_t^n | \hat{\mathcal{I}}_{t-1}) = \mathcal{N}(\mathcal{I}_t^n; \hat{\mathcal{I}}_{t-1}, \psi) \tag{7}$$

where $\mathcal{N}$ denotes the Gaussian function, ψ is a diagonal covariance matrix whose elements are the corresponding variances of affine parameters, i.e., $\sigma_x^2, \sigma_y^2, \sigma_\theta^2, \sigma_s^2, \sigma_\alpha^2, \sigma_\phi^2$. The candidates in $\mathcal{C}_t$ are sampled based on Eq. (7).

5 Fusion with Diffusion

5.1 Single Graph on Consecutive Frames

Since we have multiple image features, for each feature $\alpha \in \{1, \ldots, Q\}$, the similarity S_α can be obtained by Eq. (6). Given the target model M_{t-1} and candidate set $\mathcal{C}_t$, an edge-weighted graph $G_\alpha = (\mathcal{V}, S_\alpha)$ is built on the set $V = \{M_{t-1}\} \cup \mathcal{C}_t$, the vertices $\mathcal{V}$ represent the image patches in the set V, S_α represents the edge weight determined based on feature α, where each entry of S_α reflects the reconstruction similarity between pairs of image patches as introduced in Eq. (6).

Given a single graph $G_\alpha = (\mathcal{V}, S_\alpha)$ on consecutive frames, a reversible Markov chain can be constructed with the transition probability defined as

$$P_\alpha(i, j) = S_\alpha(i, j)/D_i \tag{8}$$

where $D_i = \sum_{j=1}^N S_\alpha(i, j)$ is the degree of each vertex. Then the transition probability $P_\alpha(i, j)$ inherits the positivity-preserving property $\sum_{j=1}^N P_\alpha(i, j) = 1,\ i = 1, \ldots, N$.

The graph G_α is a fully connected graph in many applications. To reduce the influence of noisy points, i.e., cluttered background patches in tracking, a local transition probability is used:

$$(P_{k,\alpha})(i, j) = \begin{cases} P_\alpha(i, j) & j \in k\mathrm{NN}(i) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Hence the number of non-zero elements in each row is not larger than k, which implies $\sum_{j=1}^n (P_{k,\alpha})(i, j) < 1$. This inequality is important in our framework, since it guarantees the convergence of the diffusion process on the tensor product graph presented in the next section.

5.2 Tensor Product Graph of Two Similarities

Given are two single graphs $G_\alpha = (\mathcal{V}, P_{k,\alpha})$ and $G_\beta = (\mathcal{V}, P_{k,\beta})$ defined in Sect. 5.1, we can define their **Tensor Product Graph** (**TPG**) as

$$G_\alpha \otimes G_\beta = (\mathcal{V} \times \mathcal{V}, \mathbb{P}), \tag{10}$$

where $\mathbb{P} = P_{k,\alpha} \otimes P_{k,\beta}$ is the Kronecker product of matrices defined as $\mathbb{P}(a, b, i, j) = P_{k,\alpha}(a, b)\ P_{k,\beta}(i, j)$. Thus, each entry of $\mathbb{P}$ relates four image patches. When $P_{k,\alpha}$ and $P_{k,\beta}$ are two $N \times N$ matrices, then $\mathbb{P}$ is a $N^2 \times N^2$ matrix. However, as we will see in the next subsection, we actually never compute $\mathbb{P}$ explicitly.

5.3 Diffusion Process on Tensor Product Graph

We utilize a diffusion process on TPG to combine the two similarity measures $P_{k,\alpha}$ and $P_{k,\beta}$. We begin with some notations. The vec operator creates a column vector from a matrix M by stacking the column vectors of M below one another. More formally $vec : \mathbb{R}^{N\times N} \to \mathbb{R}^{N^2}$ is defined as $vec(\mathrm{M})_g = (\mathrm{M})_{ij}$, where $i = \lfloor (g-1)/N \rfloor + 1$ and $j = g \bmod N$. The inverse operator vec^{-1} that maps a vector into a matrix is often called the reshape operator. We define a diagonal $N \times N$ matrix as

$$\Delta(i,i) = \begin{cases} 1 & i = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

Only the entry representing the target model M_{t-1} is set to one and all other entries are set to zero in Δ.

We observe that $\mathbb{P}$ is the adjacency matrix of TPG $G_\alpha \otimes G_\beta$. We define a q-th iteration of the diffusion process on this graph as

$$\mathbb{P}^q = \sum_{e=0}^{q} (\mathbb{P})^e vec(\Delta). \tag{12}$$

As shown in Yang and Latecki (2011), this iterative process is guaranteed to converge to a nontrivial solution given by

$$\lim_{q\to\infty} \sum_{e=0}^{q} (\mathbb{P})^e vec(\Delta) = (\mathrm{I} - \mathbb{P})^{-1} vec(\Delta), \tag{13}$$

where I is a identity matrix. Following Yang and Latecki (2011), we define

$$\mathrm{P}^*_{\alpha,\beta} = \mathrm{P}^* = vec^{-1}((\mathrm{I} - \mathbb{P})^{-1} vec(\Delta)) \tag{14}$$

We observe that our solution P^* is a $N \times N$ matrix. We call the diffusion process to compute P^* a Fusion with Diffusion (FD) process, since diffusion on TPG $G_\alpha \otimes G_\beta$ is used to fuse two similarity measures S_α and S_β.

Since $\mathbb{P}$ is a $N^2 \times N^2$ matrix, FD process on TPG as defined in Eq. (12) may be computationally too demanding. To compute P^* effectively, instead of diffusing on TPG directly, we show in Sect. 5.4 that FD process on TPG is equivalent to an iterative process on $N \times N$ matrices only. Consequently, instead of an $O(n^6)$ time complexity, we obtain an $O(n^3)$ complexity. Then in Sect. 5.5 we further reduce it to an efficient algorithm with time complexity $O(n^2)$, which can be used in real time tracking algorithms.

5.4 Iterative Algorithm for Computing P*

We define $\mathrm{P}^1 = P_{(k,\alpha)} P^T_{(k,\beta)}$ and

$$\mathrm{P}^{q+1} = P_{k,\alpha} (P_{k,\alpha})^q (P^T_{k,\beta})^q P^T_{k,\beta} + \Delta. \tag{15}$$

We iterate Eq. (15) until convergence, and as we prove in Proposition 1, we obtain

$$\mathrm{P}^* = \lim_{q\to\infty} \mathrm{P}^q = vec^{-1}((\mathrm{I} - \mathbb{P})^{-1} vec(\Delta)) \tag{16}$$

The iterative process in Eq. (15) is a generalization of the process introduced in Yang and Latecki (2011), where TPG of the a single graph with itself is only considered.

Algorithm 1: Iterative Fusion with Diffusion Process

Input: Two matrices $P_{k,\alpha}$, $P_{k,\beta} \in \mathbb{R}^{N\times N}$
Output: Diffusion result $\mathrm{P}^* \in \mathbb{R}^{N\times N}$

```
Compute P* = Δ.
Compute u_α = first column of P_{k,α}, u_β = first column of P_{k,β}
Compute P* ← P* + u_α u_β^T.
for i = 2, 3, ... do
    Compute u_α ← P_{k,α} u_α
    Compute u_β ← P_{k,β} u_β
    Compute P* ← P* + u_α u_β^T
end
```

Proposition 1

$$\begin{aligned} vec\left(\lim_{q\to\infty} P^{(q+1)}\right) &= \lim_{t\to\infty} \sum_{e=0}^{q-1} \mathbb{P}^e vec(\Delta) \\ &= (I - \mathbb{P})^{-1} vec(\Delta). \end{aligned} \tag{17}$$

Proof please see the appendix.

5.5 FD Algorithm

To effectively compute P^*, we propose an iterative algorithm that takes the advantage of the structure of matrix Δ. Let $\mathbf{u}_\alpha$ be a $N \times 1$ vector containing the first column of $P_{k,\alpha}$. We write $P_{k,\alpha} = [\mathbf{u}_\alpha | R]$ and $P_{k,\alpha}\Delta = [\mathbf{u}_\alpha | 0]$. It follows then that $P_{k,\alpha}\ \Delta\ P^T_{k,\beta} = \mathbf{u}_\alpha \mathbf{u}^T_\beta$. Furthermore, if we denote $(P_{k,\alpha})^j\ \Delta\ (P^T_{k,\beta})^j = \mathbf{u}_{\alpha,j}\mathbf{u}^T_{\beta,j}$, with $\mathbf{u}_{\alpha,j}$ being $N \times 1$, and $\mathbf{u}^T_{\beta,j}$ being $1 \times N$, it follows that

$$\begin{aligned} P^{j+1}_{k,\alpha}\ \Delta\ (P^T_{k,\beta})^{j+1} &= P_{k,\alpha}(P^j_{k,\alpha}\ \Delta\ (P^T_{k,\beta})^j) P^T_{k,\beta} \\ &= P_{k,\alpha}\mathbf{u}_{\alpha,j}\mathbf{u}^T_{\beta,j} P^T_{k,\beta} \\ &= (P_{k,\alpha}\mathbf{u}_{\alpha,j})(P_{k,\beta}\mathbf{u}_{\beta,j})^T \\ &= \mathbf{u}_{\alpha,j+1}\mathbf{u}^T_{\beta,j+1}. \end{aligned}$$

Hence, we replaced one of the two $N \times N$ matrix products with one matrix product between an $N \times N$ matrix and $N \times 1$ vector, and the other with a product of an $N \times 1$ by an $1 \times N$ vector. This reduces the complexity of our algorithm from $O(n^3)$ to $O(n^2)$. The final algorithm is shown in Algorithm 1.

6 Online Weight Estimation

The weight $\omega_{i,\alpha,\beta}$ of measure $S_{\alpha,\beta}$ is inversely proportional to the reconstruction error of the high level features α, β, where i is the frame number. We utilize $\mathbf{c}_{(i,\alpha)}$ defined in Eq. (5) in Sect. 4, which is a $1 \times M$ dimensional vector, to define the weight as

Table 1 Average running time (FPS)

	Frag	OAB	IVT	PLS	LGT	CT	MIL
FPS	5	20	13	8	6	25	18
Code	C	C	M-C	M-C	M-C	M-C	M-C
	TLD	APG	MTT	IMT	VTS	VTD	**Our**
FPS	23	1	0.2	1	3	3	5
Code	M-C	M-C	M-C	M-C	M-C	C	M-C

Symbol 'C' indicates that the tracker is implemented in C/C++, 'M-C' indicates that the tracker is implemented in Matlab with some MAX files

$$\hat{c}_{(i,\alpha)} = \exp\left(-\frac{\sum_M \mathbf{c}_{(i,\alpha)}}{\varrho}\right), \tag{18}$$

where $\sum_M \mathbf{c}_{(i,\alpha)}$ is the sum of all dimensions of the reconstruction error vector. A smaller sum $\sum_M \mathbf{c}_{(i,\alpha)}$ indicates that the corresponding image representation is more reliable.

The weight of each fused similarity $S_{\alpha,\beta}$ for $\alpha \in \{1,\ldots,Q\}, \beta \in \{1,\ldots,Q\}$, is then defined as

$$\omega_{i,\alpha,\beta} = \frac{\hat{c}_{i,\alpha} + \hat{c}_{i,\beta}}{\sum_{\alpha=1}^{Q}\sum_{\beta=1}^{Q}(\hat{c}_{i,\alpha} + \hat{c}_{i,\beta})}. \tag{19}$$

The larger is the value of $\hat{c}_{i,\alpha} + \hat{c}_{i,\beta}$, the larger is the weight $\omega_{\alpha,\beta}$. A large weight has the intuitive meaning that the image representations of higher level features α, β have large discriminative ability.

The weights are computed for every frame i in order to accommodate appearance changes of the tracked object.

7 Experimental Results

In this section, we extensively validate our tracking algorithm on challenging video datasets, and compare our method with state-of-the-art trackers to demonstrate the effectiveness and efficiency of our method. HOG (Dalal and Triggs 2005), LBP (Ojala et al. 2002) and Gray-Level features are employed to represent the image patches. We learn the appearance representation for each of the three features, respectively. Our tracker is implemented in MATLAB with some mex files and runs at 5 frames per second (FPS) on a 2.53 GHz Intel Core i3 CPU with 4 GB memory. In addition, the comparison on average running time has been reported in Table 1. For the experimental parameters, the affine parameters are set as $(x, y, \theta, s, \alpha, \phi) = (6, 6, 0.01, 0.001, 0.001, 0.001)$, which are fixed in all the experiments. The bin size of HOG is set to 4, and the iteration number in Algorithm 1 is set to 200. The setting of the nearest neighbor k and the candidate number are discussed in Sect. 7.3.

7.1 Datasets

We evaluate the tracking performance on 30 challenging videos from Babenko et al. (2011) and Wu et al. (2013): *Car4*, *CarDark*, *CarScale*, *Coke*, *CouponBook*, *David*, *David2*, *David3*, *Deer*, *Dog1*, *Doll*, *Dudek*, *FaceOcc2*, *Fish*, *FleetFace*, *Football*, *Football1*, *Freeman1*, *Freeman3*, *Girl*, *Jumping*, *Mhyang*, *Surfer*, *Suv*, *Sylvester*, *Tiger1*, *Tiger2*, *Trellis*, *Walking*, *Walking 2*. These videos cover common challenging situations in visual object tracking including large appearance in-plane or out-plane variations, deformation, fast motion, occlusion, moving camera, large scale changes, and complex background.

7.2 Baselines

We compare our method to 13 recent state-of-the-art tracking algorithms including Multiple Instance Learning tracker (MIL) (Babenko et al. 2011), Fragment tracker (Frag) (Adam et al. 2006), IVT tracker (Ross et al. 2008), Online Adaboost tracker (OAB) (Grabner et al. 2006), Multi-Task Tracker (MTT) (Zhang et al. 2013), Compressive Tracker (CT) (Zhang et al. 2012), Local-Global Tracker (LGT) (Čehovin et al. 2013), Tracking-Learning-Detection tracker (TLD) (Kalal et al. 2012), Partial Least Squares tracker (PLS) (Wang et al. 2012), APG tracker (Bao et al. 2012) VTS (Kwon and Lee 2010), VTD (Kwon and Lee 2011), IMT (Yoon et al. 2012). The results of these trackers are obtained by carefully running the released source codes provided by the authors with the same parameter settings reported in their papers.

In addition, to clearly show the performance gain of our similarity fusing algorithm, we report the experimental results by using each feature individually in our algorithm without diffusion step, namely HOG, LBP, IVT (Gray) in short. We also test the performance with simple weighted linear combination of multiple features (denoted as Linear). For HOG and LBP, we use the HOG subspace and LBP subspace introduced in Sect. 4 for tracking, respectively. For Linear, we obtain the original affinity matrix by Eq. (6) and then simply calculate the average affinity, which is then used in the diffusion process in Yang et al. (2013)

7.3 Parameter Discussion

There are two parameters that influence the performance of our approach, i.e., the neighborhood number and the candidate number. As shown in Tables 2 and 3, the performance of our approach is sensitive to the neighborhood number. Desired results can be obtained by setting an appropriate neighborhood number. The internal reason is intuitive, since fewer neighbors may be insufficient to enhance the similarity measure. However, excessive number of neighbors may induce undesirable noise into the similarity measure, which

Table 2 Average Precision Score in 30 videos with varying the neighborhood number (candidate number = 600)

Neighborhood number	2	3	4	5	6
Accuracy Precision Score	0.73	0.81	0.86	0.82	0.79

Table 3 Average Overlap Rate in 30 videos with varying the neighborhood number (candidate number = 600)

Neighborhood number	2	3	4	5	6
Accuracy Overlap Rate	0.68	0.70	0.72	0.69	0.65

Table 4 Average Precision Score in 30 videos with varying the candidate number (neighborhood number = 4)

Candidate number	400	500	600	700	800
Accuracy Precision Score	0.81	0.84	0.86	0.87	0.89

Table 5 Average Overlap Rate in 30 videos with varying the candidate number (neighborhood number = 4)

Candidate number	400	500	600	700	800
Accuracy Overlap Rate	0.68	0.70	0.72	0.72	0.73

may reduce the performance. In Tables 4 and 5, the average PS and average OR are given. Similar with several existing generative approaches, e.g., Mei and Ling (2011), Mei et al. (2011), Zhang et al. (2013), etc., improving the candidate number may slightly increase the accuracy of the tracker, which is the common situation in existing approaches. However, as discussed in Zhang et al. (2013), increasing the candidate number can greatly influence the speed of the tracker, hence the trade-off between the accuracy and speed is considered in this paper. In the following, the neighborhood number is set to 4 and the candidate number is set to 600 in all the experiments.

7.4 Evaluation Methodology

To impartially and comprehensively compare our algorithm with the state-of-the-art trackers, we use three kinds of quantitative measures: firstly, the *Average Center Location Error* (ACLE) is utilized, which is frequently employed to measure the performance in the literature, e.g., Babenko et al. (2011), Wang et al. (2012), The results are shown in Table 6. Moreover, both *Precision Score* (PS) (Babenko et al. 2011), and *Success Score* (SS, indicating the average bounding box overlap ratio) are utilized in this paper. The results are shown in Tables 7 and 8, respectively.

In addition, three kinds of curve evaluation methodologies are also used: *Center Location Error* (CLE), *Precision Plots* (PP) and *Success Plot* (SP).[1] The results are shown from Figs. 3, 4, 5, 6, 7 and 8.

7.5 Quantitative Comparison

7.5.1 Comparison to the Baseline Methods

In our approach, the subspace of HOG, LBP and Gray are employed to express the target, respectively. We treat the trackers using single features as the baseline approaches, IVT can be regarded as the special case of our approach which utilizes the Gray feature without any diffusion. As shown in Tables 6, 7 and 8, our tracker can always outperform these baselines, which clearly demonstrates that different image features reflect different levels of robustness to various changes of the target, and in practice, no one feature can obtain satisfactory tracking results under all conditions. However, our diffusion process can adaptively fuse such features under different conditions, which results in consistently stable performance on various kinds of videos. In addition, we also compare with the linear combination (Linear). For this baseline, the average similarity is used and the diffusion process in Yang and Latecki (2011) is used to improve the similarity measure. As can be observed, linear combination performs worse than our method, since it treats all the features equally and can not fully explore the complementary property among different features. With linear combination, once the target is lost by one feature, it will influence the whole tracking process. In contrast, our approach can effectively excavate the complementary characteristics among these features by adaptively updating the weight for each feature frame by frame.

7.5.2 Comparison to Fusion Based Methods

IMT, VTS and VTD are all fusion based approaches, and their results are shown in Tables 6, 7 and 8. Our method almost always performs better than VTS and VTD, and slightly better than IMT. This is reasonable, since our method not only excavates the integration characteristics among several features, but also utilizes the context information of the unlabeled data in the forthcoming frame to enhance the similarity measure. As shown in Table 1, the speed of the proposed algorithm is 5 FPS, which is much faster than IMT (1 FPS).

7.5.3 Comparison to Matching Based Methods

As the proposed method is based on template matching, we also compare it to several popular matching based tracking algorithms including Frag, IVT, PLS, APG and MTT to

[1] More details about the meaning of Precision Plot and Success Plot can be found in Wu et al. (2013).

Table 6 *Average Center Location Error* (ACLE measured in pixels)

Video	Frag	OAB	IVT	PLS	LGT	CT	MIL	TLD	APG	MTT	IMT	VTS	VTD	HOG	LBP	Linear	**Our**
Car4	94.3	56.7	19.4	304	133	81.2	54.2	13.5	133	31.9	**3.0**	36.6	36.8	9.6	43.1	107	*2.4*
CarDark	38.3	23.9	2.5	38.3	27.9	120	44.7	15.2	14.9	1.7	*1.0*	2.75	16.1	32.9	27.7	41.8	**1.6**
CarScale	23.0	32.3	11.2	25.0	65.7	25.9	39.8	133	14.6	24.8	15.0	35.1	38.0	**6.6**	28.7	16.8	*6.4*
Coke	69.1	25.1	37.3	10.4	33.2	16.4	18.0	14.3	59.8	10.3	**7.9**	31.9	15.8	47.2	9.4	75.1	*5.9*
Coup. Book	56.4	17.8	32.3	5.4	6.2	19.6	5.9	68.2	67.7	*3.1*	6.2	19.8	20.5	8.5	29.3	6.3	**4.8**
David	70.0	51.3	8.1	64.5	13.8	13.3	19.7	8.0	58.9	7.6	51.3	15.6	15.6	**5.9**	13.0	73.3	*3.4*
David2	6.7	4.8	10.1	71.5	7.1	79.6	25.1	55.5	4.7	2.3	**1.7**	3.1	3.3	1.9	2.0	2.5	*1.6*
David3	13.4	236	**8.1**	103	72.1	88.9	231	118	81.9	346	73.0	54.6	66.7	30.0	35.2	49.3	*7.1*
Deer	86.2	216	204	153	209	235	227	41.3	237	11.1	*4.6*	219	136	8.2	18.0	81.1	**7.5**
Dog1	21.6	10.4	6.7	7.7	7.3	10.2	18.5	**3.9**	4.5	255	4.4	11.5	11.2	4.1	6.0	5.3	*3.5*
Doll	13.6	84.4	23.4	75.5	13.4	22.5	77.9	146	83.7	87.6	*2.4*	7.6	**7.3**	12.0	19.3	23.1	10.2
Dudek	99.3	130	9.8	107	19.7	19.9	147	16.3	45.0	10.1	10.1	*9.7*	10.4	20.0	29.3	42.3	**9.7**
Faceocc2	15.1	**12.3**	19.6	34.6	32.8	66.7	17.4	16.4	12.7	15.3	42.1	13.9	19.2	36.2	18.3	38.8	*5.6*
Fish	43.3	27.4	5.3	41.8	12.9	25.7	37.2	20.3	37.9	**3.6**	3.6	7.3	16.9	4.2	14.8	10.4	*3.1*
FleetFace	67.5	144	73.8	26.4	44.4	54.3	143	36.1	43.0	70.7	**25.9**	42.2	45.9	28.1	32.1	48.6	*22.1*
Football	**1.0**	221	1.2	1.2	5.32	4.5	219	2.6	1.6	1.0	*0.9*	2.7	3.0	1.2	3.5	2.1	1.1
Football1	13.4	35.1	24.6	16.1	*3.0*	10.4	7.5	97.0	25.8	13.5	**6.8**	10.4	7.0	39.3	24.1	59.7	11.8
Freeman1	30.6	10.7	10.7	94.5	63.6	10.2	110	60.3	8.8	**8.8**	11.4	10.1	10.3	9.1	14.4	8.9	*8.6*
Freeman3	7.2	50.3	5.1	16.6	5.40	61.6	83.3	52.4	**4.7**	17.6	58.4	17.8	22.9	5.4	23.9	13.8	*3.1*
Girl	20.9	10.6	20.0	83.9	13.6	18.4	14.7	8.7	*4.4*	7.9	10.0	12.6	**8.6**	38.1	25.3	48.0	15.9
Jumping	17.7	7.5	77.1	42.5	42.6	46.2	7.3	**4.8**	34.9	63.8	4.8	41.8	42.5	5.8	29.8	87.9	*4.7*
Mhyang	12.5	14.2	2.0	5.4	13.6	24.1	19.6	10.1	3.5	3.7	**1.8**	3.9	4.4	3.3	16.2	9.8	*1.6*
Surfer	127	13.2	18.4	21.8	8.3	152	**4.4**	4.9	12.2	10.4	4.4	12.1	5.4	14.3	19.0	27.3	*4.3*
Suv	41.7	73.7	79.5	202	84.4	69.9	74.5	24.2	88.4	99.7	*3.5*	55.4	57.3	7.7	23.0	36.7	**5.1**
Sylvester	8.7	21.7	83.6	13.0	9.6	11.7	7.3	7.1	80.8	6.3	**5.6**	10.3	9.2	5.4	18.1	46.8	*3.4*
Tiger1	39.8	38.9	50.2	20.1	40.1	21.4	*8.4*	22.5	46.6	26.6	10.5	12.9	14.4	19.3	34.1	68.2	**10.3**
Tiger2	38.7	13.3	98.5	87.9	10.2	15.6	**7.5**	17.9	38.8	25.2	90.6	40.6	11.6	48.5	19.0	23.1	*5.9*
Trellis	60.2	53.4	18.2	62.9	11.3	50.8	62.1	59.3	37.5	50.5	*2.9*	24.3	32.2	24.2	37.0	19.3	**3.8**
Walking	31.0	7.6	2.8	2.8	15.9	5.4	4.3	115	2.8	3.0	**2.2**	5.4	5.8	5.0	6.2	4.3	*2.2*
Walking2	63.1	28.8	6.2	45.7	42.9	59.6	59.2	20.9	3.6	2.9	**2.9**	53.3	45.8	45.0	39.8	99.2	*2.3*

Italicized value indicates best performance, Boldface value indicates second best

Table 7 *Precision Score* (precision at the fixed threshold of 10)

Video	Frag	OAB	IVT	PLS	LGT	CT	MIL	TLD	APG	MTT	IMT	VTS	VTD	HOG	LBP	Linear	**Our**
Car4	0.15	0.32	0.26	0.09	0.21	0.34	0.35	0.38	0.08	0.27	*1.00*	0.32	0.33	**0.45**	0.27	0.01	*1.00*
CarDark	0.10	0.57	*1.00*	0.48	0.65	0.05	0.19	0.61	**0.72**	*1.00*	*1.00*	0.69	*1.00*	0.49	0.51	0.21	*1.00*
CarScale	0.62	0.41	0.73	0.66	0.62	0.65	0.53	0.20	0.71	0.67	0.67	0.46	0.44	**0.79**	0.53	0.65	*0.82*
Coke	0.05	0.14	0.12	0.42	0.08	0.29	0.26	0.48	0.06	0.79	**0.85**	0.05	0.27	0.09	0.47	0.04	*0.89*
Coup. Book	0.23	0.13	0.15	0.92	0.77	0.13	**0.96**	0.37	0.35	*1.00*	**0.96**	0.13	0.14	0.82	0.23	0.94	*1.00*
David	0.18	0.06	0.79	0.29	0.41	0.30	0.09	0.83	0.24	0.76	**0.90**	0.03	0.38	0.53	0.82	0.73	*1.00*
David2	0.86	*1.00*	0.39	0.11	0.84	0.03	0.37	0.54	**0.99**	*1.00*	*1.00*	*1.00*	*1.00*	0.53	0.81	*1.00*	*1.00*
David3	0.52	0.19	**0.72**	0.33	0.11	0.21	0.19	0.09	0.33	0.09	0.44	0.62	0.46	0.71	0.69	0.65	*0.75*
Deer	0.01	0.05	0.02	0.02	0.05	0.01	0.08	0.73	0.02	0.46	*0.98*	0.02	0.02	0.71	0.39	0.02	**0.76**
Dog1	0.64	0.55	0.95	0.74	0.95	0.71	0.29	0.96	0.94	0.54	**0.94**	0.68	0.69	0.74	0.81	0.89	*0.97*
Doll	0.74	0.39	0.75	0.31	0.38	0.11	0.18	0.05	0.11	0.38	*0.97*	0.78	0.79	0.87	0.72	0.76	**0.92**
Dudek	0.37	0.06	*0.72*	0.17	0.24	0.21	0.05	0.28	0.49	0.59	**0.66**	0.63	0.59	0.63	0.52	0.51	0.64
Faceocc2	0.39	0.48	0.03	0.25	0.14	0.01	0.19	0.49	0.58	**0.62**	0.49	0.44	0.17	0.39	0.69	0.41	*0.91*
Fish	0.41	0.03	**0.92**	0.24	0.27	0.10	0.06	0.73	0.03	0.04	*1.00*	0.76	0.48	0.29	0.42	0.61	*1.00*
FleetFace	0.14	0.01	0.18	0.28	0.03	0.15	0.13	0.18	*0.50*	0.29	0.36	**0.37**	0.35	0.27	0.28	0.21	0.33
Football	*1.00*	0.19	*1.00*	**0.98**	*1.00*	*1.00*	0.13	0.97	*1.00*	*1.00*	*1.00*	*1.00*	*1.00*	*1.00*	0.93	*1.00*	*1.00*
Football1	0.33	0.54	0.68	0.35	*0.98*	0.35	0.76	0.21	0.16	0.58	0.62	0.43	**0.78**	0.41	0.53	0.29	0.71
Freeman1	0.56	0.38	0.50	0.32	0.30	0.51	0.23	0.30	0.53	0.27	*0.61*	0.56	0.54	0.55	0.42	0.53	**0.60**
Freeman3	0.68	0.62	0.90	0.85	0.81	0.01	0.02	0.61	0.80	0.65	**0.98**	0.49	0.50	0.81	0.52	0.74	*1.00*
Girl	0.57	0.46	0.38	0.01	0.45	0.17	0.38	0.71	**0.88**	*0.91*	0.83	0.53	0.65	0.37	0.49	0.26	0.62
Jumping	0.83	0.82	0.11	0.08	0.05	0.03	0.80	*0.97*	0.21	0.06	**0.91**	0.09	0.06	0.85	0.26	0.03	*0.97*
Mhyang	0.58	0.19	**0.99**	0.84	0.54	0.06	0.24	0.78	0.90	0.96	0.98	0.94	0.91	0.93	0.42	0.69	*1.00*
Surfer	0.16	0.31	0.23	0.32	0.64	0.05	0.94	**0.96**	0.55	0.57	1.00	0.69	0.90	0.82	0.61	0.53	*1.00*
Suv	0.68	0.08	0.10	0.03	0.19	0.08	0.07	0.63	0.53	0.51	*0.95*	0.50	0.91	0.72	0.63	0.91	**0.93**
Sylvester	0.70	0.29	0.04	0.67	0.61	0.58	0.74	0.88	0.35	0.87	**0.95**	0.52	0.59	0.93	0.74	0.37	*1.00*
Tiger1	0.16	0.07	0.01	0.46	0.23	0.62	**0.74**	0.39	0.13	0.37	*0.75*	0.36	0.52	0.54	0.65	0.01	0.63
Tiger2	0.07	0.39	0.08	0.21	0.69	0.54	**0.84**	0.34	0.06	0.24	0.20	0.32	0.48	0.21	0.38	0.26	*0.89*
Trellis	0.01	0.18	0.69	0.04	0.48	0.03	0.11	0.01	0.32	0.22	**0.96**	0.34	0.35	0.23	0.30	0.21	*0.97*
Walking	0.49	0.73	**0.99**	0.97	0.42	0.97	**0.99**	0.29	**0.99**	0.06	**0.99**	0.98	0.95	0.83	0.81	0.98	*1.00*
Walking2	0.01	0.54	0.86	0.39	0.19	0.24	0.41	0.26	0.96	**0.99**	1.00	0.39	0.36	0.35	0.20	0.01	*1.00*

Italicized value indicates best performance, Boldface value indicates second best

further demonstrate its effectiveness. As shown in Table 6, our method clearly outperforms the other methods. Especially, APG and MTT use sparse representation, in which the over-complete dictionary is employed for reconstruction. The sparse coefficient learning process based on the over-complete dictionary is a bit time consuming. In contrast, our appearance model only learns a compact subspace for each feature representation, which is more efficient for tracking tasks.

PS results are shown in Table 7. Our method also achieves competitive performance on the testing sequences. Although we set the threshold to 10, which is very challenging for all the trackers,[2] we still obtain several 1.00 scores. In the testing sequences *CarDark*, *David2* and *Coup. Book*, MTT achieves comparable results with our method, but it works badly on other videos, which demonstrates that our fusion framework is more stable to different kinds of challenges that often appear in real videos.

The stability of our tracker can also be observed from the plots of location error as the function of frame number in Figs. 3 and 4. Our curves are relatively stable, which clearly illustrates that the targets are not easily lost during the whole tracking process. This phenomenon can be also demonstrated by the fact that our Precision is better than the other methods under different thresholds in Figs. 5 and 6.

[2] Babenko et al. (2011) sets the threshold to 20 and Zhou et al. (2012) sets the threshold to 15.

Table 8 *Average Overlap Rate*

Video	Frag	OAB	IVT	PLS	LGT	CT	MIL	TLD	APG	MTT	IMT	VTS	VTD	HOG	LBP	Linear	**Our**
Car4	0.18	0.28	0.45	0.02	0.20	0.23	0.25	0.56	0.15	0.37	**0.79**	0.37	0.36	0.69	0.23	0.11	*0.87*
CarDark	0.30	0.43	0.80	0.81	0.39	0.01	0.18	0.43	0.50	0.84	*0.87*	0.75	0.54	0.32	0.39	0.26	**0.85**
CarScale	0.42	0.43	0.45	0.58	0.43	0.42	0.40	0.18	**0.71**	0.52	0.66	0.43	0.42	0.69	0.41	0.35	*0.75*
Coke	0.06	0.18	0.12	0.48	0.13	0.37	0.33	0.40	0.06	**0.49**	0.53	0.12	0.38	0.17	0.55	0.02	*0.62*
Coup. Book	0.36	0.61	0.40	0.83	0.72	0.62	0.82	0.30	0.33	*0.89*	0.84	0.59	0.60	0.75	0.49	0.81	**0.86**
David	0.31	0.26	0.72	0.36	0.64	0.67	0.58	0.58	0.21	**0.74**	0.65	0.54	0.64	0.62	0.55	0.28	*0.80*
David2	0.24	0.67	0.53	0.08	0.59	0.01	0.37	0.42	0.29	0.63	**0.82**	0.67	0.68	0.71	0.69	0.72	*0.88*
David3	0.49	0.16	0.67	0.28	0.23	0.26	0.17	0.14	0.29	0.09	0.41	**0.54**	0.41	0.31	0.29	0.23	*0.68*
Deer	0.17	0.12	0.03	0.02	0.08	0.03	0.12	0.62	0.04	0.65	*0.78*	0.04	0.05	0.67	0.51	0.15	**0.74**
Dog1	0.54	0.47	0.55	0.47	0.45	0.51	0.44	0.50	0.67	0.32	**0.72**	0.60	0.59	0.70	0.52	0.51	*0.77*
Doll	0.53	0.27	0.32	0.31	0.52	0.42	0.25	0.04	0.14	0.27	*0.85*	**0.64**	**0.64**	0.34	0.44	0.52	0.55
Dudek	0.53	0.34	0.73	0.57	0.70	0.69	0.75	0.53	0.53	0.75	0.80	**0.81**	0.79	0.75	0.64	0.53	*0.91*
Faceocc2	0.72	**0.74**	0.54	0.47	0.49	0.67	0.68	0.47	0.38	0.70	0.50	0.72	0.65	0.27	0.49	0.22	*0.76*
Fish	0.54	0.38	0.80	0.28	0.59	0.42	0.30	0.52	0.24	0.33	*0.83*	0.69	0.55	0.68	0.43	0.55	**0.82**
FleetFace	0.46	0.25	0.44	0.58	0.46	0.55	0.24	0.46	0.55	0.51	0.61	*0.63*	**0.62**	0.55	0.51	0.31	*0.63*
Football	*0.84*	0.15	**0.83**	0.80	0.71	0.75	0.10	0.81	0.78	0.82	0.80	0.79	0.79	0.79	0.61	0.78	*0.84*
Football1	0.36	0.51	0.46	0.33	*0.74*	0.44	0.57	0.17	0.25	0.51	**0.58**	0.43	0.56	0.34	0.41	0.23	0.56
Freeman1	0.37	0.33	0.37	0.30	0.21	0.35	0.18	0.20	0.32	0.20	**0.42**	0.35	0.34	0.41	0.31	0.39	*0.49*
Freeman3	0.33	0.25	0.36	0.49	0.42	0.01	0.02	0.38	0.66	0.33	**0.72**	0.31	0.32	0.65	0.29	0.52	*0.74*
Girl	0.45	0.46	0.36	0.01	0.44	0.27	0.39	0.44	**0.49**	*0.63*	0.48	0.48	0.55	0.19	0.31	0.17	0.39
Jumping	0.67	0.58	0.11	0.09	0.06	0.05	0.59	*0.71*	0.15	0.07	**0.68**	0.14	0.12	0.65	0.42	0.03	*0.71*
Mhyang	0.65	0.60	0.79	0.66	0.58	0.43	0.53	0.65	*0.82*	*0.82*	**0.81**	0.75	0.73	0.37	0.65	0.53	*0.82*
Surfer	0.16	0.39	0.31	0.32	0.50	0.01	*0.74*	**0.73**	0.39	0.51	0.59	0.56	0.69	0.28	0.36	0.21	**0.73**
Suv	0.62	0.27	0.25	0.03	0.31	0.26	0.20	0.57	0.48	0.46	*0.84*	0.48	0.45	0.72	0.57	0.32	**0.82**
Sylvester	0.67	0.44	0.07	0.62	0.63	0.61	0.71	0.68	0.28	0.65	*0.75*	0.62	0.65	0.67	0.46	0.13	**0.73**
Tiger1	0.21	0.16	0.09	0.43	0.22	0.5	0.63	0.38	0.12	0.38	*0.58*	0.49	**0.55**	0.14	0.34	0.21	*0.58*
Tiger2	0.14	0.45	0.02	0.17	0.51	0.49	**0.62**	0.30	0.06	0.28	0.15	0.26	0.50	0.59	0.47	0.16	*0.68*
Trellis	0.27	0.29	0.53	0.19	0.59	0.28	0.27	0.10	0.32	0.29	*0.82*	0.47	0.43	0.39	0.25	0.12	**0.65**
Walking	0.53	0.51	0.53	0.26	0.40	0.54	0.53	0.23	0.69	*0.77*	**0.75**	0.61	0.60	0.51	0.52	0.57	0.64
Walking2	0.27	0.36	0.49	0.33	0.25	0.27	0.28	0.40	0.76	0.78	**0.79**	0.33	0.32	0.0.35	0.31	0.22	*0.82*

Italicized value indicates best performance, Boldface value indicates second best

7.5.4 Comparison to Classification Based Methods

Classification based tracking algorithms such as MIL, OAB and CT are another mainstream in this domain. For OAB, on-line Adaboost is used to train the classifier for the foreground and background classification. MIL combines multiple instance learning with on-line Adaboost . Haar-like features are adopted in such methods. These methods have limitations in handling the complex affine parameter spaces. They often can estimate the translation of target well, but may not accurately localize the target with serious rotation/scale changes. In contrast, our tracker does not suffer from such limitations, and consequently works better than the other methods in Tables 6, 7 and 8.

7.5.5 Comparison to Other Methods

Besides the above methods, we compare our method to TLD and LGT. TLD formulates the tracking task as a tracking-learning-detection process, while LGT uses an adaptive coupled-layer visual model for tracking. Our method performs always better than TLD and LGT as shown in Tables 6, 7, and 8.

7.6 Qualitative Comparison

7.6.1 Qualitative Comparison on Car Sequences

In Fig. 9, frame results for three moving car videos are shown: (A) *Car4*, (B) *CarDark*, (C) *CarScale*. *Car4* is challenging

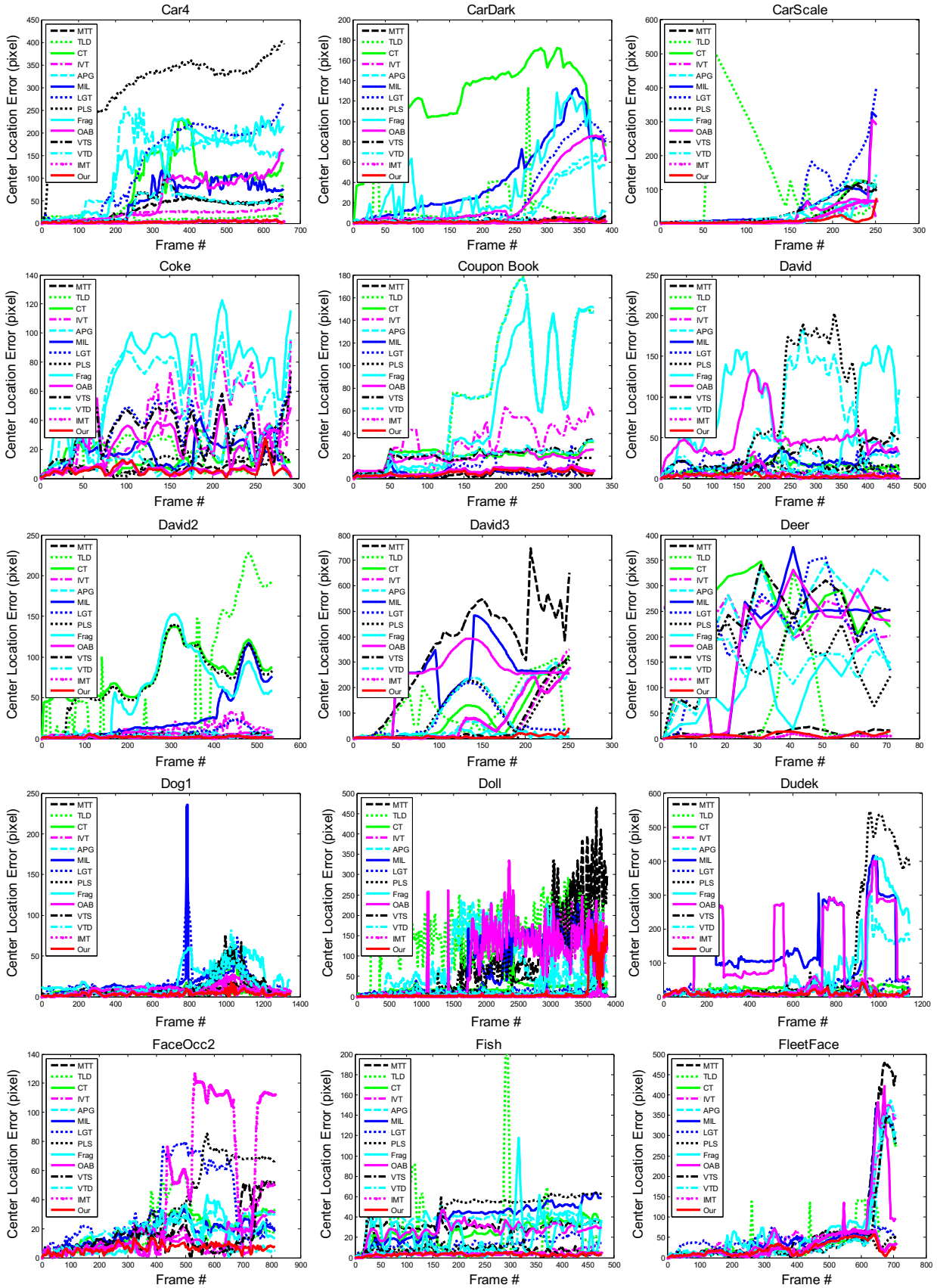

Fig. 3 *Center Location Error* (CLE) versus frame number

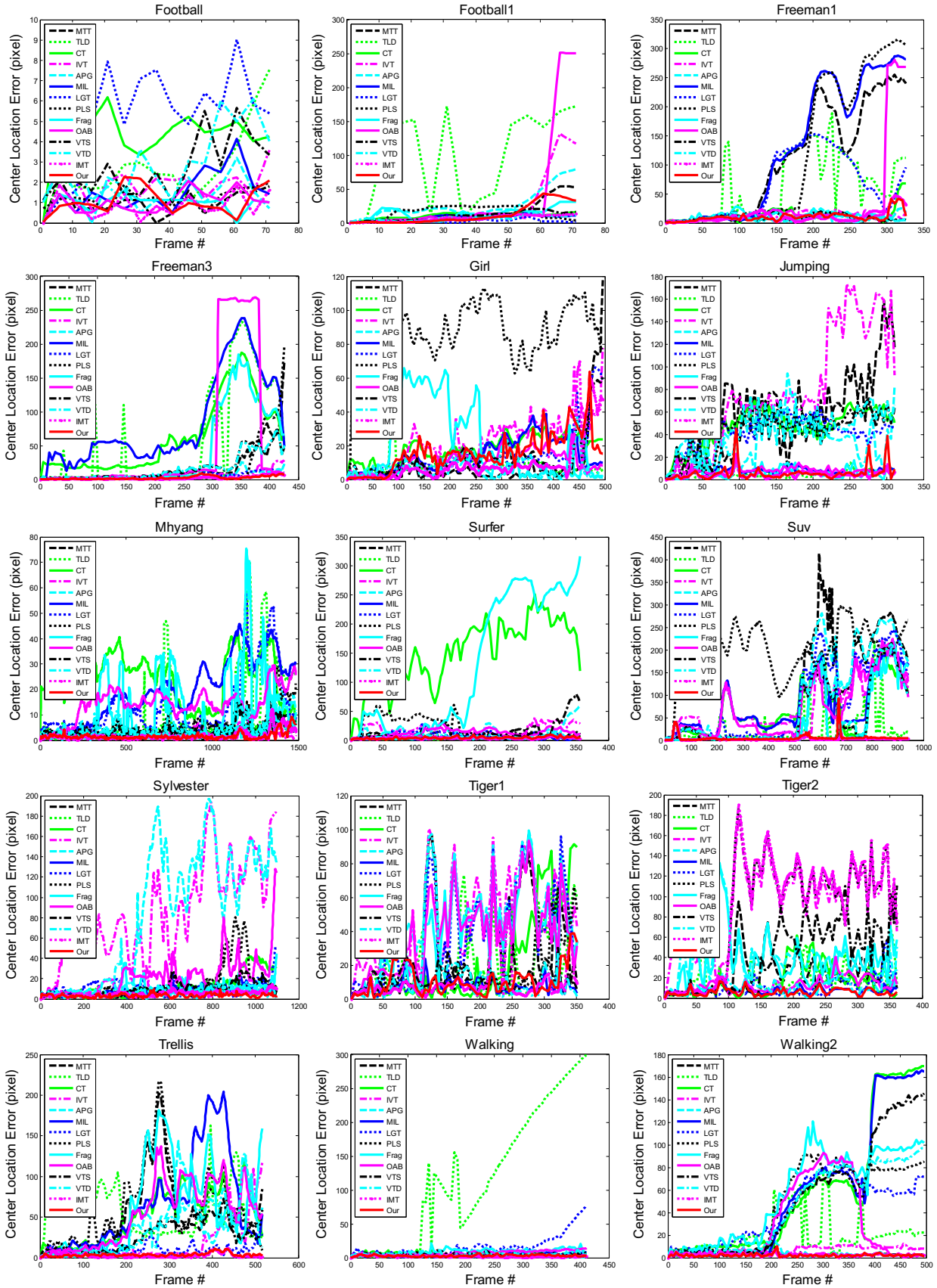

Fig. 4 *Center Location Error* (CLE) versus frame number

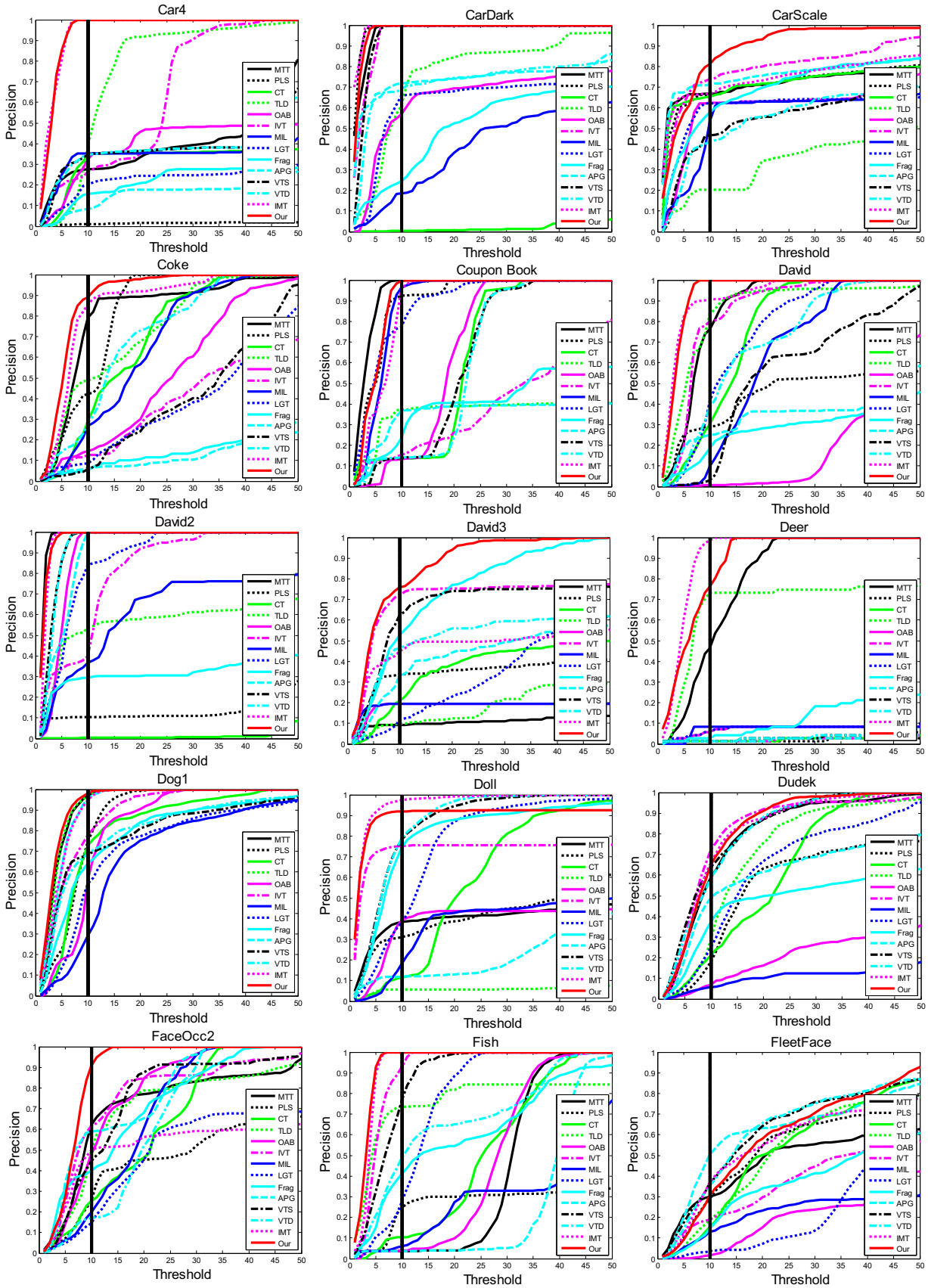

Fig. 5 *Precision Plots*. The threshold is set to 10 in our experiments (Color figure online)

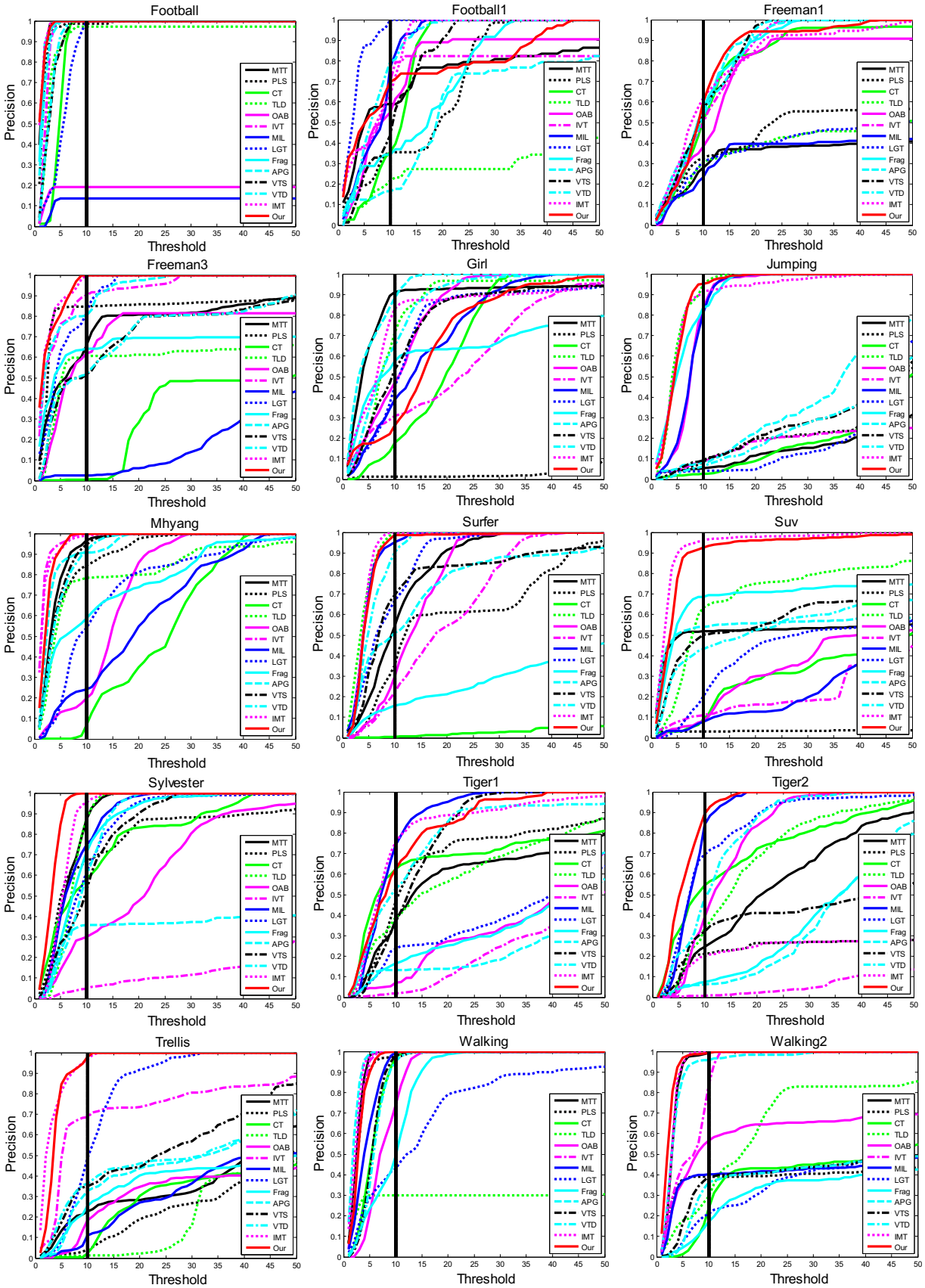

Fig. 6 *Precision Plots*. The threshold is set to 10 in our experiments (Color figure online)

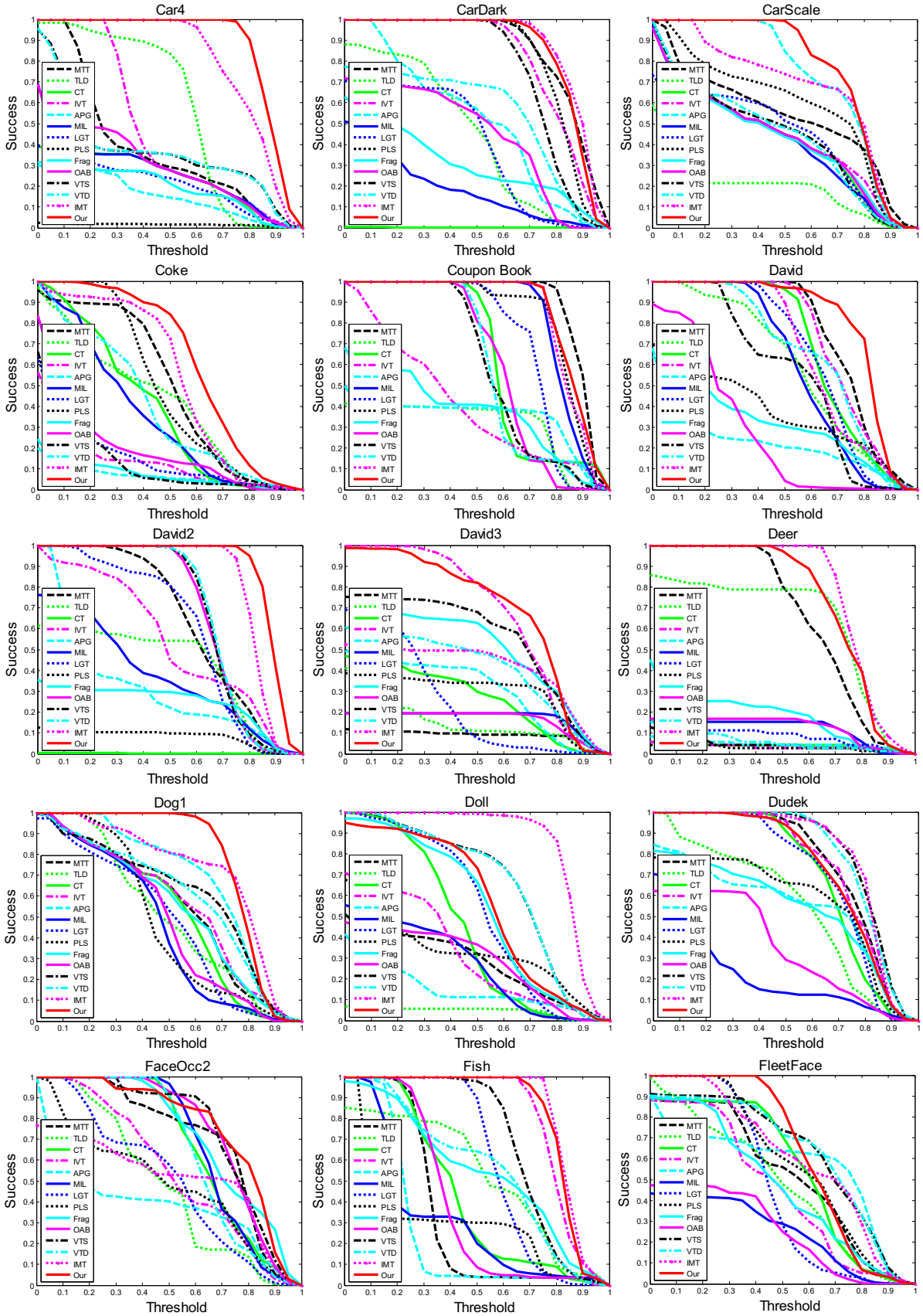

Fig. 7 *Success Plots* (Color figure online)

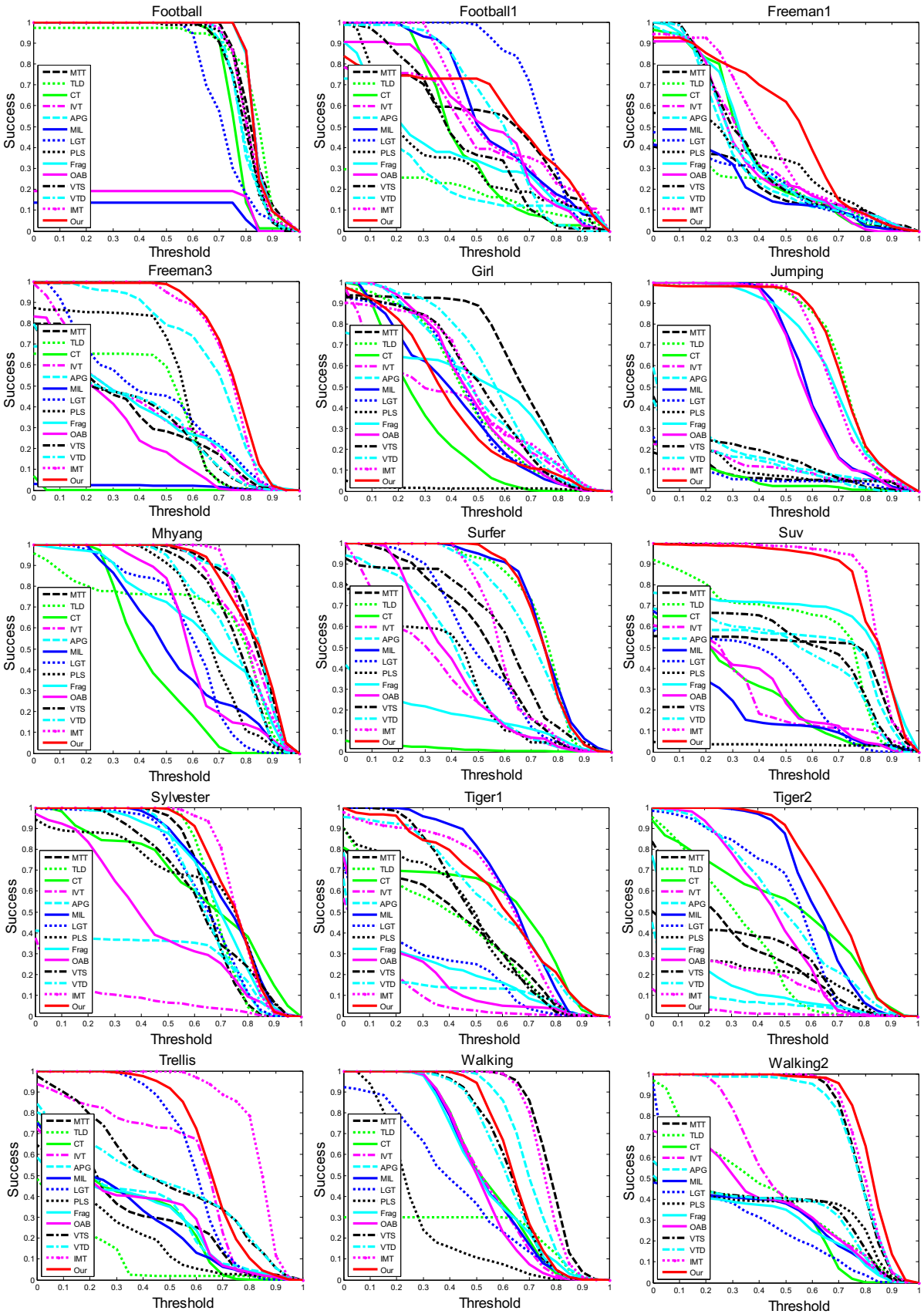

Fig. 8 *Success Plots* (Color figure online)

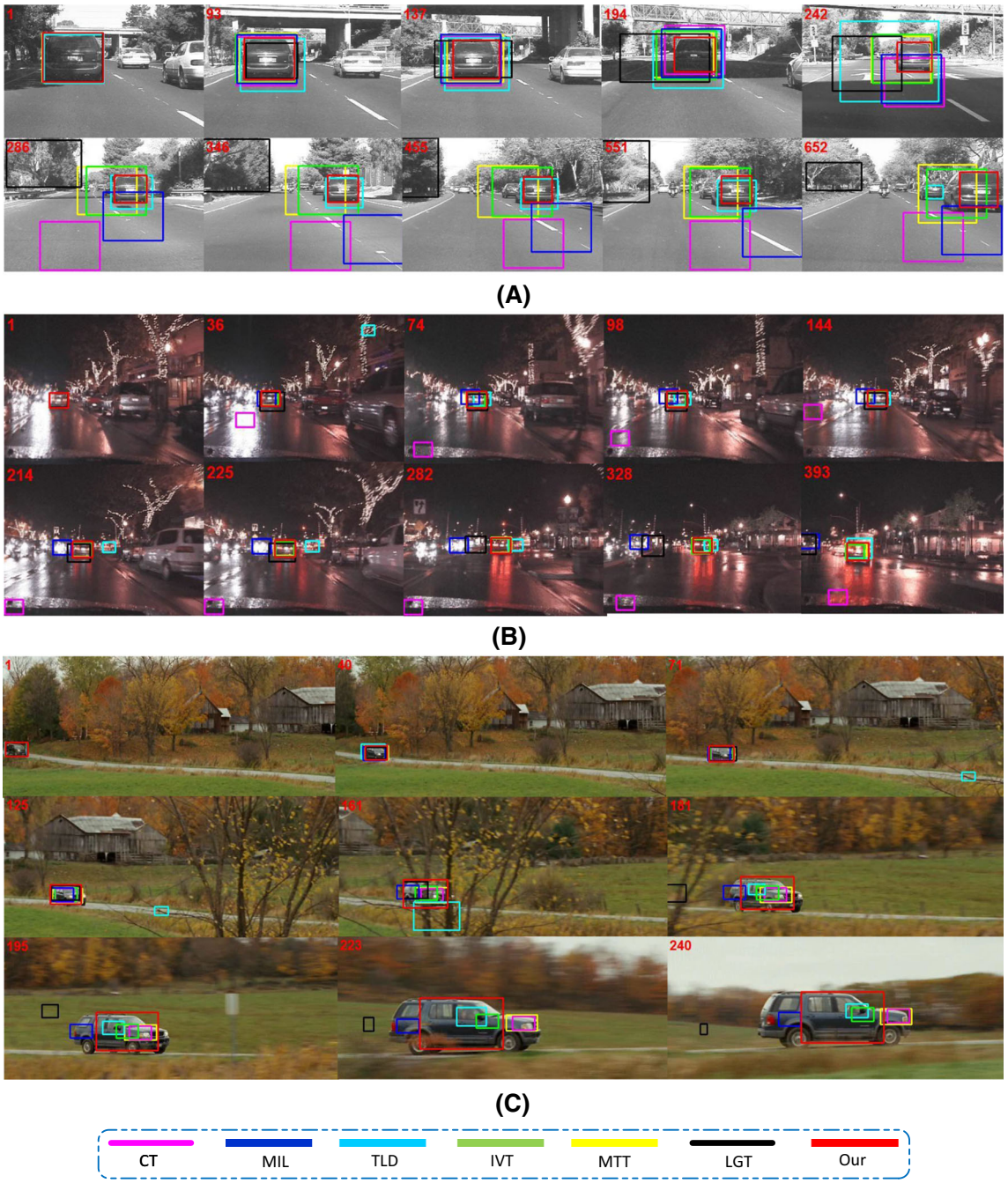

Fig. 9 Tracking frame results on car sequences, **a** *Car 4*, **b** *Car Dark*, **c** *Car Scale*

for scale variations, illumination variations; *CarDark* is challenging for cluttered background and scale variations. *CarScale* is challenging for largely scale variations.

For Fig. 9a, at the beginning of this video, e.g., frames 93 and 137, since the scale of the car is changed, TLD and LGT cannot locate the car well because they do not have a good scale estimation. In frame 194, the illumination condition of the car is greatly changed because the car is moving under a bridge, only our tracker can accurately locate the car in this case. LGT incorrectly locates the car. In frame 242, when the illumination is changed again, MIL, CT, TLD, LGT lose track of the car, IVT and MTT can track the target but have the incorrect scale. In contrast, our tracker works well until this frame. In frames 268, 346, 455 and 551, TLD tracks the car again. Since this tracker has a online learning and detection process, it can re-detect the target again, while other trackers almost lose track of the target. However, in frame 652, TLD loses the car since another car is also very similar to the

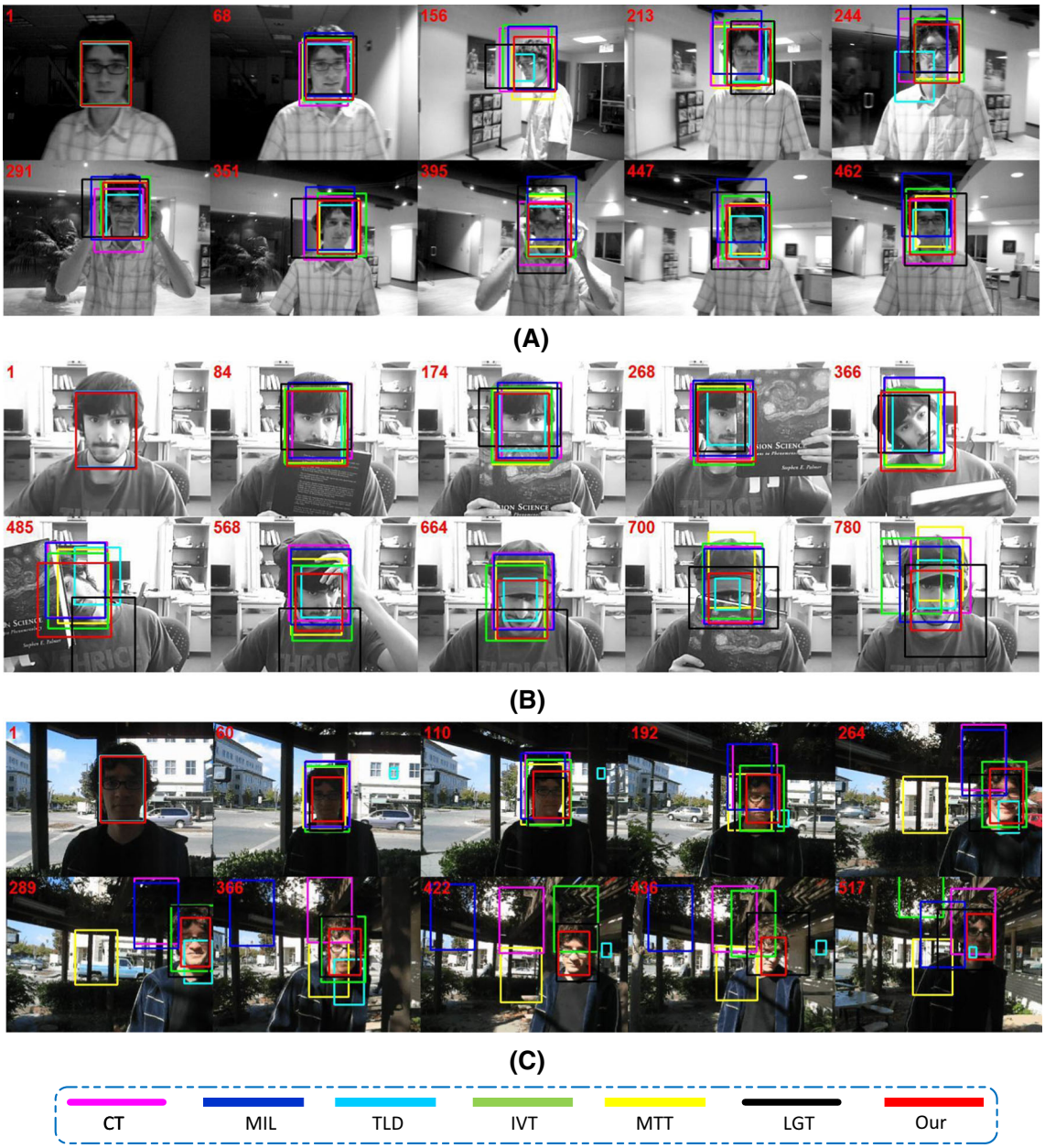

Fig. 10 Tracking results on face sequences. **a** *David Indoor*, **b** *Faceocc2*, **c** *Trellis*

training samples. Our tracker can accurately locate the car in the whole sequence.

For Fig. 9b, the background is very cluttered. In frame 36, TLD and CT lose track of the car, CT loses the target until the end of this sequence. In frame 74, 98 and 114, TLD re-detects the car again, but MIL partly drifts. In frame 214 and 225, MIL incorrectly locates the target. In frame 282, LGT loses track of the target. In frame 328 and 393, TLD partly drifts. Our tracker can locate the car in the whole sequence, MTT and IVT achieve comparable results on this sequence.

For Fig. 9c, the scale of the car is largely changed. In frame 71, TLD locates the target far away from the real target, since the detection process misleads the tracker. In frame 125 and 161, CT and MIL only partly track the target because those two trackers do not take the scale variation into consideration, especially in frame 161, where a tree brings in some noises. In frame 181, only our tracker can estimate the scale of the car. In frame 195, 223 and 240, all the other trackers only partly locate the target. Our method can well overcome the scale change for two reasons: (1) Subspace representation can overcome the scale variations in most of the cases. However, in particular cases, e.g., when the abrupt scale variation appears in consecutive frames, almost all the trackers may fail to track the target, including our approach; (2) Our

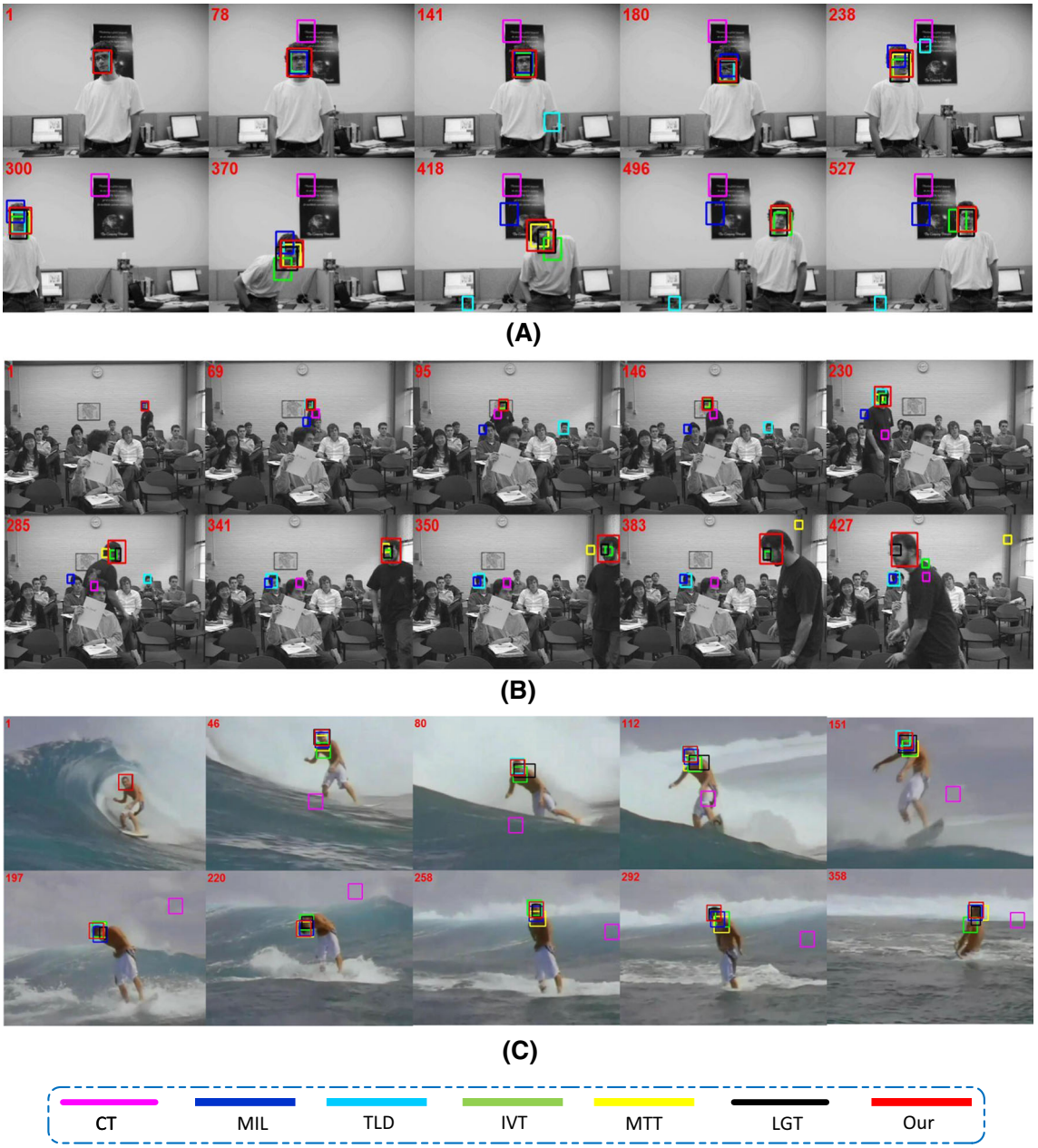

Fig. 11 Tracking frame results on small target sequences. **a** *David2*, **b** *Freeman3*, **c** *Surfer*

fusing algorithm can provide a more robust distance measure with the enhancement explored from the contexts, which can effectively select the most reliable candidate as the real target under several challenging conditions.

7.6.2 Qualitative Comparison on Face Sequences

In Fig. 10, frame results from three face sequences are shown: (A) *David Indoor*, (B) *Faceocc2*, (C) *Trellis*. The main challenges of *David Indoor* are illumination change and large pose and scale variations. *Faceocc2* is challenging for appearance variation, rotation and partial occlusion. *Trellis* is challenging for large illumination variation and pose change.

In Fig. 10a, *David Indoor* sequence, until frame 68, all the trackers almost track the target correctly. However, in frame 156, when the scale and appearance are both changed, only our tracker can locate the target accurately, we observe that HOG subspace works well in this case. In frame 213, MIL, CT and MTT drift from the target. In frame 291, 351, 395, 447, 462, while the other trackers drift more or less, our tracker can always track the target.

In Fig. 10b, *Faceocc2* sequence, when the target is occluded by a book in frame 174 and 268, MIL and CT drift

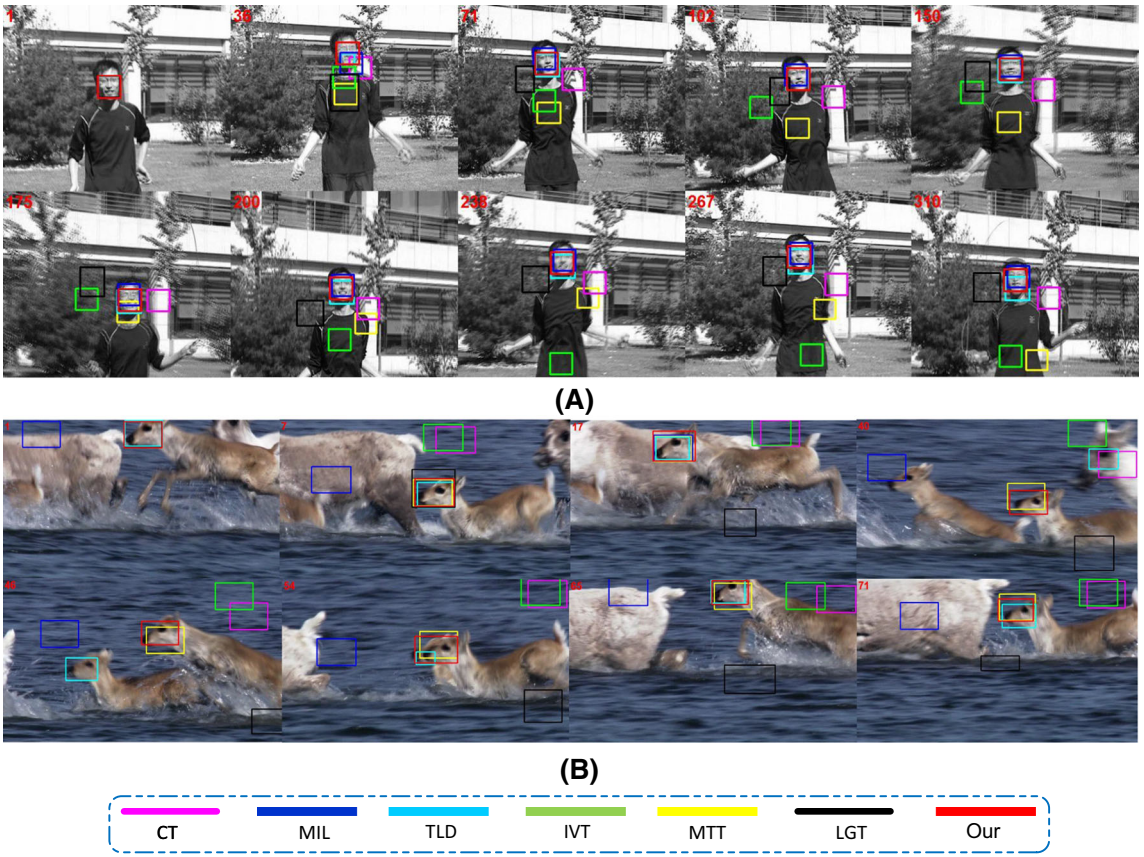

Fig. 12 Tracking frame results on fast motion sequences. **a** *Jumping*, **b** *Deer*

a bit, but our tracker does not suffer from this occlusion. In frames 366 and 485, when the target is rotated and occluded again, our tracker can also track the target correctly. In frames 586, 664, the appearance of the target is greatly changed, and only our tracker can locate the target accurately. In frame 700, our tracking result is much better than the other trackers, and in frame 780, IVT, MTT, LGT and MIL all drift away from the real target.

In Fig. 10c, *Trellis* sequence, many trackers lose track of the target due to great illumination, scale and appearance variations. In frame 192, MIL, CT and MTT are all drifted. In frame 289, MIL, CT and MTT lose track of the target and TLD also drifts. In frame 436 and 517, only our tracker can correctly track the target.

7.6.3 Qualitative Comparison on Small Target Sequences

In Fig. 11, frame results from three small target sequences are shown, (A) *David2*, (B) *Freeman3*, (C) *Surfer*. The main challenges of *David2* is appearance variance, *Freeman3* is challenging for appearance and scale variation, cluttered background, and *Surfer* is challenging for cluttered background.

In Fig. 11a, *David2* sequence, in frame 78, CT loses the target, TLD loses the target in frame 141. In frame 238, MIL drifts away. In frames 418, 496 and 527, when the appearance of the target is changed, only MTT and our tracker work well, MTT obtains comparable results on this sequence.

In Fig. 11b, *Freeman3* sequence, at the beginning, the target is very small, and the target's scale changes frame to frame. Also many other objects in this sequence are very similar to the target. As shown in frame 95 and 146, MIL and TLD locate the wrong target, in frames 285, 341, 350, 383 and 427, only our tracker can always track the target correctly.

In Fig. 11c, *Surfer* sequence, the goal is to track the head of the surfer. The background changes a lot and the pose of the surfer varies greatly. In frame 46, many trackers drift greatly. However, we can see that our tracker can always track the target accurately.

7.6.4 Qualitative Comparison on Fast Motion Sequences

In Fig. 12, the tracking frame results on two fast motion sequences are shown: (A)*Jumping*, (B)*Deer*. Fast motion leads to blur conditions which are very challenging for all the trackers.

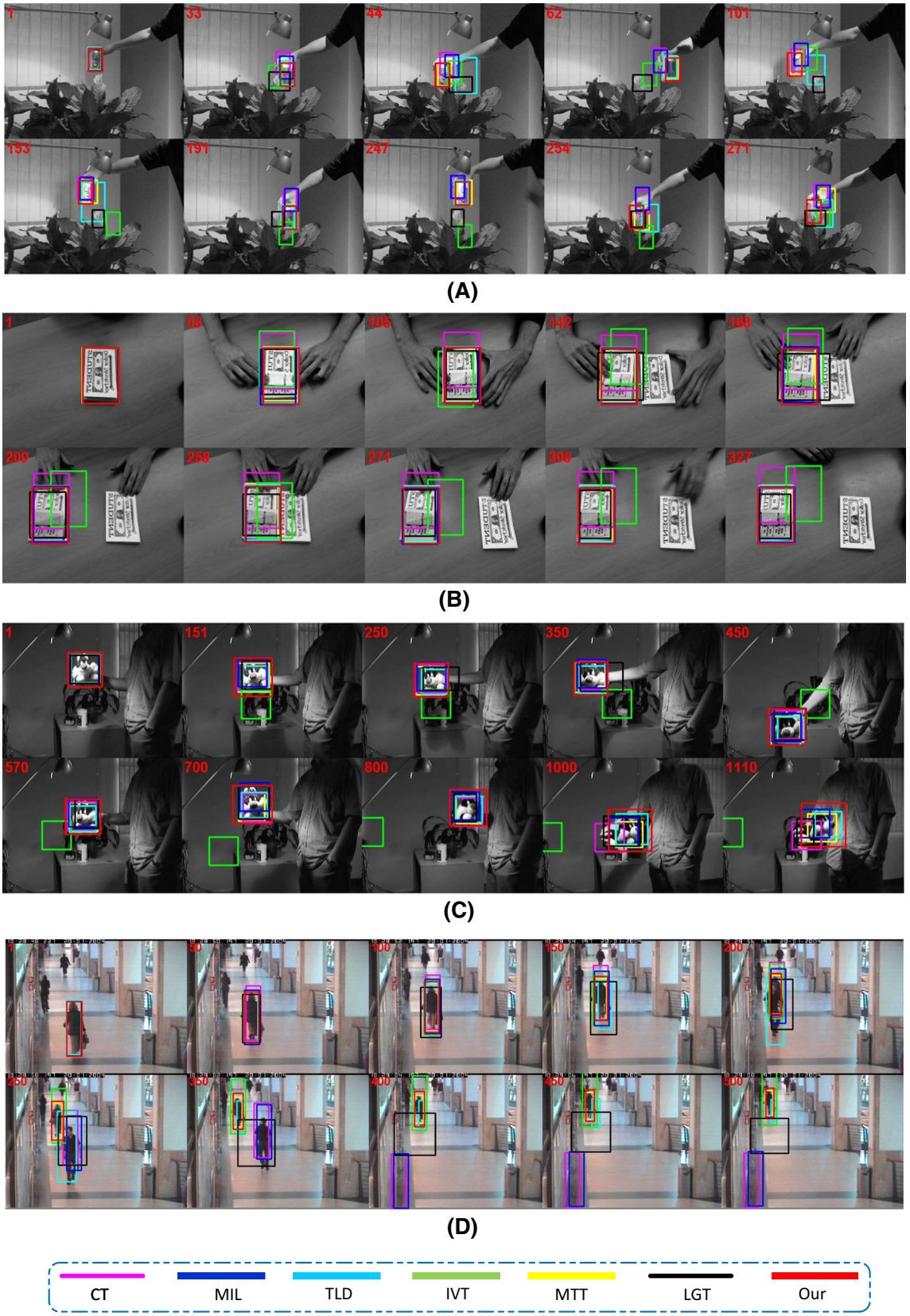

Fig. 13 Tracking frame results on other sequences: **a** *Coke Can*, **b** *Coup. Book*, **c** *Sylvester*, **d** *Walking2*

In Fig. 12a, *Jumping* sequence, a boy is jumping. we can see that in frame 36, all the other trackers lose track of the target, MTT, CT, LGT, IVT fail to track the target in the rest of the frames, LGT can re-detect the target and MIL also can work well in the rest of frames.

In Fig. 12b, *Deer* sequence, we can see that in frame 36, all the other trackers lose the target, MTT, CT, LGT, IVT fail to track the target in the rest of the frames, LGT can re-detect the target and MIL also can work well in the rest of frames.

7.6.5 Qualitative Comparison on Other Sequences

In Fig. 13, four other tracking results are shown: (A)*Coke Can*, (B)*Coup. Book*, (C)*Sylvester*, (D)*Walking2*. *Coke Can* is challenging for lighting change, appearance variation, partial or full occlusion. *Coup. Book* is challenging because there is another target very similar to the real target in this sequence. *Sylvester* is challenging for lighting variation and pose deformation.

In Fig. 13a, *Coke Can*. In frame 33, both IVT and LGT drift from the target, partial occlusion occurred before frame 44. It caused LGT and TLD to lose track of the target completely, MIL and CT slightly drift. Before 247, the target is fully occluded, however, when the target appears again, our tracker can still track the target correctly.

In Fig. 13b, *Coup. Book* sequence. In frame 58, when the target appearance has changed, CT and IVT both drift. In rest of the frames, LGT also cannot estimate the target well. MIL, MTT, TLD can obtain similar results with our tracker.

In Fig. 13c, *Sylvester* sequence, our tracker can track the target in the whole frames, the other trackers lose track of the target more or less.

In Fig. 13d, *Walking2* sequence, the scale of the target is changed, in frame 150, LGT and CT both drift a bit, in frame 200, another target appears, and the real target is partially occluded, hence, in frames 250 and 350, MIL, CT and LGT incorrectly track the wrong target.

8 Limitation and Future Work

As above-mentioned, the key contributions of our approach lie in designing a novel framework for fusing multiple representation instead of building a tracking system. For simplicity and generality, we only adopt the basic features like LBP, HOG, etc. in a whole template to clearly prove its effectiveness. Specifically, our approach do not include any occlusion handling process, and hence, our tracking approach cannot address the longtime full occlusion well currently. This is the main limitation of our approach. However, as demonstrated in the experimental result, even though using such simple representations, our method can more or less overcome the issues caused by the partial occlusion due to the multiple subspaces presented in our paper.

Furthermore, we believe fusing the more flexible representation or models in our framework can better handle the videos with longtime full occlusion, e.g., a possible solution is to utilize the sparse based representation (Bao et al. 2012; Hong et al. 2013; Jia et al. 2012; Mei and Ling 2011; Mei et al. 2011; Zhong et al. 2012), etc. Since in these approaches, the trivial template is adopted to handle occlusion, and the reconstruction error is also utilized to measure the similarity. Hence these representations can be easily integrated in our framework.

9 Conclusions

In this paper, a novel fused similarity measure is proposed for robust visual tracking. Multiple cues are integrated to a single uniform similarity measure to enhance the discriminative ability of the measurement. A novel diffusion process on a tensor product graph is utilized to achieve this goal. In addition, the context structure of the candidates in the forthcoming frame is utilized to further improve the similarity measurement. The proposed method has a quadratic time complexity, which makes it suitable for real time tracking. An extensive experimental evaluation on several challenging videos, clearly demonstrates that the proposed method significantly outperforms a large number of state-of-the-art tracking algorithms.

Acknowledgments This work was primarily supported by National Natural Science Foundation of China (NSFC) (Nos. 61222308, 61572207, and 61573160), and in part by Program for New Century Excellent Talents in University (No. NCET-12-0217), NSF grants OIA-1027897 and IIS-1302164, China 973 Program under Grant No. 2012CB316300, National Natural Science Foundation of China (NSFC) (No. 61173120).

Appendix: Proof of Proposition 1

Proof of Proposition 1 Equation (15) can be rewritten as

$$\begin{aligned}
\mathrm{P}^{(q+1)} &= P_{k,\alpha}\,(P_{k,\alpha})^q\,(P_{k,\beta}^T)^q\,P_{k,\beta}^T + \Delta \\
&= P_{k,\alpha}[P_{k,\alpha}\,(P_{k,\alpha})^{(q-1)}(P_{k,\beta}^T)^{(q-1)}\,P_{k,\beta}^T + \Delta]P_{k,\beta}^T + \Delta \\
&= (P_{k,\alpha})^2\,(P_{k,\alpha})^{(q-1)}(P_{k,\beta}^T)^{(q-1)}\,(P_{k,\beta}^T)^2 + P_{k,\alpha}\,\Delta\,P_{k,\beta} + \Delta \\
&= \cdots \\
&= (P_{k,\alpha})^q\,P_{k,\alpha}P_{k,\beta}^T\,(P_{k,\beta}^T)^q + (P_{k,\alpha})^{q-1}\,\Delta\,(P_{k,\beta}^T)^{q-1} + \cdots + \Delta \\
&= (P_{k,\alpha})^q\,P_{k,\alpha}P_{k,\beta}^T\,(P_{k,\beta}^T)^q + \sum_{e=0}^{q-1}(P_{k,\alpha})^e\,\Delta\,(P_{k,\beta}^T)^e
\end{aligned} \tag{20}$$

Lemma 1 $\lim_{q\to\infty}(P_{k,\alpha})^q\,P_{k,\alpha}P_{k,\beta}^T\,(P_{k,\beta}^T)^q = 0$

Proof It suffices to show that $(P_{k,\alpha})^q$ and $(P_{k,\beta}^T)^q$ go to 0, when $q \to \infty$. This is true if and only if every eigenvalue

of $P_{k,\alpha}$ and $P_{k,\beta}$ is less than one in absolute value. Since $P_{k,\alpha}$ and $P_{k,\beta}$ has nonnegative entries, this holds if its row sums are all less than one. As described in Sect. 5.1, we have $\sum_{b=1}^{N}(P_{k,\alpha})_{a,b} < 1$ and $\sum_{j=1}^{N}(P_{k,\beta})_{i,j} < 1$.

Lemma 1 shows that the first summand in Eq. (20) converges to zero, and consequently we have

$$\lim_{t\to\infty} \mathrm{P}^{(q+1)} = \lim_{t\to\infty} \sum_{e=0}^{q-1} (P_{k,\alpha})^e \; \Delta \; (P_{k,\beta}^T)^e \tag{21}$$

Lemma 2 *$vec\left((P_{k,\alpha})^e \; \Delta \; (P_{k,\beta}^T)^e\right) = (P)^e vec(\Delta)$ for $e = 1, 2, \ldots$.*

Proof Our proof is by induction.

Suppose $(\mathrm{P})^l vec(\Delta) = vec\left((P_{k,\alpha})^l \; \Delta \; (P_{k,\beta}^T)^l\right)$ is true for $e = l$, then for $e = l + 1$ we have

$$\begin{aligned}(\mathrm{P})^{l+1} vec(\Delta) &= \mathrm{P}\,\mathrm{P}^l vec(\Delta) \\ &= vec\left(P_{k,\alpha}\; vec^{-1}(\mathrm{P}^l vec(\Delta))\; P_{k,\beta}^T\right) \\ &= vec\left(P_{k,\alpha}\; ((P_{k,\alpha})^l \; \Delta \; (P_{k,\beta}^T)^l)\; P_{k,\beta}^T\right) \\ &= vec\left((P_{k,\alpha})^{l+1} \; \Delta \; (P_{k,\beta}^T)^{l+1}\right)\end{aligned}$$

and the proof of Lemma 2 is complete. □

By Lemma 1 and Lemma 2, we obtain that

$$vec\left(\sum_{e=0}^{q-1} (P_{k,\alpha})^e \; \Delta \; (P_{k,\beta}^T)^e\right) = \sum_{e=0}^{q-1} (\mathrm{P})^e vec(\Delta) \tag{22}$$

The following useful identity holds for the Kronecker Product (Vishwanathan et al. 2010):

$$\begin{aligned}vec(P_{k,\beta}\Delta P_{k,\alpha}^T) &= (P_{k,\alpha} \otimes P_{k,\beta}) vec(\Delta) \\ &= (\mathbb{P}) vec(\Delta)\end{aligned} \tag{23}$$

Putting together (21), (22), (23), we obtain

$$\begin{aligned}vec\left(\lim_{q\to\infty} \mathrm{P}^{(q+1)}\right) &= vec\left(\lim_{q\to\infty} \sum_{e=0}^{q-1} (P_{k,\alpha})^e \; \Delta \; (P_{k,\beta}^T)^e\right) \\ &= \lim_{t\to\infty} \sum_{e=0}^{q-1} \mathbb{P}^e vec(\Delta) \\ &= (\mathrm{I} - \mathbb{P})^{-1} vec(\Delta) \\ &= vec(\mathrm{P}^*).\end{aligned} \tag{24}$$

This proves Proposition 1.

References

Adam, A., Rivlin, E., & Shimshoni, I. (2006). Robust fragment-based tracking using the integral histogram. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 798–805).

Avidan, S. (2004). Support vector tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26(8)*, 1064–1072.

Avidan, S. (2007). Ensemble tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29(2)*, 261–271.

Babenko, B., Yang, M., & Belongie, S. (2011). Robust object tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(8), 1619–1632.

Badrinarayanan, V., Perez, P., Clerc, F. L., & Oisel, L. (2007). Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues. In *Proceedings of IEEE Interestial Conference on Computer Vision (ICCV)*.

Bai, X., Wang, B., Yao, C., Liu, W., & Tu, Z. (2010a). Co transduction for shape retrieval. *IEEE Transactions on Image Processing*, *32*(5), 861–874.

Bai, X., Yang, X., Latecki, L. J., Liu, W., & Tu, Z. (2010b). Learning context sensitive shape similarity by graph transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(5), 861–874.

Bao, C., Wu, Y., Ling, H., & Ji, H. (2012). Real time robust l1 tracker using accelerated proximal gradient approach. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Belkin, M., & Niyogi, P. (2004). Semi-supervised learning on riemannian manifolds. *Machine Learning*, *56*(special Issue on clustering), 209–239.

Čehovin, L., Kristan, M., & Leonardis, A. (2013). Robust visual tracking using an adaptive coupled-layer visual model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(4), 941–953.

Collins, R., Liu, Y., & Leordeanu, M. (2005). Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(10), 1631–1643.

Comaniciu, D., Member, V. R., & Meer, P. (2003). Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(5), 564–575.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 886–893).

Fan, J., Shen, X., & Wu, Y. (2012). Scribble tracker: A matting-based approach for robust tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(8), 1633–1644.

Fan, J., Wu, Y., & Dai, S. (2010). Discriminative spatial attention for robust tracking. In *Proceedings of European Conference on Computer Vision (ECCV)*.

Fan, Z., Yang, M., Wu, Y., Hua, G., & Yu, T. (2006). Efficient optimal kernel placement for reliable visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Francesc Moreno-Noguer, A. S., & Samaras, D. (2008). Dependent multiple cue integration for robust tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(4), 670–685.

Grabner, H., Grabner, M., & Bischof, H. (2006). Real-time tracking via on-line boosting. In *British Machine Vision Conference (BMVC)* (pp. 47–56).

Grabner, H., Leistner, C., & Bischof, H. (2008). Semi-supervised on-line boosting for robust tracking. In *Proceedings of European Conference on Computer Vision (ECCV)* (pp. 234–247).

Hong, Z., Mei, X., Prokhorov, D., & Tao, D. (2013). Tracking via robust multi-task multi-view joint sparse representation. In *Proceedings of IEEE Interestial Conference on Computer Vision (ICCV)*.

Hu, W., Li, X., Zhang, X., Shi, X., Maybank, S., & Zhang, Z. (2010). Incremental tensor subspace learning and its applications to foreground segmentation and tracking. *International Journal of Computer Vision*, *91*(3), 303–327.

Jia, X., Lu, H., & Yang, M. H. (2012). Visual tracking via adaptive structural local sparse appearance model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiang, N., Liu, W., & Wu, Y. (2011). Learning adaptive metric for robust visual tracking. *IEEE Transactions on Image Processing*, *20*(8), 2288–2300.

Jiang, N., Liu, W., & Wu, Y. (2012). Order determination and sparsity-regularized metric learning for adaptive visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kalal, Z., Mikolajczyk, K., & Matas, J. (2012). Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(7), 1409–1422.

Kwon, J., & Lee, K. M. (2010). Visual tracking decomposition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kwon, J., & Lee, K. M. (2011). Tracking by sampling trackers. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.

Kwon, J., & Lee, K. M. (2013). Highly non-rigid object tracking via patch-based dynamic appearance modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(10), 2427–2441.

Leichter, I. (2012). Mean shift trackers with cross-bin metrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(4), 695–706.

Li, X., Dick, A., Shen, C., van den Hengel, A., & Wang, H. (2013). Incremental learning of 3d-dct compact representations for robust visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(3), 863–881.

Li, X., Dick, A., Shen, C., Zhang, Z., van den Hengel, A., & Wang, H. (2013). Visual tracking with spatio-temporal dempster-shafer information fusion. *IEEE Transactions on Image Processing*, *22*(8), 3028–3040.

Li, X., Hu, W., Shen, C., Zhang, Z., Dick, A., & van den Hengel, A. (2013). A survey of appearance models in visual object tracking. *ACM Transactions on Intelligent Systems and Technology*, *4*(4), 58.

Liu, B., Huang, J., Kulikowski, C., & Yang, L. (2012). Robust visual tracking using local sparse appearance model and k-selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(5), 861–874.

Ma, J., Qiu, W., Zhao, J., Ma, Y., Yuille, A., & Tu, Z. (2015). Robust l2e estimation of transformation for non-rigid registration. *IEEE Transactions on Signal Processing*, *63*(5), 1115–1129.

Ma, J., Zhao, J., Tian, J., Yuille, A. L., & Tu, Z. (2014). Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, *23*(4), 1706–1721.

Mei, X., & Ling, H. (2011). Robust visual tracking and vehicle classification via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(11), 2259–2272.

Mei, X., Ling, H., Wu, Y., Blasch, E., & Bai, L. (2011). Minimum error bounded efficient l1 tracker with occlusion detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(7), 971–987.

Porikli, F., Tuzel, O., & Meer, P. (2006). Covariance tracking using model update based on lie algebra. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ross, D., Kim, J., Lin, R. S., & Yang, M. H. (2008). Incremental learning for robust visual tracking. *International Journal of Computer Vision*, *77*(1), 125–141.

Santner, J., Leistner, C., Saffari, A., Pock, T., & Bischof, H. (2010). Prost: Parallel robust online simple tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sinha, K., & Belkin, M. (2009). Semi-supervised learning using sparse eigenfunction bases. In *Advances in Neural Information Processing Systems (NIPS)*.

Spengler, M., & Schiele, B. (2003). Towards robust multi-cue integration for visual tracking. *Machine Vision and Applications*, *14*, 50–58.

Stenger, B., Woodley, T., & Cipolla, R. (2009). Learning to track with multiple observers. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tsagkatakis, G., & Savakis, A. (2011). Online distance metric learning for object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, *21*(12), 1810–1821.

Vishwanathan, S., Schraudolph, N., Kondor, R., & Borgwardt, K. (2010). Graph kernels. *Journal of Machine Learning Research*, *11*(4), 1201–1242.

Wang, D., Lu, H., & Yang, M. H. (2013). Online object tracking with sparse prototypes. *IEEE Transactions on Image Processing*, *22*(1), 314–325.

Wang, J., & Yagi, Y. (2008). Integrating color and shape-texture features for adaptive real-time object tracking. *IEEE Transactions on Image Processing*, *17*(2), 235–240.

Wang, M., Qiao, H., & Zhang, B. (2011). A new algorithm for robust pedestrian tracking based on manifold learning and feature selection. *IEEE Transactions on Intelligent Transportation Systems*, *12*(4), 1195–1208.

Wang, Q., Chen, F., Xu, W., & Yang, M. H. (2012). Object tracking via partial least squares analysis. *IEEE Transactions on Image Processing*, *21*(10), 4454–4465.

Wang, S., Lu, H., Yang, F., & Yang, M. H. (2011). Superpixel tracking. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.

Wang, X., Hua, G., & Han, T. X. (2010). Discriminative tracking by metric learning. In *Proceedings of European Conference on Computer Vision (ECCV)*.

Wu, Y., Cheng, J., Wang, J., Lu, H., Wang, J., Ling, H., et al. (2012). Real-time probabilistic covariance tracking with efficient model update. *IEEE Transactions on Image Processing*, *21*(5), 2824–2837.

Wu, Y., & Fan, J. (2009). Contextual flow. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wu, Y., Lim, J., & Yang, M. H. (2013). Online object tracking: A benchmark. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yang, M., Hua, G., & Wu, Y. (2009). Context-aware visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(7), 1195–1209.

Yang, M., Yuan, J., & Wu, Y. (2007). Spatial selection for attentional visual tracking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yang, X., & Latecki, L. J. (2011). Affinity learning on a tensor product graph with applications to shape and image retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*.

Yang, X., Prasad, L., & Latecki, L. J. (2013). Affinity learning with diffusion on tensor product graph. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 28–38.

Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A survey. *ACM Computing Surveys (CSUR)*, *38*(4), 1201–1242.

Yoon, J. H., Kim, D. Y., & Yoon, K. J. (2012). Visual tracking via adaptive tracker selection with multiple features. In *Proceedings of European Conference on Computer Vision (ECCV)*.

Zelnik-Manor, L., & Perona, P. (2004). Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 1601–1608).

Zhang, K., Zhang, L., & Yang, M. H. (2012). Real-time compressive tracking. In *Proceedings of European Conference on Computer Vision (ECCV)*.

Zhang, T., Ghanem, B., Liu, S., & Ahuja, N. (2013). Robust visual tracking via structured multi-task sparse learning. *International Journal of Computer Vision*, *101*(2), 367–383.

Zhao, Q., Yang, Z., & Tao, H. (2010). Differential earth mover's distance with its applications to visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(2), 274–287.

Zhong, W., Lu, H., & Yang, M. H. (2012). Robust object tracking via sparsity-based collaborative model. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhou, D., Bousquet, O., Lal, T., Weston, J., & Scholkopf, B. (2004). Learning with local and global consistency. In *Advances in Neural Information Processing Systems (NIPS)*.

Zhou, Y., Bai, X., Liu, W., & Latecki, L. J. (2012). Fusion with diffusion for robust visual tracking. In *Advances in Neural Information Processing Systems (NIPS)*.

Zhu, X. (2006). *Semi-supervised learning literature survey.* Technical Report 1530, Department of Computer Sciences, University of Wisconsin, Madison.