Engineering Applications of Artificial Intelligence 55 (2016) 363–374

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Non-linear eigenspace visual object tracking

Iftikhar Majeed *, Omar Arif

School of Electrical Engineering and Computer Sciences, National University of Sciences and Technology, Islamabad, Pakistan

ARTICLE INFO

Article history:
Received 26 November 2015
Received in revised form
12 June 2016
Accepted 6 August 2016

Keywords:
Visual tracking
Kernel methods
Particle filter
Machine learning
Vision systems

ABSTRACT

This paper presents a visual object tracking algorithm using an eigenspace representation. Previous approaches to eigenspace methods for object tracking use vectorized image regions as observations. Here, feature vectors associated to pixels of the target template are considered to be individual observations of the target object. The collection of observations is learned using non-linear subspace projection to arrive at an eigenspace representation. This representation allows tracking pixel-wise but the pixels are tied together using subspace representation which provides a robust and compact representation of the object. Localization and segmentation are carried out by deriving a similarity function in the eigenspace representation. A gradient descent and mean-shift based techniques are derived to maximize the similarity function with respect to the transformation parameters. The de-noising and clustering capabilities of the eigenspace representation lead to a localization procedure that is robust to noise and partial occlusion. For fast moving objects and to recover from total occlusion, a probabilistic search strategy, based on particle filter, is also developed. A unique feature of our approach is that it permits segmentation in addition to localization when multiple templates of the target are given. The performance of the algorithm is tested on real world tracking examples.

1. Introduction

Visual tracking is the process of locating an object of interest through a sequence of successive images. It is a fundamental problem in computer vision with wide areas of applications such as surveillance, traffic monitoring, human computer interaction, vehicle navigation, and robot planning. Some of the problems associated with object tracking are image noise, occlusion, background clutter, complex object shapes, etc.

Target objects can be represented by their appearances, such as color, texture, edges, and shape information, which provide characteristic information about the object. This characteristic information, gathered from a template or a set of templates of the target object, is encoded into a cost or similarity function. Tracking algorithms then determine the correspondence of the object region in consecutive images by optimizing the pre-determined similarity functional. Methods following this approach are called *generative methods* (Kwon and Lee, 2001; Liu et al., 2011; Gao et al., 2014), as they search the image space to find the region most similar to the target model. In contrast to generative methods, *discriminative methods* formulate the problem as a binary classification problem. From positive and negative samples, a classifier is trained to separate the foreground from the background. This can be carried out at pixel level (Godec et al., 2013; Avidan, 2007) or at region level (Hare et al., 2011; Henriques et al., 2015). The latter methods are also called *tracking by detection*. Tracking by detection methods are more suitable for tracking rigid methods while methods that pose tracking as pixel wise classification problem are more suitable for non-rigid objects as they ignore the spatial information (Oron et al., 2012).

Algorithms can also be grouped on the basis of different encoding strategies employed to track rigid and non-rigid objects. Tracking rigid objects call for methods that encode the spatial geometry of the object in similarity function. In the simplest case of template matching, the similarity function is reduced to per pixel difference between the template and the target region. More involved methods use multiple templates (Kwon and Lee, 2001), sparse representations (Mei et al., 2011; Wang et al., 2015; Zhang et al., 2015) and subspace models (Ross et al., 2004; Sun and Liu, 2010). For deformable objects, histogram based approaches are more suitable (Oron et al., 2012), reducing the similarity between target and candidate region to similarity between color distributions. For example, Comaniciu et al. (2003) use a histogram weighted by a spatial kernel as a probability density function of the object region. The correspondence of the target object between sequential frames is established at the region level by optimizing the Bhattacharya coefficient between the target and the

* Corresponding author.
E-mail addresses: 13mscsimajeed@seecs.edu.pk (I. Majeed), omar.arif@seecs.edu.pk (O. Arif).

http://dx.doi.org/10.1016/j.engappai.2016.08.005

candidate distributions using mean-shift (Cheng, 1995). Histograms discard spatial information, which becomes problematic when faced with occlusions and the presence of target features in the background. Attempts to incorporate spatial information into the descriptor have been proposed in Hager et al. (2004), Zhimin et al. (2007), Birchfield and Rangarajan (2005). The aforementioned algorithms require computing the probability density functions (histograms), which becomes computationally expensive for higher dimensions. An additional problem associated with computing probability density functions is the sparseness of the observations within the feature space, especially when the sample set size is small. Methods, such as Yang and Duraiswami (2005), Singh et al. (2004), Elgammal et al. (2003) define similarity functions between kernel density estimates of the template and target distribution in a joint feature-spatial space. Since these methods employ non-parametric density comparison techniques, the intermediate step of estimating the density function is not carried out.

Essential to improved tracking is the derivation of a model that can capture the relationship between the purely image-based observations and the spatial content associated to said observations. Some generative methods use unsupervised learning techniques such as principal component analysis (PCA) to measure the correlation among the pixels of the templates of the target. The templates are vectorized to form a matrix $D = [I_1, I_2, \ldots I_n]$, where each column is a vectorized template. The covariance matrix obtained from the data in D is diagonalized to obtain a low dimensional eigenspace representation of the target. This representation has been used for tracking in Black and Jepson (1998) and Lim et al. (2004). In Lim et al. (2004), the subspace is also incrementally updated to account for appearance and illumination change. Tsai et al. (2003) perform PCA on a collection of vectorized signed distance maps of the training shapes to incorporate the shape model into the segmentation procedure. Similarly (Sun and Liu, 2010) uses incremental Kernel PCA with particle filtering framework to track objects that perform pose variation or undergo occasional occlusion. In all the subspace methods discussed here, each vectorized template I_i is an observation with the implicit assumption that the image appearance will remain similar to the training templates. However, under partial or extensive occlusions this assumption will not hold and the tracker may give erroneous results.

In this paper, we propose to use each pixel vector of the target template as an observation, unlike the aforementioned methods, where vectorized target template is considered as an observation. A pixel vector is a concatenation of appearance (color value, gradient, edge value etc.) and spatial location. The method is schematically described in Fig. 1. Pixel vectors extracted from a single template can be used for learning the model. Such an approach is not feasible for other eigenspace methods, where several to many vectorized object templates are required. The representation is powerful in the sense that tracking is done pixel-wise, but the pixels are tied together in eigenspace representation.

Contributions: This paper connects non-parametric, kernel-based methods with statistical eigenspace methods to derive a target localization strategy. Each feature vector associated to a pixel of the target object describes an observation of the target object, whose overall joint appearance-spatial geometry is learned using non-linear eigenspace representation associated to the collection of feature vectors forming the target. The eigenspace representation provides a compact and robust description of the target being tracked. A gradient descent and mean-shift based optimization techniques are developed to robustly track the object. These strategies provide reliable solution under partial occlusion. To track through total occlusion and to track fast moving objects, we also develop particle filtering based approach (Arulampalam et al.) to sequentially estimate the state variables, which in our case are the transformation parameters of the rectangular target object region. Particle filter performance increases when sample size is large. However, increasing the samples also increases the computational cost. In this paper, we also derive kernel integral image based formulation that allows us to densely sample the state space without loosing computational efficiency. The speedup obtained is of about 2.5 orders of magnitude over non-integral image implementation. This paper improves upon and extends an earlier basic version of the paper (Arif and Vela, 2009).

We describe the target eigenspace representation in Section 2. Similarity measure is explained in Section 3 followed by object tracking methods described in Section 4. Experimental validation of the proposed algorithm is carried out in Section 5.

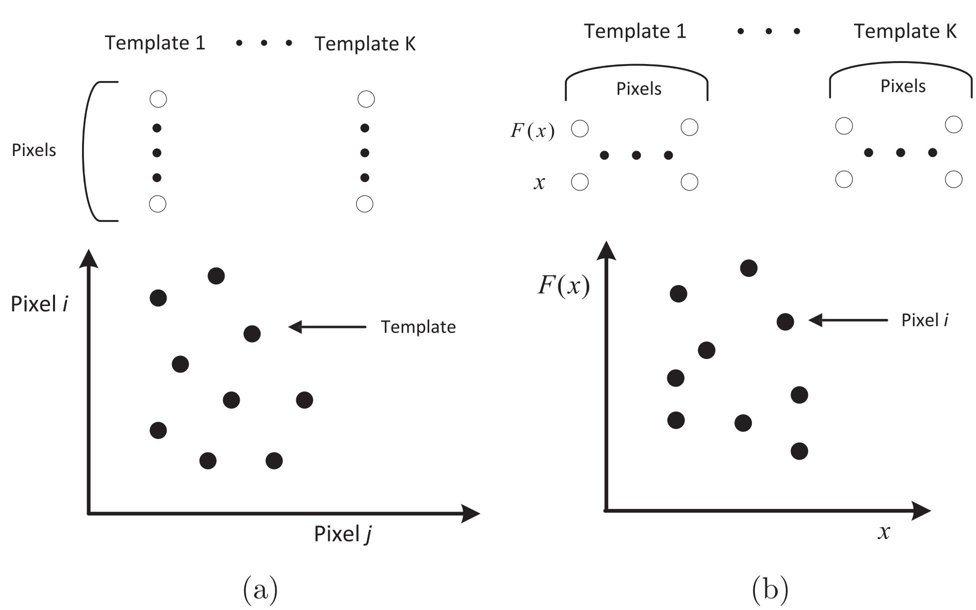

Fig. 1. Target representations: template-wise vs. pixel-wise. (a) Each template is an observation. Training templates are vectorized, stacked together, and learned. (b) Proposed approach: Each pixel vector (appearance+spatial location) is an observation. Pixel vectors from all training templates are amassed and learned.