Optical Engineering

SPIEDigitalLibrary.org/oe

Online visual tracking based on selective sparse appearance model and spatiotemporal analysis

Ming Xue
Shibao Zheng
Hua Yang
Yi Zhou
Zhenghua Yu

SPIE

Online visual tracking based on selective sparse appearance model and spatiotemporal analysis

Ming Xue,[a,*] Shibao Zheng,[a] Hua Yang,[a] Yi Zhou,[b] and Zhenghua Yu[c]
[a]Shanghai Jiao Tong University, Institute of Image Communication and Network Engineering, Shanghai 200240, China
[b]Dalian Maritime University, Department of Electronics Engineering, Dalian 116026, China
[c]Bocom Smart Network Technologies Inc., Shanghai 200233, China

Abstract. To tackle robust visual tracking in complex environment, an online algorithm based on generative model is proposed. The target is represented with overlapped and selected local patches based on key point proportion ranking, and its location is estimated by spatiotemporal analysis. Temporally, a propagated affine warping dynamical model is newly introduced. Spatially, an observation model based on weighted sparse representation and geometric confidence inference is newly established. Both selection pattern and templates are periodically updated to adapt the target's appearance variation. Experiments demonstrate that the proposed approach achieves more favorable performance compared with classical works on challenging image sequences. [DOI: 10.1117/1.OE.53.1.013103]

Keywords: online visual tracking; key-point-based patch; geometric confidence propagation; state analysis; sparse representation; particle filtering; Bayesian inference.

Paper 130854 received Jun. 11, 2013; revised manuscript received Nov. 24, 2013; accepted for publication Nov. 27, 2013; published online Jan. 23, 2014.

1 Introduction

Visual tracking is an important topic in the computer vision community and has been intensively investigated during the recent decades. It lays the foundation for high-level visual problems such as motion analysis and behavior understanding. Generally speaking, visual tracking is applied in the tasks of motion-based recognition, automated surveillance, human–computer interaction (HCI), vehicle navigation, video indexing, etc.[1]

Historically, visual trackers proposed in early years always kept the target appearance model fixed throughout the whole image sequence.[2–4] Recently, methods proposed to track targets while evolving the appearance model in an online manner, called online visual tracking, have been popular.[5] An online visual tracking method typically follows the Bayesian inference framework[6] and mainly consists of three components: an appearance representation scheme, a dynamical model (or state transition model), and an observation model. In these components, the first one considers the formulation uniqueness of target appearance, the second one aims to describe the target states and their interframe relationship, and the third one evaluates the likelihood of an observed image patch belonging to the object class. Obviously, appearance model variation introduces several challenges. For example, the evolution incurs the risk of including wrong measurements and thus causes the tracking window to drift from the target. Moreover, the tracker must be able to online evaluate the quality of estimated results in the last frame, so that it could adjust its contributions to model update in the current frame. Although visual tracking has been intensively investigated, there are still many challenges such as partial occlusion, appearance variation, scale change, significant motion, cluttered background, etc. These challenges make the establishment of efficient online visual tracking a difficult task.

In this paper, an online visual tracking algorithm is proposed based on selective sparse appearance model and spatiotemporal analysis. Compared with other online tracking methods, main contributions of this work are concluded as follows: (1) For the representation aspect, a selective sparse appearance model is novelly proposed based on key patch selection, which establishes a balance between flexibility and uniqueness in target representation. (2) Temporally, an adaptive dynamical model is newly introduced based on target state analysis and joint-Gaussian propagation. The sampling covariance matrix is timely updated in view of the previous tracking results, which is different from the parameter-fixed proposals in other tracking algorithms. (3) Spatially, a geometric inference method is proposed to measure the appearance similarity for observation modeling. Different from the maximum-a-posterior (MAP) estimation in other generative works, target location estimation in this paper is conducted based on confidence inference using a portion of most similar candidates. Evaluations on numerous image sequences have been conducted, and the results demonstrate a more satisfactory performance compared with state-of-the-art online algorithms.

The remainder of this paper is organized as follows. Related works are presented in Sec. 2. In Sec. 3, general description of the proposed algorithm is introduced. Accordingly, details on target representation scheme are described in Sec. 4, whereas the tracking framework based on Bayesian inference is proposed in Sec. 5. Experimental results and discussions are given in Sec. 6. In Sec. 7, concluding remarks and possible directions for future research are provided.

*Address all correspondence to: Ming Xue, E-mail: silas_xue@sjtu.edu.cn

2 Related Works and Context

Visual tracking has been studied for several decades. In this section, studies related to our work are summarized. A thorough survey can be found in the related references.[1,7]

2.1 *Appearance Representation in Visual Tracking*

Representation of the target is basic but important to appearance-based visual tracking. Discrimination capability, computational efficiency, and occlusion resistance are generally considered as three main aspects for evaluation. Old tracking works construct the scheme in the form of feature point,[8] contour,[2] or silhouette.[3] For online visual tracking, the schemes are classified into patch-based schemes (e.g., holistic gray-level image vector[9,10] and fragments[11–13]), feature-based schemes,[14–17] statistics-based schemes[18–21] and their combinations. In patch-based schemes, Yang et al.[12] propose an attentional visual tracking algorithm by early extracting a pool of attentional regions that have good localization properties. Zhou et al.[13] explore the informative fragments based on human detectors to compose the reference model during the tracking process.

In target representation, taking the whole target region could be a good choice, since it collects all the visual information from the target and can be directly implemented without additional processing. However, such scheme could be blunt and lack flexibility, especially when the target appearance sharply varies, or when occlusion or abrupt motion occurs. Moreover, since the target is labeled using rectangles, the region inside the labeling rectangle but outside the target area could negatively affect the tracking performance. Practically, all visual data of the target is needed to be further processed, which results in heavy computation. Discovering the features or regions with little variance in scale, rotation, and translation is important in visual tracking.[8] Feature points take advantages in their uniqueness and flexibility on appearance representation. However, numbers of previous works merely transform visual information into data statistics, which lacks generalization capability. It also prevents further processing directly from the visual aspect. Moreover, intrinsic visual characteristics, such as continuity and sparsity, cannot be further exploited. Though targets could be jointly represented based on features and holistic regions, complicated calculations might cause slow processing speed.

2.2 *Particle Filtering for Online Visual Tracking*

Particle filtering is a Bayesian sequential importance sampling technique for the posterior distribution estimation of state variables characterizing a dynamical system. For visual tracking, various improved works have been proposed since the condensation algorithm.[2] In online visual tracking currently, it is regarded as a dynamical modeling method. Ross et al.[9] propose a variant of the condensation algorithm called affine warping. They model the target state $\mathbf{X}_t$ by a Gaussian distribution around the previous state $\mathbf{X}_{t-1}$, $p(\mathbf{X}_t|\mathbf{X}_{t-1}) = \mathcal{N}(\mathbf{X}_t; \mathbf{X}_{t-1}, \Psi)$, where Ψ is an affine covariance vector. Kwon et al.[22] propose a geometric method, where the two-dimensional (2-D) affine motion of a given target is estimated by means of coordinate-invariant particle filtering on the 2-D affine lie group $Aff(2)$. Mei and Ling[19] treat the local target motion as a constant velocity model and add the latest horizontal and vertical velocities to the translation parameters.

However, these works only consider the target state in the latest frame, which could be regarded as an one-dimensional (1-D) Markovian chain. They fail to make use of more previous tracking results. Moreover, they predefine the covariance matrix manually corresponding to different image sequences and keep it fixed in the whole tracking process. Therefore, they separate covariance and tracking result from each other and could prevent sampling from searching better candidates so that the tracking performance might be negatively affected.

2.3 *Online Generative Visual Tracking with Sparse Representation*

Observation modeling refers to a similarity evaluation process between the sampling candidates and the target and could be classified into three categories:[7] generative methods, discriminative methods, and hybrid methods. Generative methods focus on the exploration of a target observation with minimal predefined error based on specific evaluation criteria, whereas discriminative ones make attempts to maximize the margin between the target and nontarget regions using classification techniques. Hybrid trackers often integrate the two methods above into a combination framework. Specifically, generative visual trackers could be summarized including mixture models,[23,24] integral histogram,[11] subspace learning,[9,10] sparse representation,[19–21,25–27] visual tracking decomposition,[28] covariance tracking,[29] etc. They often drive the localization procedure by a maximum-likelihood or a MAP formulation relying on the target appearance model. Jepson et al.[23] design an elaborate mixture model with an online expectation-maximization algorithm to explicitly model the appearance changes during tracking. Adam et al.[11] decompose the template into fragments and vote on the possible positions and scales of the target by comparing their histograms with the corresponding candidate counterparts. Ross et al.[9] propose a generalized tracking framework based on the incremental principal component analysis subspace learning method with a sample mean update. Li et al.[29] explore the log-Euclidean Riemannian metric for statistics based on the covariance matrices of target features. Kwon and Lee[28] decompose the target observation model into multiple basic object models and then a compound tracking scheme is established by information integration and exchange via interactive Markov chain Monte Carlo (IMCMC). Cruz-Mota et al.[10] introduce spatial and temporal weights to the algorithm proposed by Ross et al.[9] and establish an incremental temporally weighted visual tracking algorithm with spatial penalty (ITWVTSP) for visual tracking.

Sparse representation follows the native linear combination characteristics and could capture the region similarity in a more efficient way.[30,31] It is first introduced to visual tracking by Mei and Ling.[19] They propose a l_1 minimization tracking algorithm, where the target is approximately spanned by target templates and trivial templates. The candidate with the smallest projection error is considered as the estimated tracking result. Liu et al.[20] model the target appearance based on a static sparse dictionary and a dynamically updated basis distribution, which is learned by K-selection and sparse-constraint-regularized mean-shift. Bao et al.[32] apply the accelerated proximal gradient

(APG) optimization approach to realize the real-time tracking performance. Bai and Li[26] construct the target appearance using a sparse linear combination of structured subspace unions, which consists of a learned eigen template set and a partitioned occlusion template set. Jia et al.[27] propose a structural local sparse appearance model to represent the target and introduce an alignment pooling method for location estimation.

3 General Description of Proposed Online Visual Tracking Algorithm

In this paper, we continue to explore the partial selection routines in appearance representation inspired by Yang et al.[12] and Zhou et al.,[13] and a generative online visual tracking algorithm is proposed based on selective sparse appearance model and spatiotemporal analysis. The workflow diagram is shown in Fig. 1. Once the target region is divided into overlapped patches, key patches would be selected as the representation of the target based on key point proportion ranking (KPPR). Accordingly, masked sparse representation is introduced to compute the patch coefficients based on elastic net regularization. In dynamical modeling, candidates are sampled based on affine temporal affine warping propagation. State analysis is conducted based on the joint Gaussian assumption and tracking information in the previous frames, and a parameter update scheme is introduced to adjust the dynamical model. Then, in observation modeling, the masked sparse representation is conducted to obtain the coefficients of the candidates, and their p-norms of kernel-weighted traces are established as the confidence scores for ranking. Most similar candidates obtained would be further used to estimate the target location based on Gaussian approximation. As time evolves, both selection pattern and template are periodically updated to adapt the target's appearance.

The proposed formulation has the following advantages. First, the proposed target representation scheme takes advantages of not only feature points in uniqueness and flexibility but also holistic region in comprehensiveness and efficiency. Second, the proposed affine propagation method temporally flexiblizes the covariance matrix of the distribution and provides more opportunities in searching better candidates. Third, the proposed process solves the linear approximation based on a masked and weighted convex optimization with elastic net regularizer, and thus manual setting of l_1 norm constraints is not necessary. The proposed p-norm of kernel-weighted trace function can capture the overall infinitesimal change in volume of the sparse coding output. Fourth, the proposed inference scheme has little negative influence in tracking accuracy but shows its spatial robustness against various visual challenges, especially cluttered background and severe occlusion.

4 Target Representation Based on Selective Sparse Appearance Model

In this section, we propose a selective sparse appearance model for target representation. Definition of a key patch and the KPPR algorithm is introduced and then the corresponding sparse representation scheme based on selected patches is presented.

4.1 *KPPR for Patch Selection*

We define a KEY patch for better selection of the target patches as follows:

Definition 1 In an image $\mathbf{Y}$, a region $\mathbf{P}$ is defined as a KEY patch when and only when the following conditions are satisfied:

1. At least the location and size of $\mathbf{P}$ have been defined inside $\mathbf{Y}$;
2. At least there is one key point $\mathbf{p}^{\mathrm{KEY}}$ in $\mathbf{P}$: $\mathbf{p}^{\mathrm{KEY}} \in \mathbf{P}$.

Thus, suppose L key feature points $\mathbf{p}_i^{\mathrm{KEY}}, i = 1, 2, \ldots, L$ have been detected in the target region, and K patches

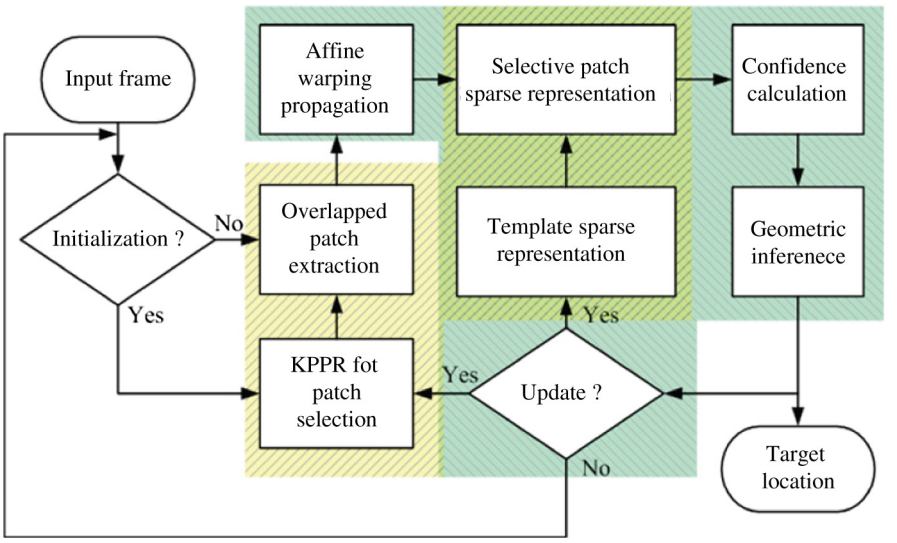

Fig. 1 Workflow of proposed algorithm. The proposed selective sparse appearance model based on KPPR in yellow shading is detailed in Sec. 4, whereas the proposed affine warping propagation, confidence calculation, geometric inference, and update process colored in green shading are described from Sec. 5.1 to 5.4.

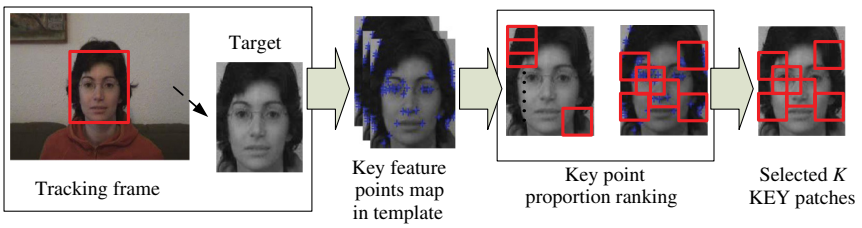

Fig. 2 Key point proportion ranking (KPPR). Key point features are firstly extracted and then the point proportions for each patch are calculated and ranked to boost the selected K KEY patches.

$\mathbf{P}_j, j = 1, 2, \ldots, K, K \leq N$ have been defined, the KEY patch $\mathbf{P}_j^{\text{KEY}}$ is generated as follows:

$$\mathbf{P}_j^{\text{KEY}} = \{\mathbf{P}_j, \mathbf{p}_i^{\text{KEY}} | \mathbf{p}_i^{\text{KEY}} \in \mathbf{P}_j, i = 1, 2, \ldots, L, j = 1, 2, \ldots, K\}. \tag{1}$$

In the rest of this paper, we would use $\mathbf{P}_j$ to represent a KEY patch $\mathbf{P}_j^{\text{KEY}}$ for simplification if there is no additional comment. Obviously, if there are key feature points for each patch, all the patches are regarded as key patches. Moreover, the number of key feature points in each patch can be different, and it could be assumed that the importance of a patch is positively proportional to the number of feature points that it contains. This assumption naturally follows the characteristics of features and could also be considered reasonable from a context perspective. Heuristically, if a key point is found, its local neighborhood could be also regarded as an important and representative region. Therefore, more feature points in a fix-size region infer that the neighborhoods connect with each other and compose a larger important region. In the extreme case, each pixel in the patch is decided as a feature point, and thus the whole region uniquely represents itself. For feature point extraction in this paper, the Shi-Tomasi corner detector method is chosen.[8] It finds points with large response function

$$\text{Res} = \min(\rho_1, \rho_2), \tag{2}$$

where ρ_1, ρ_2 are eigenvalues of a structured tensor $A = [x \quad y]\begin{bmatrix} g_x & g_{xy} \\ g_{xy} & g_y \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$. g_x, g_y and g_{xy} are the horizontal, vertical, and diagonal image gradients convolved with a circularly weighted window function. Other well-known feature point extraction methods might also be available. To select the most important patches, a key point proportion (KPP) is defined as follows.

Definition 2 For a KEY patch $\mathbf{P}_j, j = 1, 2, \ldots, K$ with L key feature points, its KPP is L, $\text{KPP}_j \triangleq L$, when and only when Eq. (1) is satisfied. Feature points are important in invariance capture for visual tracking, and it could be concluded that the more feature points a region contains, the more important it is. Thus, KPP is applied to evaluate the importance of a patch, and a KPPR is further presented to select the most important KEY patches, which is illustrated in Fig. 2 and summarized in Algorithm 1, namely, patches with the most key feature points are chosen. Once the KEY patches are decided, the selection pattern would be fixed in the next few frames before update.

4.2 *Target Sparse Representation Based on Selected Overlapped Patches*

The global appearance of an object under different illumination and viewpoint conditions is known to lie approximately in a low-dimensional subspace.[19] In this work, it is assumed that good target could be sparsely represented with a projecting residual by its selected overlapped patches in the target template subspace.

Suppose at time t, the target region $\mathbf{Y}_t$ with size $s_x, s_y, s = s_x \times s_y$ is sampled into N overlapped patches $\mathbf{Y}_t = [\mathbf{P}_t^1, \mathbf{P}_t^2, \ldots, \mathbf{P}_t^N]$, whose size is $d = d_x \times d_y, d_x \leq s_x, d_y \leq s_y$, and K patches are selected based on KPPR described above. Moreover, there exist a set of templates $\mathbf{T}_t = [\mathbf{t}_t^1, \mathbf{t}_t^2, \ldots, \mathbf{t}_t^M] \in \mathbb{R}^{(d_x \times d_y \times K) \times M}$, where M refers to the number of the templates. The corresponding patches, $\mathbf{t}_t^j = [\mathbf{b}_j^1, \mathbf{b}_j^2, \ldots, \mathbf{b}_j^K] \in \mathbb{R}^{d \times K}, j = 1, 2, \ldots, M$, have been stacked, normalized, and vectorized. They share the same patch sampling and selection scheme with that of the target candidates. Then, any patch of a target candidate $\mathbf{P}_t^i \in \mathbb{R}^d, i = 1, 2, \ldots, K$ in current frame will approximately lie in the linear span of the corresponding template patches in the past M frames

$$\mathbf{P}_t^i = \mathbf{b}_t^1 \beta_1^i + \mathbf{b}_t^2 \beta_2^i + \cdots + \mathbf{b}_t^{M \times K} \beta_{M \times K}^i \tag{3}$$

for some scalars, $\beta_k^i \in \mathbb{R}^K, i = 1, 2, \ldots, N, k = 1, 2, \ldots, M \times K$.

Algorithm 1 Key point proportion ranking (KPPR) for KEY patch selection.

Input:

Target region $\mathbf{Y}_t$, required KEY patch number K.

Predefined overlapped patches number N,patch size d_x, d_y, overlap rate R_o.

Output:

Selected KEY patch $\mathbf{P}_j^{\text{KEY}}, j = 1, 2, \ldots, K$.

1: Sample region $\mathbf{Y}_t$ into N patches $\mathbf{P} = \{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N\}$, with d_x, d_y and R_o.

2: Compute the key feature points $\mathbf{p}$ for $\mathbf{Y}_t$.

3: Obtain KPP_j for each patch $\mathbf{P}_j, j = 1, 2, \ldots, N$ by Eq. 1.

4: Rank $\text{KPP} = \{\text{KPP}_1, \text{KPP}_2, \ldots, \text{KPP}_N\}$ in descending order.

5: Obtain patch indexes corresponding to the first K KPP values.

Thus, a target patch $\mathbf{P}_t^i, i = 1, 2, \ldots, K$ is represented based on the dictionary composed of the corresponding templates by solving mask convex optimization problem based on elastic net regularization:[33,34]

$$\min_{\beta_j \in \mathbb{R}^{(M\times K)}} \frac{1}{2}\|\mathrm{diag}(\sigma_j)(\mathbf{Y}_t - \mathbf{D}\beta_j)\|_2^2 + \lambda_1 \frac{\|\gamma\|_0}{(d\times K)}\|\beta_j\|_1 + \frac{\lambda_2}{2}\|\beta_j\|_2^2,$$
$$\text{s.t.}\beta_j \geq 0, j = 1,2,\ldots,K, \tag{4}$$

where $\mathrm{diag}(\sigma_j)$ refers to the diagonal matrix supported by σ_j, and ϕ_j belongs to a block circulant mask matrix $\sum = [\sigma_1, \sigma_2, \ldots, \sigma_N] \in \mathbb{R}^{d\times N}$. Each column of $\sum$ corresponds to a vector compose of d successive "1" elements and $s - d$ "0" elements. λ_1 and λ_2 are regularization constants. Therefore, only K columns would be selected, and $\mathbf{D} = [\mathbf{t}_t^1, \mathbf{t}_t^1, \ldots, \mathbf{t}_t^K] = [\mathbf{b}_1^1, \mathbf{b}_1^2, \ldots, \mathbf{b}_1^K, \mathbf{b}_2^1, \mathbf{b}_2^2, \ldots, \mathbf{b}_2^K, \ldots, \mathbf{b}_M^1, \mathbf{b}_M^2, \ldots, \mathbf{b}_M^K] \in \mathbb{R}^{d\times(M\times K)}$ refers to the dictionary, whose columns are composed of the template patches according to the KPPR selection scheme described above.

5 Generative Visual Tracking Process Based on Spatiotemporal Analysis

An online visual tracking process could be interpreted as a Bayesian recursive and sequential inference task in a Markov model with hidden state variables. It could be further divided into cascaded estimation of dynamical model and observation model.[9] Suppose a set of target images $\mathbf{Y}_t = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t\}$ have been provided up to time t, the hidden state variable of the target $\mathbf{X}_t$ could be estimated as follows:

$$p(\mathbf{X}_t|\mathbf{Y}_t) \propto p(\mathbf{y}_t|\mathbf{X}_t) \int p(\mathbf{X}_t|\mathbf{X}_{t-1}) p(\mathbf{X}_{t-1}|\mathbf{Y}_{t-1}) \mathrm{d}\mathbf{X}_{t-1}, \tag{5}$$

where $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ refers to the dynamical model between two consecutive states and $p(\mathbf{y}_t|\mathbf{X}_t)$ denotes the observation model related to the likelihood estimation of $\mathbf{y}_t$ based on the state $\mathbf{X}_t$. The target state in this paper is approximately parameterized using a six-tuple set introduced by Ross et al.,[9] $\mathbf{X}_t = \{x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t\}$. The elements, respectively, denote horizontal and vertical translation, rotation angle, scale, aspect ratio, and skew direction.

5.1 *Dynamical Modeling: Temporal Propagation Based on Affine State Analysis*

In this paper, we analyze the current target state based on previous ones with a joint Gaussian assumption proposed below. The comparison of original and proposed affine warping is shown in Fig. 3. Correspondingly, a theorem is described as follows with informal proof afterwards.

Theorem Suppose $\mathbf{X}_t = \{x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t\}, t \geq 0$, where each element is time-varying random variable, $\mathbf{X}_t$ is joint Gaussian.

Proof Since the joint distribution of single gaussian variables is still Gaussian,[35] based on Gaussian assumption proposed by Ross et al.[9] and the target state definition, the theorem holds. □

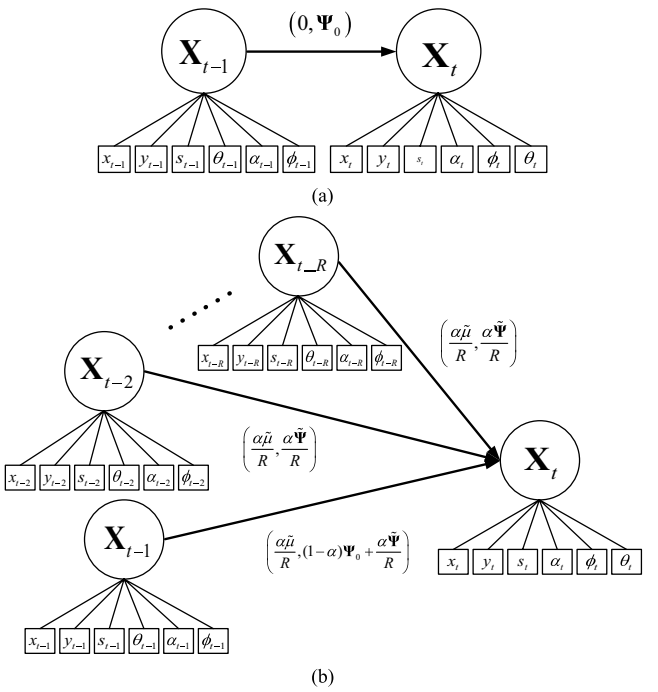

Fig. 3 Affine warping comparison. The original affine warping introduced by Ross et al.[9] in (a) only considers the target state in the latest frame, and the proposed one in (b) consider more previous target state with a nonfixed covariance update.

Thus, the dynamical model could be updated based on the analysis of previous target states in a joint Gaussian way, the new model is presented as

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = \mathcal{N}(\mathbf{X}_t; \tilde{\mathbf{X}}_{t-1} + (1-\alpha)\tilde{\mu}, (1-\alpha)\Psi_0 + \alpha\tilde{\Psi}), \tag{6}$$

where α is an update rate parameter, and Ψ_0 contains the initial affine variances of six elements. To tackle unexpected motion variation, the target states in previous R frames are approximately considered as the input for $\tilde{\Psi}$ calculation in this paper. Correspondingly, suppose $\mathbf{X}_R = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_R]^T$, $\tilde{\mu}$ and $\tilde{\Psi}$ up to time t could be computed following Gaussian kernel estimation by

$$\tilde{\mu} \approx \tilde{\mu}_R = \frac{1}{R}\sum_{i=1}^{R} \tilde{\Psi}_R, \tilde{\Psi} \approx \tilde{\Psi}_R = \mathrm{var}(\mathbf{X}_R), \tag{7}$$

where $\mathrm{var}(\mathbf{X}_R)$ refers to the variance of $\mathbf{X}_R$. $\tilde{\mathbf{X}}_{t-1}$ is computed detailed in Sec. 5.3.

The proposed dynamical model could also be viewed as a weighted multidimensional Markovian chain form for affine warping, which transforms the 1-D Markovian chain to a weighted R-D form. Moreover, it is also a sample-biased estimation. Though the general dynamical assumption between two target states in the indefinite time process follows a Gaussian distribution without bias, the states of a specific target are predictable given motion continuity assumption, and thus the estimation could be biased associated with previous target states given fixed time interval.

5.2 *Observation Modeling: Confidence Calculation Based on Weighted Sparse Representation*

Based on the selective sparse appearance model described above, we introduce a patch-view form of Eq. (4) as

$$\min_{\beta_j \in \mathbb{R}^{(M\times K)}} \frac{1}{2}\|\mathbf{P}_t^j - \mathbf{D}\beta_j\|_2^2 + \lambda_1\|\beta_j\|_1 + \frac{\lambda_2}{2}\|\beta_j\|_2^2. \quad \text{s.t.}\beta_j \geq 0, j = 1, 2, \ldots, K. \tag{8}$$

Equation (8) can be solved by the least angle regression (LARS) algorithm to compute the coefficients $\beta_j = [\beta_1^j, \beta_2^j, \ldots, \beta_{M\times K}^j]$. The details of the LARS algorithm could be referred to Ref. 36.

Earlier templates could be more similar with the initial appearance of the target, but it might influence the target appearance approximation in abrupt variation. Thus, a temporal weight $\mathbf{W}$ is introduced as

$$\mathbf{W} = [\mathbf{w}, \mathbf{w}, \ldots, \mathbf{w}] \in \mathbb{R}^{M\times K}, \tag{9}$$

$$\mathbf{w} = \left[\frac{e^{-\eta_0}}{\sum_{j=0}^{K-1} e^{-(\eta_0-\eta j)}}, \frac{e^{-(\eta_0-\eta)}}{\sum_{j=0}^{K-1} e^{-(\eta_0-\eta j)}}, \ldots, \frac{e^{-(\eta_0-(K-1)\eta)}}{\sum_{j=0}^{K-1} e^{-(\eta_0-\eta j)}}\right]^T, \tag{10}$$

where η_0 and η are constants to control the weights. Thus, Eq. (8) changes to

$$\min_{\beta_j \in R^d} \frac{1}{2}\|\mathbf{P}_t^j - \langle \mathbf{D}, \mathbf{W}\rangle\beta_j\|_2^2 + \lambda_1\|\beta_j\|_1 + \frac{\lambda_2}{2}\|\beta_j\|_2^2, \quad \text{s.t.}\beta_j \geq 0, j = 1, 2, \ldots, K, \tag{11}$$

where $\langle\cdot,\cdot\rangle$ refers to the inner product. Equation (11) could also be solved by LARS.[36] The difference is that each column is premultiplied with a weight $\mathbf{w}$. It should be noted that though the template-based sparse representation has recently been discussed,[19–21,26,27] all of them fail to consider the issue of template importance from a temporal perspective.

The fragmented tracking algorithm[11] applies the kernel-weighted scheme, which assigns low weights to the pixels far from the target's center. These pixels are more likely to contain background information or occluding objects, and thus their contributions to location estimation should be diminished. In this paper, we apply this conception to coefficient-based confidence modeling. Suppose $\beta = \{\beta_1, \beta_2, \ldots, \beta_M\} \in \mathbb{R}^{(M\times K)\times M}$ have been obtained and the corresponding trace is

$$\mathbf{e} = \text{trace}\left(\sum_{j=1}^{M}\beta_j\right), \tag{12}$$

a p-norm of kernel-weighted trace for β is presented for confidence calculation, and the confidence score L_v for a certain target candidate is defined as

$$\mathrm{L}_v \triangleq \|\mathbf{k}_i\mathbf{e}_i\|_p = \left(\sum_{i=1}^{K}(\kappa e)^p\right)^{\frac{1}{p}}, \tag{13}$$

where $\mathbf{e}_i$ refers to the i'th element. $\mathbf{k}$ is defined as $\mathbf{k} = \{\kappa_i\}_{i=1}^K$, where κ_i refers to the i'th value of a vectorized Gaussian kernel function κ. It follows the same selection pattern with that of the patch described in Sec. 4.

5.3 *Observation Modeling: Geometric Inference of Candidate Confidence*

Compared with the maximal scheme in previous works, we construct the observation estimation based on the spatial distribution of top candidates in the confidence ranking results. To begin with, a 3-D confidence-coordinate space (CCS) is introduced as follows.

Definition 3 Given a set of target candidates (x_t^k, y_t^k), $k \in \mathbb{Z}^+$, and the corresponding normalized confidence scores are L_k, the CCS is defined as

$$\mathbf{CCS} = \{\mathbf{O}_k | \mathbf{O}_k = (x_t^k, y_t^k, L_k), k \in \mathbb{Z}^+\}. \tag{14}$$

If we illustrate the distribution of top candidate confidence scores in a local area around the true target location shown in Fig. 4, it could be found that without noise introduced, the more candidates we obtain, the more Gaussian the confidence distribution would be. This could be proved by classical center limit theorem, and each candidate is regarded as a sample of confidence. Suppose there is only one point with the maximal confidence corresponding to the target in the current frame, and each candidate is sampled following a Gaussian distribution around the target, the confidence would gradually drop as it moves away from the extreme point. Based on these conceptions, we assume that the

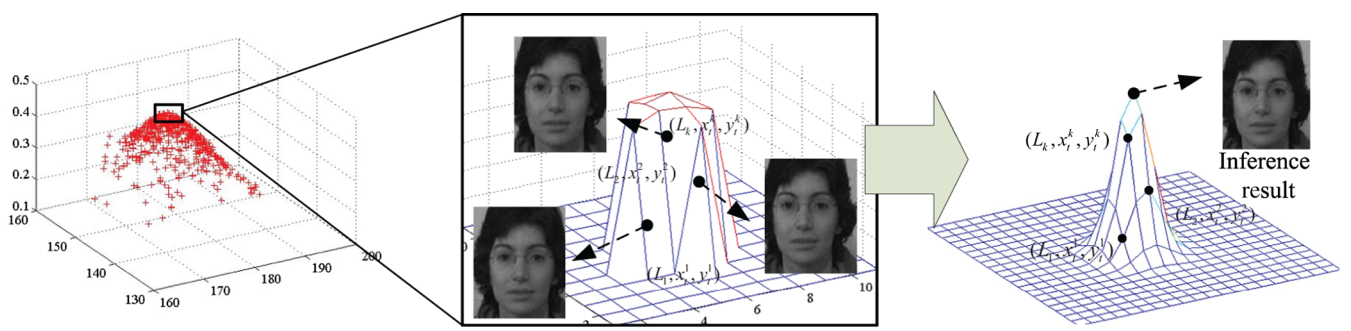

Fig. 4 Confidence scores distribution in a local area and inference result by Gaussian approximation. Without noise introduced, the more candidates we obtain, the more Gaussian the distribution of the confidence would be.

confidence values follow a Gaussian distribution in CCS of a local limited region.

Then, a geometric inference method is presented to estimate the target location. Suppose Q points with highest confidence scores are known, the observation in this paper approximates a 2-D Gaussian function in CCS to find the peak. Furthermore, the observation estimation for a certain target candidate is proportional to the geometric confidence inference output defined as

$$p(\mathbf{y}_t|\mathbf{X}_t) \propto I(L, \mathbf{X}_t), \tag{15}$$

where, $I(\cdot)$ refers to the inference result. It should be noted that Q should not be large, since noises could be introduced, and therefore the assumption above might not be met. In this paper, only the minimum of Q is predefined, and the sample number finally used is subject to increase. The geometric inference process is summarized in Algorithm 2. Each time, Q points in CCS are obtained, a Gaussian fit is conducted. The inference results would be checked and used to compose the target state in current frame, otherwise inference would be applied for another maximal $M-1$ times with Q update per time. Eventually, if there is no suitable result, the target state used for sampling would be updated with a predicted bias vector Δ computed by constant velocity approximation.

5.4 *Template and Selection Pattern Update*

Long-time fixed templates might negatively affect the tracking performance in dynamic scenes, and an update is essential. In this paper, we propose to periodically replace one of the templates set $\mathbf{t}_t^i, i = 1, 2, \ldots, M$ by sparse representation. A template $\tilde{\mathbf{t}}$ could be obtained by sparsely representing the estimated target vector $\tilde{\mathbf{Y}}_t$ using a linear combination of eigen-basis vectors based on elastic net. The equation is

$$\min_{\mathbf{c}\in\mathbb{R}^s} \frac{1}{2}\|(\mathbf{C}_t - \mathbf{H}\mathbf{c})\|_2^2 + \lambda_1\|\mathbf{c}\|_1 + \frac{\lambda_2}{2}\|\mathbf{c}\|_2^2, \tag{16}$$

where $\mathbf{H} = [\mathbf{U}\mathbf{I}], \mathbf{c} = [\mathbf{q}\mathbf{e}]$. $\mathbf{U}$ is the matrix composed of eigen-basis vectors computed following the method by Ross et al.,[9] $\mathbf{q}$ refers to the coefficients of eigen-basis vectors, and $\mathbf{e}$ represents trivial noises. A similar process also appears in Ref. 27. Comparatively, we do not apply the l_1 constraint but the elastic net one. This process could also be viewed as template denoising with underlying formulation $\mathbf{t} = \mathbf{U}\mathbf{q} + \mathbf{e}$, so that reconstruction errors in Eqs. (4) and (8) due to appearance variation can be effectively reduced. If deformation occurs, the selected patch would regularly change to adapt the appearance variation. Since the target is labeled in rectangles, some areas that do not belong to the target might be within the rectangles. However, it would not affect the final tracking performance because these areas are limited. The overlapped patches within the target region cover the major areas and would eliminate the noise. The template update strategy is summarized in Algorithm 3.

In this paper, it is assumed that the first M templates of the target are known, which can be generated by manual labeling or other trackers. In the mean time, the KPPR algorithm would be reapplied on the tracking result to re-select the KEY patches.

Algorithm 2 Spatial confidence inference based on 2-D Gaussian approximation in CCS.

Input:

Q points $\mathbf{O}_k = (x_t^k, y_t^k, L_k), k = 1, 2, \ldots, Q$, where $Q \ll V$, maximal iteration number M, initial covariance vector $\Psi_\mathbf{0}$, fitting tolerance Tol.

Output:

Target state $\mathbf{X}_t$, state for sampling $\tilde{\mathbf{X}}_t$.

1: Initialize $\tilde{\mathbf{C}} = \{\tilde{x}, \tilde{y}\}$ by $\tilde{x} = +\inf, \tilde{y} = +\inf, F_c = 0$.

2: Compute the average value $\bar{\mathbf{C}} = \{\bar{x}, \bar{y}\}$ of Q points as $\bar{x} = \frac{1}{Q}\sum_{k=1}^{Q} x_k, \bar{y} = \frac{1}{Q}\sum_{k=1}^{Q} y_k$.

3: Obtain the centralized Q values as $\bar{x}_k = x_k - \bar{x}, \bar{x}_k = y_k - \bar{y}, k = 1, 2, \ldots, Q$.

4: **While** $\|c_x - \bar{x}\|_2^2 > \Psi_x^2$ or $\|c_y - \bar{y}\|_2^2 > \Psi_y^2$ **do**

5: Obtain c_x or c_y by Gaussian fit of Q points with Tol using Least Square Fit.

6: **if** c_x or c_y is null **then**

7: Update the inference flag vector $\mathbf{F} \leftarrow [\mathbf{F}, 1]$.

8: Update the inference flag counter $F_c \leftarrow F_c + 1$.

9: Update $Q \leftarrow Q + 1$.

10: **else**

11: Update the inference flag vector $\mathbf{F} \leftarrow [\mathbf{F}, 0]$.

12: Obtain the inference result $\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \tilde{\mathbf{C}} + \bar{\mathbf{C}}$.

13: Break

14: **end if**

15: **if** $F_c = M$ **then**

16: Obtain $\mathbf{X}_t$ through the candidate with maximal confidence in Eq. 13.

17: Obtain a predicted bias vector $\Delta = [\frac{1}{M}\sum_{i=0}^{M-1}(x_{t-i} - x_{t-i-1}), \frac{1}{M}\sum_{i=0}^{M-1}(y_{t-i} - y_{t-i-1}), 0, 0, 0, 0]$.

18: Update the state for sampling $\tilde{\mathbf{X}}_t \leftarrow \mathbf{X}_t + (\sum_{i=0}^{M_0} F_{t-i})\Delta, M_0 = \lfloor \min(\frac{\Psi_x}{2}, \frac{\Psi_y}{2}) \rfloor$.

19: Break.

20: **else**

21: Update the state for sampling $\tilde{\mathbf{X}}_t \leftarrow \mathbf{X}_t$.

22: **end if**

23: **end while**

24: Obtain $\theta_t, s_t, \alpha_t, \phi_t$ through the candidate with maximal confidence in Eq. 13.

25: Obtain the estimated target state $\mathbf{X}_t = \{x_t, y_t, \theta_t, s_t, \alpha_t, \phi_t\}$.

5.5 *Summary of Algorithm*

The proposed algorithm is integrated in Algorithm 4.

Qualitatively, in Algorithm 4, sparse coding in confidence score calculation and template update are the most time-

Algorithm 3 Template update based on elastic net regulation.

Input:

Estimated target vector $\tilde{\mathbf{Y}}_t$, eigen-basis vectors $\mathbf{U}$, template set $\mathbf{T}_t = \{\mathbf{t}\}_t^i, i = 1, 2, \ldots, M$ and regularization parameter λ_1, λ_2.

Output:

New template set $\mathbf{T}_t$.

1: Solve Eq. 16 to obtain $\mathbf{q}$.

2: Generate a random integral number $i \in [2, M]$ to index the template to be replaced.

3: Replace the template $\mathbf{t}_t^i$ with $\tilde{\mathbf{t}} = \mathbf{Uq}$.

4: Normalize the template set $\mathbf{T}_t$.

Algorithm 4 Proposed online visual tracking algorithm.

Input:

Image sequence with T frames, initial target state $\mathbf{X}_0$, particle numbers V, inference point number Q, template and selection update frequency U_f, template weight $\mathbf{W}$, fitting tolerance Tol, template number M, overlapped percentage, state number for analysis R, update rate α, constant λ_1, λ_2, d_x, d_y, s_x, s_y, η_0, η.

Output:

Current target state $\mathbf{X}_t$, template set $\mathbf{T}_t$.

1: *(Initialization)* Track the target in the first M frames to obtain the state $\mathbf{X}_{1:M}$ and template set $T_{1:M}$.

2: *(Patch Selection)* Obtain the K KEY patches based on KPPR by Algorithm 1.

3: **for** $t = M + 1 \rightarrow T$ **do**

4: *(Dynamical Modeling)* Obtain V target candidates $\{\tilde{\mathbf{y}}_t\}_{k=1}^V$ based on affine warping propagation by Eq. 6 and Eq. 7.

5: *(Observation Modeling)* Obtain $\{\beta^k\}_{k=1}^V$ based on sparse coding by Eq. 11.

6: *(Observation Modeling)* Obtain $\{\mathrm{L}_{\mathrm{v}}^k\}_{k=1}^V$ based on confidence score by Eq. 12 and 13.

7: *(Observation Modeling)* Conduct geometric inference to obtain $\mathbf{X}_t$ and $\tilde{\mathbf{X}}_t$ based on $\{x^k, y^k, \mathrm{L}_{\mathrm{v}}^k, \}_{k=1}^Q$, $Q < V$ in CCS by Algorithm 2. The maximal iteration number is set M.

8: **if** $t/U_f = 0$ **then**

9: *(Template Update)* Obtain new template set $\mathbf{T}_t$ by Algorithm 3.

10: *(Selection Pattern Update)* Update the K KEY patches based on KPPR by Algorithm 1.

11: **end if**

12: **end for**

consuming part, and the proposed spatial confidence inference process ranks second. The dynamical modeling and patch selection part take the least running time. To speed up processing, we apply a C implementation of elastic net regulation proposed by Mairal et al.[34] Moreover, we define an inference flag counter F_c in the proposed confidence inference algorithm. It controls the maximal iteration number so that the algorithm would not take infinite time to search for an inference result. Further quantitative analysis is described in the next section.

6 Experiment and Discussion

In this section, we present experiments on test image sequences to demonstrate the efficiency and effectiveness of the proposed algorithm. Both qualitative and quantitative evaluations are presented as follows, and additionally, separate evaluations and analysis on the number of patch selection, the confidence inference algorithm and the computation complexity are also conducted.

6.1 *Experiment Setup*

The proposed algorithm is implemented in MATLAB and C/C++, which runs at 1.0 to 1.6 fps on a 2.5-GHz machine with 2 GB RAM. For parameter configuration, the target region is normalized to 32×32 pixels, $d_x = d_y = 32, d = 1024$, and the patch size is set to 16×16 pixels, $s_x = s_y = 16, s = 256$, while the overlapped percentage of neighbored patch is 0.5. Thus, totally nine overlapped patches are sampled, $N = 9$. Six hundred particles are used for dynamical modeling, $V = 600$. Target states of the latest eight frames are used for propagation, $R = 8$, and the update rate parameter is set to 0.1, $\alpha = 0.1$. $M = 10$, where the target at the first frame is manually labeled and the other $M - 1$ frames are labeled based on the tracking results by a KD-tree forest visual tracker.[37] The regularization constants λ_1 and λ_2 are set to be 0.01, and $Q = 5$ for the initial number of particle inference. The inference tolerance is set to be 0.1, Tol $= 0.1$. Both the template and KEY patch selection pattern are set to be updated for every five frames, $U_f = 5$. The weight parameters in Eq. (10) are $\eta_0 = 1, \eta = 0.1$. In all the experiments of this paper except Sec. 6.4, six patches are selected, $K = 6$.

It should be noted that the settings on V, M, U_f, λ_1, and λ_2 above are based on the setup of classical online visual tracking algorithms so as for better performance comparison.[9,10,16,17,19,38] The overlapped percentage of neighbored patch is related to the appearance variation of the target region. Since low percentage number would lead to lower efficiency, and the benchmark video is of various kinds, an unbiased number 0.5 is set. Q is set considering the least numbers for Gaussian fitting. Other parameters including R, α, η_0, η, and Tol are established after times of experiments with reference to the balance between accuracy and efficiency. Increasing them would lead to lower accuracy, while high R and α would cause the sampling location drift away, resulting in unfavorable adaption for fast motion and occlusion handling.

For tracking performance evaluation, 14 image sequences, totally more than 6,000 frames, are used in the experiments, where the target locations through all the frames are already manually labeled as ground truth. Comparatively, the proposed tracker is evaluated against eight state-of-the-art algorithms based on the source codes provided by the authors, including Frag,[11] IVT,[9] VTD,[28] L1T,[19] MIL,[16] TLD,[17] ITWVTSP,[10] and PLS.[38] These image sequences

described above are also separatively obtained from their web sites. Their parameter settings are shown in Table 1. Since implicit stochasticity exists in all of the algorithms, each quantitative score below is averagely computed considering the results of five independent runs of the corresponding algorithm. Live video demos and more results can be obtained from the authors.

6.2 *Qualitative Evaluation*

Qualitative analysis and discussions are provided as follows in common use of tracking human bodies, vehicles, and human and animal faces. The visual challenges include heavy occlusion, illumination change, scale change, fast motion, cluttered background, pose variation, motion blur, and low contrast.

6.2.1 *Human bodies*

Tracking human bodies is widely used in motion-based recognition and automated surveillance. The sequences used for evaluation include *Caviar 1*, *Caviar 2*, and *Singer*.

It is shown in Fig. 5 that IVT,[9] ITWVTSP,[10] MIL,[16] L1T,[19] and PLS[38] do not perform well in *Caviar 1*. They fail to discover the target when it is occluded by a similar object (e.g., #0133 and #0192). Only the proposed tracker, VTD,[28] Frag,[11] and TLD[17] handle the heavy occlusion successfully. However, VTD[28] and Frag[11] cannot smoothly adapt the scale changes of the person (e.g., #0133 and #0367). In *Caviar 2*, almost all the trackers evaluated except PLS[38] and MIL[16] can follow the target. However, many of them including IVT,[9] VTD,[28] ITWVTSP,[10] and TLD[17] cannot adapt the scale as the human moves near to the camera (e.g., #0220 and #0455). By contrast, our algorithm performs well in terms of position estimation and scale adaptation.

Table 1 Main parameter settings for eight state-of-the-art algorithms.

Method	Main parameter settings
Frag	16 bins in the histograms, 36 horizontal and vertical patches, 25% of patches for vote map combination
IVT	Patch size 32×32, forgetting term 0.95, a maximum of 16 eigenvectors, a block update of 5 frames, 600 particles
VTD	4 types of features, 5 patches, 8 basic trackers
L1T	Template size 16×16, similarity function threshold 0.5, 8 target templates, 600 particles
MIL	45 positive image patches, 65 negative image patches, learning rate 0.85, 50 out of 250 weak classifiers are chosen
TLD	Patch size 15×15, threshold for NN classifier 0.6, classification margin threshold 0.1, scale step 1.2, bounding box size 20 pixels, Gaussian kernel, 100 positive patches
ITWVTSP	Error threshold 0.07, spatial weight 2.0 for "spec"; others are the same as IVT
PLS	Patch size 32×32, 1 positive sample, 30 negative samples, 10 weight vectors, a maximum of 5 appearance models, forgetting factor 0.8, 600 particles

In *Singer* shown in Fig. 5(c), only the results of partial trackers (e.g., proposed and VTD)[28] are satisfactory, while the others cannot adjust the scale [e.g., Frag,[11] L1T,[19] and MIL[16]] or accurately locate the target [e.g., TLD[17] at #098, #0116 and #0226, IVT[9] at #0126]. Both drastic scale and location deviation occur when lighting condition changes. Especially, PLS[38] cannot capture the scale variation of the target through all the frames of *Singer*. The ITWVTSP[10] algorithm performs much better than the IVT algorithm[9] in this video. Comparatively, the proposed algorithms can locate the target more accurately and robustly against illumination variation.

6.2.2 *Human and animal faces*

Face detection and tracking are very important in HCI and animal monitoring application. In the experiments, five videos are used including *David Indoor*, *Occlusion 1*, *Occlusion 2*, *Girl*, and *Deer*.

Figure 6 shows that in *Occlusion 1*, all the evaluation algorithms can follow the target approximately correctly, yet some trackers drift from the face when occlusion occurs [e.g., MIL[16] at #0300, #0565, and #0833, ITWVTSP[10] at #0565 and #0833, IVT,[9] L1T,[19] Frg,[11] TLD,[17] and VTD[28] at #0565]. In *Occlusion 2*, the differences are more obvious. It can be found that L1T[19] drifts more from the target compared with other algorithms [e.g., MIL[16] at #0576, and #0713], and IVT[9] and TLD[17] cannot adapt the appearance during occlusion and head rotation (e.g., #0713). Though the VTD[28] and ITWVTSP[10] could locate the face center more accurately, they could not cover the occluded area due to pose variation (e.g., #0713). PLS[38] cannot continuously follow the target, while MIL[16] and Frag[11] estimate the target less accurately than the proposed algorithm.

In *Girl*, it is found in Fig. 7 that only the proposed algorithm, Frag,[11] TLD,[17] and VTD[28] can consistently follow the face, while the proposed method can estimate the location more accurately (e.g., at #0310 and #0345). The other trackers gradually drift from the target to the surroundings. In *David Indoor*, some algorithms [e.g., Frag[11] and PLS[38]] drift away from the target during the tracking process, while some algorithms cannot adapt the scale when out-of-plane rotation occurs [e.g., MIL[16] and L1T[19] at #0175 and #0389, VTD[28] and ITWVTSP[10] at #00389]. In *Deer*, the successful trackers only include the proposed algorithms, VTD[28] and PLS,[38] while the others fail to capture the head of *deer* when it jumps up and down repeatedly. Comprehensively and qualitatively speaking, the proposed algorithms perform the best.

6.2.3 *Vehicles*

In vehicle navigation, especially self-driving technology, the basic role is to steadily track the rear of vehicles against different kinds of weather conditions and road environments. The sequences used for evaluation include *Car 4* and *Car 11*, which are separately recorded in the day and at night. It is shown in Fig. 8 that Frag[11] and MIL[16] do not perform well in the first two sequences. When the car goes into or out of the shadows, there is a drastic lighting change, which causes the estimated locations by VTD[28] and L1T[19] to drift (e.g., at #0312 and #0429). The ITWVTSP[10] tracker can locate the target center accurately but fails to adapt the scale change.

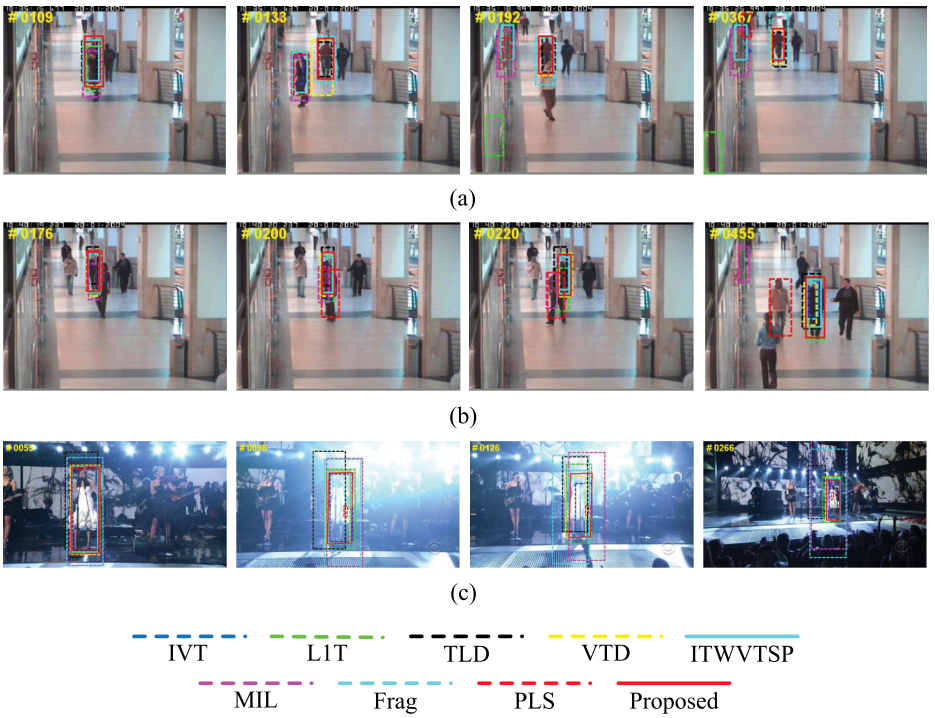

Fig. 5 Qualitative evaluation of (a) *Caviar 1*, (b) *Caviar 2*, and (c) *Singer*, where object appearances change drastically due to heavy occlusion, scale change, and light variation. Similar objects also appear in the scenes. Six patches are selected for the proposed algorithm.

In *Car 11*, only IVT,[9] ITWVTSP,[10] PLS,[38] and the proposed algorithm successfully track the target in the whole sequence. The remaining trackers drift away or take the surroundings as the target [e.g., MIL[16] at #0182 and #0269 and VTD[28] and L1T[19] at #0269 and #0336].

6.3 *Quantitative Evaluation*

Besides qualitative evaluation, quantitative evaluation of the tracking results is also an important issue which typically computes the difference between the predicted and the manually labeled ground truth information. Similar with other classical works, two performance criteria are applied to compare the proposed tracker with other reference trackers. The first one refers to center error (CE) evaluation, which is the CE based on Euclidean distance from the tracking location to the ground truth center at each frame. The second one refers to the overlap ratio evaluation, which is also used in object detection[39] and defined as the share area proportion of the box obtained by tracker and the one by ground truth at each frame.

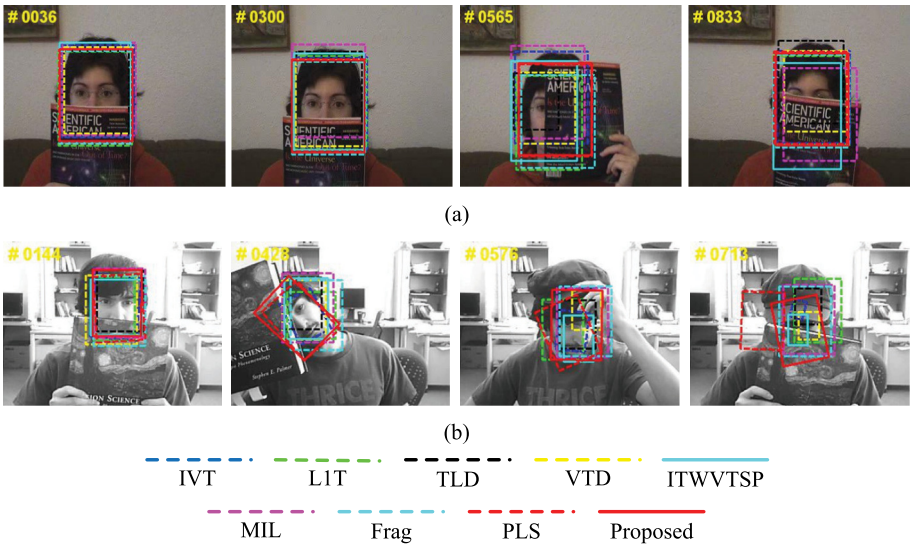

Fig. 6 Qualitative evaluation of (a) *Occlusion 1* and (b) *Occlusion 2*, where object appearances change drastically due to heavy occlusion and pose variation. Six patches are selected for the proposed algorithm.

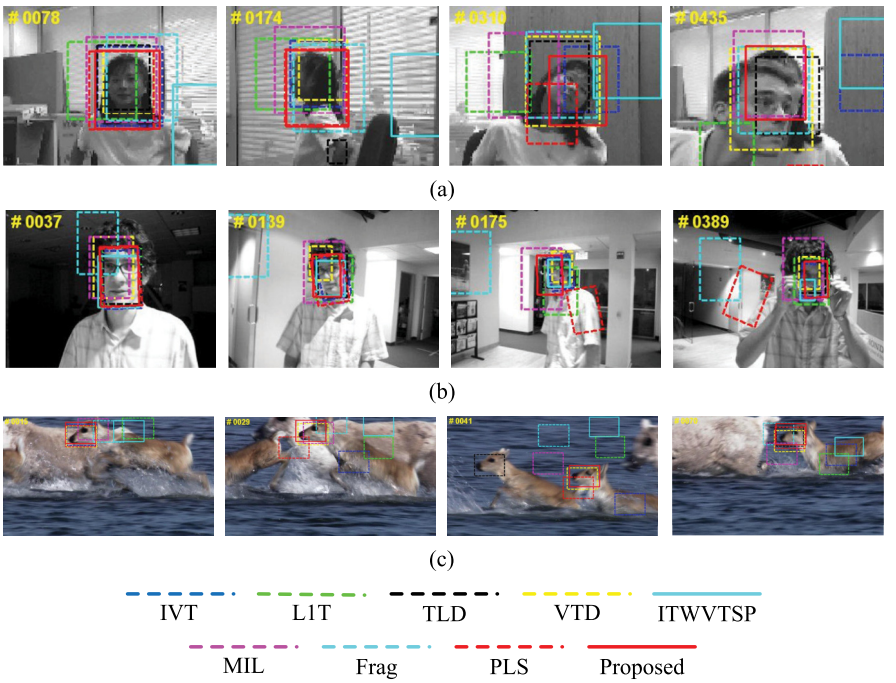

Fig. 7 Qualitative evaluation of (a) *Girl*, (b) *David Indoor*, and (c) *Deer*, where object appearances change drastically due to fast motion, pose variation, light variation, scale change, cluttered background and motion blur. Six patches are selected for the proposed algorithm.

Furthermore, in this paper, the average CE (ACE) and average overlap rate (AOR) are introduced, which are defined as

$$\mathrm{ACE} = \frac{1}{T}\sum_{i=1}^{T} \|\mathbf{c}_{\mathrm{eval}}^{i} - \mathbf{c}_{gt}^{i}\|_2^2, \tag{17}$$

$$\mathrm{AOR} = \frac{1}{T}\sum_{i=1}^{T} \frac{\mathbf{A}_{\mathrm{eval}}^{i} \cap \mathbf{A}_{gt}^{i}}{\mathbf{A}_{\mathrm{eval}}^{i} \cup \mathbf{A}_{gt}^{i}}, \tag{18}$$

where $\mathbf{c}_{\mathrm{eval}}^{i}, \mathbf{c}_{gt}^{i} \in \mathbb{R}^{2\times 1}$ refer to the horizontal and vertical center coordinates of the evaluation and ground-truth labeling results at the i'th frame, respectively, and $\mathbf{A}_{\mathrm{eval}}^{i}, \mathbf{A}_{gt}^{i} \in \mathbb{R}^{+}$ are corresponding areas of the target in one test sequence.

The results of ACE and AOR for 10 sequences above are summarized in Table 2. For each sequence, the first line refers to ACE, whereas the second refers to AOR. It can be concluded that the proposed tracking method runs the best or the second-best performance on ACE and AOR in all the tested trackers. Though some CE values are higher, the gaps are limited, and all the AORs of proposed tracker

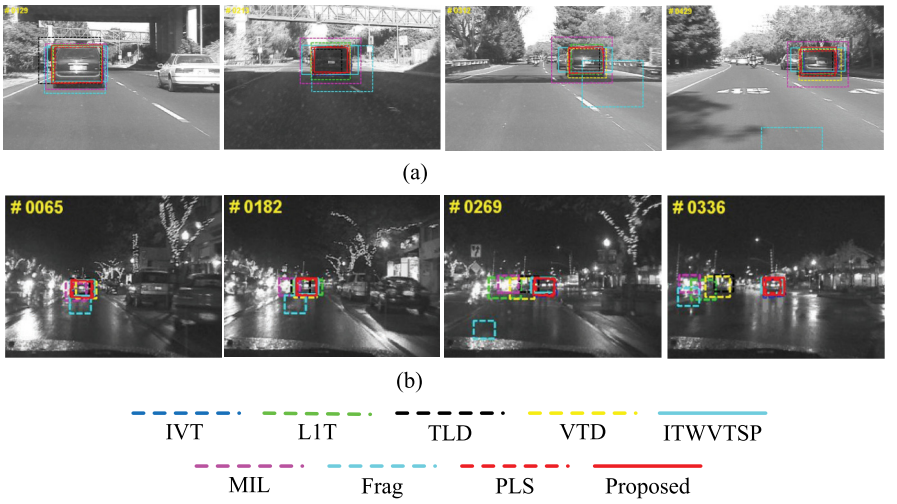

Fig. 8 Qualitative evaluation of (a) *Car 4* and (b) *Car 11*, where object appearance changes drastically due to scale change, abrupt illumination variation, cluttered background, and low contrast. Six patches are selected for the proposed algorithm.

Table 2 ACE (pixels) and average OR of tracking methods. The best two results are in bold and italics.

	Frag	IVT	VTD	L1T	TLD	MIL	ITWVTSP	PLS	Proposed
Caviar 1	5.699	45.245	*3.909*	119.932	5.593	48.499	42.342	47.393	**1.197**
	0.682	0.277	*0.834*	0.278	0.704	0.255	0.287	0.268	**0.860**
Caviar 2	5.569	8.641	4.724	*3.243*	8.514	70.269	4.569	32.431	**1.784**
	0.557	0.452	0.671	*0.811*	0.658	0.255	0.673	0.365	**0.813**
Singer	22.034	8.483	*4.057*	4.571	32.690	15.171	5.129	14.199	**4.008**
	0.341	0.662	*0.790*	0.703	0.413	0.337	0.779	0.212	**0.798**
Occlusion 1	5.621	9.175	11.135	6.500	17.648	32.260	18.855	**4.596**	*5.590*
	0.899	0.845	0.775	0.876	0.649	0.594	0.713	*0.904*	**0.907**
Occlusion 2	15.491	10.212	10.408	11.119	18.588	14.058	*9.982*	46.186	**6.631**
	0.604	0.588	0.592	*0.672*	0.493	0.612	0.588	0.471	**0.750**
David Indoor	76.691	**3.589**	13.552	7.630	9.671	16.146	14.118	64.335	*4.800*
	0.195	*0.712*	0.525	0.625	0.602	0.448	0.446	0.278	**0.761**
Deer	92.089	127.467	*11.920*	171.468	25.652	66.457	176.825	20.198	**6.116**
	0.076	0.217	*0.577*	0.039	0.412	0.213	0.024	0.510	**0.624**
Girl	*18.046*	48.474	21.442	62.435	23.158	32.209	125.698	53.368	**11.756**
	0.689	0.426	0.512	0.326	0.577	0.520	0.052	0.451	**0.735**
Car 4	179.775	**2.866**	12.290	4.081	18.797	60.104	7.831	10.163	*3.991*
	0.223	**0.922**	0.734	0.843	0.637	0.344	0.720	0.780	*0.884*
Car 11	63.922	2.106	27.055	33.252	25.113	43.465	2.066	**1.691**	*2.048*
	0.086	*0.808*	0.432	0.435	0.376	0.175	0.753	0.769	**0.812**
ACE average	51.877	35.656	*14.372*	46.776	18.542	39.864	40.450	29.456	**4.795**
AOR average	0.435	0.591	*0.644*	0.561	0.552	0.375	0.504	0.501	**0.794**

except *Car 4* are better than those of the others. Moreover, based on the ACE and AOR performance averages across all the experimental sequences, it can be concluded that the proposed performs comprehensively more favorably than the other methods. The details of the "center error" and "overlap rate" plot can be obtained from the authors.

Table 3 Computation complexity and processing time (seconds) of tracking methods.

Algorithm	Computational Complexity	Time (16×16)	Time (32×32)
IVT	$O(dM)$	0.019 s	0.074 s
ITWVTSP	$O(dM)$	0.021 s	0.076 s
LIT	$O(d^2 + dM)$	0.320 s	0.842 s
Proposed	$O(KsM + 9QM)$	0.247 s	0.535 s

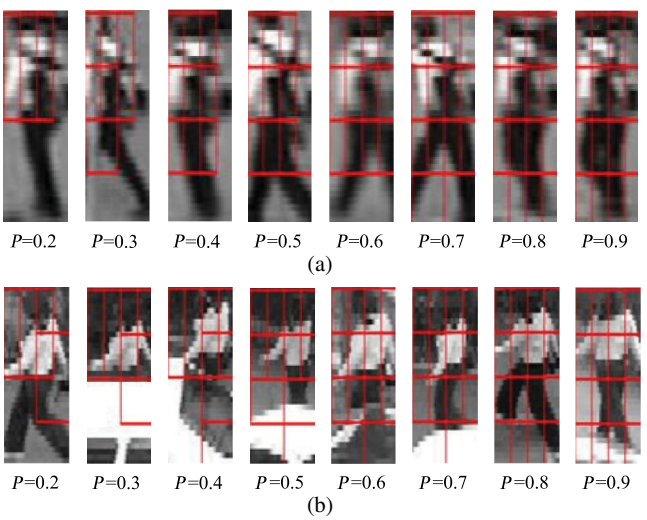

Fig. 9 Patch selection results with different P values, which vary from 0.2 to 0.9. (a) PETS2001, (b) Woman.

6.4 *Robustness on Patch Selection*

The number of selected patches is one of the key issues related to tracking performance in the proposed algorithm. An experiment is conducted to evaluate its robustness. A number selection rate P is introduced to fluctuate the selection number K, $K = \text{round}(K_0 \times P)$, where K_0 refers to the patch number without selection, and round$(\cdot)$ is the approximation function. The rate P varies from 0.2 to 0.9, while the other parameters are the same with the settings above. Two challenging sequences *PETS2001*[25] and *Woman*[11] are used.

Results of patch selection are shown in Fig. 9. Correspondingly, the CE and OR values are shown in Fig. 10. It is shown that the proposed tracker can generally follow the target with different selection rates rather than totally lose it. As P decreases, the performance does not deteriorate much. It is obvious that too limited information of the target could prevent the tracker from uniquely and successfully modeling the target's appearance, and thus the tracker fails to estimate the location with high accuracy. However, our proposed tracker could still find the likely location. Suppose a target is regarded as being successfully tracked when the OR is >0.5, a threshold line is added to the figure. Similar criterion is also applied in PASCAL VOC.[39] It is found in Fig. 10 that the proposed method is able to successfully track the target with limited selected patches, where P is not <0.4 empirically.

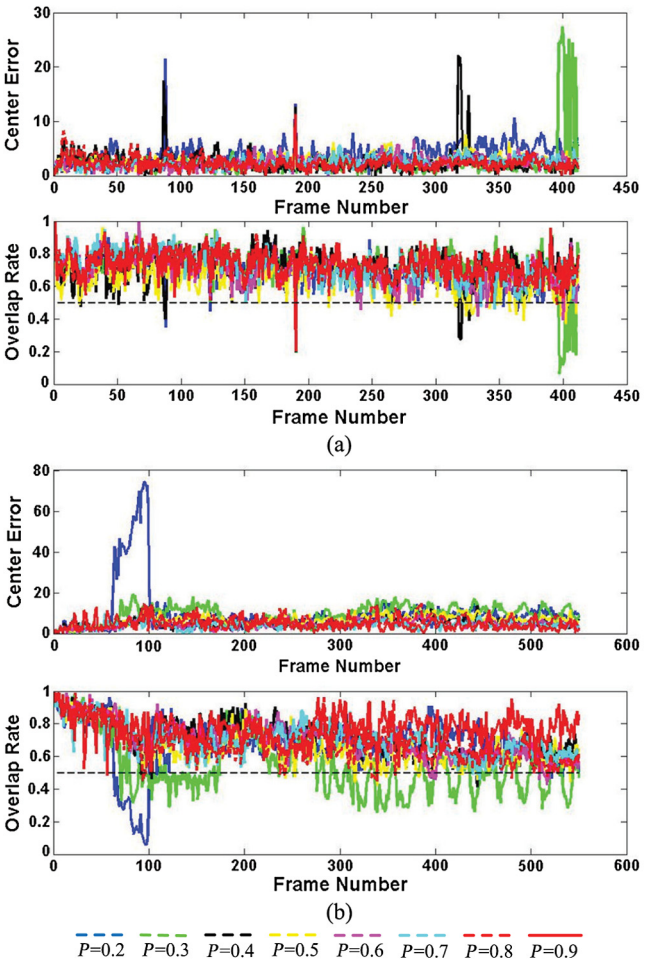

Fig. 10 Center error (CE) and overlap rate (OR) with different P values, which vary from 0.2 to 0.9. (a) PETS2001, (b) Woman.

6.5 *Comparison between Maximal and Proposed Inference Scheme*

In Sec. 5.3, a geometric inference method is proposed to locate the target. Since the final target location would not be decided by the candidate with highest confidence score but with the inference output of highest candidates in CCS, it might affect the tracking accuracy. However, we argue that the influence is quite limited, and more favorable performance compared with other works has been obtained as described above. More importantly, the proposed scheme is quite useful in cluttered background and complete occlusion environment when it is integrated with covariance variation in dynamic modeling. Heuristically, it could be viewed as a soft and local abnormality detection scheme. In cluttered background, the tracker is subject to the target's outside distraction. Under the motion continuity assumption, the proposed scheme obtains spatial cues from the most confident candidates to stabilize and centralize the location. In the complete occlusion situation, the scheme provides

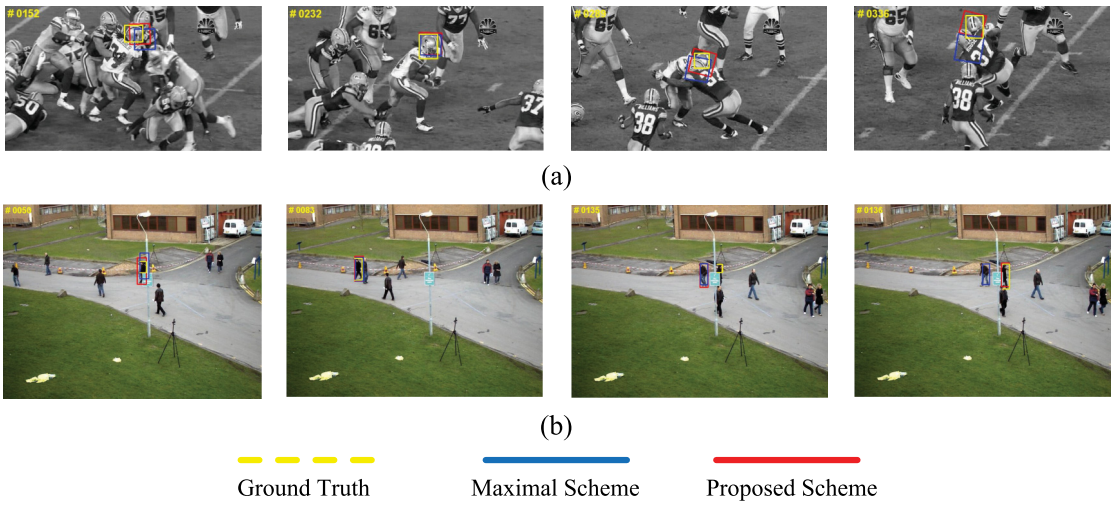

Fig. 11 Geometric inference comparison between maximal and proposed scheme. (a) Football, (b) PETS2009.

extra opportunities to detect the target in a wider area. To demonstrate such characteristics and advantages over the maximal scheme, two challenging sequences *Football*[28] and *Pets2009*[40] are used.

The selected qualitative results are shown in Fig. 11. In Fig. 11(a) on sequence *Football*, it is found that without geometric inference, the tracker gradually drifts to the surrounding areas of the player's head due to the neighborhood similarity (e.g., #0289 and #336), while more stable performance is obtained with the proposed geometric inference scheme. The sequence *Pets2009* is quite challenging; because when the target is heavily occluded, another pedestrian is passing by him. The tracker with the maximal scheme eventually follows a wrong object. In the proposed method, although the tracker mistakes the wrong pedestrian for the target in the first several frames, sparse coefficients of the false target would scatter the points distribution in the CCS, violate the inference condition, and cause the sampling state $\tilde{\mathbf{X}}_t$ to be much biased. Based on these unacceptable inference results, the searching range is extended according to Algorithm 2. When the true target appears again without much appearance variation, the tracker re-detects it and continues with correct location estimation in the following sequences.

6.6 *Computational Complexity Analysis*

In Sec. 5, it could be found that sparse coding and the proposed spatial confidence inference algorithm are most time consuming. Thus, we also compare the computation complexity and processing time with three representative trackers including IVT,[9] ITWVTSP,[10] and L1T which is show in Table 3.[19] Suppose d refers to the dimension of a vectorized image, and M is the number of eigen vectors or templates, $d \gg M$, the computational complexity of IVT[9] and ITWVTSP[10] is $O(dM)$, for they mainly involve matrix–vector multiplication. The computational load of L1T[19] is $O(d^2 + dM)$, while the load of the proposed algorithm is $O(KsM + 9QM)$. The first part is related to sparse coding, where K refers to the number of selected patches, and s is the patch size, $d^2 \gg Ks > d$. The second part is related to geometric inference, Q is inference point number, $M < 9QM \ll d$. Moreover, processing times of different normalized image sizes (16×16 and 32×32) for solving one image are also presented. It can be found that enlarging the normalized size of the target region increases the computation time. Both the L1T[19] algorithm and the proposed one apply sparse representation and yet the proposed tracker is much faster than the L1T[19] tracker. Although the proposed algorithm is slower than the IVT[9] and ITWVTSP[10] algorithm, it achieves a better performance in accuracy evaluation.

7 Conclusion

This paper presents a generative tracking algorithm based on sparse representation of selected overlapped patches via KPPR and spatiotemporal geometric inference of candidate confidences sampled by propagated affine motion modeling. Not only qualitative and quantitative evaluations but also the analysis on selected patch number and geometric inference process are conducted. The experiments demonstrate that on challenging image sequences, our proposed tracking algorithm comprehensively performs more favorably against state-of-the-art online tracking algorithms. The future work might include exploring more efficient l_1 minimization algorithms (e.g., APG)[32] for real-time application and extending this algorithm to multiple-object tracking given certain application environments. Currently, the temporal weight matrix in Eq. (10) is fixed during the tracking process. More information could be introduced for its adaption to the latest tracking conditions.

Acknowledgments

This work is supported by NSFC (No. 61171172 and No. 61102099), National Key Technology R&D Program (No. 2011BAK14B02), and STCSM (No. 10231204002, No. 11231203102 and No. 12DZ2272600). We acknowledge Dr. Javier Cruz-Mota and the other authors for sharing the source codes of evaluated trackers. We also give our sincere thanks to the anonymous reviewers for their comments and suggestions.

References

1. A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Comput. Surv.* **38**, 1–45 (2006).
2. M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," *Int. J. Comput. Vision* **29**(1), 5–28 (1998).
3. I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 809–830 (2000).
4. P. Pérez et al., "Color-based probabilistic tracking," in *Proc. 7th European Conference on Computer Vision-Part I*, pp. 661–675, Springer-Verlag, London, UK (2002).
5. Q. Wang et al., "An experimental comparison of online object-tracking algorithms," *Proc. SPIE* **8138**, 81381A (2011).
6. M. Arulampalam et al., "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Trans. Signal Process.* **50**(2), 174–188 (2002).
7. S. Salti, A. Cavallaro, and L. Di Stefano, "Adaptive appearance modeling for video tracking: survey and evaluation," *IEEE Trans. Image Process.* **21**, 4334–4348 (2012).
8. J. Shi and C. Tomasi, "Good features to track," in *1994 IEEE Conf. Computer Vision and Pattern Recognition (CVPR'94)*, pp. 593–600 (1994).
9. D. A. Ross et al., "Incremental learning for robust visual tracking," *Int. J. Comput. Vision* **77**(1–3), 125–141 (2008).
10. J. Cruz-Mota, M. Bierlaire, and J. Thiran, "Sample and pixel weighting strategies for robust incremental visual tracking," *IEEE Trans. Circuits Syst. Video Technol.* **23**(5), 898–911 (2013).
11. A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *2006 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR 2006)*, pp. 798–805, IEEE Computer Society, New York (2006).
12. M. Yang, J. Yuan, and Y. Wu, "Spatial selection for attentional visual tracking," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8 (2007).
13. Y. Zhou, H. Snoussi, and S. Zheng, "Bayesian variational human tracking based on informative body parts," *Opt. Eng.* **51**(6), 067203 (2012).
14. H. Grabner and H. Bischof, "On-line boosting and vision," in *IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Vol. 1, pp. 260–267 (2006).
15. H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. 10th European Conf. Computer Vision: Part I*, pp. 234–247, Springer-Verlag, Berlin, Heidelberg (2008).
16. B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**, 1619–1632 (2011).
17. Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(7), 1409–1422 (2012).
18. S. Wang et al., "Superpixel tracking," in *IEEE Int. Conf. Computer Vision*, pp. 1323–1330 (2011).
19. X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(11), 2259–2272 (2011).
20. B. Liu et al., "Robust tracking using local sparse appearance model and k-selection," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1313–1320 (2011).

21. H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1305–1312 (2011).
22. J. Kwon, K. M. Lee, and F. Park, "Visual tracking via geometric particle filtering on the affine group with optimal importance functions," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 991–998 (2009).
23. A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(10), 1296–1311 (2003).
24. S. K. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Trans. Image Process.* **13**(11), 1491–1506 (2004).
25. X. Mei et al., "Minimum error bounded efficient l1 tracker with occlusion detection," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1257–1264 (2011).
26. T. Bai and Y. Li, "Robust visual tracking with structured sparse representation appearance model," *Pattern Recognit.* **45**(6), 2390–2404 (2012).
27. X. Jia, H. Lu, and M.-H. Yang, "Visual tracking via adaptive structural local sparse appearance model," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1822–1829 (2012).
28. J. Kwon and K.-M. Lee, "Visual tracking decomposition," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1269–1276 (2010).
29. X. Li et al., "Visual tracking via incremental log-Euclidean riemannian subspace learning," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8 (2008).
30. J. Wright et al., "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(2), 210–227 (2009).
31. M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st ed., Springer Publishing Company, Inc., Berlin, Heidelberg (2010).
32. C. Bao et al., "Real time robust l1 tracker using accelerated proximal gradient approach," in *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1830–1837 (2012).
33. H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. R. Stat. Soc. Ser. B* **67**(2), 301–320 (2005).
34. J. Mairal et al., "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.* **11**(2), 19–60 (2010).
35. C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, MIT Press, Cambridge, MA (2005).
36. B. Efron et al., "Least angle regression," *Ann. Stat.* **32**(2), 407–499 (2004).
37. M. Xue and S. Zheng, "A bayesian online object tracking method using affine warping and random kd-tree forest," in *Multimedia and Signal Processing*, *Communications in Computer and Information Science*, F. Wang et al., Eds., Vol. 346, pp. 275–282, Springer, Berlin, Heidelberg (2012).
38. Q. Wang et al., "Object tracking via partial least squares analysis," *IEEE Trans. Image Process.* **21**(10), 4454–4465 (2012).
39. M. Everingham et al., "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision* **88**(2), 303–338 (2010).
40. J. SanMiguel, A. Cavallaro, and J. Martinez, "Adaptive online performance evaluation of video trackers," *IEEE Trans. Image Process.* **21**(5), 2812–2823.

Ming Xue received his BE degree from Shanghai University, Shanghai, China, in 2007 and his MS degree from Xidian University, Xi'an, China, in 2010. He is currently working toward the PhD degree in information and communication engineering from Shanghai Jiaotong University, Shanghai, China. From 2008 to 2009, he was a visiting student at Israel Institute of Technology (Technion), Haifa, Israel, sponsored by Chinese-Israel Governmental Exchange Scholarship. His current research interests include intelligent surveillance, multimedia systems, and machine learning.

Shibao Zheng received both the BS and MS degrees in electronic engineering from Xidian University, Xi'an, China, in 1983 and 1986, respectively. He is currently the professor and vice director of Elderly Health Information and Technology Institute of Shanghai Jiao Tong University (SJTU), Shanghai, China. His current research interests include urban image surveillance system, intelligent video analysis, spatial information system, and elderly health technology, etc.

Hua Yang received her PhD degree in communication and information from Shanghai Jiaotong University, in 2004, and both the BS and MS degrees in communication and information from Haerbin Engineering University, China, in 1998 and 2001, respectively. She is currently an associate professor in the Department of Electronic Engineering, Shanghai Jiaotong University, China. Her current research interests include video coding and networking, machine learning, and smart video surveillance.

Yi Zhou received his PhD degree in information and communication engineering from both Shanghai Jiaotong University and University of Technology of Troyes, France, in 2012, and both the BS and MS degrees in electronic engineering from Dalian Maritime University, Dalian, China, in 2003 and 2006. He is currently a lecturer in the Department of Electronics Engineering, Dalian Maritime University. His research interests include signal processing, computer vision, and machine learning.

Zhenghua Yu received his BEng and MEng from Southeast University, China, in 1993 and 1996, respectively. He received the PhD in pattern recognition and intelligent control from Shanghai Jiaotong University in 1999. He held senior research positions at Motorola Labs, Sydney, and National ICT Australia from 2000 to 2006. He is currently the chief scientist of Bocom Smart Network Technologies Inc. His current research interests include computer vision, machine learning, image and video processing, and their industrial applications.