Graph mode-based contextual kernels for robust SVM tracking

Xi Li[†], Anthony Dick[†], Hanzi Wang[◦], Chunhua Shen[†‡], Anton van den Hengel[†]
[†]School of Computer Sciences, University of Adelaide, Australia
[‡]NICTA*, Canberra Research Laboratory, Australia
[◦]Center for Pattern Analysis and Machine Intelligence, Xiamen University, China, 361005

Abstract

Visual tracking has been typically solved as a binary classification problem. Most existing trackers only consider the pairwise interactions between samples, and thereby ignore the higher-order contextual interactions, which may lead to the sensitivity to complicated factors such as noises, outliers, background clutters and so on. In this paper, we propose a visual tracker based on support vector machines (SVMs), for which a novel graph mode-based contextual kernel is designed to effectively capture the higher-order contextual information from samples. To do so, we first create a visual graph whose similarity matrix is determined by a baseline visual kernel. Second, a set of high-order contexts are discovered in the visual graph. The problem of discovering these high-order contexts is solved by seeking modes of the visual graph. Each graph mode corresponds to a vertex community termed as a high-order context. Third, we construct a contextual kernel that effectively captures the interaction information between the high-order contexts. Finally, this contextual kernel is embedded into SVMs for robust tracking. Experimental results on challenging videos demonstrate the effectiveness and robustness of the proposed tracker.

1. Introduction

Recently, visual tracking has attracted much research attention. It remains a challenging problem due to issues such as complicated appearance and illumination change, occlusion, cluttered background etc. To build a robust tracker, a variety of appearance models using different learning techniques have been proposed in the literature. According to the learning techniques, these appearance models may be roughly classified into two categories: generative learning based and discriminative learning based appearance models. Generative learning based appearance models (GLMs) mainly concentrate on how to construct robust object representation in specified feature spaces, including the integral histogram [1], kernel density estimation [3], spatial-color mixture of Gaussians [2], subspace learning [18, 19, 20], sparse representation [4, 23], visual tracking decomposition [17] and so on. A drawback of these methods is that they often ignore the influence of background, and consequently suffer from distractions caused by the background regions with similar appearance to foreground objects.

In contrast, discriminative learning based appearance models (DLMs) aim to maximize the inter-class separability between the object and non-object regions using discriminative learning techniques, including SVMs [7, 8, 21], boosting [5, 6], random forest [14], multiple instance learning [9], spatial attention learning [13], etc. Most of these DLMs have used the pairwise interaction information from samples for object/non-object classification, and consequently ignore the higher-order contextual interaction information (e.g., the interaction between two contexts in Fig. 1(d)). As a result, they may achieve unstable or unsuccessful tracking performance since the pairwise interactions are easily corrupted by complicated factors such as noises, outliers, background clutters and so on. In this paper, we show that the contextual information can play an important role in DLMs based tracking.

Motivation Typically, the pairwise similarity of two samples is based only on the individual samples themselves. Once one sample is corrupted by noise, their pairwise similarity will change significantly, so their true affinity cannot be computed in a stable way. Thus, designing a robust similarity measure is a key issue in visual tracking.

In this paper, we show that the high-order contextual information from samples can help to alleviate this issue. Usually, a high-order context is defined as a group of samples having some common properties. Each sample in the high-order context is influenced by other samples in the same high-order context. In this case, the similarity measure depends on not only two individual samples but also their corresponding contexts. Thus, although the pairwise interaction between two individual samples is sensitive to noise, the interaction (illustrated in Fig. 1(d)) between their

*NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Center of Excellence program.

978-1-4577-1102-2/11/$26.00

high-order contexts is much more stable because it considers more cross-link information from their high-order contexts. Even if the pairwise similarity is corrupted, the high-order contextual interaction can still provide complementary information to counteract the impact of noise.

Here we propose a robust tracker that is based upon a graph mode-based contextual kernel for SVM tracking. Our main contributions are three-fold.

1. We introduce the high-order context into the visual tracking process. The high-order context of a sample is defined as a set of samples with similar visual content to the sample. Moreover, we design a contextual similarity measure (defined in Eq. (7)) between two high-order contexts to capture their interaction information.
2. The problem of building the high-order context is converted to that of graph mode seeking, which can automatically discover the modes (i.e., dense subgraphs) of a given visual graph characterized by a baseline visual kernel. According to the information provided by the graph modes, we can find that the graph vertexes belonging to the same vertex community have some common visual properties, as illustrated in Fig. 2. Such vertex communities correspond to *high-order contexts*.
3. We design a novel contextual kernel that fully combines the information from the baseline visual kernel and the high-order contexts. The contextual kernel takes a large value when samples share not only the similar visual content but also the mutually correlated high-order contexts. We also prove that the designed contextual kernel is a Mercer kernel. Therefore, we can naturally embed the contextual kernel into an SVM for robust tracking.

Related work *Discriminative learning based tracking* An online AdaBoost classifier [5] is developed for discriminative feature selection, which enables the tracker to adapt to appearance variations caused by out-of-plane rotations and illumination changes. Later, Grabner *et al.* [6] present a semi-supervised online boosting algorithm for tracking. This algorithm can significantly alleviate the model drifting problem caused during updating the model for the online AdaBoost classifier. Avidan [10] constructs a confidence map by pixel-wise classification using an ensemble of online learned weak classifiers, and mean shift is used to locate the mode of the confidence map. Collins *et al.* [11] propose online feature selection for robust tracking. They try to find the most discriminative linear combination of the RGB color channels at each frame. Liu and Yu [12] propose an efficient online boosting algorithm based on gradient-based feature selection for pedestrian tracking. Babenko *et al.* [9] present a tracking system based on online multiple instance boosting. Their tracker is able to update the appearance model with a set of image patches, which is robust but can lose accuracy if none of the patches precisely capture the object appearance information.

Avidan [7] proposes an off-line SVM-based tracking algorithm for distinguishing a target vehicle from backgrounds. Since the algorithm needs many labeled training data, extending the algorithm to general object tracking is difficult. Tian *et al.* [8] utilize the ensemble of linear SVM classifiers for visual tracking. These classifiers can be adaptively weighted according to their discriminative abilities during different periods. Tang *et al.* [21] present an online semi-supervised learning based tracker. The method constructs two feature-specific SVM classifiers in a co-training framework, and thus is capable for improving each individual classifier using the information from other features. However, all of these tracking algorithms do not take the contextual information into account.

Context-aware tracking Yang *et al.* [22] propose a context-aware tracking algorithm, which considers a set of auxiliary objects in the tracking process. As the context of the target, these auxiliary objects need to satisfy the following three conditions: (a) persistent co-occurrence with the target; (b) consistent motion correlation to the target; and (c) easy to track. However, these conditions may not be easily satisfied in practice.

2. The proposed visual tracker

The workflow of the proposed visual tracker is listed in Algorithm 1. For better illustration, we elaborate the important components of the proposed visual tracker in this section, including contextual kernel design and training sample selection.

Contextual kernel design For discriminative learning-based visual tracking, a large amount of unlabeled samples in each frame can provide rich useful contextual information for object/non-object classification. Figs. 1 (a) and (b) illustrate that the contextual information from unlabeled samples may have a great influence on the SVM learning results. Hence, it is necessary to consider the influence of both labeled and unlabeled samples in the process of designing a contextual kernel for SVM classification. Inspired by this, we design a contextual kernel as follows.

First, we introduce some notation used hereinafter. Let $\mathbb{Z} = \{\mathbf{z}_i\}_{i=1}^{N}$ denote a sample set, $\mathbb{Z}_l = \{\mathbf{z}_i\}_{i=1}^{\gamma}$ denote the labeled sample subset of $\mathbb{Z}$, $\mathbb{Y}_l = \{y_i\}_{i=1}^{\gamma}$ denote the corresponding label set of $\mathbb{Z}_l$ for $y_i \in \{-1, +1\}$, and $\mathbb{Z}_u = \{\mathbf{z}_i\}_{i=\gamma+1}^{N}$ denote the unlabeled sample subset of $\mathbb{Z}$. Based on $\mathbb{Z} = \{\mathbf{z}_i\}_{i=1}^{N}$, we create a visual graph G with N vertexes. Mathematically, the graph G can be denoted as $G = (V, E, W)$, where $V = \{v_i\}_{i=1}^{N}$ is the vertex set corresponding to $\{\mathbf{z}_i\}_{i=1}^{N}$, $E \subseteq V \times V$ is the edge set, and W is the edge-weight function returning the affinity value between two vertexes. In practice, the graph G is formulated

Algorithm 1 Contextual kernel-based SVM tracking

Input: Frame t, the object state $\mathbf{L}_{t-1}^{*}$ in the frame $t-1$, previous labeled sample set $\mathbb{Z}_l$ from previous observed frames, positive maximum buffer size $\mathbb{T}^{+}$, and negative maximum buffer size $\mathbb{T}^{-}$.

1: Sample a number of candidate object states $\mathcal{L}_t = \{\mathbf{L}_t^i\}$ using the particle filters (referred to [18]).
2: Crop out the corresponding normalized image regions of $\mathcal{L}_t$ by affine warping.
3: Extract the corresponding HOG feature set $\mathbb{Z}_u = \{\mathbf{z}_i\}$.
4: Construct the graph mode-based contextual kernel CK^* from (12) using the data set $\mathbb{Z} = \{\mathbb{Z}_l, \mathbb{Z}_u\}$.
- Compute the baseline visual kernel K from (4).
- Build a visual graph G whose similarity matrix A is defined in (3).
- Solve the optimization problem (5) to seek a set of graph modes $\{\mathcal{M}_i\}_{i=1}^{Q}$ by graph shift.
- Calculate the contextual affinity matrix $\mathcal{C}$ from (9).
- Compute the contextual kernel CK^* from (12).

5: Train an SVM classifier $h(\mathbf{z})$ from (14) over $\mathbb{Z}_l$ by solving the optimization problem (13).
6: Determine the optimal object state $\mathbf{L}_t^*$ by the MAP (maximum a posterior) estimation in the particle filters, where the observation model is defined as:

$$p\left(\mathbf{z}_i|\mathbf{L}_t^i\right) \propto \frac{1}{1+\exp\left(-\mu h(\mathbf{z}_i)\right)} \tag{1}$$

where μ is a scaling factor.
7: Extract the corresponding positive and negative support vector sets of $h(\mathbf{z})$, i.e., $\mathbb{S}^{+}$ and $\mathbb{S}^{-}$.
8: Select positive samples $\mathbb{Z}_t^{+}$ and negative samples $\mathbb{Z}_t^{-}$ from $\mathbb{Z}_u$.
9: Update the labeled sample set $\mathbb{Z}_l$ with $\mathbb{Z}_l^{+} \bigcup \mathbb{Z}_l^{-}$, where $\mathbb{Z}_l^{+} = \mathbb{S}^{+} \bigcup \mathbb{Z}_t^{+}$ and $\mathbb{Z}_l^{-} = \mathbb{S}^{-} \bigcup \mathbb{Z}_t^{-}$.
- If $|\mathbb{Z}_l^{+}| > \mathbb{T}^{+}$, then $\mathbb{Z}_l^{+}$ is truncated to keep the last $\mathbb{T}^{+}$ elements occurring recently.
- If $|\mathbb{Z}_l^{-}| > \mathbb{T}^{-}$, then $\mathbb{Z}_l^{-}$ is truncated to keep the last $\mathbb{T}^{-}$ elements occurring recently.

10: **return** The object state $\mathbf{L}_t^*$ and the updated labeled sample set $\mathbb{Z}_l$.

as a weighted similarity matrix $A = (a_{ij})_{N\times N}$:

$$a_{ij} = \begin{cases} W(v_i, v_j) & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $W(v_i, v_j) = K(\mathbf{z}_i, \mathbf{z}_j)$ with $K(\cdot,\cdot)$ being a baseline visual kernel function. Since G is a graph without self-loops, $A = (a_{ij})_{N\times N}$ is a matrix whose diagonal elements are all zeros. As a result, the similarity matrix $A = (a_{ij})_{N\times N}$ can be reformulated as:

$$a_{ij} = \begin{cases} K(\mathbf{z}_i, \mathbf{z}_j) & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The above graph creation is independent of the choice of kernel functions. It is easy to incorporate various kernel functions into the above graph creation process. In our case, the Gaussian RBF kernel function is used as the baseline visual kernel, which evaluates the visual similarity between two image regions. Furthermore, each image region is represented as a HOG feature descriptor (referred to [16]) in the five spatial block-division modes (like [20]). More specifically, given two image regions, we extract their corresponding HOG feature vectors $\mathbf{z}_i = (h_\ell^i)_{\ell=1}^r$ and $\mathbf{z}_j = (h_\ell^j)_{\ell=1}^r$, where r is the HOG feature dimension. The Gaussian RBF kernel function $K(\mathbf{z}_i, \mathbf{z}_j)$ is formulated as:

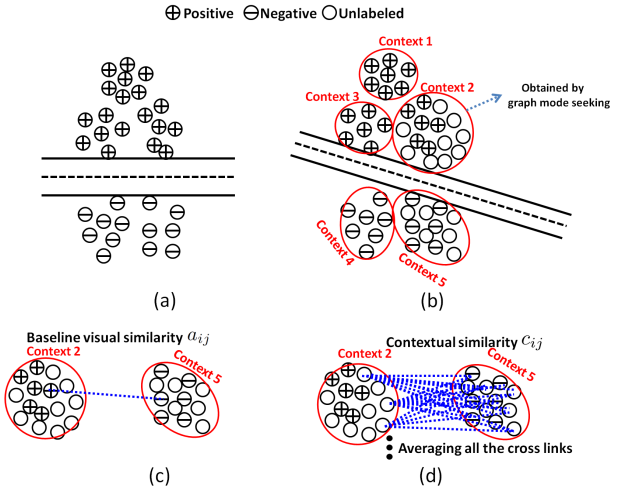

Figure 1: Illustration of the proposed contextual SVM learning. (a) shows the SVM classification boundary without using the contextual information from samples; (b) shows the SVM classification boundary using the contextual information of samples; (c) depicts the process of computing the baseline visual similarity a_{ij} (see Eq. (3)); and (d) illustrates the process of computing the contextual similarity c_{ij} (see Eq. (7)).

$$K(\mathbf{z}_i, \mathbf{z}_j) = \exp\left(-\beta \sum_{\ell=1}^{r} (h_\ell^i - h_\ell^j)^2\right). \tag{4}$$

where β is a scaling factor. Since $K(\mathbf{z}_i, \mathbf{z}_j)$ is equal to $K(\mathbf{z}_j, \mathbf{z}_i)$, a_{ij} is equal to a_{ji}. Consequently, the weighted similarity matrix A of the graph G is a symmetric matrix in our case.

Given the graph G defined above, how to effectively capture the useful contextual information from G plays a vital role in contextual kernel design. Due to the robustness to noises and outliers (as claimed in [15]), a dense subgraph of G, that is a coherent subset of vertexes in a graph, can be used as a contextual information carrier. In the dense subgraph, the vertexes are cohesively interconnected so that their intrinsic properties are similar to each other. Consequently, each vertex in the dense subgraph is greatly influenced by the other vertexes in the same dense subgraph. Thus, the dense subgraph is essentially a context providing contextual information to its own vertexes. In graph theory, the cohesiveness of a subset of vertexes in the graph G is measured by a graph density whose local maxima correspond to the modes of the graph G. Mathematically, the graph density is formulated as: $g(x) = \sum_{i,j} a_{ij} x_i x_j = x^T A x$, where x indicates a probabilistic cluster of vertexes, and the i-th component x_i of x reflects the probability of v_i belonging to this cluster. To seek the modes of the graph G, we need to optimize the following quadratic programming problem:

$$\begin{array}{ll} \text{maximize} & g(x) = x^T A x \\ \text{s.t.} & x \in \triangle^N \end{array} \tag{5}$$

where $\triangle^N = \{x \in \mathcal{R}^N | x \geq 0 \text{ and } \|x\|_1 = 1\}$. The optimization problem (5) can be efficiently solved by graph

shift [15][1]. More specifically, let $\{\mathcal{M}_i\}_{i=1}^{Q}$ be the Q graph modes sought by graph shift, and $\mathcal{M}_i$ be the vertex index set of the i-th graph mode. Without loss of generality, we suppose the size of $\mathcal{M}_i$ is q_i. Consequently, $\mathcal{M}_i$ is equivalent to $\{\mathbb{I}_\ell^i\}_{\ell=1}^{q_i}$ where $\mathbb{I}_\ell^i$ indexes the ℓ-th vertex of the i-th graph mode. Based on $\mathbb{I}_\ell^i$, we introduce an $N \times 1$ graph mode vector $\mathbf{s}_i$ whose element is defined as:

$$\mathbf{s}_i(n) = \begin{cases} 1 & \text{if } n \in \{\mathbb{I}_\ell^i\}_{\ell=1}^{q_i} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $n \in \{1, 2, \ldots, N\}$. In this case, the affinity value between the i-th and j-th graph modes can be measured by the average similarity between these two graph modes (as illustrated in Fig. 1(d)):

$$c_{ij} = \frac{1}{q_i q_j} \sum_{m \in \{\mathbb{I}_\ell^i\}_{\ell=1}^{q_i}} \sum_{n \in \{\mathbb{I}_\ell^j\}_{\ell=1}^{q_j}} a_{mn} \tag{7}$$

For simplicity, Eq. (7) can be transformed into its matrix form:

$$c_{ij} = \frac{1}{q_i q_j} \mathbf{s}_i^T A \mathbf{s}_j = \frac{1}{q_j q_i} \mathbf{s}_j^T A^T \mathbf{s}_i = \frac{1}{q_j q_i} \mathbf{s}_j^T A \mathbf{s}_i = c_{ji} \tag{8}$$

In this case, we have a cross-mode affinity matrix $\mathbb{C}_{ij} \in \mathcal{R}^{N \times N}$ formulated as: $\mathbb{C}_{ij} = c_{ij} \mathbf{s}_i \mathbf{s}_j^T$. Thus, we can obtain all the cross-mode affinity matrices: i.e., $\{\{\mathbb{C}_{ij}\}_{i=1}^{Q}\}_{j=1}^{Q}$. By taking the average of these cross-mode affinity matrices, a unified contextual affinity matrix $\mathcal{C}$ is computed to evaluate the total contextual similarity between any two vertexes:

$$\mathcal{C} = \frac{1}{Q^2} \sum_{i=1}^{Q} \sum_{j=1}^{Q} \mathbb{C}_{ij} = \frac{1}{Q^2} \sum_{i=1}^{Q} \sum_{j=1}^{Q} c_{ij} \mathbf{s}_i \mathbf{s}_j^T \tag{9}$$

Based on the contextual affinity matrix $\mathcal{C}$ and the Gaussian RBF kernel (4), a contextual kernel CK is designed as follows:

$$CK(\mathbf{z}_i, \mathbf{z}_j) = K(\mathbf{z}_i, \mathbf{z}_j) \exp(\mathcal{C}(\mathbf{z}_i, \mathbf{z}_j)) \tag{10}$$

For simplicity, Eq. (10) can be transformed into its matrix form: $CK = K \circ \exp(\mathcal{C})$, where $\circ$ is the element-wise matrix multiplication operator and $\exp(\cdot)$ is the element-wise exponent operator. To avoid singularity, the term $\exp(\mathcal{C})$ can be replaced with $\exp(\mathcal{C}) + \xi I_N$, where I_N is an $N \times N$ identity matrix and ξ is a control variable which is determined by:

$$\xi = \begin{cases} -\lambda^* & \text{if } \lambda^* < 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where λ^* is the minimum eigenvalue of $\exp(\mathcal{C})$. So we have the final contextual kernel CK^*:

$$CK^* = K \circ (\exp(\mathcal{C}) + \xi I_N) \tag{12}$$

Proposition 2.1. *The contextual affinity matrix $\mathcal{C}$ is a symmetric matrix.*

[1] http://sites.google.com/site/lhrbss/

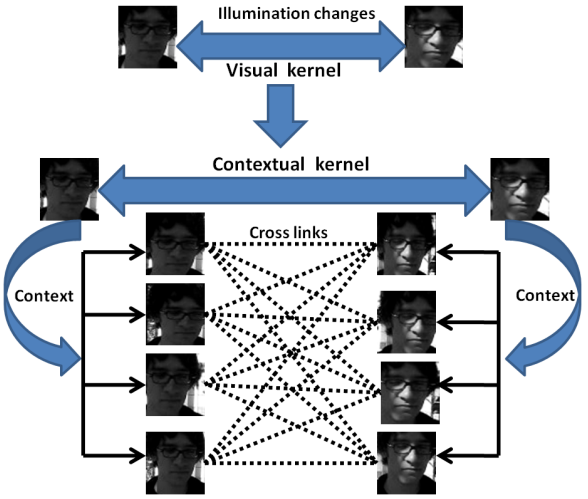

Figure 2: An example of illustrating the differences between the visual kernel and the proposed contextual kernel. The upper part corresponds to the visual kernel while the lower part is associated with the proposed contextual kernel, which captures the contextual interaction relationships (i.e., cross links).

Proof. The mathematical derivation is formulated as:

$$\mathcal{C}^T = \frac{1}{Q^2} \sum_{i=1}^{Q} \sum_{j=1}^{Q} c_{ij} \mathbf{s}_j \mathbf{s}_i^T = \frac{1}{Q^2} \sum_{i=1}^{Q} \sum_{j=1}^{Q} c_{ji} \mathbf{s}_j \mathbf{s}_i^T$$
$$= \frac{1}{Q^2} \sum_{i=1}^{Q} \sum_{j=1}^{Q} \mathbb{C}_{ji} = \frac{1}{Q^2} \sum_{i=1}^{Q} \sum_{j=1}^{Q} \mathbb{C}_{ij} = \mathcal{C}$$

□

Proposition 2.2. *The contextual kernel CK^* is a Mercer kernel.*

Proof. For simplicity, the term $(\exp(\mathcal{C}) + \xi I_N)$ is abbreviated as $f_{\mathcal{C}}$. Since $\mathcal{C}$ is a symmetric matrix, $f_{\mathcal{C}}$ is also a symmetric matrix:

$$f_{\mathcal{C}}^T = \exp(\mathcal{C}^T) + (\xi I_N)^T = \exp(\mathcal{C}) + \xi I_N = f_{\mathcal{C}}$$

As a positive semi-definite matrix, $f_{\mathcal{C}}$ can be decomposed into $U^T \Lambda U = (U \Lambda^{\frac{1}{2}})^T (U \Lambda^{\frac{1}{2}})$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_N)$ and $U = (\mathbf{u}_i)_{i=1}^N$ correspond to the eigenvalues and eigenvectors of $f_{\mathcal{C}}$, respectively. As a result, $f_{\mathcal{C}}(\mathbf{z}_i, \mathbf{z}_j)$ can be rewritten as the inner product of two terms: $\phi(\mathbf{z}_i)^T \phi(\mathbf{z}_j)$, where $\phi(\mathbf{z}_\ell)$ $(1 \leq \ell \leq N)$ is a kernel mapping such that $\phi(\mathbf{z}_\ell) = (\sqrt{\lambda_1}\mathbf{u}_1(\ell), \ldots, \sqrt{\lambda_N}\mathbf{u}_N(\ell))^T$. Therefore, $f_{\mathcal{C}}$ is a Mercer kernel. As is known to us, the Gaussian RBF kernel K is also a Mercer kernel. Being the element-wise product of two Mercer kernels K and $f_{\mathcal{C}}$, CK^* is consequently a Mercer kernel. □

Fig. 2 gives an example of illustrating the differences between the visual kernel K and the proposed contextual kernel CK^*. Specifically, the face regions of the same person in two different frames have different visual properties due to illumination changes. Their visual kernel K is only determined by the visual appearance properties of the two face regions, as shown in the upper part of Fig. 2. However, the proposed contextual kernel CK^* considers not only the visual affinity relationship between the two face regions but also the contextual interaction relationships (i.e., cross links) between their corresponding contexts, as shown in the lower part of Fig. 2.

Based on the contextual kernel CK^*, we can derive a standard SVM optimization problem over the labeled sample set $\mathbb{Z}_l$. Mathematically, the dual form of the SVM optimization problem can be formulated as:

$$\begin{array}{ll}\max\limits_{\boldsymbol{\alpha}} & \sum_{i=1}^{\gamma}\alpha_i - \frac{1}{2}\sum_{i=1}^{\gamma}\sum_{j=1}^{\gamma} y_i y_j \alpha_i \alpha_j CK^*(\mathbf{z}_i, \mathbf{z}_j) \\ \text{s.t.} & \sum_{i=1}^{\gamma} y_i \alpha_i = 0;\ 0 \le \alpha_i \le D \end{array} \quad (13)$$

where $y_i \in \{-1, 1\}$, γ is the size of $\mathbb{Z}_l$ and D is a regularization factor. After solving the optimization problem (13) using the libsvm tool, an SVM classification function is obtained as:

$$h(\mathbf{z}) = \sum_{i=1}^{\gamma} \alpha_i y_i CK^*(\mathbf{z}, \mathbf{z}_i) + b. \quad (14)$$

In real tracking applications, the cardinality of $\mathbb{Z}_l$ can become larger and larger as new labeled samples from different frames are added to $\mathbb{Z}_l$. For computational efficiency, we have to discard some older samples and retain the recently added samples in the SVM training process. Specifically, we define a threshold $\mathbb{T}^+$ (or $\mathbb{T}^-$) to limit the positive (or negative) sample size of $\mathbb{Z}_l$. If exceeding $\mathbb{T}^+$ (or $\mathbb{T}^-$), the positive (or negative) sample size will be reduced to only retain the last $\mathbb{T}^+$ (or $\mathbb{T}^-$) elements occurring recently. Algorithm 1 gives more details.

Training sample selection In the proposed tracker, we take a conventional strategy for training sample selection. Namely, the image regions from a small neighborhood around the object location are selected as positive samples, and the negative samples are generated by selecting the image regions which are relatively far from the object location. Specifically, an ascending sort for the samples from $\mathbb{Z}_u$ is made according to their spatial distances to the current object location, resulting in a sorted sample set $\mathbb{Z}_u^s$. By selecting the top (or bottom) 10% of the samples from $\mathbb{Z}_u^s$, we have a subset denoted as $\mathbb{Z}_t^+$ (or $\mathbb{Z}_t^-$) , which is the final positive (or negative) sample set. When $t = 1$ (i.e., in the first frame), the object location is manually labeled. $\mathbb{Z}_l$ is equal to $\mathbb{Z}_1^+ \bigcup \mathbb{Z}_1^-$.

3. Experiments

Video sequences We evaluate the proposed contextual SVM tracker on six challenging videos, which are composed of 8-bit grayscale images. All of the six videos are captured with moving cameras in different scenes.

Video 1 A car runs fast in a dark road scene with background clutters and varying lighting conditions.

Video 2 A car runs in a highway with many shadows (caused by bridges or trees). When the car crosses a bridge, its appearance drastically changes because of the shadow disturbance. Furthermore, the pose of the car gradually changes over time.

Video 3 A man walks under a treillage. Meantime, several events take place simultaneously, including drastic illumination changes and head pose variations.

Video 4 A girl rushes along a pavement. Her appearance varies with significant scale changes. In particular, she collides with a man in the middle of the video, and then her body drastically rotates.

Video 5 There is an ice hockey match. During the match, there exist several events such as partial occlusions, out-of-plane rotations, body pose variations, abrupt motion and so on.

Video 6 Several skaters are dancing in a dark scene with illumination changes. Their appearances drastically vary due to several factors such as partial occlusions, body pose variations, illumination changes and so forth.

Implementation details Six experiments on the abovementioned challenging six videos are conducted to demonstrate the advantages of the proposed tracker using the graph mode-based contextual kernel, referred to as CKST. The proposed CKST is implemented in Matlab on a workstation with an Intel Core 2 Duo 2.66GHz processor and 3.24G RAM. The average running time of the proposed CKST is about two seconds per frame. The main computational time of CKST is spent at Steps 3-5 in Algorithm 1. In practice, the parameters $\mathbb{T}^+$ and $\mathbb{T}^-$ in Algorithm 1 are both set to 500. For the sake of computational efficiency, we only consider the object state information in 2D translation and scaling in the particle filtering module. The particle number at Step 1 in Algorithm 1 is set to 200. The scaling factor μ defined in Eq. (1) is set to 1. The above parameter settings remain the same in all the experiments.

We also compare the performance of CKST against six other state-of-the-art trackers. Specifically, the six competing trackers are referred to as SVMWC (SVM without using the contextual information), MIBT (multiple instance boosting-based tracker [9]), VTD (visual tracking decomposition [17]), OAB (online AdaBoost [5]), IPCA (incremental PCA [18]), and L1T (ℓ_1-minimization tracker [4]). We use the source code of MIBT[2], VTD[3], OAB[4], IPCA[5], and L1T[6] from their websites.

The reasons for selecting the six competing trackers are as follows. First, SVMWC is close to CKST while SVMWC does not use the contextual information. More specifically, SVMWC directly uses the Gaussian RBF kernel (4) for SVM classification while CKST uses the contextual kernel (12) for SVM classification. Thus, the purpose of comparing CKST with SVMWC is to verify the superiority of CKST using the contextual information over SVMWC. Second, MIBT is a recently proposed discriminant learning-based tracker, which uses multiple instance boosting for object/non-object classification. To deal with

[2] http://vision.ucsd.edu/∼bbabenko/project_miltrack.shtml
[3] http://cv.snu.ac.kr/research/∼vtd/
[4] http://www.vision.ee.ethz.ch/boostingTrackers/download.htm
[5] http://www.cs.utoronto.ca/∼dross/ivt/
[6] http://www.ist.temple.edu/∼hbling/

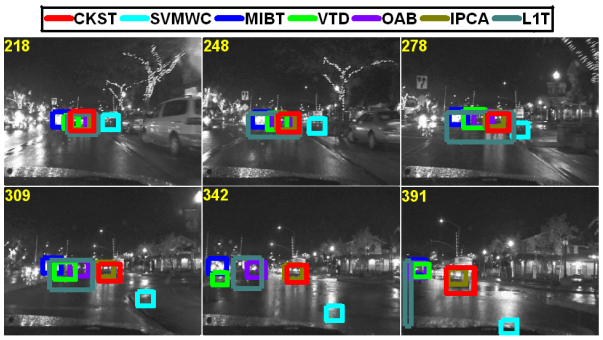

Figure 3: The tracking results of the seven trackers over the representative frames (218, 248, 278, 309, 342, 391) of Video 1 in the scenarios with varying lighting conditions and background clutters.

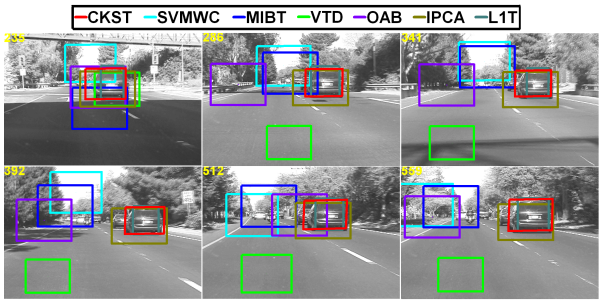

Figure 4: The tracking results of the seven trackers over the representative frames (235, 286, 341, 392, 512, 559) of Video 2 in the scenarios with shadow disturbance and pose variation.

the inherent ambiguity of object localization, MIBT utilizes multiple instances for object representation. In comparison, OAB utilizes online boosting for object/non-object classification. Thus, comparing CKST with MIBT and OAB can demonstrate the discriminative capabilities of CKST in handling large appearance variations. Third, VTD is a recently proposed tracker based on visual tracking decomposition. Using sparse principal component analysis, VTD decomposes the observation (or motion) model into a set of basic observation (or motion) models, each of which covers a specific type of object appearance (or motion). IPCA uses incremental principal component analysis to construct the eigenspace-based observation model for visual tracking. L1T treats visual tracking as a sparse approximation problem using ℓ_1-regularized minimization. Thus, comparing CKST with VTD, IPCA, and L1T can show their capabilities of tolerating complicated appearance changes.

Tracking results Figs. 3–8 show the corresponding tracking results (highlighted by the bounding boxes in different colors) of the seven trackers over the representative frames of the six videos.

Video 1 After frame 271, VTD loses the car due to illumination changes. Distracted by background clutters, SVMWC, MIBT, L1T, and OAB fail to track the car after frames 196, 211, 288, and 289, respectively. In comparison, only CKST and IPCA succeed in tracking the car throughout the video sequence.

Video 2 In face of both shadow disturbance and pose

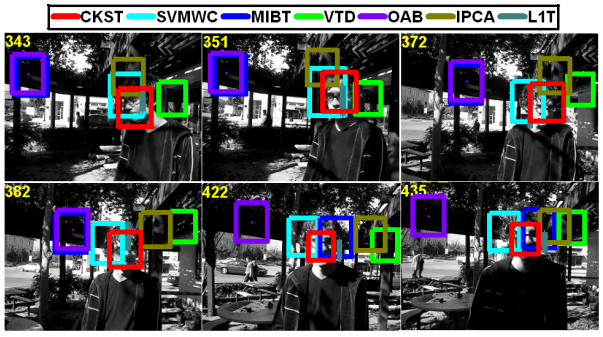

Figure 5: The tracking results of the seven trackers over the representative frames (343, 351, 372, 382, 422, 435) of Video 3 in the scenarios with drastic illumination changes and head pose variations.

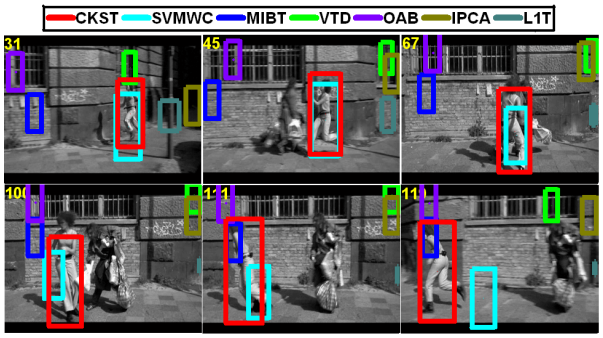

Figure 6: The tracking results of the seven trackers over the representative frames (31, 45, 67, 100, 111, 119) of Video 4 in the scenarios with drastic scale changes and body pose variations.

variation, SVMWC, MIBT, and OAB break down when they track the car. In contrast, VTD is able to track the car before frame 240. However, it tracks the car inaccurately or unsuccessfully after frame 240. On the contrary, CKST can track the car effectively in the situations of shadow disturbance and pose variation throughout the video sequence. Both IPCA and L1T achieve less accurate tracking results than CKST.

Video 3 Under the circumstances of drastic changes in environmental illumination and head pose, SVMWC loses the face after frame 441 while MIBT, VTD, and OAB fail to track the face after frames 201, 148, and 180, respectively. IPCA loses the face after frame 201. L1T begins to lose the face from frame 249. Compared with these competing trackers, CKST can successfully track the face all the time.

Video 4 After colliding with another human body, the human body significantly rotates, leading to the drastic appearance changes. In this case, MIBT, VTD, OAB, L1T, and IPCA begin to lose the human body from frame 23. SVMWC has difficulties in accurately tracking the human body from frame 71 to frame 113, and it loses the human body after frame 114. In contrast, CKST can successfully deal with the difficulties caused by drastic body pose variation, and achieve robust tracking results.

Video 5 Four main factors cause the object appearance changes, including partial occlusions, out-of-plane rotations, body pose variations, and abrupt motion. From frame

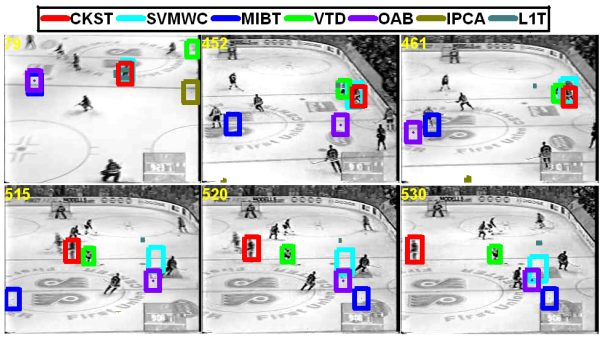

Figure 7: The tracking results of the seven trackers over the representative frames (79, 452, 461, 515, 520, 530) of Video 5 in the scenarios with partial occlusions, out-of-plane rotations, body pose variations, and abrupt motion.

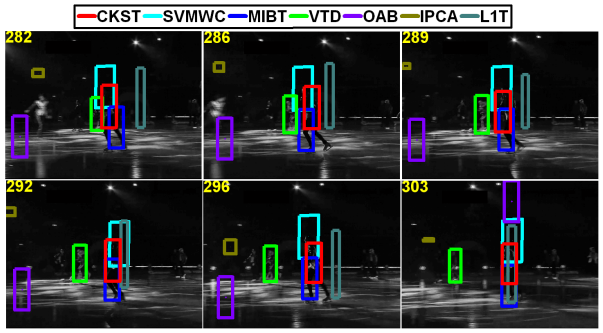

Figure 8: The tracking results of the seven trackers over the representative frames (282, 286, 289, 292, 296, 303) of Video 6 in the scenarios with partial occlusions, body pose variations, and illumination changes.

36 to frame 328, MIBT achieves bad tracking performance. After frame 408, MIBT loses the ice hockey player. VTD and SVMWC lose the ice hockey player after frames 512 and 468, respectively. Both OAB and IPCA fail to track the ice hockey player after frame 56. In contrast, CKST can adapt to the object appearance changes caused by the four factors, and achieve the most accurate tracking results.

Video 6 Starting from frame 287, VTD fails to track the dancer while SVMWC and MIBT achieve inaccurate tracking results. IPCA, L1T, and OAB lose the dancer after frames 48, 275, and 279, respectively. In comparison, CKST can adapt to the object appearance changes, and achieve robust the most accurate tracking results in the situations of partial occlusions, body pose variations, and illumination changes.

Quantitative comparison The object center locations are labeled manually and used as the ground truth. Hence, we can quantitatively evaluate the tracking performances of the seven competing trackers by computing their corresponding pixel-based tracking location errors to the ground truth. Fig. 9 plots the tracking location error plots (highlighted in different colors) obtained by the seven trackers in the six experiments. Further, we also compute the mean of the tracking location errors in the six experiments, and report the results in Table 1. From Fig. 9 and Table 1, we can see that the proposed CKST achieves the most robust and accurate tracking performance over the six video sequences.

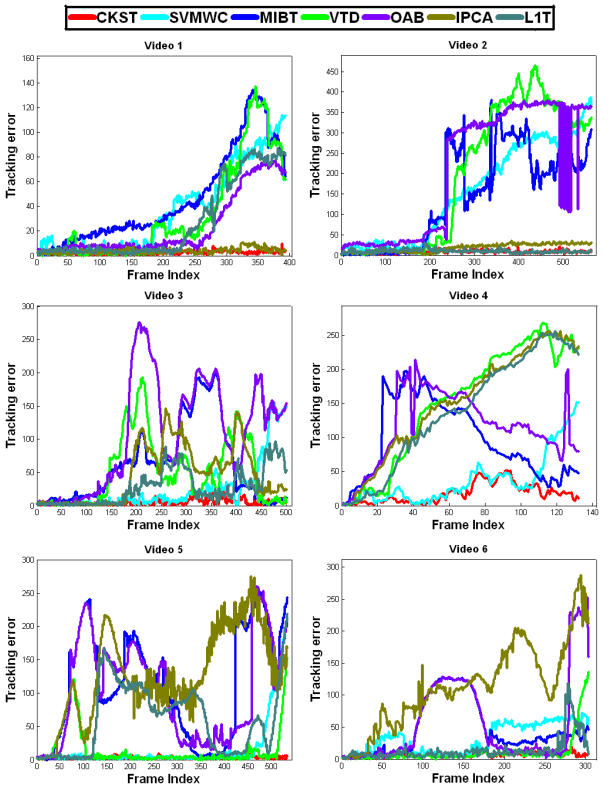

Figure 9: The tracking location error plots obtained by the seven trackers over the six videos.

Discussion The reasons why the proposed CKST outperforms the six other competing trackers are briefly analyzed as follows. Both VTD and IPCA are based on principal component analysis which is sensitive to noises and outliers. During tracking, large appearance changes caused by complex factors (e.g., partial occlusions, abrupt motion, drastic illumination changes, or drastic pose variations) can introduce a large number of outliers to the training data. Using the polluted training data, both VTD and IPCA may learn an inaccurate or wrong subspace model for object representation, leading to inaccurate or wrong tracking results. Moreover, neither of VTD and IPCA takes the object/non-object discriminative information into account, resulting in the distractions by background. L1T performs dynamic template learning for object tracking. In the presence of drastic appearance variations caused by occlusions, L1T may utilize incorrect appearance information in learning and updating templates. OAB learns an AdaBoost classifier for object tracking. Since ignoring the spatial contextual relationships among samples, OAB is sensitive to noises and outliers. MIBT learns a multiple-instance boosting classifier for object tracking. Since representing an object as multiple instances, MIBT may introduce many false positive samples in the process of boosting learning, leading to tracking degradations or even failures. Without considering the contextual information among samples, SVMWC

Table 1: Quantitative comparison results of the seven trackers over the six videos (referred to as V_k for $k = 1, \ldots, 6$). We report the mean of their tracking location errors over the six videos.

	V_1	V_2	V_3	V_4	V_5	V_6
CKST	**2.7**	**8.1**	**5.2**	**17.7**	**3.3**	**6.9**
SVMWC	36.6	154.6	25.7	37.2	15.6	32.2
MIBT	45.5	144.7	59.9	95.4	111.9	46.1
VTD	32.7	194.5	51.1	151.2	14.8	11.4
OAB	22.8	209.1	106.5	114.9	102.8	56.1
IPCA	3.9	17.9	46.4	152.1	127.9	113.4
L1T	26.6	9.9	27.7	139.2	59.8	14.2

cannot effectively adjust the SVM's separating hyperplane to adapt to complicated object appearance changes during tracking. By the graph mode seeking procedure, the proposed CKST is able to capture the intrinsic contextual information (i.e., graph modes and their interactions) from samples. The intrinsic contextual information can reflect the high-order interaction relationships among samples (illustrated in Fig. 1(d)), leading to the robustness of CKST to noises, outliers, and complicated appearance changes. By embedding the intrinsic contextual information into SVM, CKST is capable of learning an effective separating hyperplane for object/non-object classification, leading to robust tracking results.

4. Conclusion

We have proposed a graph mode-based contextual kernel for robust SVM tracking. In this work, a set of high-order contexts are discovered and combined into the tracking process. The problem of discovering these high-order contexts is formulated as graph mode seeking, which can be efficiently solved using graph shift [15]. Each graph mode corresponds to a mode-specific vertex context, in which the graph vertexes share some common visual properties. Furthermore, we design a contextual kernel to capture the interaction information between the mode-specific vertex contexts. We theoretically prove that this contextual kernel is a Mercer kernel. Therefore, embedding this contextual kernel into the standard SVM, the global optimum is guaranteed. We incorporate this contextual kernel into the object/non-object SVM classification scheme for visual tracking. Compared with several state-of-the-art trackers, the proposed CKST tracker is more robust to illumination changes, pose variations, partial occlusions, background clutters, and complicated appearance changes. Experimental results on challenging videos have demonstrated the effectiveness and robustness of the proposed CKST.

Acknowledgments

This work is partially supported by ARC Discovery Project (DP1094764). H. Wang's participation in this work is supported by Xiamen Science & Technology Planning Project Fund (3502Z20116005) of China.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni, "Robust Fragments-based Tracking using the Integral Histogram," in *Proc. IEEE Int. Conf. Comp. Vis. & Pattern Recognition*, pp.798-805, 2006.

[2] H. Wang, D. Suter, K. Schindler, and C. Shen, "Adaptive Object Tracking Based on an Effective Appearance Filter," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 29, pp.1661-1667, 2007.

[3] C. Shen, J. Kim, and H. Wang, "Generalized Kernel-based Visual Tracking," *IEEE Trans. Circ. Syst. Video Tech.,* vol. 20, pp. 119-130, 2010.

[4] X. Mei and H. Ling, "Robust Visual Tracking using ℓ_1 Minimization," in *Proc. IEEE Int. Conf. Comp. Vis.,* pp. 1436-1443, 2009.

[5] H. Grabner, M. Grabner, and H. Bischof, "Real-time Tracking via On-line Boosting," in *Proc. British Machine Vis. Conf.,* pp. 47-56, 2006.

[6] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised On-line Boosting for Robust Tracking," in *Proc. Euro. Conf. Comp. Vis.,* pp.234-247, 2008.

[7] S. Avidan, "Support Vector Tracking," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 26, pp. 1064-1072, 2004.

[8] M. Tian, W. Zhang, and F. Liu, "On-Line Ensemble SVM for Robust Object Tracking," in *Proc. Asian Conf. Comp. Vis.,* pp. 355-364, 2007.

[9] B. Babenko, M. Yang, and S. Belongie, "Visual Tracking with Online Multiple Instance Learning," *IEEE Trans. Pattern Anal. Mach. Intell.,* 2010.

[10] S. Avidan, "Ensemble Tracking," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 29, pp. 261-271, 2007.

[11] R. T. Collins, Y. Liu, and M. Leordeanu, "Online Selection of Discriminative Tracking Features," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 27, pp.1631-1643, 2005.

[12] X. Liu and T. Yu, "Gradient Feature Selection for Online Boosting," in *Proc. IEEE Int. Conf. Comp. Vis.,* pp. 1-8, 2007.

[13] J. Fan, Y. Wu, and S. Dai, "Discriminative Spatial Attention for Robust Tracking," in *Proc. Euro. Conf. Comp. Vis.*, pp. 480-493, 2010.

[14] J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "PROST: Parallel Robust Online Simple Tracking,", in *Proc. IEEE Int. Conf. Comp. Vis. Pattern Recognition,* pp.723-730, 2010.

[15] H. Liu and S. Yan, "Robust Graph Mode Seeking by Graph Shift," in *Proc. Int. Conf. Mach. Learn.*, 2010.

[16] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proc. IEEE Int. Conf. Comp. Vis. Pattern Recognition*, 2005.

[17] J. Kwon and K. M. Lee, "Visual Tracking Decomposition," in *Proc. IEEE Int. Conf. Comp. Vis. Pattern Recognition*, 2010.

[18] D. A. Ross, J. Lim, R. Lin, and M. Yang. "Incremental Learning for Robust Visual Tracking," *Int. J. Comp. Vis., vol. 77, pp. 125-141,* 2008.

[19] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo, "Robust Visual Tracking Based on Incremental Tensor Subspace Learning," in *Proc. IEEE Int. Conf. Comp. Vis.,* 2007.

[20] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, and J. Cheng, "Visual Tracking Via Incremental Log-Euclidean Riemannian Subspace Learning," in *Proc. IEEE Int. Conf. Comp. Vis. Pattern Recognition,* 2008.

[21] F. Tang, S. Brennan, Q. Zhao, and H. Tao, "Co-Tracking Using Semi-Supervised Support Vector Machines," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2007.

[22] M. Yang, Y. Wu, and G. Hua, "Context-Aware Visual Tracking," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 31, pp.1195-1209, 2009.

[23] H. Li, C. Shen, and Q. Shi, "Real-time Visual Tracking with Compressed Sensing," in *Proc. IEEE Int. Conf. Comp. Vis. Pattern Recognition,* 2011.