Image and Vision Computing 51 (2016) 84–94

Contents lists available at ScienceDirect

Image and Vision Computing

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

Online unsupervised feature learning for visual tracking☆

Fayao Liu, Chunhua Shen*, Ian Reid, Anton van den Hengel

School of Computer Science, The University of Adelaide, SA5005, Australia

ARTICLE INFO

Article history:
Received 6 March 2015
Received in revised form 4 February 2016
Accepted 19 April 2016
Available online 3 May 2016

Keywords:
Object tracking
Unsupervised feature learning
Dictionary learning

ABSTRACT

We propose a method for visual tracking-by-detection based on online feature learning. Our learning framework performs feature encoding with respect to an over-complete dictionary, followed by spatial pyramid pooling. We then learn a linear classifier based on the resulting feature encoding. Unlike previous work, we learn the dictionary online and update it to help capture the appearance of the tracked target as well as the background. In more detail, given a test image window, we extract local image patches from it and each local patch is encoded with respect to the dictionary. The encoded features are then pooled over a spatial pyramid to form an aggregated feature vector. Finally, a simple linear classifier is trained on these features. Our experiments show that the proposed powerful—albeit simple—tracker, outperforms all the state-of-the-art tracking methods that we have tested. Moreover, we evaluate the performance of different dictionary learning and feature encoding methods in the proposed tracking framework, and analyze the impact of each component in the tracking scenario. In particular, we show that a small dictionary, learned and updated online is as effective and more efficient than a huge dictionary learned offline. We further demonstrate the flexibility of feature learning by showing how it can be used within a structured learning tracking framework. The outcome is one of the best trackers reported to date, which facilitates the advantages of both feature learning and structured output prediction. We also implement a multi-object tracker, which achieves state-of-the-art performance.

1. Introduction

Robust visual tracking is an important topic in computer vision, with applications from object recognition to behavior analysis. Given the initial state (*e.g.*, bounding box) of a target in a video sequence, visual tracking aims to infer the states of the target in succeeding frames. Despite profound recent progress [1–9], there remain significant challenges such as the inevitable changes in target appearance over time, and confounding backgrounds or occlusions (see [10] for a categorization of these challenges).

To address the issue of appearance and background variation, many sophisticated appearance models have been proposed, which may be roughly categorized as either generative or discriminative. Generative model-based trackers build a robust appearance model for the tracked object and search for the best matching candidate regions. Examples that fall into this category are incremental subspace learning [11], sparse representation based tracking [1,8,12–14], distribution fields representation based tracking [15]. In contrast, tracking methods based upon discriminative learning typically operate within a classification framework, using a classifier to distinguish the target from its surroundings. Representative methods include support vector machines (SVM) [6], boosting ensemble tracking [16], online multiple instance learning [17], bootstrapping binary classifier tracker [18], and structured output tracking [2]. Our proposed tracker belongs to this latter category.

In recent years, unsupervised feature learning methods have been successfully applied to many vision tasks such as image classification [19,20], object recognition [21], and scene categorization [22]. A standard feature learning pipeline typically comprises three steps: (a) learning an over-complete dictionary; (b) encoding the features with respect to the dictionary; (c) spatial pooling of the encoded features over a pyramid of regular spatial grids. The dictionary learning process is typically unsupervised, using a method such as K-means, K-SVD [23], sparse coding, sparse/denoising autoencoder, or even random sampling. As for the encoding, soft thresholding, soft assignment, sparse coding, localized soft assignment [24] are commonly applied. It has been shown in [20] that for a sufficiently large dictionary, the specific dictionary learning method has little impact on the classification performance. Rather, it is the encoding step that is pivotal to final performance.

☆ This paper has been recommended for acceptance by Seong-Whan Lee, PhD.
* Corresponding author.
E-mail address: chunhua.shen@adelaide.edu.au (C. Shen).

http://dx.doi.org/10.1016/j.imavis.2016.04.008
0262-8856/© 2016 Elsevier B.V. All rights reserved.

The success of this approach has inspired us to adapt the image classification pipeline to object tracking. The main contributions of this work are as follows: (i) We propose a feature learning based tracker using online dictionary learning [25]. Not only do we update the classifier model over time, but also we update the dictionary words online. This gives improved adaptation to foreground and background appearance compared to a large offline dictionary and is more efficient. Despite the simplicity of the proposed tracker, it outperforms almost all state-of-the-art trackers in the literature. (ii) We evaluate the performance of a selection of widely-used dictionary learning and feature encoding methods within the proposed tracking framework. We conclude that a compact dictionary learned from the sequence frames is powerful enough for good performance, which is different from the image classification case [20]. The reason can be attributed to the relatively simple classification problem due to the sequential property of the tracking problem. (iii) To further demonstrate the superior performance of learned features over traditional hand-crafted features in visual tracking we show how online feature learning can be easily incorporated into a structured learning based tracker [2] and demonstrate improved tracking accuracy as a result.

We briefly review some recent work that is most relevant to ours. The appearance model is a critical component of any tracking system and has attracted extensive study as a result. Besides the traditional hand-crafted features, such as texture [16], HOG [3,4], Haar-like features [2,5,17], and similar, sparse representations have also been widely used. It is this sparsity-based approach which motivates the feature learning based tracker proposed here. The sparse representations based tracking [8,26] solves the standard sparse coding problem in order to sparsely represent the tracking object and the tracking is implemented using the reconstruction residual. Note that the representations in these methods are holistic (which means that they are based on templates rather than patches and therefore spatial pooling cannot be applied), and that the dictionaries are constructed using simple methods such as sampling or principal component analysis. This is in contrast to the method which we propose here. Ours is mainly motivated by the success of the generic image classification pipeline using unsupervised feature learning, in which features are encoded on local patches and spatial pooling plays a critical role. For example, our experiments show that pooling, which confers some invariance to local illumination changes and spatial misalignment—has a significant positive impact on performance. Moreover, sparse representations need to solve the computationally expensive L_0 or L_1 optimization at each frame [8,26]. Although the work of [1,8] proposed faster solvers, in general, they are still slow.

Recent work of [27] proposes learning a dictionary of SIFT descriptors extracted from natural images using sparse coding. Their work is close to ours in that both follow the generic image classification pipeline. The main differences are as follows: (i) We initially learn image features from the first frame of tracking video, rather than learning generic image features from natural images as in [27]. Our experiments show the advantage of learning features from the tracking video. (ii) Hand-crafted SIFT is used as the low-level feature in [27]. Instead, we learn the image features from raw pixels. (iii) More importantly, the method in [27] encodes the features by solving a group sparsity regularized coding problem, which is computationally expensive. In contrast, we use simple closed-form encoding. (iv) We update both the dictionary and the classifier online so that the tracker can better adapt to the changes of the target and background. Partially due to the lack of online dictionary updating, a large-sized dictionary has to be used in [27], which leads to a final representation of high dimension (14336 in their case). This considerably slows down the classifier evaluation. The above issues severely limit the practical value of [27] in tracking. We show that by using online dictionary learning on pixels with simple but extremely efficient encoding methods, along with spatial pooling, our method outperforms almost all state-of-the-art trackers.

2. Unsupervised feature learning for tracking

We follow the well-known tracking-by-detection framework (*e.g.*, [16]), which attempts to learn a classifier to discriminate the target object from the background. Unlike almost all prior work, which uses raw pixels or hand-crafted features, we propose to learn features in an online, unsupervised fashion [25], tailored for tracking. First, we learn a dictionary $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_n] \in \mathcal{R}^{m \times n}$ of size n where each column $\mathbf{d}_j$ represents a basis vector[1] and m is dimension. Note that if $n > m$, then $\mathbf{D}$ is over-complete. The dictionary is learned from N image patches extracted from the current frame: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathcal{R}^{m \times N}$, where $\mathbf{x}_i$ denotes a vector formed by stacking all pixel values of an image patch. The dictionary is then updated online during tracking when necessary. Note that the dictionary could equally be formed from local image descriptors, but in contrast to the claim in [27] which learns a dictionary over SIFT features, we observed that such an approach did not perform as well in the tracking task (see Section 3.1).

Due to its efficiency and ease of implementation, the soft threshold (ST) coding strategy is applied here, which writes

$$\mathbf{C} = \max\{0, \mathbf{D}^{\top}\mathbf{X} - s\},$$

where s is a threshold. $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]^{\top} \in \mathcal{R}^{n \times N}$ are the encoded features representing the original image patches $\mathbf{X}$ over which we then perform max pooling to produce the final feature vectors $\hat{\mathbf{X}}$. This is based on the theoretical and empirical evaluation in [28] which showed that max-pooling generally yields more discriminative features for classification, compared to sum or average pooling. Finally the learned features $\hat{\mathbf{X}}$ are then used to train a linear SVM for detection.

The framework of our feature learning based tracking is illustrated in Fig. 1 and the algorithm is summarized in Algorithm 1.

Algorithm 1. Online unsupervised feature learning for tracking.

```
Input: Initial dictionary $\mathbf{D}_{\text{curr}}$; image patch size $p$; step size $q$; length of sequence $T$.
for $t = 1$ to $T$ do
    Extract image patches $\mathbf{X}_t$ of size $p \times p$ at step size $q$ from frame $t$ and do contrast normalization.
    If dictionary update required (see Sec(2.2)) do
        do_update=true.
        Update dictionary $\mathbf{D}_{\text{curr}}$ by Algorithm 2 with $\mathbf{D}^{(0)} = \mathbf{D}_{\text{curr}}$.
    end if
    Sample a set of image patches around the previous target location estimate.
    Encode the raw pixel features of the patches extracted within each sampled box using $\mathbf{D}_{\text{curr}}$ by soft threshold coding.
    Perform max-pooling over a spatial pyramid of multiple layers.
    If $mod(t, 5) = 0$ or do_update=true then retrain LS-SVM classifier by solving (7).

    Run the classifier over search window and select as the target location the bounding box with the highest confidence rating.
end for
```

2.1. Online dictionary learning

Although recent work [19,20] has shown that simple dictionary learning methods can be very effective for large dictionary sizes, the computational limitations imposed by the visual tracking problem demand a compact representation. However a fixed dictionary is generally not sufficient to cope with the inevitable appearance changes of the tracked object as well as the background over time. Our solution to this is to employ online dictionary

[1] We call the element in a dictionary basis, although it is not necessarily orthogonal.