Pattern Recognition Letters 80 (2016) 157–164

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Time Varying Metric Learning for visual tracking[☆]

Jiatong Li[a,b,*], Baojun Zhao[a], Chenwei Deng[a], Richard Yi Da Xu[b]

[a] Beijing Institute of Technology, 5 South Zhongguancun Street, Haidian District, Beijing 100081, PR China
[b] University of Technology, Sydney, 81 Broadway, Ultimo, NSW 2007, Australia

ARTICLE INFO

Article history:
Received 23 February 2016
Available online 7 July 2016

Keywords:
Metric learning
Visual tracking
Wishart process

ABSTRACT

Traditional tracking-by-detection based methods treat the target and the background as a binary classification problem. This two class classification method suffers from two main drawbacks. Firstly, the learning result may be unreliable when the number of training samples is not large enough. Secondly, the binary classifier tends to drift because of the complex background tracking conditions. In this paper, we propose a new model called Time Varying Metric Learning (TVML) for visual tracking. We adopt the Wishart Process to model the time varying metrics for target features, and apply the Recursive Bayesian Estimation (RBE) framework to learn the metric from the data with "side information contraint". Metric learning with side information is able to omit the clustering of negative samples, which is more preferable in complex background tracking scenarios. The recursive Bayesian model ensures the learned metric is accurate with limited training samples. The experimental results demonstrate the comparable performance of the TVML tracker compared to state-of-the-art methods, especially when there are background clutters.

1. Introduction

Visual object tracking is a typical research topic in computer vision that has been widely applied in many areas, such as video surveillance, intelligent traffic and human computer interaction. Although much work has been done on visual tracking, it is still a very challenging problem, and the tracking results may be influenced by many factors including background clutters, appearance changes, illumination variations and abrupt motions.

Over the last few years, a tracking approach called tracking-by-detection (TBD) is proposed which achieves excellent performance [30]. TBD methods have attracted much attention since they are robust to appearance variations, and have great discriminative capacity. The TBD methods usually learn the appearance of an object by training an online binary classifier, where a discriminative model is adopted in the tracking procedure [2,3,11,12,36]. Another main concern of tracking methods is the model update problem. Many effective model update mechanisms have been proposed, such as incremental subspace update [24], online boosting [8,9,21], PN learning [15] and structured SVM [35].

Most of the above existing tracking methods employ a fixed, pre-specified metric during the entire tracking process. For instance, Euclidean metric is most commonly used for feature comparison and chi-square distance for calculating histogram distances. If we can obtain features that are discriminative enough from the distracters, the result is satisfactory in many instances. However, under many circumstances, we often observe that a candidate with the closest match by using pre-specified metric do not always turns out to be the true target-of-interest. Thus finding an appropriate metric is necessary.

Metric learning based tracking methods aim to learn the appropriate metric to better handle the classification between object and the background [14,17,27]. Tsagkatakis and Savakis [27] combine the online metric learning method with nearest neighbor classification to boost the tracking performance. Jiang et al. [14] propose an adaptive metric learning tracking method, where its goal is to find the best extended nearest classifier to maximize the expected number of training data that are correctly classified. Traditional metric learning based methods treat the object and background as binary classification. However, in the real tracking scenarios, there is no need to exert constraint on the entire background patches as one negative class. Imagine that there are two or more background distracters which are distant apart from each other in feature space, then we must learn a classifier that is capable of distinguishing the target from mixture of clusters in the negative class. To avoid this unnecessary complication, in this paper, we

[☆] "This paper has been recommended for acceptance by Dr. D. Dembele.".

[*] Corresponding author at: University of Technology, Sydney, 81 Broadway, Ultimo, NSW 2007, Australia.

E-mail address: Jiatong.Li-3@student.uts.edu.au, ljt_bit@bit.edu.cn (J. Li).

http://dx.doi.org/10.1016/j.patrec.2016.06.017
0167-8655/© 2016 Elsevier B.V. All rights reserved.

use the side information that presents a set of pairwise constraints on training data: equivalence constraints that include pairs of “similar” data and inequivalence constraints that include pairs of “dissimilar” data, which can omit background clustering.

Our key motivation is that, in a tracking scenario, it is unnecessary to assume the classes of background. Therefore, it is desirable to provide the so-called “side information” to indicate which data are “similar” or “dissimilar”. This side information based metric learning was first introduced in [31]. Furthermore, Yang et al. [33] extend the work and prove that it performs effectively in the Bayesian learning framework with limited training data. In this paper, we introduce the side information based metric learning method into visual tracking, and show that it can effectively promote the tracking performance.

Our main contribution is that we propose a new time varying metric learning model and its Sequential Monte Carlo (SMC) solution for visual tracking. Our method introduces the Wishart Process to model the time varying metric transition and adopts the side information constraint to train the model in a dynamic setting, which provides an appropriate means of defining pair-wise distance between feature samples. In addition, we present a Recursive Bayesian Estimation (RBE) framework to estimate the metric, which can achieve effective learning result with limited training data.

2. Related work

Visual object tracking is one of the traditional research topics in computer vision. During decades of evolution, numerous methods have been proposed. The recent survey and tracking benchmark further promote the development of the field [20,25,30,34].

Many trackers train an online binary classifier to distinguish the object from background. One representative method is Support Vector Tracker [2], which uses the idea of SVM combined with optical flow to enhance the performance. Hare et al. [10] further extent the work to Structured SVM to learn the samples with structured labels, which achieves promising result. Babenko et al. [3] introduce multiple instance learning to collect positive and negative samples into bags to learn a discriminative classifier, so as to overcome the drift problem. Zhang et al. [36] adopt the random projection method to reduce the feature dimension which achieves real-time tracking.

In addition to the binary classifier method. Incremental subspace learning and boosting methods are also introduced to the online tracker. Ross et al. [24] propose an incremental learning method for object tracking based on PCA representation. This method can efficiently learn and update a low-dimensional subspace which is composed by PCA eigenvectors. Boosting-based appearance models have been widely used for visual tracking [8,9,15,21] due to their efficient discriminative learning capabilities. Practically, discriminative haar-like features are selected, and weak classifiers are correspondingly generated and pooled together as a strong classifier for object location. Grabner et al. [8] demonstrate that an online boosting-based feature selection and classification method can improve tracking performance dramatically than off-line classifier based algorithms. In [37], an efficient instance probability optimization based feature selection scheme was exploited for better tracking performance with low computational complexity.

Multiple models or trackers are further adopted to obtain more robust performance. Wei et al. [28] propose to combine generative model with discriminative model to jointly handle complex tracking conditions. Kwon and Lee [18] enrich the Bayesian tracking model to multiple state so as to adapt to complicated tracking scenarios. Kalal et al. [16] address the long term tracking problem by using dual experts to handle the positive and negative distracters, which achieves excellent result. In [19,35], multiple trackers are integrated to form a tracker ensemble to enhance the tracking performance.

Our work inherits the advantages of tracking-by-detection methods but with great differences. Firstly, our method trains the samples with side information constraint, which focuses on discriminating the object from background by omitting the negative samples clustering. Secondly, unlike the gradient based metric learning methods [14,27], our method estimates the metric accurately with limited samples in the Bayesian framework, which is more suitable to the visual tracking.

The rest of the paper is organized as follows: in Section 3, we first introduce the Wishart Process and side information constraint, and then propose our Time Varying Metric Learning model with its SMC solution. In Section 4, we explain how to apply the proposed model to visual tracking. In Section 5, model validation is conducted on synthetic data, then the proposed tracker is evaluated in the 50 sequences tracking benchmark. Finally, we conclude the paper.

3. Proposed model

In this paper, we aim to learn a distance metric to calculate the distance between two feature vectors $\mathbf{x}$ and $\mathbf{y}$, which can be defined as:

$$d_M(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_M = \sqrt{(\mathbf{x}-\mathbf{y})^T M(\mathbf{x}-\mathbf{y})}, \tag{1}$$

where M is a positive semi-definite (PSD) matrix.

3.1. Wishart process

Wishart Process [23] is a stochastic process which is able to generate a sequence of random PSD matrices M_1, ..., M_t over the time. According to the definition of Wishart Process, the relationship between M_t and M_{t-1} is defined as:

$$\begin{aligned} &M_t|\nu, S_{t-1} \sim Wishart(\nu, S_{t-1}) \\ &where \quad S_{t-1} = \nu(\frac{1}{\nu}A^{\frac{1}{2}})(M_{t-1})^d(\frac{1}{\nu}A^{\frac{1}{2}})^T, \end{aligned} \tag{2}$$

where M_t is the PSD matrix at time t, and $Wishart(\nu, S_{t-1})$ is the Wishart distribution parameterized by ν and S_{t-1}, which are the number of degrees of freedom and the time-dependent scale parameter respectively. A is a positive definite symmetric parameter matrix that is decomposed by Cholesky decomposition as $A = (A^{\frac{1}{2}})(A^{\frac{1}{2}})^T$. d is the scalar parameter.

Wishart Process is able to model the dynamic behavior of a set of PSD matrices across time. The scale parameter S_t not only defines the time variation of the PSD matrix but also ensures the proposed matrices are positive definite. The parameters A and d control the variation behavior of the PSD matrices. A is interpreted as revealing how each element of PSD matrix depends on the previous PSD matrix. While parameter d denotes the overall strength of the metric evolution relationship. d is proved to be theoretically bound between $(-1, 1)$ [23]. And in practice, d is usually within (0, 1). More details about Wishart Process and its parameter interpretation is referred to [23,29].

3.2. Side information constraint

As mentioned above, we aim to learn a metric to better distinguish positive (target) samples from negative (background) samples by putting no constraint to the negative training data. To achieve this goal, we propose to use pair-wise data constraint instead of treating all the training data as two classes. As a result, by setting the similar pair-wise constraint to positive data, and the dissimilar constraint between pair-wise positive and negative data,

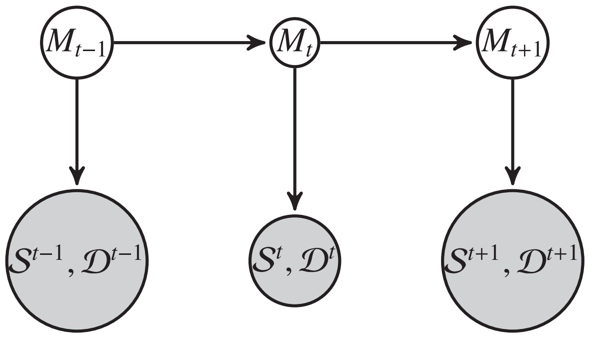

M_{t-1}
M_t
M_{t+1}
$\mathcal{S}^{t-1}, \mathcal{D}^{t-1}$
$\mathcal{S}^{t}, \mathcal{D}^{t}$
$\mathcal{S}^{t+1}, \mathcal{D}^{t+1}$

Fig. 1. Time Varying Metric Learning Graphical Model.

we can omit the negative clusters and put all the energy in discriminating the positive data from the negative. Therefore, given a metric, it is necessary to define a probability to describe the similar and dissimilar relationship for data pairs.

Motivated by [33], the probability for two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ to form a similar or dissimilar set under a given distance metric M can defined as:

$$P(c_{i,j}|\mathbf{x}_i, \mathbf{x}_j, M, \mu) = \frac{1}{1 + \exp(c_{i,j}(\| \mathbf{x}_i - \mathbf{x}_j \|_M^2 - \mu))} \qquad \text{where} \quad c_{i,j} = \begin{cases} +1 & (\mathbf{x}_i, \mathbf{x}_j) \in S \\ -1 & (\mathbf{x}_i, \mathbf{x}_j) \in D, \end{cases} \tag{3}$$

where S and D denote the similar and dissimilar set respectively. μ is the threshold, which means that two data are more likely to be in the same constraint set when their distance is less than the threshold μ. Note the above formula is a discrete distribution, whose sum over random variable $c_{i,j} \in \{+1, -1\}$ equals 1. Assuming each pair of the training data is independent, the whole likelihood probability for all the constraints in S and D is define as:

$$P(S, D|M, \mu) = \prod_{(i,j)\in S} \frac{1}{1 + \exp(\| \mathbf{x}_i - \mathbf{x}_j \|_M^2 - \mu)} \times \prod_{(i,j)\in D} \frac{1}{1 + \exp(-(\| \mathbf{x}_i - \mathbf{x}_j \|_M^2 - \mu))} \tag{4}$$

3.3. Graphical model for time varying metric learning

Unlike most existing metric learning frameworks which learn the data from a single time instance, we introduce a Bayesian framework to learn the metric recursively. Given a set of labelled “similar” and “dissimilar” constraints for training data at each time step t, we propose the Time Varying Metric Learning (TVML) graphical model as illustrated in Fig. 1. The TVML likelihood probability $P(S_t, D_t|M_t)$ is defined as Eq. (4), whereas the transition probability $P(M_t|M_{t-1})$ is naturally defined as the Wishart Process shown in Eq. (2).

The proposed TVML model is capable of estimating the metric behaviour that is well suited for distinguishing the similar data from the dissimilar ones while minimizing the sum of distances in the similar set.

At each time step, the prior and the likelihood of the model interacts. Because the prior carries forward its state information from the previous time step, TVML needs only a few data to estimate the current state M_t.

3.4. The sequential Monte Carlo solution

After the TVML model is a established, our aim is to estimate metric M_t at each time step given all the previous equivalence and inequivalence knowledge, i.e. $P(M_t|S_{1:t}, D_{1:t})$.

The above can be viewed as a filtering problem, we use the Sequential Monte Carlo method (SMC) [7,26] to solve it. According to SMC method, the posterior distribution is represented by the following Monte Carlo approximation:

$$\hat{P}(M_t|S_{1:t}, D_{1:t}) \propto \sum_{i=1}^{N} P(S_t, D_t|\hat{M}_t^i)\delta_{\hat{M}_t^i}(m_t) = \sum_{i=1}^{N} w_t^i \delta_{\hat{M}_t^i}(m_t) \tag{5}$$

where “ ˆ ” is the estimation symbol and the particles weighted by $P(S_t, D_t|\hat{M}_t^i)$ are used to approximate the posterior at each time t. The proof is detailed in the supplementary material. In every recursive step, we use the transition $P(M_t|M_{t-1})$ as the proposal to propagate the particles. Finally we conduct the resampling step to overcome the degeneracy problem.

As shown in [7], when the variance of the particles is small, the resampling step might be unnecessary. Therefore, in this paper, the SMC solution conducts the resampling step only when the variance of weights is larger than the pre-specified threshold. This is assessed by the so-called Effective Sample Size (ESS): $ESS = \left(\sum_{i=1}^{N}(w_t^i)^2\right)^{-1}$. The ESS varies between 1 and N and resampling is triggered only when it is below a threshold N_T, typically $N_T = N/2$ [7].

4. Metric learning for tracking

In this section, we show how to apply the proposed TVML model into visual tracking. We first describe the proposed tracking framework, and then introduce the target representation method and the online model update scheme. Finally, the algorithm flow is summarized.

4.1. Tracking framework

As shown in Fig. 2, the whole tracking process is divided into three main steps: training, testing and model update.

Training: Given the previous target location, a set of image patches are collected as training data. Image patches close to the target area are considered as positive samples, while the far away image patches are treated as negative samples. This collection method is quite useful especially for overcoming the target distracting problem. As the learned metric can well distinguish the true target from the distracters before they interact or occlude. After collecting the positive and negative image patches, feature extraction is conducted to all the patches, which is discussed in the Section 4.2. Then the extracted feature vectors are fed into the TVML model to learn the metric for testing.

The key point of training step lies in the training constraint. Unlike traditional methods that only exert two class label to the training data, our method utilize the side information that constrains each pair of the positive samples as similar, i.e., having smaller distance, and each pair of positive and negative samples as dissimilar, i.e., having larger distance. The method does not place any constraints over pairs of negative samples, as it concentrates its entire effort in distinguishing the target from background.

Testing: Instead of searching for the target with smallest distance to the previous object region, we adopt a collection of previous target appearances to form a template library for online model [27], so as to adapt to the object appearance variation. Using the template library and the learned metric, the location of the target for the current frame is obtained by locating the region that has the smallest distance with the template library.

Finally, we update the model as described in Section 4.2.

4.2. Target representation and model update

Target Representation, also known as feature extraction, is a critical step in visual tracking. In most trackers, low level features,

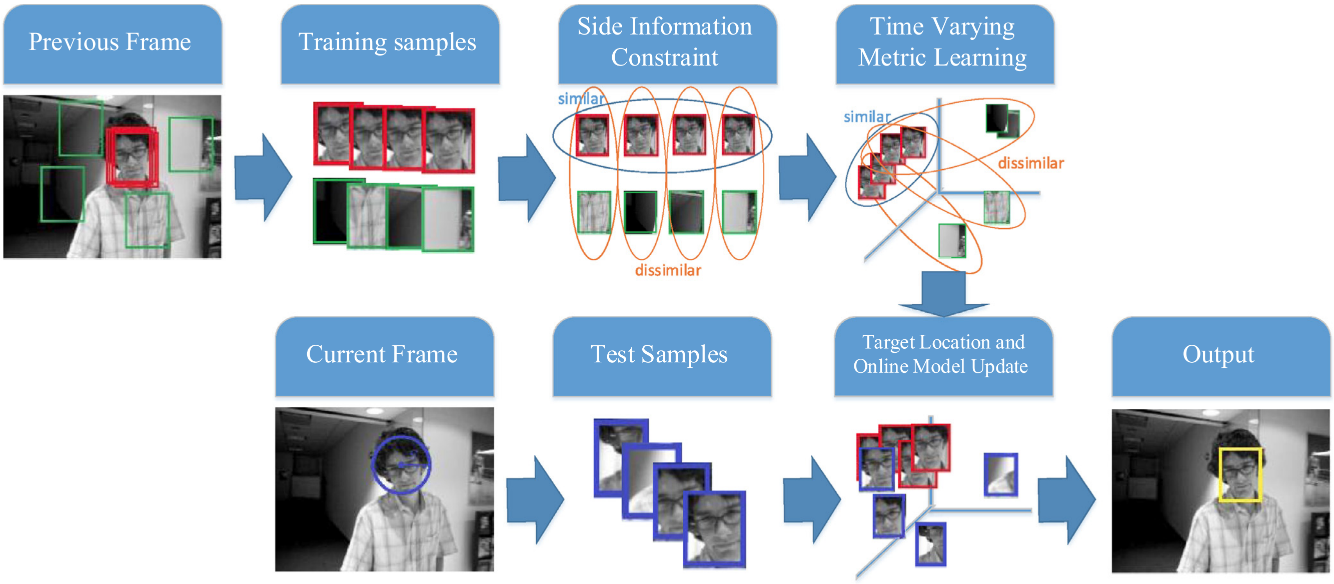


Fig. 2. TVML for tracking. There are three main steps: training (the flows in the first row), testing and model update (the flows in the second row). The training step collect the training samples and learn the metric. The testing step then use the learned metric to search the current target location followed by the step of online model update.

such as Haar-like features or gray-scale features are adopted [3,20,36]. To handle the target appearance more effectively, we adopt the dense Scale Invariant Feature Transform (SIFT) [5] to represent the target. SIFT feature is famous for its scale and rotation invariance, and it is also invariant to small changes of the object appearance. It has been further shown that extracting SIFT feature on the regular grid, also called dense SIFT, is capable of outperforming that obtained by keypoint detection [32]. In this paper, we adopt the dense SIFT feature to represent image patches.

The bottleneck of the dense SIFT feature is its computation load because of the relatively high dimension, since each grid point on the image patch will generate a 128-dimension feature vector. To overcome this problem, we adopt random projection as in [27] to reduce the feature dimension. Random Projection (RP) is a data-independent method for dimension reduction, which do not needs the prior knowledge of the object appearance.

To tackle the target appearance variation issue, we introduce an online model to adapt to appearance changes similar to [27]. The online model maintains a template library consisting of previous target appearances, and update the library by replacing old templates with new ones if necessary. Let the feature vector of the image patch denoted by $\mathbf{x}$, and the template library containing feature vectors of the past target appearances denoted by $\mathbf{L} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots\}$. The distance between a target candidate $\mathbf{x}$ and the template library is defined as the minimum distance of $\mathbf{x}$ to each of the template element in $\mathbf{L}$ given the metric M, i.e. $d = \min_{\mathbf{x}_i \in \mathbf{L}} d_M(\mathbf{x}, \mathbf{x}_i)$.

More details of the above strategies can be referred to [1,27].

Based on the above discussion, the algorithm flow is summarized as follows.

Algorithm 1: TVML Tracker.

```
input : Initial target state $\mathbf{s}_1$
output: Estimated target state $\mathbf{s}_t = (\hat{x}_t, \hat{y}_t)$
repeat
    Given target state $\mathbf{s}_{t-1} = (\hat{x}_{t-1}, \hat{y}_{t-1})$, collect training data from frame $t-1$;
    Set the side information constraint to $S_{t-1}$ and $D_{t-1}$ according to Section 4.1;
    Learn metric $M_{t-1}$ using SMC as indicated in Eq. (5);
    Update the online model $\mathbf{L}_{t-1}$ according to Section 4.2;
    Estimate the target state $\mathbf{s}_t = (\hat{x}_t, \hat{y}_t)$ in frame $t$ using metric $M_{t-1}$ and online model $\mathbf{L}_{t-1}$;
until Last frame of video sequence;
```

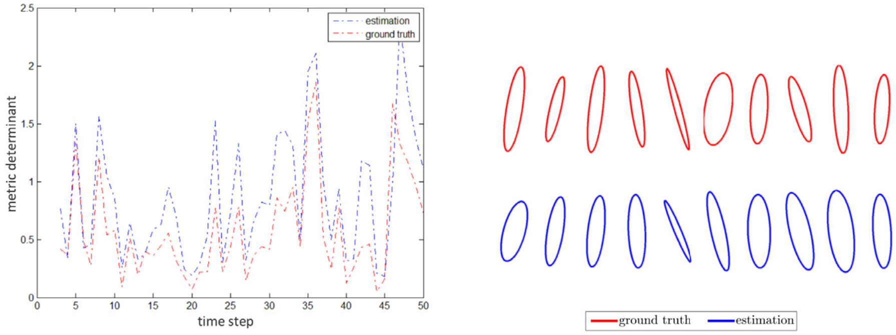
5. Experiments

The whole experiment is divided into two parts. Firstly, we conduct the model validation using the synthetic data. Secondly, we evaluate our proposed model in the 50 sequences tracking benchmark [30], and compare its performance with the state-of-the-art methods. The experimental platform is on a 3.20GHz CPU with 8GB RAM.

5.1. Model validation

In order to demonstrate the effectiveness of our TVML model, we first use the synthetic data to validate that it is able to catch various metrics. We generate synthetic data by sampling the likelihood probability shown in Eq. (4). For efficiency and simplicity, we choose the Metropolis-Hasting sampling method [22]. To validate the generalization of our model, we set the parameters of Wishart Process for ground truth and test model differently, where the parameters of the former are $\nu = 5, d = 0.3, A = I$, and those of the latter are $\nu = 9, d = 0.5, A = I * 0.8$ (I is the identity matrix). In order to illustrate the experimental results clearly, we set the data dimension to 2, so as to plot the covariance matrix as the ellipse [4,29].

Fig. 3(a) shows the determinant of the metric for both the ground truth and its estimation, and Fig. 3(b) shows the randomly selected time-dependent elliptical representations, which is more accurate to illustrate the estimation and ground truth. It can be seen that the trend of estimated metric determinant is very close to the ground truth even their parameters are very different. The figure for elliptical representation further illustrates the estimation accuracy of the mode. The above results demonstrate



(a) metric determinant

(b) elliptical representation

Fig. 3. TVML on synthetic data.

that TVML can capture the metric both from the trend and precision. In addition, the estimated matrix determinant in Fig. 3(a) is overall larger than the ground truth because of the monotonicity of the likelihood function. It is worth noting that the likelihood monotonicity is more reasonable since it is better for the distances among the S set smaller and D set larger.

5.2. Experiment for visual tracking

In this section, the proposed tracker is evaluated on the 50 sequences tracking benchmark [30]. The compared methods are the 27 trackers provided by the benchmark. In the following sections, we firstly describe the implementation details of the proposed TVML tracker and the evaluation criteria, then report the experimental results for further evaluation.

5.2.1. Experiment setup

In this paper, we extract dense SIFT feature in the resized $45*45$ image patch with $2*2$ features spread in the patch uniformly. The feature dimension of one image patch is then $128*4=512$. We then use RP to reduce its dimension to 128.

As indicated in Section 3.1, the parameter A and d play an important role in determining the dynamic behavior of the covariance structure. The parameter d indicates the overall strength of matrix evolution. d close to 0 means a weak overall effect of the current volatility on future values, and d close to 1 indicates high persistence. In tracking scenario, the target and background changes are relatively continuous and stable. Therefore, we make a trade-off to set $d=0.5$. Parameter A reveals how each element of the PSD matrix depends on the elements of the previous PSD matrix. As it's hard to predict the weight relation between the elements of the feature vectors, we do not give too much pre-defined constraint and set $A=I*1.2$. Therefore, the parameters of the Wishart Process are set to $d=0.5, A=I*1.2, \nu=4.5,$ where we find the setting of ν has little influence of the performance in practice. The parameter μ of likelihood is evaluated from the first frame, which is set to the mean of the median value of S set and D set. The number of the positive and negative training examples are 15 and 20 respectively. For the SMC solution, the particles are initialized as the identity matrix and are propagated according to the Wishart Process. Resampling is conducted when the ESS is below the threshold of $N/2$, where N is the number of particles. In this paper, N is set to 200. To keep a rich representation for the object, the template library contains five templates elements. To tackle the target occlusion issue, we set a pre-defined threshold ξ. If $d_t > \xi$, then the target occlusion is detected, and the model update pauses temporarily to prevent bad update. The algorithm is implemented in Matlab & C. The code has not been optimized and runs at roughly $1\sim3$ s per frame at the current state.

Two performance evaluation metrics are used in the experiment: Overlap Success Rate (OSR) and Center Location Error (CLE). The OSR is defined as $score=\frac{ROI_T \bigcap ROI_G}{ROI_T \bigcup ROI_G}$, where ROI_T is the area of the estimated bounding box and ROI_G is area of the ground truth bounding box. CLE is the center distance between the tracking result and the ground truth bounding box.

5.2.2. Experimental results

Overall performance. The overall performance of the benchmark is illustrated in two plots: precision plot and success plot. The former plot shows the percentage of frames whose CLE is within the given threshold distance. The latter plot shows the ratios of successful frames at the thresholds from 0 to 1. The successful frames are counted if the OSR is larger than 0.5.

The overall performance is shown in Fig. 4. For clear illustration, only the top-10 trackers are shown in the plots. From the plots, it is shown that TVML ranks first in both the evaluation methods, and TLD [16] ranks the second. Our method has improved by 7.6% and 7.1% for the precision and success measure respectively compared to the second rank tracker. Note that VTD [18], VTS [19] and ASLA [13] are all based on Bayesian framework, which demonstrate the effectiveness of our tracker. Among the compared trackers, VTD and VTS use the image intensity, color and shape as the combined feature. TLD adopts patch template as features. It demonstrates to some extent that the high level dense SIFT feature performs better than the low level features.

The computation load of the top-10 trackers for overall performance are shown in Table 1. From the table, it is observed that FPS of sampling based method have higher computation load than grid search methods. Particularly, CSK runs much faster than the other methods since it is based on Fast Fourier Transform (FFT). In addition, TLD ranks second in FPS because the whole algorithm is based on gray scale feature, which needs less computations than those of high level features. Our method is mainly based on SMC sampling, and it has relatively the same computation load than the other sampling based trackers (Note the model solver is based on sampling, and the search scheme is grid search). Moreover, during the experiments, it is observed that the dense SIFT computation takes about 40% time in our whole algorithm. Taken the whole computation load into consideration, we draw conclusion that the proposed method has normal computation complexity among the sampling based methods.

Attribute-based performance. All the sequences in the benchmark are then divided into four groups according to their attributes, including background clutter, occlusion, deformation and scale variation. The attributes experimental results are shown in

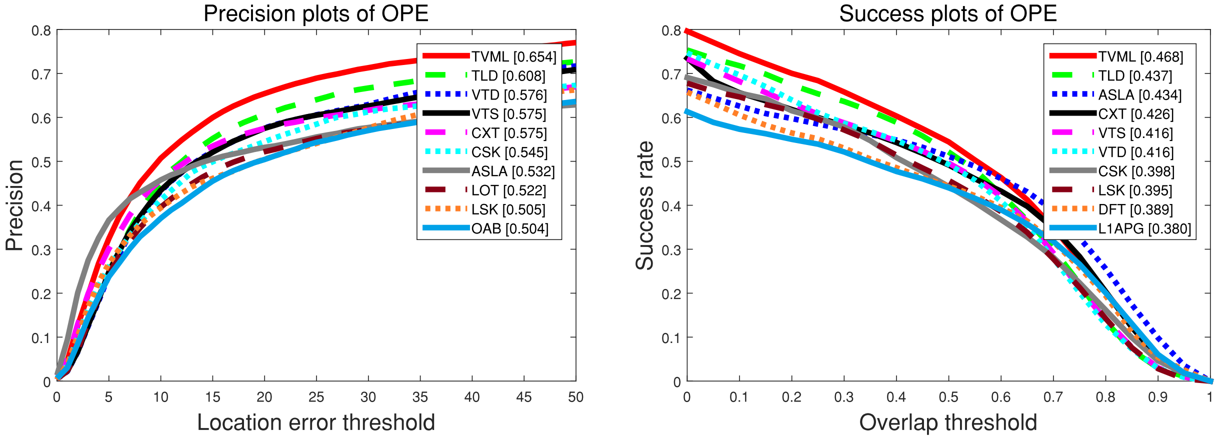


Fig. 4. The precision plot and success plot over 50 sequences benchmark using one pass evaluation (OPE). The legend illustrates the score of the threshold 20 for the precision plot, and the area under curve (AUC) for the success plot. Legend of the same color denotes the same rank.

Table 1
Computation loads of the top-10 trackers in Fig. 4 are presented in three aspects, including frames per second (FPS), tracking method and the implementation. For method, S: Sampling based method, GS: Grid search based method. For implementation, M: Matlab, MC: Matlab + C, E: executable code.

	TVML	TLD	VTD	VTS	CXT	CSK	ASLA	LOT	LSK	OAB
FPS	2.56	25.74	2.75	2.72	10.67	268.45	3.48	0.47	0.072	5.13
Method	S	GS	S	S	GS	GS	S	S	GS	GS
Implementation	MC	MC	MC (E)	MC (E)	C	MC	MC	M	M (E)	C

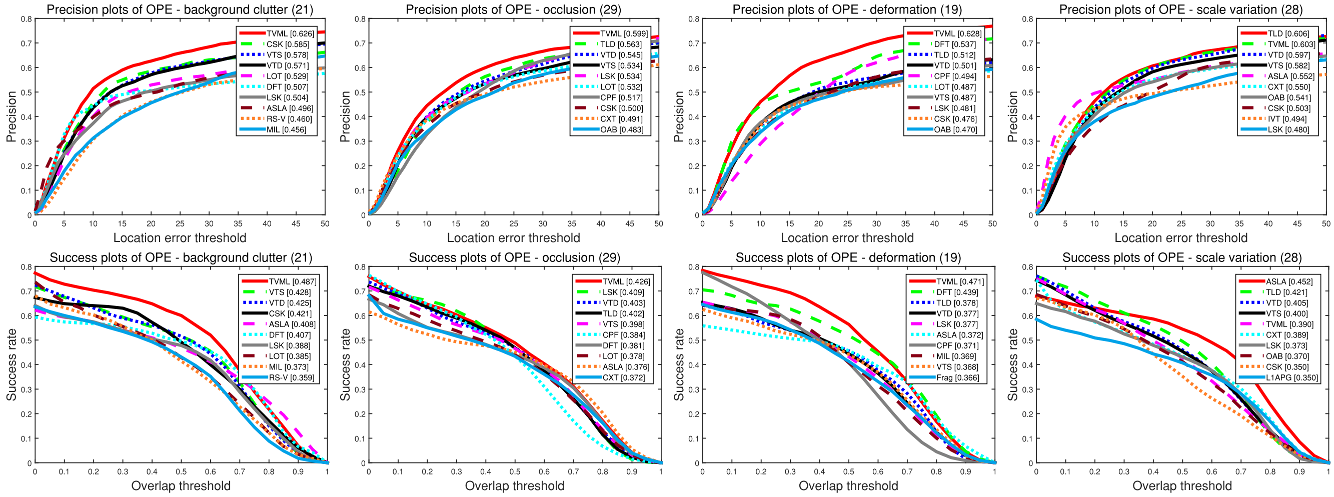


Fig. 5. The precision plots and success plots for four main attributes of the benchmark, i.e. background clutter, occlusion, deformation and scale variation. The legend illustrates the score of threshold 20 for precision and AUC score for success rate of each tracker. The same color indicates the same rank.

Fig. 5. In accordance with our assumption, one of the most effective scenarios handled by TVML tracker should be background clutter. From the plots in Fig. 5, it is illustrated that both the precision and the success rate of the proposed method rank first among the compared trackers. To be more specific, in the background clutter attribute sequences, TVML has improved the precision and success score by 7% and 12.8% respectively relative to the second tracker, which demonstrate our method is very effective in handling complex background clutters. In addition, in the other two attributes of occlusion and deformation, TVML also outperforms against other methods. In the scale attribute sequences, TVML ranks second in the precision measure, while ranks fifth in the success rate metric. This is reasonable since all of the trackers of TLD [16], ASLA [13], VTD [18] and VTS [19] have scale estimation. Therefore, their success scores are higher. However, only TLD outperforms TVML in scale attribute measured by precision metric. This is because the robustness of dense SIFT feature extracted on the image grid.

Lastly, the snapshots of some typical sequences of the whole benchmark are shown in Fig. 6. Among the eight sequences, *Basketball*, *Football* and *Singer2* go through severe background clutters. TVML performs well in the above sequences, which demonstrate the proposed metric learning framework is capable of handling these background clutter scenarios. *Bolt*, *Football* and *SUV* have similar target distracters. In the *Bolt* sequence, most of the trackers drift at the last few frames while TVML can deal with such appearance changes. This also demonstrate our online model mechanism is effective. At the last snapshot for *Football*, most of the trackers drift to the nearby distracter, but our method is able to catch the true target due to the online metric learning strategy. The same conclusion can be draw from *David*, *SUV* and *Jogging*. In

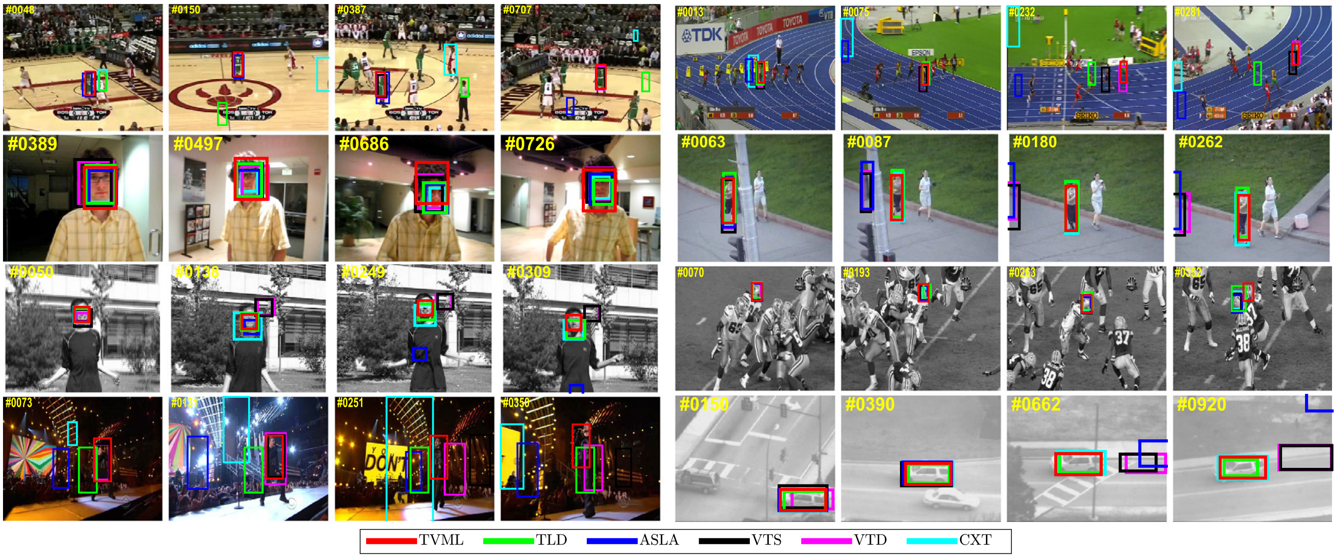


Fig. 6. Tracking snapshots of the top six algorithms among the overall performance including TVML, TLD [16], ASLA [13], VTS [19], VTD [18], CXT [6] over eight sequences. The illustration example videos from top-left to bottom-right are *Basketball, Bolt, David, Jogging, Jumping, Football, Singer2 and SUV*.

practice, we find that in most cases, the proposed tracker can catch the target even with partial occlusion. This is because of the robustness of the dense SIFT feature as well as the online model.

6. Conclusion and discussion

This paper proposes a novel Time Varying Metric Learning model for object tracking. The proposed model is under the Recursive Baysian Estimation framework, which is capable of learning the metric with limited number of training data. Wishart Process is introduced as the transition model to capture the dynamic metrics under side information constraint. Our model is suitable but not limited to visual tracking application. The future work is to further learn the parameters of Wishart Process, and extend our model to other non-tracking applications.

Acknowledgments

This project is supported by International Graduate Exchange Program of Beijing Institute of Technology.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.patrec.2016.06.017

References

[1] D. Achlioptas, Database-friendly random projections: Johnson-lindenstrauss with binary coins, J. Comput. Syst. Sci. 66 (4) (2003) 671–687.
[2] S. Avidan, Support vector tracking, IEEE Trans. Pattern Anal. Mach. Intell. 26 (8) (2004) 1064–1072.
[3] B. Babenko, M.H. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1619–1632.
[4] C.M. Bishop, Pattern Recognition and Machine Learning, springer, 2006.
[5] A. Bosch, A. Zisserman, Image classification using random forests and ferns, in: Proceedings of the IEEE International Conference on Computer Vision(ICCV), 2007.
[6] T.B. Dinh, N. Vo, G. Medioni, Context tracker: Exploring supporters and distracters in unconstrained environments, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1177–1184.
[7] A. Doucet, A.M. Johansen, A tutorial on particle filtering and smoothing, Handbook of Nonlinear Filtering, 12, Oxford University Press, 2009, pp. 656–704.
[8] H. Grabner, M. Grabner, H. Bischof, Real-time tracking via on-line boosting, Proceedings of British Machine Vision Conference (BMVC) 1 (2006) 47–56.
[9] H. Grabner, C. Leistner, H. Bischof, in: Semi-supervised on-line boosting for robust tracking, 2008, pp. 234–247.
[10] S. Hare, A. Saffari, P.H. Torr, Struck: Structured output tracking with kernels, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 263–270.
[11] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels, in: Proceedings of the European Conference on Computer Vision (ECCV), 2012, pp. 702–715.
[12] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 583–596.
[13] X. Jia, H. Lu, M.H. Yang, Visual tracking via adaptive structural local sparse appearance model, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1822–1829.
[14] N. Jiang, W. Liu, Y. Wu, Learning adaptive metric for robust visual tracking, IEEE Trans. Image Process. 20 (8) (2012) 2288–2300.
[15] Z. Kalal, J. Matas, K. Mikolajczyk, Pn learning: bootstrapping binary classifiers by structural constraints, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 49–56.
[16] Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (7) (2012) 1409–1422.
[17] B. Kulis, Metric learning: a survey, Found. Trends Mach. Learn. 5 (4) (2012) 287–364.
[18] J. Kwon, K.M. Lee, Visual tracking decomposition, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 1269–1276.
[19] J. Kwon, K.M. Lee, Tracking by sampling and integrating multiple trackers, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2014) 1428–1441.
[20] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, A.V.D. Hengel, A survey of appearance models in visual object tracking, ACM Trans. Intell. Syst. Technol. (TIST) 4 (4) (2013) 58.
[21] X. Liu, T. Yu, Gradient feature selection for online boosting, in: Proceedings of IEEE International Conference on Computer Vision (ICCV), 2007.
[22] K.P. Murphy, Machine Learning: A Probabilistic Perspective, MIT press, 2012, pp. 848–855.
[23] A. Philipov, M.E. Glickman, Multivariate stochastic volatility via wishart processes, J. Bus. Econ. Stat. 24 (3) (2006) 313–328.
[24] D.A. Ross, J. Lim, R.S. Lin, M.H. Yang, Incremental learning for robust visual tracking, Int. J. Comput. Vis. 77 (1–3) (2008) 125–141.
[25] A.W.M. Smeulders, D.W. Chu, R. Cucchiara, S. Calderara, A. Denhghan, M. Shah, Visual tracking: an experimental survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2014) 1442–1468.
[26] A. Smith, A. Doucet, N.D. Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer Science & Business Media, 2013, pp. 79–86.
[27] G. Tsagkatakis, A. Savakis, Online distance metric learning for object tracking, IEEE Trans. Circ. Syst. Video Technol. 21 (12) (2011) 1810–1821.
[28] W. Wei, H. Lu, M.H. Yang, Robust object tracking via sparsity-based collaborative model, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 1838–1845.
[29] A.G. Wilson, Z. Ghahramani, Generalised wishart processes, in: Proceedings of Uncertainty in Artificial Intelligence (UAI), 2010, pp. 736–744.
[30] Y. Wu, J. Lim, M.H. Yang, Online object tracking: A benchmark, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2411–2418.

[31] E.P. Xing, M.I. Jordan, S. Russell, A.Y. Ng, Distance metric learning with application to clustering with side-information, in: Advances in neural information processing systems (NIPS), 2003, pp. 505–512.

[32] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 1794–1801.

[33] L. Yang, R. Jin, R. Sukthanar, Bayesian active distance metric learning, in: Proceedings of Uncertainty in Artificial Intelligence (UAI), 2012. 442–229

[34] A. Yilmaz, O. Javed, M. Shah, Object tracking: a survey, ACM Comput. Surv. (CSUR) 38 (4) (2006) 13.

[35] J. Zhang, S. Ma, S. Sclaroff, Meem: robust tracking via multiple experts using entropy minimization, in: Proceedings of the European Conference on Computer Vision (ECCV), 2014, pp. 188–203.

[36] K. Zhang, L. Zhang, M.H. Yang, Real-time compressive tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2012, pp. 864–877.

[37] K. Zhang, L. Zhang, M.H. Yang, Real-time object tracking via online discriminative feature selection, IEEE Trans. Image Process. 22 (12) (2013) 4664–4677.