A Survey of Appearance Models in Visual Object Tracking

Xi Li, NLPR, Institute of Automation, Chinese Academy of Sciences, China
The University of Adelaide, SA 5005, Australia
Weiming Hu, NLPR, Institute of Automation, Chinese Academy of Sciences, China
Chunhua Shen, The University of Adelaide, SA 5005, Australia
Zhongfei Zhang, State University of New York, Binghamton, NY 13902, USA
Anthony Dick, The University of Adelaide, SA 5005, Australia
Anton van den Hengel, The University of Adelaide, SA 5005, Australia

Visual object tracking is a significant computer vision task which can be applied to many domains such as visual surveillance, human computer interaction, and video compression. Despite extensive research on this topic, it still suffers from difficulties in handling complex object appearance changes caused by factors such as illumination variation, partial occlusion, shape deformation, and camera motion. Therefore, effective modeling of the 2D appearance of tracked objects is a key issue for the success of a visual tracker. In the literature, researchers have proposed a variety of 2D appearance models.

To help readers swiftly learn the recent advances in 2D appearance models for visual object tracking, we contribute this survey, which provides a detailed review of the existing 2D appearance models. In particular, this survey takes a module-based architecture that enables readers to easily grasp the key points of visual object tracking. In this survey, we first decompose the problem of appearance modeling into two different processing stages: visual representation and statistical modeling. Then, different 2D appearance models are categorized and discussed with respect to their composition modules. Finally, we address several issues of interest as well as the remaining challenges for future research on this topic.

The contributions of this survey are four-fold. First, we review the literature of visual representations according to their feature-construction mechanisms (i.e., local and global). Second, the existing statistical modeling schemes for tracking-by-detection are reviewed according to their model-construction mechanisms: generative, discriminative, and hybrid generative-discriminative. Third, each type of visual representations or statistical modeling techniques is analyzed and discussed from a theoretical or practical viewpoint. Fourth, the existing benchmark resources (e.g., source codes and video datasets) are examined in this survey.

Categories and Subject Descriptors: I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis–Tracking

General Terms: Algorithms, Performances

Additional Key Words and Phrases: Visual object tracking, appearance model, features, statistical modeling

1. INTRODUCTION

One of the main goals of computer vision is to enable computers to replicate the basic functions of human vision such as motion perception and scene understanding. To achieve the goal of intelligent motion perception, much effort has been spent on visual object tracking, which is one of the most important and challenging research topics in computer vision. Essentially, the core of visual object tracking is to robustly estimate the motion state (i.e., location, orientation, size, etc.) of a target object in each frame of an input image sequence.

In recent years, a large body of research on visual object tracking has been published in the literature. Research interest in visual object tracking comes from the fact that it has a wide range of real-world applications, including visual surveillance, traffic flow monitoring, video compression, and human-computer interaction. For example, visual object tracking is successfully applied to monitor human activities in residential areas, parking lots, and banks (e.g., W^4 system [Haritaoglu et al. 2000] and VSAM project [Collins et al. 2000]). In the field of traffic transportation, visual object tracking is also widely used to cope with traffic flow monitoring [Coifman et al. 1998], traffic accident detection [Tai et al. 2004], pedestrian counting [Masoud and Papanikolopoulos 2001], and so on. Moreover, visual object tracking is utilized by the MPEG-4 video compression standard [Sikora 1997] to automatically detect and track moving objects in videos. As a result, more encoding bytes are assigned to moving objects while fewer encoding bytes are for redundant backgrounds. Visual object tracking also has several human-computer interaction applications such as hand gesture recognition [Pavlovie et al. 1997] and mobile video conferencing [Paschalakis and Bober 2004].

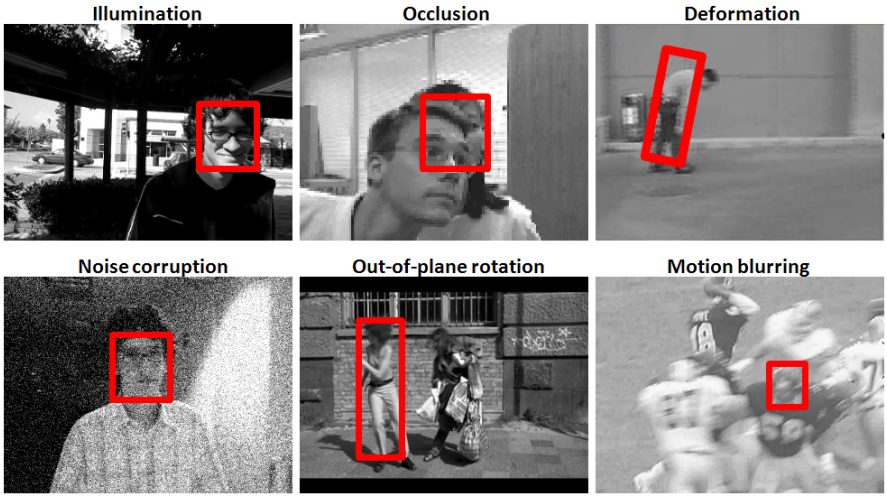

Fig. 1: Illustration of complicated appearance changes in visual object tracking.

Note that all the above applications heavily rely on the information provided by a robust visual object tracking method. If such information is not available, these applications would be no longer valid. Therefore, robust visual object tracking is a key issue to make these applications viable.

1.1. Overview of visual object tracking

In general, a typical visual object tracking system is composed of four modules: object initialization, appearance modeling, motion estimation, and object localization.

— *i) object initialization*. This may be manual or automatic. Manual initialization is performed by users to annotate object locations with bounding boxes or ellipses. In contrast, automatic initialization is usually achieved by object detectors (e.g., face or human detectors).
— *ii) appearance modeling*. This generally consists of two components: visual representation and statistical modeling. Visual representation focuses on how to construct robust object descriptors using different types of visual features. Statistical modeling concentrates on how to build effective mathematical models for object identification using statistical learning techniques.
— *iii) motion estimation*. This is formulated as a dynamic state estimation problem: $x_t = f(x_{t-1}, v_{t-1})$ and $z_t = h(x_t, w_t)$, where x_t is the current state, f is the state evolution function, v_{t-1} is the evolution process noise, z_t is the current observation, h denotes the measurement function, and w_t is the measurement noise. The task of motion estimation is usually completed by utilizing predictors such as linear regression techniques [Ellis et al. 2010], Kalman filters [Kalman 1960], or particle filters [Isard and Blake 1998; Arulampalam et al. 2002].
— *iv) object localization*. This is performed by a greedy search or maximum a posterior estimation based on motion estimation.

1.2. Challenges in developing robust appearance models

Many issues have made robust visual object tracking very challenging, including (i) low-quality camera sensors (e.g., low frame rate, low resolution, low bit-depth, and color distortion); (ii) challenging factors (e.g., non-rigid object tracking, small-size object tracking, tracking a varying number of objects, and complicated pose estimation); (iii) real-time processing requirements; (iv) object tracking across cameras with non-overlapping views [Javed et al. 2008]; and (v) object appearance variations (as shown in Fig. 1) caused by several complicated factors (e.g., environmental illumination changes, rapid camera motions, full occlusion, noise disturbance, non-rigid shape deformation, out-of-plane object rotation, and pose variation). These challenges may cause tracking degradations and even failures.

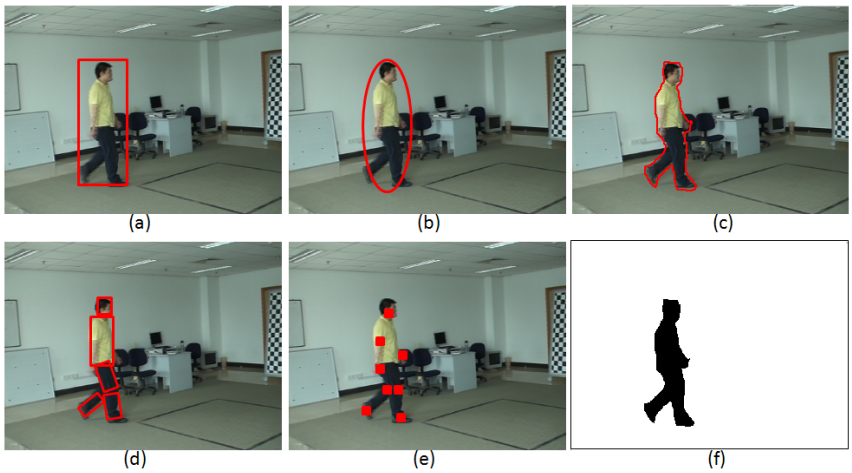

Fig. 2: Illustration of object tracking forms. (a) bounding box, (b) ellipse, (c) contour, (d) articulation block, (e) interest point, (f) silhouette.

In order to deal with these challenges, researchers have proposed a wide range of appearance models using different visual representations and/or statistical modeling techniques. These appearance models usually focus on different problems in visual object tracking, and thus have different properties and characteristics. Typically, they attempt to answer the following questions:

— What to track (e.g., bounding box, ellipse, contour, articulation block, interest point, and silhouette, as shown in Fig. 2)?
— What visual representations are appropriate and robust for visual object tracking?
— What are the advantages or disadvantages of different visual representations for different tracking tasks?
— Which types of statistical learning schemes are suitable for visual object tracking?
— What are the properties or characteristics of these statistical learning schemes during visual object tracking?
— How should the camera/object motion be modeled in the tracking process?

The answers to these questions rely heavily on the specific context/environment of the tracking task and the tracking information available to users. Consequently, it is necessary to categorize these appearance models into several task-specific categories and discuss in detail the representative appearance models of each category. Motivated by this consideration, we provide a survey to help readers acquire valuable tracking knowledge and choose the most suitable appearance model for their particular tracking tasks. Furthermore, we examine several interesting issues for developing new appearance models.

2. ORGANIZATION OF THIS SURVEY

Fig. 3 shows the organization of this survey, which is composed of two modules: visual representation and statistical modeling. The visual representation module concentrates on how to robustly describe the spatio-temporal characteristics of object appearance. In this module, a variety of visual representations are discussed, as illustrated by the tree-structured taxonomy in the left part of Fig. 3. These visual representations can capture various visual information at different levels (i.e., local and global). Typically, the local visual representations encode the local statistical information (e.g., interest point) of an image region, while the global visual representations reflect the global statistical characteristics (e.g., color histogram) of an image region. For a clear illustration of this module, a detailed literature review of visual representations is given in Sec. 3.

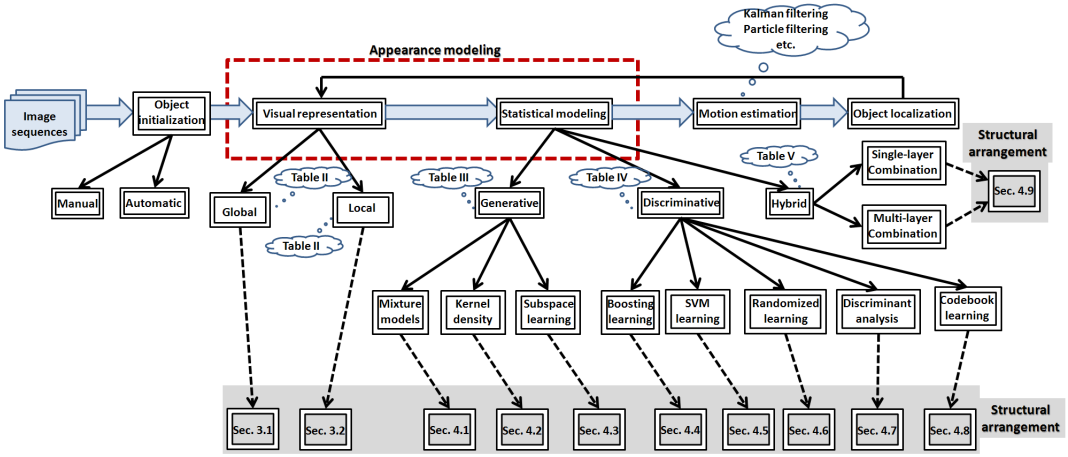

Fig. 3: The organization of this survey.

Table I: Summary of related literature surveys

Authors	Topic	Journal/conference title
[Gerónimo et al. 2010]	Pedestrian Detection	IEEE Trans. on PAMI.
[Candamo et al. 2010]	Human Behavior Recognition	IEEE Trans. on Intelligent Transportation Systems
[Cannons 2008]	Visual Tracking	Technical Report
[Zhan et al. 2008]	Crowd analysis	Machine Vision Application
[Kang and Deng 2007]	Intelligent Visual Surveillance	IEEE/ACIS Int. Conf. Comput. Inf. Sci.
[Yilmaz et al. 2006]	Visual object tracking	ACM Computing Survey
[Forsyth et al. 2006]	Human Motion Analysis	Found. Trends Comput. Graph. Vis.
[Sun et al. 2006]	Vehicle Detection	IEEE Trans. on PAMI.
[Hu et al. 2004]	Object Motion and Behaviors	IEEE Trans. on Syst., Man, Cybern. C, Appl. Rev.
[Arulampalam et al. 2002]	Bayesian Tracking	IEEE Trans. on Signal Processing

As shown in the right part of Fig. 3, the statistical modeling module is inspired by the tracking-by-detection idea, and thus focuses on using different types of statistical learning schemes to learn a robust statistical model for object detection, including generative, discriminative, and hybrid generative-discriminative ones. In this module, various tracking-by-detection methods based on different statistical modeling techniques are designed to facilitate different statistical properties of the object/non-object class. For a clear illustration of this module, a detailed literature review of statistical modeling schemes for tracking-by-detection is given in Sec. 4.

Moreover, a number of source codes and video datasets for visual object tracking are examined to make them easier for readers to conduct tracking experiments in Sec. 5. Finally, the survey is concluded in Sec. 6. In particular, we additionally address several interesting issues for the future research in Sec. 6.

2.1. Main differences from other related surveys

In the recent literature, several related surveys (e.g., [Gerónimo et al. 2010; Candamo et al. 2010; Cannons 2008; Zhan et al. 2008; Kang and Deng 2007; Yilmaz et al. 2006; Forsyth et al. 2006; Sun et al. 2006; Hu et al. 2004; Arulampalam et al. 2002]) of visual object tracking have been made to investigate the state-of-the-art tracking algorithms and their potential applications, as listed in Tab. I. Among these surveys, the topics of the surveys [Cannons 2008; Yilmaz et al. 2006] are closely related to this paper. Specifically, both of the surveys [Cannons 2008; Yilmaz et al. 2006] focus on low-level tracking techniques using different visual features or statistical learning techniques, and thereby give very comprehensive and specific technical contributions.

The main differences between these two surveys [Cannons 2008; Yilmaz et al. 2006] and this survey are summarized as follows. First, this survey focuses on the 2D appearance modeling for visual object tracking. In comparison, the surveys of [Cannons 2008; Yilmaz et al. 2006] concern all the

Table II: Summary of representative visual representations

Item No.	References	Global/local	Visual representations
1	[Ho et al. 2004; Li et al. 2004; Ross et al. 2008]	Global	Vector-based raw pixel representation
2	[Li et al. 2007]	Global	Matrix-based raw pixel representation
3	[Wang et al. 2007]	Global	Multi-cue raw pixel representation (i.e., color, position, edge)
4	[Werlberger et al. 2009; Santner et al. 2010]	Global	Optical flow representation (constant-brightness-constraint)
5	[Black and Anandan 1996; Wu and Fan 2009]	Global	Optical flow representation (non-brightness-constraint)
6	[Bradski 1998] [Comaniciu et al. 2003; Zhao et al. 2010]	Global	Color histogram representation
7	[Georgescu and Meer 2004]	Global	Multi-cue spatial-color histogram representation (i.e., joint histogram in (x, y, R, G, B))
8	[Adam et al. 2006]	Global	Multi-cue spatial-color histogram representation (i.e., patch-division histogram)
9	[Haralick et al. 1973; Gelzinis et al. 2007]	Global	Multi-cue spatial-texture histogram representation (i.e., Gray-Level Co-occurrence Matrix)
10	[Haritaoglu and Flickner 2001] [Ning et al. 2009]	Global	Multi-cue shape-texture histogram representation (i.e., color, gradient, texture)
11	[Porikli et al. 2006; Wu et al. 2008]	Global	Affine-invariant covariance representation
12	[Li et al. 2008; Hong et al. 2010] [Wu et al. 2012; Hu et al. 2012]	Global	Log-Euclidean covariance representation
13	[He et al. 2002; Li et al. 2009]	Global	Wavelet filtering-based representation
14	[Paragios and Deriche 2000; Cremers 2006] [Allili and Ziou 2007; Sun et al. 2011]	Global	Active contour representation
15	[Lin et al. 2007]	Local	Local feature-based represnetation (local templates)
16	[Tang and Tao 2008; Zhou et al. 2009]	Local	Local feature-based represnetation (SIFT features)
17	[Donoser and Bischof 2006; Tran and Davis 2007]	Local	Local feature-based represnetation (MSER features)
18	[He et al. 2009]	Local	Local feature-based represnetation (SURF features)
19	[Grabner et al. 2007; Kim 2008]	Local	Local feature-based represnetation (Corner features)
20	[Collins et al. 2005; Grabner and Bischof 2006] [Yu et al. 2008]	Local	Local feature-based represnetation (feature pools of Harr, HOG, LBP etc.)
21	[Toyama and Hager 1996] [Mahadevan and Vasconcelos 2009] [Yang et al. 2007; Fan et al. 2010]	Local	Local feature-based representations (Saliency detection-based features)
22	[Ren and Malik 2007; Wang et al. 2011]	Local	Local feature-based represnetation (Segmentation-based features)

modules shown in Fig. 3. Hence, this survey is more intensive while the surveys of [Cannons 2008; Yilmaz et al. 2006] are more comprehensive. Second, this survey provides a more detailed analysis of various appearance models. Third, the survey of [Yilmaz et al. 2006] splits visual object tracking into three categories: point tracking, kernel tracking, and silhouette tracking (see Fig. 7 in [Yilmaz et al. 2006] for details); the survey of [Cannons 2008] gives a very detailed and comprehensive review of each tracking issue in visual object tracking. In contrast to these two surveys, this survey is formulated as a general module-based architecture (shown in Fig. 3) that enables readers to easily grasp the key points of visual object tracking. Fourth, this survey investigates a large number of state-of-the-art appearance models which make use of novel visual features and statistical learning techniques. In comparison, the surveys [Cannons 2008; Yilmaz et al. 2006] pay more attention to classic and fundamental appearance models used for visual object tracking.

2.2. Contributions of this survey

The contributions of this survey are as follows. First, we review the literature of visual representations from a feature-construction viewpoint. Specifically, we hierarchically categorize visual representations into local and global features. Second, we take a tracking-by-detection criterion for reviewing the existing statistical modeling schemes. According to the model-construction mechanisms, these statistical modeling schemes are roughly classified into three categories: generative, discriminative, and hybrid generative-discriminative. For each category, different types of statistical learning techniques for object detection are reviewed and discussed. Third, we provide a detailed discussion on each type of visual representations or statistical learning techniques with their properties. Finally,

we examine the existing benchmark resources for visual object tracking, including source codes and databases.

3. VISUAL REPRESENTATION

3.1. Global visual representation

A global visual representation reflects the global statistical characteristics of object appearance. Typically, it can be investigated in the following main aspects: (i) raw pixel representation; (ii) optical flow representation; (iii) histogram representation; (iv)covariance representation; (v) wavelet filtering-based representation; and (vi)active contour representation. Tab. II lists several representative tracking methods using global visual representations (i.e., Rows 1-14).

— Raw pixel representation. As the most fundamental features in computer vision, raw pixel values are widely used in visual object tracking because of their simplicity and efficiency. Raw pixel representation directly utilizes the raw color or intensity values of the image pixels to represent the object regions. Such a representation is simple and efficient for fast object tracking. In the literature, raw pixel representations are usually constructed in the following two forms: vector-based [Silveira and Malis 2007; Ho et al. 2004; Li et al. 2004; Ross et al. 2008] and matrix-based [Li et al. 2007; Wen et al. 2009; Hu et al. 2010; Wang et al. 2007; Li et al. 2008]. The vector-based representation directly flattens an image region into a high-dimensional vector, and often suffers from a small-sample-size problem. Motivated by attempting to alleviate the small-sample-size problem, the matrix-based representation directly utilizes 2D matrices or higher-order tensors as the basic data units for object description due to its relatively low-dimensional property.

  However, raw pixel information alone is not enough for robust visual object tracking. Researchers attempt to embed other visual cues (e.g., shape or texture) into the raw pixel representation. Typically, the color features are enriched by fusing other visual information such as edge [Wang et al. 2007] and texture [Allili and Ziou 2007].

— Optical flow representation. In principle, optical flow represents a dense field of displacement vectors of each pixel inside an image region, and is commonly used to capture the spatio-temporal motion information of an object. Typically, optical flow has two branches: constant-brightness-constraint (CBC) optical flow [Lucas and Kanade 1981; Horn and Schunck 1981; Werlberger et al. 2009; Sethi and Jain 1987; Salari and Sethi 1990; Santner et al. 2010] and non-brightness-constraint (NBC) optical flow [Black and Anandan 1996; Sawhney and Ayer 1996; Hager and Belhumeur 1998; Bergen et al. 1992; Irani 1999; Wu and Fan 2009]. The CBC optical flow has a constraint on brightness constancy while the NBC optical flow deals with the situations with varying lighting conditions.

— Histogram representation. Histogram representations are popular in visual object tracking because of their effectiveness and efficiency in capturing the distribution characteristics of visual features inside the object regions. In general, they have two branches: single-cue and multi-cue.

  (i) A single-cue histogram representation often constructs a histogram to capture the distribution information inside an object region. For example, Bradski [1998] uses a color histogram in the Hue Saturation Value (HSV) color space for object representation, and then embeds the color histogram into a continuously adaptive mean shift (CAMSHIFT) framework for object tracking. However, the direct use of color histogram may result in the loss of spatial information. Following the work in [Bradski 1998], Comaniciu et al. [2003] utilize a spatially weighted color histogram in the RGB color space for visual representation, and subsequently embed the spatially weighted color histogram into a mean shift-based tracking framework for object state inference. Zhao et al. [2010] convert the problem of object tracking into that of matching the RGB color distributions across frames. As a result, the task of object localization is taken by using a fast differential EMD (Earth Mover's Distance) to compute the similarity between the color distribution of the learned target and the color distribution of a candidate region.

  (ii) A multi-cue histogram representation aims to encode more information to enhance the robustness of visual representation. Typically, it contains three main components: a) spatial-color; b) spatial-texture; c) shape-texture;

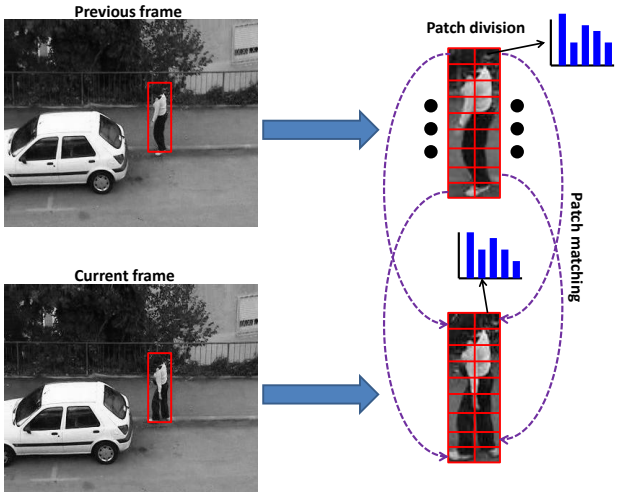

Fig. 4: Illustration of patch-division visual representation (from [Adam et al. 2006], ©2006 IEEE). The left part shows the previous and current frames, and the right part displays the patch-wise histogram matching process between two image regions.

For a), two strategies are adopted, including joint spatial-color modeling and patch-division. The goal of joint spatial-color modeling is to describe the distribution properties of object appearance in a joint spatial-color space (e.g., (x, y, R, G, B) in [Yang et al. 2005; Georgescu and Meer 2004; Birchfield and Rangarajan 2005]). The patch-division strategy is to encode the spatial information into the appearance models by splitting the tracking region into a set of patches [Adam et al. 2006; Nejhum et al. 2010]. By considering the geometric relationship between patches, it is capable of capturing the spatial layout information. For example, Adam et al. [Adam et al. 2006] construct a patch-division visual representation with a histogram-based feature description for object tracking, as shown in Fig. 4. The final tracking position is determined by combining the vote maps of all patches (represented by grayscale histograms). The combination mechanism can eliminate the influence of the outlier vote maps caused by occlusion. For the computational efficiency, Porikli [2005] introduces a novel concept of an integral histogram to compute the histograms of all possible target regions in a Cartesian data space. This greatly accelerates the speed of histogram matching in the process of mean shift tracking.

For b), an estimate of the joint spatial-texture probability is made to capture the distribution information on object appearance. For example, Haralick et al. [1973] propose a spatial-texture histogram representation called Gray-Level Co-occurrence Matrix (GLCM), which encodes the co-occurrence information on pairwise intensities in a specified direction and distance. Note that the GLCM in [Haralick et al. 1973] needs to tune different distance parameter values before selecting the best distance parameter value by experimental evaluations. Following the work in [Haralick et al. 1973], Gelzinis et al. [Gelzinis et al. 2007] propose a GLCM-based histogram representation that does not need to carefully select an appropriate distance parameter value. The proposed histogram representation gathers the information on the co-occurrence matrices computed for several distance parameter values.

For c), the shape or texture information on object appearance is incorporated into the histogram representation for robust visual object tracking. For instance, Haritaoglu and Flickner [2001] incorporate the gradient or edge information into the color histogram-based visual representation. Similar to [Haritaoglu and Flickner 2001], Wang and Yagi [2008] construct an visual representation using color and shape cues. The color cues are composed of color histograms in three different color spaces: RGB, HSV, and normalized rg. The shape cue is described by gradient orientation histograms. To exploit the textural information of the object, Ning et al. [2009] propose a joint

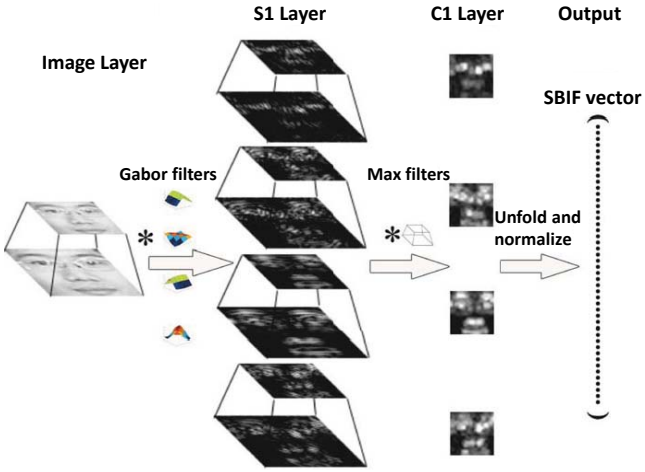

Fig. 5: Illustration of three-layer Gabor features (from [Li et al. 2009], ©2009 IEEE). The first column shows the grayscale face images aligned in a spatial pyramid way (i.e., image layer); the second column the Gabor energy maps (containing rich orientation and spatial frequency information in the image pyramid) obtained by Gabor filtering (i.e., S1 layer); the third column exhibits the response of applying max filtering (returning the maximum value; and tolerant to local distortions) to the Gabor energy maps; and the last column plots the final feature vector after unfolding and normalization.

color-texture histogram for visual representation. The local binary pattern (LBP) technique is employed to identify the key points in the object regions. Using the identified key points, they build a confidence mask for joint color-texture feature selection.

— Covariance representation. In order to capture the correlation information of object appearance, covariance matrix representations are proposed for visual representation in [Porikli et al. 2006; Tuzel et al. 2006]. According to the Riemannian metrics mentioned in [Li et al. 2008; Hu et al. 2012], the covariance matrix representations can be divided into two branches: affine-invariant Riemannian metric-based and Log-Euclidean Riemannian metric-based.

(i) The Affine-invariant Riemannian metric [Porikli et al. 2006; Tuzel et al. 2006] is based on the following distance measure: $\rho(\mathbf{C}_1, \mathbf{C}_2) = \sqrt{\sum_{j=1}^{d} \ln^2 \lambda_j(\mathbf{C}_1, \mathbf{C}_2)}$, where $\{\lambda_j(\mathbf{C}_1, \mathbf{C}_2)\}_{j=1}^{d}$ are the generalized eigenvalues of the two covariance matrices $\mathbf{C}_1$ and $\mathbf{C}_2$: $\lambda_j \mathbf{C}_1 \mathbf{x}_j = \mathbf{C}_2 \mathbf{x}_j$, $j \in \{1, \ldots, d\}$, and $\mathbf{x}_j$ is the j-th generalized eigenvector. Following the work in [Porikli et al. 2006; Tuzel et al. 2006], Austvoll and Kwolek [2010] use the covariance matrix inside a region to detect whether the feature occlusion events take place. The detection task can be completed by comparing the covariance matrix-based distance measures in a particular window around the occluded key point.

(ii) The Log-Euclidean Riemannian metric [Arsigny et al. 2006] formulates the distance measure between two covariance matrices in a Euclidean vector space. Mathematically, the Log-Euclidean Riemannian metric for two covariance matrices $\mathbf{C}_i$ and $\mathbf{C}_j$ is formulated as: $\mathrm{d}(\mathbf{C}_i, \mathbf{C}_j) = \| \log(\mathbf{C}_i) - \log(\mathbf{C}_j) \|$ where $\log$ is the matrix logarithm operator. For the descriptive convenience, the covariance matrices under the Log-Euclidean Riemannian metric are referred to as the Log-Euclidean covariance matrices. Inspired by [Arsigny et al. 2006], Li et al. [2008] employ the Log-Euclidean covariance matrices of image features for visual representation. Since the Log-Euclidean covariance matrices lie in a Euclidean vector space, their mean can be easily computed as the standard arithmetic mean. Due to this linear property, classic subspace learning techniques (e.g., principal component analysis) can be directly applied onto the Log-Euclidean covariance matrices. Following the work in [Li et al. 2008; Hu et al. 2012], Wu et al. [2009; 2012] extend the tracking problem of using 2D Log-Euclidean covariance matrices to that of using higher-order ten-

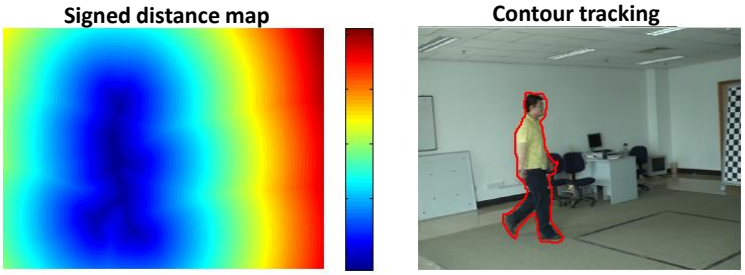

Fig. 6: Illustration of an active contour representation. The left part shows the signed distance map of a human contour; and the right part displays the contour tracking result.

sors, and aim to incrementally learn a low-dimensional covariance tensor representation. Inspired by [Li et al. 2008; Hu et al. 2012], Hong et al. [2010] propose a simplified covariance region descriptor (called Sigma set), which comprises the lower triangular matrix square root (obtained by Cholesky factorization) of the covariance matrix (used in [Li et al. 2008]). The proposed covariance region descriptor characterizes the second order statistics of object appearance by a set of vectors. Meanwhile, it retains the advantages of the region covariance descriptor [Porikli et al. 2006], such as low-dimensionality, robustness to noise and illumination variations, and good discriminative power.

— Wavelet filtering-based representation. In principle, a wavelet filtering-based representation takes advantage of wavelet transforms to filter the object region in different scales or directions. For instance, He et al. [2002] utilize a 2D Gabor wavelet transform (GWT) for visual representation. Specifically, an object is represented by several feature points with high GWT coefficients. Moreover, Li *et al.* [2009] propose a tracking algorithm based on three-layer simplified biologically inspired (SBI) features (i.e., image layer, S1 layer, and C1 layer). Through the flattening operations on the four Gabor energy maps in the C1 layer, a unified SBI feature vector is returned to encode the rich spatial frequency information, as shown in Fig. 5.

— Active contour representation. In order to track the nonrigid objects, active contour representations have been widely used in recent years [Paragios and Deriche 2000; Cremers 2006; Allili and Ziou 2007; Vaswani et al. 2008; Sun et al. 2011]. Typically, an active contour representation (shown in Fig. 6) is defined as a signed distance map $\mathbf{\Phi}$:

$$\mathbf{\Phi}(x,y) = \begin{cases} 0 & (x,y) \in C \\ d(x,y,C) & (x,y) \in R_{out} \\ -d(x,y,C) & (x,y) \in R_{in} \end{cases} \quad (1)$$

where R_{in} and R_{out} respectively denote the regions inside and outside the contour C, and $d(x,y,C)$ is a function returning the smallest Euclidean distance from point (x,y) to the contour C. Moreover, an active contour representation is associated with a energy function which comprises three terms: internal energy, external energy, and shape energy. The internal energy term reflects the internal constraints on the object contour (e.g., the curvature-based evolution force), the external energy term measures the likelihood of the image data belonging to the foreground object class, and the shape energy characterizes the shape prior constraints on the object contour.

3.1.1. Discussion. Without feature extraction, the raw pixel representation is simple and efficient for visual object tracking. Since only considering the color information on object appearance, the raw pixel representation is susceptible to complicated appearance changes caused by illumination variation.

The constant-brightness-constraint (CBC) optical flow captures the field information on the translational vectors of each pixel in a region with the potential assumption of locally unchanged brightness. However, the CBC assumption is often invalid in the complicated situations caused by image noise, illumination fluctuation, and local deformation. To address this issue, the non-brightness-

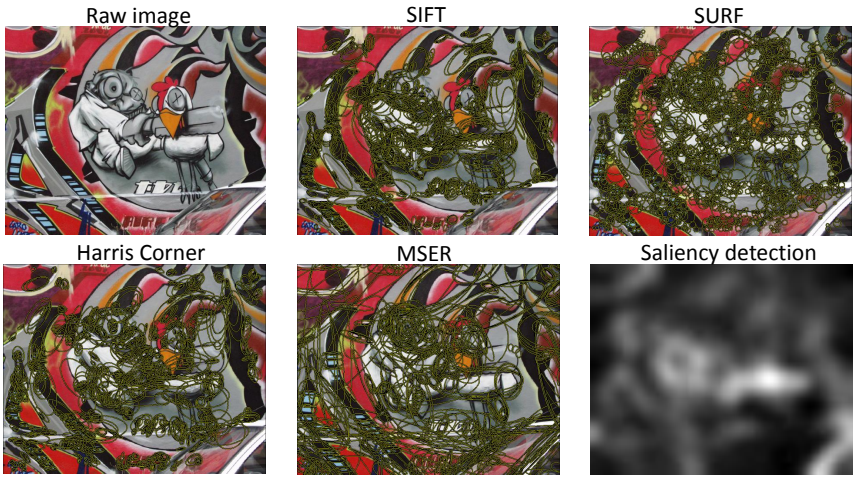

Fig. 7: Illustration of several local features (extracted by using the software which can be downloaded at http://www.robots.ox.ac.uk/∼vgg/research/affine/ and http://www.klab.caltech.edu/∼harel/share/gbvs.php.)

constraint optical flow is developed to introduce more geometric constraints on the contextual relationship of pixels.

The single-cue histogram representation is capable of efficiently encoding the statistical distribution information of visual features within the object regions. Due to its weakness in characterizing the spatial structural information of tracked objects, it is often affected by background distractions with similar colors to the tracked objects. In order to capture more spatial information, the spatial-color histogram representation is introduced for visual object tracking. Usually, it encodes the spatial information by either modeling object appearance in a joint spatial-color feature space or taking a patch-division strategy. However, the above histogram representations do not consider the shape or texture information of object appearance. As a consequence, it is difficult to distinguish the object from the background with similar color distributions. To alleviate this issue, the shape-texture histogram representation is proposed to integrate shape or texture information (e.g., gradient or edge) into the histogram representation, leading to the robustness of object appearance variations in illumination and pose.

The advantages of using the covariance matrix representation are as follows: (i) it can capture the intrinsic self-correlation properties of object appearance; (ii) it provides an effective way of fusing different image features from different modalities; (iii) it is low-dimensional, leading to the computational efficiency; (iv) it allows for comparing regions of different sizes or shapes; (v) it is easy to implement; (vi) it is robust to illumination changes, occlusion, and shape deformations. The disadvantages of using the covariance matrix representation are as follows: (i) it is sensitive to noisy corruption because of taking pixel-wise statistics; (ii) it loses much useful information such as texture, shape, and location.

A wavelet filtering-based representation is to encode the local texture information of object appearance by wavelet transform, which is a convolution with various wavelet filters. As a result, the wavelet filtering-based representation is capable of characterizing the statistical properties of object appearance in multiple scales and directions (e.g., Gabor filtering).

An active contour representation is designed to cope with the problem of nonrigid object tracking. Usually, the active contour representation adopts the signed distance map to implicitly encode the boundary information of an object. On the basis of level set evolution, the active contour representation can precisely segment the object with a complicated shape.

Fig. 8: Illustration of the local template-based visual representation using superpixels.

3.2. Local feature-based visual representation

As shown in Fig. 7, local feature-based visual representations mainly utilize interest points or saliency detection to encode the object appearance information. In general, the local features based on the interest points can be mainly categorized into seven classes: local template-based, segmentation-based, SIFT-based, MSER-based, SURF-based, corner feature-based, feature pool-based, and saliency detection-based. Several representative tracking methods using local feature-based visual representations are listed in Rows 15-22 of Tab. II.

— Local template-based. In general, local template-based visual representations are to represent an object region using a set of part templates. In contrast to the global template-based visual representation, they are able to cope with partial occlusions effectively and model shape articulations flexibly. For instance, a hierarchical part-template shape model is proposed for human detection and segmentation [Lin et al. 2007]. The shape model is associated with a part-template tree that decomposes a human body into a set of part-templates. By hierarchically matching the part-templates with a test image, the proposed part-template shape model can generate a reliable set of detection hypotheses, which are then put into a global optimization framework for the final human localization.
— Segmentation-based. Typically, a segmentation-based visual representation incorporates the image segmentation cues (e.g., object boundary [Ren and Malik 2007]) into the process of object tracking, which leads to reliable tracking results. Another alternative is based on superpixel segmentation, which aims to group pixels into perceptually meaningful atomic regions. For example, Wang et al. [2011] construct a local template-based visual representation with the superpixel segmentation, as shown in Fig. 8. Specifically, the surrounding region of an object is segmented into several superpixels, each of which corresponds to a local template. By building a local template dictionary based on the mean shift clustering, an object state is predicted by associating the superpixels of a candidate sample with the local templates in the dictionary.
— SIFT-based. Usually, a SIFT-based visual representation directly makes use of the SIFT features inside an object region to describe the structural information of object appearance. Usually, there are two types of SIFT-based visual representations: (i) individual SIFT point-based; and (ii) SIFT graph-based. For (i), Zhou et al. [2009] set up a SIFT point-based visual representation, and combine this visual representation with the mean shift for object tracking. Specifically, SIFT features are used to find the correspondences between the regions of interest across frames. Meanwhile, the mean shift procedure is implemented to conduct a similarity search via color histograms. By using a mutual support mechanism between SIFT and the mean shift, the tracking algorithm is able to achieve a consistent and stable tracking performance. However, the tracking algorithm may suffer from a background clutter which may lead to a one-to-many SIFT feature matching. In this situation, the mean shift and SIFT feature matching may make mutually contradictory decisions. For (ii), the SIFT graph-based visual representations are based on the underlying geometric contextual relationship among SIFT feature points. For example, Tang and Tao [2008] construct a relational graph

using SIFT-based attributes for object representation. The graph is based on the stable SIFT features which persistently appear in several consecutive frames. However, such stable SIFT features are unlikely to exist in complex situations such as shape deformation and illumination changes.

— MSER-based. A MSER-based visual representation needs to extract the MSER (maximally stable extremal region) features for visual representation [Sivic et al. 2006]. Subsequently, Tran and Davis [2007] construct a probabilistic pixel-wise occupancy map for each MSER feature, and then perform the MSER feature matching for object tracking. Similar to [Tran and Davis 2007], Donoser and Bischof [2006] also use MSER features for visual representation. To improve the stability of MSER features, they take temporal information across frames into consideration.

— SURF-based. With the scale-invariant and rotation-invariant properties, the SURF (Speeded Up Robust Feature) is a variant of SIFT [Bay et al. 2006]. It has similar properties to those of SIFT in terms of repeatability, distinctiveness, and robustness, but its computational speed is much faster. Inspired by this fact, He et al. [2009] develop a tracking algorithm using a SURF-based visual representation. By judging the compatibility of local SURF features with global object motion, the tracking algorithm is robust to appearance changes and background clutters.

— Corner feature-based. Typically, a corner feature-based visual representation makes use of corner features inside an object region to describe the structural properties of object appearance, and then matches these corner features across frames for object localization. For instance, Kim [2008] utilizes corner features for visual representation, and then perform dynamic multi-level corner feature grouping to generate a set of corner point trajectories. As a result, the spatio-temporal characteristics of object appearance can be well captured. Moreover, Grabner et al. [2007] explore the intrinsic differences between the object and non-object corner features by building a boosting discriminative model for corner feature classification.

— Local feature pool based. Recently, local feature pool based visual representations have been widely used in ensemble learning based object tracking. Usually, they need to set up a huge feature pool (i.e., a large number of various features) for constructing a set of weak learners, which are used for discriminative feature selection. Therefore, different kinds of visual features (e.g., color, local binary pattern [Collins et al. 2005], histogram of oriented gradients [Collins et al. 2005; Liu and Yu 2007; Yu et al. 2008], Gabor features with Gabor wavelets [Nguyen and Smeulders 2004], and Haar-like features with Haar wavelets [Babenko et al. 2009]) can be used by FSSL in an independent or interleaving manner. For example, Collins et al. [2005] set up a color feature pool whose elements are linear combinations of the following RGB components: $\{(\alpha_1, \beta_1, \gamma_1)|\alpha_1, \beta_1, \gamma_1 \in \{-2, -1, 0, 1, 2\}\}$. As a result, an object is localized by selecting the discriminative color features from this pool. Grabner and Bischof [Grabner and Bischof 2006] construct an ensemble classifier by learning several weak classifiers trained from the Haar-like features [Viola and Jones 2002], histograms of oriented gradient (HOG) [Dalal and Triggs 2005], and local binary patterns (LBP) [Ojala et al. 2002]. Babenko et al. [2009] utilize the Haar-like features to construct a weak classifier, and then apply an online multiple instance boosting to learn a strong ensemble classifier for object tracking.

— Saliency detection-based. In principle, saliency detection is inspired by the focus-of-attention (FoA) theory [Palmer 1999; Wolfe 1994] to simulate the human perception mechanism for capturing the salient information of an image. Such salient information is helpful for visual object tracking due to its distinctness and robustness. Based on saliency detection, researchers apply the biological vision theory to visual object tracking [Toyama and Hager 1996; Mahadevan and Vasconcelos 2009]. More recently, Yang et al. [2007; 2010] construct an attentional visual representation method based on the spatial selection. This visual representation method takes a two-stage strategy for spatial selective attention. At the first stage, a pool of attentional regions (ARs) are extracted as the salient image regions. At the second stage, discriminative learning is performed to select several discriminative attentional regions for visual representation. Finally, the task of object tracking is taken by matching the ARs between two consecutive frames.

3.2.1. Discussion. The aforementioned local feature-based representations use local templates, segmentation, SIFT, MSER, SURF, corner points, local feature pools, or saliency detection, re-

spectively. Due to the use of different features, these representations have different properties and characteristics. By representing an object region using a set of part templates, the local template-based visual representations are able to encode the local spatial layout information of object appearance, resulting in the robustness to partial occlusions. With the power of image segmentation, the segmentation-based visual representations are capable of well capturing the intrinsic structural information (e.g., object boundaries and superpixels) of object appearance, leading to reliable tracking results in challenging situations. Since the SIFT features are invariant to image scaling, partial occlusion, illumination change, and 3D camera viewpoint change, the SIFT-based representation is robust to appearance changes in illumination, shape deformation, and partial occlusion. However, it cannot encode precise information on the objects such as size, orientation, and pose. The MSER-based representation attempts to find several maximally stable extremal regions for feature matching across frames. Hence, it can tolerate pixel noise, but suffers from illumination changes. The SURF-based representation is on the basis of the "Speeded Up Robust Features", which has the properties of scale-invariance, rotation-invariance, and computationally efficiency. The corner-point representation aims to discover a set of corner features for feature matching. Therefore, it is suitable for tracking objects (e.g., cars or trucks) with plenty of corner points, and sensitive to the influence of non-rigid shape deformation and noise. The feature pool-based representation is strongly correlated with feature selection-based ensemble learning that needs a number of local features (e.g., color, texture, and shape). Due to the use of many features, the process of feature extraction and feature selection is computationally slow. The saliency detection-based representation aims to find a pool of discriminative salient regions for a particular object. By matching the salient regions across frames, object localization can be achieved. However, its drawback is to rely heavily on salient region detection, which is sensitive to noise or drastic illumination variation.

3.3. Discussion on global and local visual representations

In general, the global visual representations are simple and computationally efficient for fast object tracking. Due to the imposed global geometric constraints, the global visual representations are susceptible to global appearance changes (e.g., caused by illumination variation or out-of-plane rotation). To deal with complicated appearance changes, a multi-cue strategy is taken by the global features to incorporate multiple types of visual information (e.g., position, shape, texture, and geometric structure) into the appearance models.

In contrast, the local visual representations are able to capture the local structural object appearance. Consequently, the local visual representations are robust to global appearance changes caused by illumination variation, shape deformation, rotation, and partial occlusion. Since they require the keypoint detection, the interest point-based local visual representations often suffer from noise disturbance and background distraction. Moreover, the local feature pool-based visual representations, which are typically required by discriminative feature selection, need a huge number of local features (e.g., color, texture, and shape), resulting in a very high computational cost. Inspired by the biological vision, the local visual representations using biological features attempt to capture the salient or intrinsic structural information inside the object regions. This salient information is relatively stable during the process of visual object tracking. However, salient region features rely heavily on salient region detection which may be susceptible to noise or drastic illumination variation, leading to potentially many feature mismatches across frames.

4. STATISTICAL MODELING FOR TRACKING-BY-DETECTION

Recently, visual object tracking has been posed as a tracking-by-detection problem (shown in Fig. 9), where statistical modeling is dynamically performed to support object detection. According to the model-construction mechanism, statistical modeling is classified into three categories, including generative, discriminative, and hybrid generative-discriminative.

The generative appearance models mainly concentrate on how to accurately fit the data from the object class. However, it is very difficult to verify the correctness of the specified model in practice. Besides, the local optima are always obtained during the course of parameter estimation (e.g., expectation maximization). By introducing online-update mechanisms, they incrementally learn visual

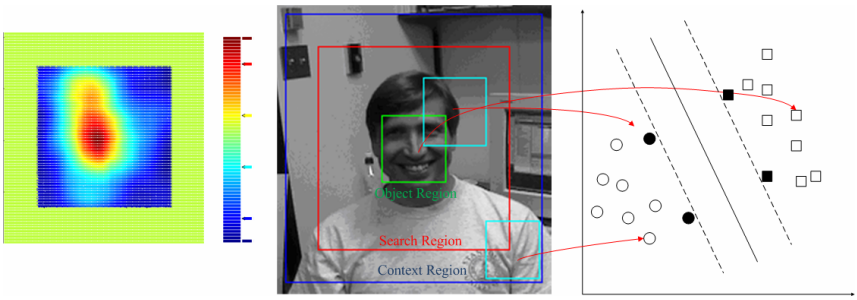

Fig. 9: Illustration of tracking-by-detection based on SVM classification (from [Tian et al. 2007], ©2007 Springer). The left subfigure shows the score map of face/non-face classification; the middle subfigure displays the search region for object localization and the context region for selecting face and non-face samples; the right subfigure plots the classification hyperplane that separates face and non-face classes.

representations for the foreground object region information while ignoring the influence of the background. As a result, they often suffer from distractions caused by the background regions with similar appearance to the object class. Tab. III lists representative tracking-by-detection methods based on generative learning techniques.

In comparison, discriminative appearance models pose visual object tracking as a binary classification issue. They aim to maximize the separability between the object and non-object regions discriminately. Moreover, they focus on discovering highly informative features for visual object tracking. For the computational consideration, online variants are proposed to incrementally learn discriminative classification functions for the purpose of object or non-object predictions. Thus, they can achieve effective and efficient predictive performances. Nevertheless, a major limitation of the discriminative appearance models is to rely heavily on training sample selection (e.g., by self-learning or co-learning). Tab. IV lists representative tracking-by-detection methods based on discriminative learning techniques.

The generative and discriminative appearance models have their own advantages and disadvantages, and are complementary to each other to a certain extent. Therefore, researchers propose hybrid generative-discriminative appearance models (HGDAMs) to fuse the useful information from the generative and the discriminative models. Due to taking a heuristic fusion strategy, HGDAMs cannot guarantee that the performance of the hybrid models after information fusion is better than those of the individual models. In addition, HGDAMs may add more constraints and introduce more parameters, leading to more inflexibility in practice. Tab. V lists representative tracking-by-detection methods based on hybrid generative-discriminative learning techniques.

4.1. Mixture generative appearance models

Typically, this type of generative appearance models adaptively learns several components to capture the spatio-temporal diversity of object appearance. They can be classified into two categories: WSL mixture models and Gaussian mixture models.

— WSL mixture models. In principle, the WSL mixture model [Jepson et al. 2003] contains the following three components: W-component, S-component, and L-component. These three components characterize the inter-frame variations, the stable structure for all past observations, and outliers such as occluded pixels, respectively. As a variant of [Jepson et al. 2003], another WSL mixture model [Zhou et al. 2004] is proposed to directly employ the pixel-wise intensities as visual features instead of using the filter responses (e.g. in [Jepson et al. 2003]). Moreover, the L-component is discarded in modeling the occlusion using robust statistics, and an F-component is added as a fixed template that is observed most often.
— Gaussian mixture models. In essence, the Gaussian mixture models [McKenna et al. 1999; Stauffer and Grimson 2000; Han and Davis 2005; Yu and Wu 2006; Wang et al. 2007] utilize a set of Gaus-

Table III: Summary of representative tracking-by-detection appearance models based on generative learning techniques.

Item No.	References	Mixture models	Kernel density estimation	Subspace learning	Used generative learning techniques
1	[McKenna et al. 1999]	color-based GMM	—	—	Gaussian mixture model (GMM) in the hue-saturation color space
2	[Yu and Wu 2006] [Wang et al. 2007]	Spatio-color GMM	—	—	Spatial-color appearance model using GMM Spatial-color mixture of Gaussians (SMOG)
3	[Jepson et al. 2003; Zhou et al. 2004]	WSL	—	—	three-component mixture models: W-component, S-component, L-component
4	[Comaniciu et al. 2003] [Leichter et al. 2010]	—	Color-driven	—	Mean shift using a spatially weighted color histogram Mean shift using multiple reference color histograms
5	[Leichter et al. 2009]	—	Shape-integration	—	Affine kernel fitting using color and boundary cues
6	[Collins 2003] [Yang et al. 2005]	—	Scale-aware	—	Mean shift considering scale changes
7	[Nguyen et al. 2007]	—	Scale-aware	—	EM-based maximum likelihood estimation for kernel-based tracking
8	[Yilmaz 2007]	—	Non-symmetric kernel	—	Asymmetric kernel mean shift
9	[Shen et al. 2007]	—	Global mode seeking	—	Annealed mean shift
10	[Han et al. 2008]	—	Sequential kernel density estimation	—	Sequential kernel-based tracking
11	[Black and Jepson 1996; Ho et al. 2004] [Ross et al. 2008; Wen et al. 2012] [Wang et al. 2012]	—	—	Vector-based linear subspace learning	Principal component analysis Partial least square analysis
12	[Wang et al. 2007; Li et al. 2007] [Wen et al. 2009; Hu et al. 2010]	—	—	Tensor-based linear subspace learning	2D principle component analysis Tensor subspace analysis
13	[Lim et al. 2006; Chin and Suter 2007]	—	—	Nonlinear subspace learning	Local linear embedding Kernel principle component analysis
14	[Mei and Ling 2009; Li et al. 2011] [Zhang et al. 2012; Jia et al. 2012] [Bao et al. 2012]	—	—	Sparse representation	ℓ_1 sparse approximation
15	[Li et al. 2012]	—	—	Non-sparse representation	Metric-weighted least-square regression
16	[Li et al. 2013]	—	—	3D-DCT representation	Signal compression
17	[Lee and Kriegman 2005; Fan et al. 2008] [Kwon and Lee 2010]	—	—	Multiple subspaces	bi-subspace or multi-subspace learning
18	[Hou et al. 2001] [Sclaroff and Isidoro 2003] [Matthews and Baker 2004]	—	—	Active appearance models	Shape and appearance 3D mesh fitting

sian distributions to approximate the underlying density function of object appearance, as shown in Fig. 10. For instance, an object appearance model [Han and Davis 2005] using a mixture of Gaussian density functions is proposed to automatically determine the number of density functions and their associated parameters including mean, covariance, and weight. Rectangular features are introduced by averaging the corresponding intensities of neighboring pixels (e.g., 3×3 or 5×5) in each color channel. To capture a spatial-temporal description of the tracked objects, Wang et al. [2007] present a Spatial-color Mixture of Gaussians (referred to as SMOG) appearance model, which can simultaneously encode both spatial layout and color information. To enhance its robustness and stability, Wang et al. further integrate multiple cues into the SMOG appearance model, including three features of edge points: their spatial distribution, gradient intensity, and size. However, it is difficult for the Gaussian mixture models to select the correct number of components. For example, adaptively determining the component number k in a GMM is a difficult task in practice. As a result, the mixture models often use ad-hoc or heuristic criteria for selecting k, leading to the tracking inflexibility.

4.2. Kernel-based generative appearance models (KGAMs)

Kernel-based generative appearance models (KGAMs) utilize kernel density estimation to construct kernel-based visual representations, and then carry out the mean shift for object localization, as shown in Fig. 11. According to the mechanisms used for kernel construction or mode seeking, they may be split into the following six branches: color-driven KGAMs, shape-integration KGAMs, scale-aware KGAMs, non-symmetric KGAMs, KGAMs by global mode seeking, and sequential-kernel-learning KGAMs.

— Color-driven KGAMs. Typically, a color-driven KGAM [Comaniciu et al. 2003] builds a color histogram-based visual representation regularized by a spatially smooth isotropic kernel. Using the Bhattacharyya coefficient as the similarity metric, a mean shift procedure is performed for object

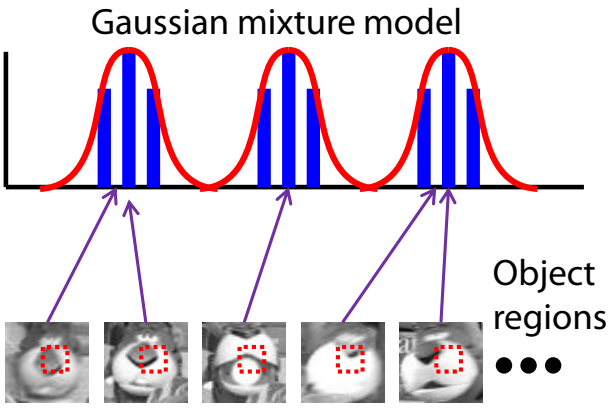

Fig. 10: Illustration of Gaussian mixture generative appearance models.

localization by finding the basin of attraction of the local maxima. However, the tracker [Comaniciu et al. 2003] only considers color information and therefore ignores other useful information such as edge and shape, resulting in the sensitivity to background clutters and occlusions. Another color-driven KGAM [Leichter et al. 2010] is developed to handle multi-view color variations by constructing the convex hull of multiple view-specific reference color histograms.

— Shape-integration KGAMs. In general, the shape-integration KGAMs aim to build a kernel density function in the joint color-shape space. For example, a shape-integration KGAM [Leichter et al. 2009] is proposed to capture the spatio-temporal properties of object appearance using color and boundary cues. It is based on two spatially normalized and rotationally symmetric kernels for describing the information about the color and object boundary.

— Scale-aware KGAMs. In essence, the scale-aware KGAMs are to capture the spatio-temporal distribution information on object appearance at multiple scales. For instance, a scale-aware KGAM [Collins 2003] using the difference of Gaussian based mean shift features is presented to cope with the problem of kernel scale selection by detecting local maxima of the Difference-of-Gaussian (DOG) scale-space filters formulated as:

$$\mathrm{DOG}(x;\sigma) = \frac{1}{2\pi\sigma^2/1.6}\exp(-\frac{\|x\|^2}{2\sigma^2/1.6}) - \frac{1}{2\pi\sigma^2(1.6)}\exp(-\frac{\|x\|^2}{2\sigma^2(1.6)}) \quad (2)$$

where σ is a scaling factor. Based on a new probabilistic interpretation, another scale-aware KGAM [Nguyen et al. 2007] is proposed to solve a maximum likelihood problem, which treats the coordinates for the pixels as random variables. As a result, the problem of kernel scale selection is converted to that of maximum likelihood optimization in the joint spatial-color space.

— Non-symmetric KGAMs. The conventional KGAMs use a symmetric kernel (e.g., a circle or an ellipse), leading to a large estimation bias in the process of estimating the complicated underlying density function. To address this issue, a non-symmetric KGAM [Yilmaz 2007] is developed based on the asymmetric kernel mean shift with adaptively varying the scale and orientation of the kernel. In contrast to the symmetric mean shift (only requiring the image coordinate estimate), the non-symmetric KGAM needs to simultaneously estimate the image coordinates, the scales, and the orientations in a few number of mean shift iterations. Introducing asymmetric kernels can generate a more accurate representation of the underlying density so that the estimation bias is reduced. Furthermore, the asymmetric kernel is just a generalization of the previous radially symmetric and anisotropic kernels.

— KGAMs by global mode seeking. Due to the local optimization property of the mean shift, large inter-frame object translations lead to tracking degradations or even failures. In order to tackle this problem, Shen et al. [2007] propose an annealed mean shift algorithm motivated by the success of the annealed importance sampling, which is essentially a way of assigning the weights to the

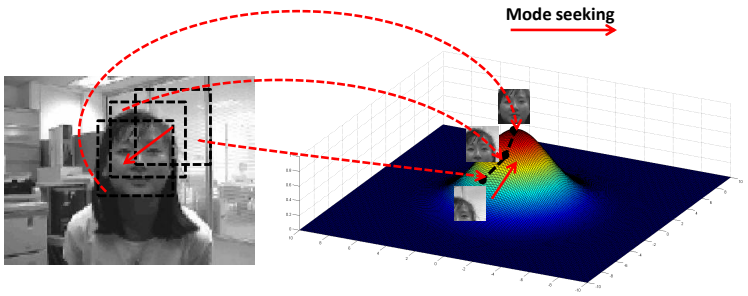

Fig. 11: Illustration of the mode seeking process by mean shift.

states obtained by multiple simulated annealing runs [Neal 2001]. Here, the states correspond to the object positions while the simulated annealing runs are associated with different bandwidths for the kernel density estimation. The proposed annealed mean shift algorithm aims to make a progressive position evolution of the mean shift as the bandwidths monotonically decrease (i.e., the convergence position of mean shift with the last bandwidth works as the initial position of the mean shift with the next bandwidth), and finally seeks the global mode.

— Sequential-kernel-learning KGAMs. Batch-mode kernel density estimation needs to store the nonparametric representations of the kernel densities, leading to a high computational and memory complexity. To address this issue, Han et al. [2008] develop a sequential kernel density approximation (SKDE) algorithm for real-time visual object tracking. The SKDE algorithm sequentially learns a nonparametric representation of the kernel density and propagates the density modes over time.

— Discussion. The color-driven kernel-based tracking algorithms mainly take the color information into consideration. However, complicated factors may give rise to drastic tracking degradations, including scale changes, background clutters, occlusions, and rapid object movements. To address this issue, various algorithmic extensions have been made. The aim of scale-aware tracking algorithms is to capture the multi-scale spatial layout information of object appearance. Thus, they are capable of effectively completing the tracking task under the circumstance of drastic scaling changes. Moreover, the edge or shape information is very helpful for accurate object localization or resisting background distraction. Motivated by this consideration, shape-driven kernel-based tracking algorithms have been developed to integrate the edge or shape information into the kernel design process. Normally, the kernel-based tracking algorithms utilize symmetric kernels (e.g., a circle or an ellipse) for object tracking, resulting in a large estimation bias for complicated underlying density functions. To tackle this problem, non-symmetric kernel-based tracking algorithms are proposed to construct a better representation of the underlying density. Conventional kernel-based tracking algorithms tend to pursue the local model seeking, resulting in tracking degradations or even failures due to their local optimization properties. To address this issue, researchers borrow ideas from both simulated annealing and annealed importance sampling to obtain a feasible solution to global mode seeking. In practice, the factors of computational complexity and memory consumption have a great effect on real-time kernel-based tracking algorithms. Thus, sequential techniques for kernel density estimation have been developed for online kernel-based tracking.

4.3. Subspace learning-based generative appearance models (SLGAMs)

In visual object tracking, a target is usually associated with several underlying subspaces, each of which is spanned by a set of basis templates. For convenience, let τ denote the target and $(\mathbf{a}_1\ \mathbf{a}_2 \ldots \mathbf{a}_N)$ denote the basis templates of an underlying subspace. Mathematically, the target τ can be linearly represented in the following form:

$$\tau = c_1\mathbf{a}_1 + c_2\mathbf{a}_2 + \cdots + c_N\mathbf{a}_N = (\mathbf{a}_1\ \mathbf{a}_2 \ldots \mathbf{a}_N)(c_1\ c_2 \ldots c_N)^T, \tag{3}$$

where $(c_1\ c_2 \ldots c_N)$ is the coefficient vector. Therefore, subspace learning-based generative appearance models (SLGAMs) focus on how to effectively obtain these underlying subspaces and their associated basis templates by using various techniques for subspace analysis. For instance, some

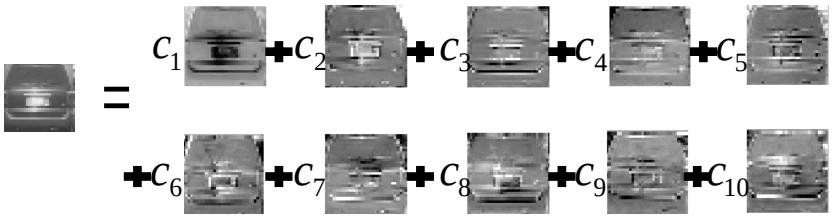

Fig. 12: Illustration of linear PCA subspace models. The left part shows a candidate sample, and the right part displays a linear combination of eigenbasis samples.

SLGAMs utilize eigenvalue decomposition or linear regression for subspace analysis, and others construct multiple subspaces to model the distribution characteristics of object appearance. According to the used techniques for subspace analysis, they can be categorized into two types: conventional and unconventional SLGAMs.

4.3.1. Conventional subspace models. In general, conventional subspace models can be split into the following two branches: linear subspace models and non-linear subspace models.

— Linear subspace models. In recent years, linear subspace models (LSMs) have been widely applied to visual object tracking. According to the dimension of the used feature space, LSL can be divided into (i) lower-order LSMs and (ii) higher-order LSMs. The lower-order LSMs [Black and Jepson 1996; Ho et al. 2004; Li et al. 2004; Skocaj and Leonardis 2003; Wen et al. 2012] needs to construct vector-based subspace models (e.g., eigenspace by principal component analysis shown in Fig. 12) while the higher-order LSMs needs to build matrix-based or tensor-based subspace models (e.g., 2D eigenspace by 2D principal component analysis and tensor eigenspace by tensor analysis).

For (i), several incremental principal component analysis (PCA) algorithms are proposed to make linear subspace models more efficient. For instance, an incremental robust PCA algorithm [Li et al. 2004] is developed to incorporate robust analysis into the process of subspace learning. Similar to [Li et al. 2004], Skocaj and Leonardis [2003] embed the robust analysis technique into the incremental subspace learning framework, which makes a sequential update of the principal subspace. The learning framework considers the weighted influence of both individual images and individual pixels within an image. Unlike the aforementioned robust PCA algorithm based on weighted residual errors, the incremental subspace learning algorithms in [Levy and Lindenbaum 2000; Brand 2002] utilize incremental singular value decomposition (SVD) to obtain a closed-form solution to subspace learning. However, these incremental PCA algorithms cannot update the sample mean during subspace learning. To address this issue, a subspace model based on R-SVD (i.e., rank-R singular value decomposition) is built with a sample mean update [Ross et al. 2008]. Moreover, Wang et al. [2012] apply partial least square analysis to learn a low-dimensional feature subspace for object tracking. In theory, the partial least square analysis is capable of modeling relations between sets of variables driven by a small number of latent factors, leading to robust object tracking results.

For (ii), a set of higher-order LSMs are proposed to address the small-sample-size problem, where the number of samples is far smaller than the dimension of samples. Therefore, many researchers begin to build matrix-based or tensor-based subspace models. For instance, Wang et al. [2007] directly analyze the 2D image matrices, and construct a 2DPCA-based appearance model for object tracking. In addition to the foreground information, they also consider background information to avoid the distractions from the background clutters. Moreover, Li et al. [2007; 2010] and Wen et al. [2009] take advantage of online tensor decomposition to construct a tensor-based appearance model for robust visual object tracking.

— Nonlinear subspace models. If the training data lie on an underlying nonlinear manifold, the LSM-based tracking algorithms may fail. Therefore, researchers attempt to employ nonlinear subspace learning to capture the underlying geometric information from target samples. For the robust human tracking, a nonlinear subspace model [Lim et al. 2006] is built using nonlinear dimension reduc-

tion techniques (i.e., Local Linear Embedding). As a nonlinear generalization of PCA, a nonlinear subspace model [Chin and Suter 2007] based on kernel principal component analysis (KPCA) is constructed to capture the kernelized eigenspace information from target samples.

4.3.2. Unconventional subspace models. In general, unconventional subspace models can also be used for visual object tracking. Roughly, they can be divided into three categories: sparse/non-sparse representation, autoregressive modeling, and multi-subspace learning.

— Sparse/non-sparse representation. Typically, a set of target samples is associated with an underlying subspace spanned by several templates. The likelihood of a candidate sample belonging to the object class is often determined by the residual between the candidate samples and the reconstructed samples derived from a linear representation. To ensure a sparse linear representation, an ℓ_1-regularized optimization procedure is adopted to obtain a sparse linear representation solution [Mei and Ling 2009]. Based on the sparse representation technique in [Mei and Ling 2009], Jia et al. [2012] propose a tracking method that further improves the tracking accuracy by using the block-division spatial pooling schemes (e.g., average pooling, max pooling, and alignment pooling). Moreover, Zhang et al. [2012] present a multi-task sparse optimization framework based on a $\ell_{p,q}$-regularized least-square minimization cost function. Instead of treating test samples independently, the framework explores the interdependencies between test samples by solving a $\ell_{p,q}$-regularized group sparsity problem. When $p = q = 1$, the framework degenerates to the popular ℓ_1 tracker [Mei and Ling 2009].

To achieve a real-time performance of the ℓ_1 tracker [Mei and Ling 2009], a subspace model [Li et al. 2011] based on compressive sensing is built by solving an orthogonal matching pursuit (OMP) optimization problem (i.e., random projections), which is about 6000 times faster than [Mei and Ling 2009]. Similar to [Li et al. 2011], Zhang et al. [2012] make use of compressive sensing (random projections) to generate a low-dimensional compressive feature descriptor, leading to a real-time tracking performance. Alternatively, Bao et al. [2012] take advantage of the popular accelerated proximal gradient (APG) approach to optimize the ℓ_1-regularized least square minimization problem, which has a quadratic convergence property to ensure the real-time tracking performance. Another way of improving the efficiency of the ℓ_1 tracker [Mei and Ling 2009] is to reduce the number of ℓ_1 minimizations in the process of evaluating test samples [Mei et al. 2011]. This task is accomplished by estimating the minimal error bound of the likelihood function in particle filtering, resulting in a moderate improvement in tracking efficiency. From an viewpoint of signal compression, Li et al. [Li et al. 2013] construct a compact 3D-DCT object representation based on a DCT subspace spanned by cosine basis functions. With the power of fast Fourier Transform (FFT), the proposed 3D-DCT object representation is capable of efficiently adapting to spatio-temporal appearance variations during tracking, leading to robust tracking results in complicated situations.

On the other hand, the sparsity of the linear representation is unnecessary for robust object tracking as long as an adequate number of template samples are provided, as pointed out in [Li et al. 2012]. Therefore, a non-sparse metric weighted linear representation (with a closed-form solution) is proposed to effectively and efficiently model the intrinsic appearance properties of the tracked object [Li et al. 2012].

— Autoregressive modeling. Since tracking is a time-dependent process, the target samples from adjacent frames are mutually correlated. To characterize the time dependency across frames, a variety of appearance models are proposed in recent years. For instance, a dynamical statistical shape representation is proposed to capture the temporal correlation information on human silhouettes from consecutive frames [Cremers 2006]. The proposed representation learns a linear autoregressive shape model, where the current silhouette is linearly constrained by the previous silhouettes. The learned shape model is then integrated into the level-set evolution process, resulting in robust segmentation results.

— Multi-subspace learning. In order to capture the distribution diversity of target samples, several efforts establish the double or multiple subspaces for visual representation. For example, Fan et al. [2008] present a bi-subspace model for visual tracking. The model simultaneously considers two visual cues: color appearance and texture appearance. Subsequently, the model uses a co-

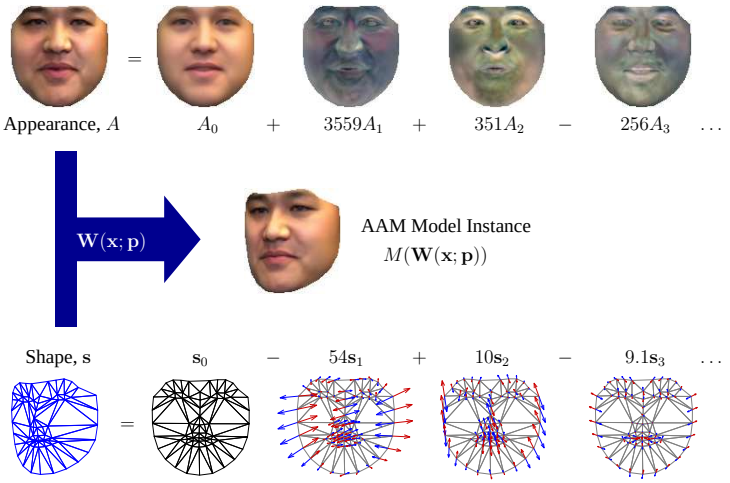

Fig. 13: Illustration of active appearance models (from [Matthews and Baker 2004], ©2004 Springer). The upper part shows that an appearance A is linearly represented by a base appearance A_0 and several appearance images; the middle part displays the piecewise affine warp $\mathbf{W}(x;\mathbf{p})$ that transforms a pixel from a base shape into the active appearance model; and the lower part exhibits that a shape s is linearly represented by a base shape s_0 and several shapes $(\mathrm{s}_i)_{i=1}^n$.

training strategy to exchange information between two visual cues. For video-based recognition and tracking, Lee and Kriegman [2005] present a generic appearance model that seeks to set up a face appearance manifold consisting of several sub-manifolds. Each sub-manifold corresponds to a face pose subspace. Furthermore, Kwon and Lee [2010] construct a set of basic observation models, each of which is associated with a specific appearance manifold of a tracked object. By combining these basic observation models, a compound observation model is obtained, resulting in a robustness to combinatorial appearance changes.

— Active appearance models (AAMs). Usually, AAMs [Hou et al. 2001; Sclaroff and Isidoro 2003; Matthews and Baker 2004] need to incorporate two components: a) shape and b) appearance, as shown in Fig. 13. For a), the shape s of an AAM can be expressed as a linear combination of a base shape s_0 and several shape vectors $(\mathrm{s}_i)_{i=1}^n$ such that $\mathrm{s} = \mathrm{s}_0 + \sum_{i=1}^n p_i \mathrm{s}_i$ where the shape s denotes $(x_1, y_1, x_2, y_2, \ldots, x_v, y_v)$ that are the coordinates of the v vertices making up the mesh. For b), the appearance of the AAM can be represented as a linear combination of a base appearance $A_0(x)$ and several appearance images $(A_i(x))_{i=1}^m$ such that $A(x) = A_0(x) + \sum_{i=1}^m \lambda_i A_i(x)$ where $x \in \mathrm{s}_0$ is a pixel lying inside the base mesh s_0. Therefore, given a test image, the AAM needs to minimize the following cost function for the model fitting:

$$\sum_{x \in \mathbf{S}_0} \left[A_0(x) + \sum_{i=1}^{m} \lambda_i A_i(x) - I(\mathbf{W}(x;\mathbf{p})) \right], \tag{4}$$

where $\mathbf{W}(x;\mathbf{p})$ denotes a piecewise affine warp that transforms a pixel $x \in \mathrm{s}_0$ into AAM.

4.3.3. Discussion. The lower-order linear subspace models (LSMs) usually learn vector-based visual representations for visual object tracking. For the tracking efficiency, several incremental LSMs (e.g., incremental PCA) are developed for online visual object tracking. Since the vector-based visual representations suffer from the small-sample-size problem, researchers construct higher-order matrix-based or tensor-based visual representations. However, the above LSMs potentially assume that object appearance samples lie on an underlying linear manifold. In practice, this assumption is often violated because of complex extrinsic/intrinsic appearance changes. Motivated by this consideration, non-linear subspace models are developed for visual representation. However, the problem with these non-linear subspace models is that they are computationally expensive due to the non-linear subspace learning (e.g., nonlinear dimension reduction).

Table IV: Summary of representative tracking-by-detection methods based on discriminative learning techniques

Item No.	References	Boosting	SVM	Randomized learning	Discriminant analysis	Codebook learning	Used Discriminative learning techniques
1	[Grabner et al. 2006] [Grabner and Bischof 2006] [Liu and Yu 2007]	Self-learning single-instance	—	—	—	—	Boosting with feature ranking-based feature selection
2	[Avidan 2007]	Self-learning single-instance	—	—	—	—	Boosting with feature weighting-based feature selection
3	[Visentini et al. 2008]	Self-learning single-instance	—	—	—	—	Dynamic ensemble based boosting
4	[Leistner et al. 2009]	Self-learning single-instance	—	—	—	—	Noise-insensitive boosting
5	[Okuma et al. 2004; Wang et al. 2005]	Self-learning single-instance	—	—	—	—	Particle filtering integration based boosting
6	[Wu et al. 2012; Luo et al. 2011]	Self-learning single-instance	—	—	—	—	Transfer learning based boosting
7	[Levin et al. 2007; Grabner et al. 2008] [Liu et al. 2009]	Co-learning single-instance	—	—	—	—	Semi-supervised co-learning boosting
8	[Babenko et al. 2009; Li et al. 2010]	Self-learning Multi-instance	—	—	—	—	Multiple instance boosting
9	[Zeisl et al. 2010]	Co-learning Multi-instance	—	—	—	—	Semi-supervised Multiple instance boosting
10	[Avidan 2004; Williams et al. 2005] [Tian et al. 2007]	—	Self-learning single-instance	—	—	—	Single SVM classifier or SVM ensemble classifiers
11	[Bai and Tang 2012]	—	Self-learning single-instance	—	—	—	Ranking SVM learning
12	[Hare et al. 2011; Yao et al. 2012]	—	Self-learning single-instance	—	—	—	Structured SVM learning
13	[Tang et al. 2007]	—	Co-learning single-instance	—	—	—	Semi-supervised SVM classifiers
14	[Saffari et al. 2009; Godec et al. 2010] [Leistner et al. 2010]	—	—	Self-learning single-instance	—	—	Random forests or Random Naive Bayes classifiers
15	[Lin et al. 2004; Nguyen and Smeulders 2006] [Li et al. 2008]	—	—	Single-modal self-learning single-instance	—	—	Fisher Linear Discriminant Analysis
16	[Wang et al. 2010; Jiang et al. 2011] [Jiang et al. 2012]	—	—	Single-modal self-learning single-instance	—	—	Discriminant metric learning
17	[Zhu and Martinez 2006; Xu et al. 2008]	—	—	—	Multi-modal self-learning single-instance	—	Subclass Discriminant Analysis
18	[Zhang et al. 2007; Zha et al. 2010]	—	—	—	Graph-driven self-learning single-instance	—	Graph embedding Graph transductive learning
19	[Collins et al. 2005]	—	—	—	—	Self-learning single-instance	Feature ranking based feature selection
20	[Gall et al. 2010]	—	—	—	—	Instance-specific codebook	Discriminative codebook learning

In recent years, unconventional subspace models have been proposed for visual object tracking. These models either enforce the sparsity constraints on the linear representation solution or have different assumptions of subspace properties. However, the sparsity-constrained linear representation typically induces a high optimization complexity, which motivates researchers to develop an efficient optimization method (e.g., APG and OMP) for a real-time tracking performance. Without the conventional single-subspace assumption, bi-subspace or multi-subspace algorithms are proposed to more precisely model the distribution diversity of the target samples, but at the cost of an additional computation.

4.4. Boosting-based discriminative appearance models

In the last decade, boosting-based discriminative appearance models (BDAMs) have been widely used in visual object tracking because of their powerful discriminative learning capabilities. According to the learning strategies employed, they can be categorized into self-learning and co-learning BDAMs. Typically, the self-learning BDAMs utilize the discriminative information from single source to guide the task of object/non-object classification, while the co-learning BDAMs exploit the multi-source discriminative information for object detection. More specifically, the self-learning BDAMs first train a classifier over the data from the previous frames, and subsequently use the trained classifier to evaluate possible object regions at the current frame. After object localization, a set of so-called "positive" and "negative" samples are selected to update the classifier. These "positive" and "negative" samples are labeled by the previously trained classifier. Due to tracking errors, the training samples obtained in the tracking process may be polluted by noise. Therefore, the labels for the training samples are unreliable. As the tracking process proceeds, the tracking error may be accumulated, possibly resulting in the "drift" problem. In contrast, the co-learning BDAMs often takes a semi-supervised strategy for object/non-object classification (e.g., co-training by building multiple classifiers).

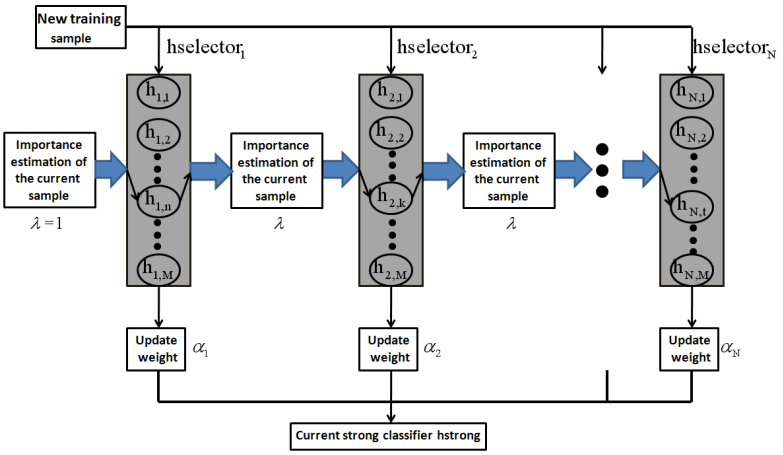

Fig. 14: Illustration of online boosting for feature selection.

On the other hand, BDAMs also take different strategies for visual representation, i.e., single-instance and multi-instance ones. The single-instance BDAMs require precise object localization. If a precise object localization is not available, these tracking algorithms may use sub-optimal positive samples to update their corresponding object or non-object discriminative classifiers, which may lead to a model drift problem. Moreover, object detection or tracking has its own inherent ambiguity, that is, precise object locations may be unknown even for human labelers. To deal with this ambiguity, the multi-instance BDAMs are proposed to represent an object by a set of image patches around the tracker location. Thus, they can be further classified into single-instance or multi-instance BDAMs.

4.4.1. Self-learning single-instance BDAMs. Based on online boosting [Oza and Russell 2001], researchers have developed a variety of computer vision applications such as object detection [Viola and Jones 2002] and visual object tracking [Grabner et al. 2006; Grabner and Bischof 2006]. In these applications, the variants of boosting are invented to satisfy different demands.

— Conventional BDAMs. As shown in Fig. 14, the conventional BDAMs first make a discriminative evaluation of each feature from a candidate feature pool, and then select the top-ranked features to conduct the tracking process [Grabner et al. 2006; Grabner and Bischof 2006]. To accelerate the feature selection process, Liu and Yu [2007] utilize gradient-based feature selection to construct a BDAM. But this BDAM requires an initial set of weak classifiers to be given in advance, leading to difficulty in general object tracking. The above-mentioned BDAMs often perform poorly in capturing the correlation information between features, leading to the redundancy of selected features and the failure to compensate for the tracking error caused by other features.
To address this issue, a feature weighting strategy is adopted to attach all the features from the feature pool with different weights, and then performs weighted fusion for object tracking. For instance, Avidan [2007] constructs a confidence map by pixel classification using an ensemble of online learned weak classifiers, which are trained by a feature weighting-based boosting approach. Since needing to store and compute all the features during feature selection, the feature weighting-based boosting approach is computationally expensive. Furthermore, Parag et al. [2008] build a feature weighting-based BDAM for object tracking, where the weak classifiers themselves are adaptively modified to adapt to scene changes. Namely, the parameters of the weak classifiers are adaptively changed instead of replacement when the new data arrive. The common property of the feature weighting-based BDAMs is that they depend on a fixed number of weak classifiers. However, this property may restrict the flexibility of the trackers in practice.
— Dynamic ensemble-based BDAMs. The conventional BDAMs need to construct a fixed number of weak learners in advance, and select these weak learners iteratively as the boosting procedure proceeds. However, due to the time-varying property of visual object tracking, they are incapable

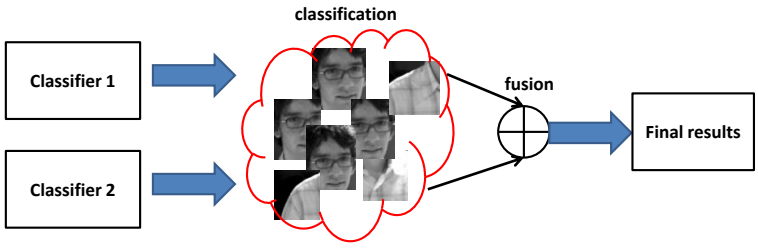

Fig. 15: Illustration of a typical co-learning problem.

of effectively adapting to dynamic object appearance changes. To address this problem, a dynamic ensemble-based BDAM [Visentini et al. 2008] is proposed to dynamically construct and update the set of weak classifiers according to the ensemble error value.

— Noise-insensitive BDAMs. To make visual object tracking more robust to noise corruption, a set of BDAMs are proposed in the literature. For instance, Leistner et al. [2009] point out that the convex loss functions typically used in boosting are highly sensitive to random noise. To enhance robustness, Leistner et al. [2009] develop a generic BDAM called online GradientBoost, which contains a set of noise insensitive loss functions. In essence, this BDAM is an extension of the GradientBoost algorithm [Friedman 2001] and works similarly to the AnyBoost algorithm [Mason et al. 1999].

— Particle filtering integration-based BDAMs. To make visual object tracking more efficient, researchers embed feature selection into the particle filtering process. For example, Wang et al. [2005] and Okuma et al. [2004] propose two online feature selection-based BDAMs using particle filtering, which generate the candidate state set of a tracked object, and the classification results of AdaBoost is used to determine the final state.

— Transfer learning-based BDAMs. Typically, most existing BDAMs have an underlying assumption that the training samples collected from the current frame follow a similar distribution to those from the last frame. However, this assumption is often violated when the "drift" problem takes place. To address the "drift" problem, a number of novel BDAMs [Wu et al. 2012; Luo et al. 2011] are proposed to categorize the samples into two classes: auxiliary samples (obtained in the last frames) and target samples (generated in the current frame). By exploring the intrinsic proximity relationships among these samples, the proposed BDAMs are capable of effectively transferring the discriminative information on auxiliary samples to the discriminative learning process using the current target samples, leading to robust tracking results.

4.4.2. Co-learning single-instance BDAMs. In general, the self-learning BDAMs suffer from the "model drift" problem due to their error accumulation caused by using the self-learning strategy. In order to address this problem, researchers adopt the semi-supervised learning techniques [Zhu 2005] for visual object tracking. For instance, Grabner et al. [2008] develop a BDAM based on semi-supervised online boosting. Its main idea is to formulate the boosting update process in a semi-supervised manner as a fused decision of a given prior and an online classier, as illustrated in Fig. 15. Subsequently, Liu et al. [2009] make use of the co-training strategy to online learn each weak classifier in boosting instead of only the final strong classifier. The co-training strategy dynamically generates a series of unlabeled samples for progressively modifying the weak classifiers, leading to the robustness to environmental changes. It is proven that the co-training strategy can minimize the boosting error bound in theory.

4.4.3. Multi-instance BDAMs. To deal with the underlying ambiguity of object localization, multiple instance learning is used for object tracking, as illustrated in Fig. 16. In principle, it represents an object by a set of image patches around the tracker location.

— Self-learning multi-instance BDAMs. For example, Babenko et al. [2009] represent an object by a set of image patches, which correspond to an instance bag with each instance being an image

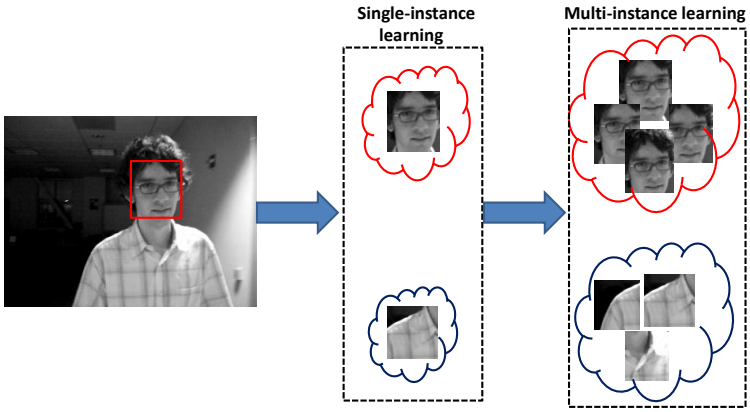

Fig. 16: Illustration of single-instance multi-instance learning. The left part shows the tracking result; the middle part displays the positive and negative samples used by the single-instance learning; and the right part exhibits the positive and negative sample bags used by the multi-instance learning.

patch. Based on online multiple instance boosting, a tracking system is developed to characterize the ambiguity of object localization in an online manner. The tracking system assumes that all positively labelled instances are truly positive, but this assumption is sometimes violated in practice. Furthermore, the tracking system trains the weak classifiers based only on the current frame, and is likely to be over-fitting. Instead of equally treating the samples in each bag [Babenko et al. 2009], Zhang et al. [2012] propose an online weighted multiple instance tracker, which incorporates the sample importance information (i.e., the samples closer to the current tracker location are of greater importance) into the online multi-instance boosting learning process, resulting in robust tracking results. To characterize the cumulative loss of the weak classifiers across multiple frames instead of the current frame, Li et al. [2010] propose an online multi-instance BDAM using the strong convex elastic net regularizer instead of the ℓ_1 regularizer, and further prove that the proposed multiple instance learning (MIL) algorithm has a cumulative regret (evaluating the cumulative loss of the online algorithm) of $\mathcal{O}(\sqrt{T})$ with T being the number of boosting iterations.

— Co-learning multi-instance BDAMs. Zeisl et al. [2010] and Li et al. [2013] combine the advantages of semi-supervised learning and multiple instance learning in the process of designing a BDAM. Semi-supervised learning can incorporate more prior information, and multiple instance learning focuses on the uncertainty about where to select positive samples for model updating.

4.4.4. Discussion. As mentioned previously, BDAMs can be roughly classified into: self-learning based and co-learning based ones. Self-learning based BDAMs adopt the self-learning strategy to learn object/non-object classifiers. They utilize previously learnt classifiers to select "positive" and "negative" training samples, and then update the current classifiers with the selected training samples. As a result, tracking errors may be gradually accumulated. In order to tackle this problem, co-learning based BDAMs are developed to capture the discriminative information from many unlabeled samples in each frame. They generally employ semi-supervised co-learning techniques to update the classifiers with both labeled and unlabeled samples in an interleaved manner, resulting in more robust tracking results.

On the other hand, conventional BDAMs take a single-instance strategy for visual representation, i.e., one image patch for each object. The drawback of this single-instance visual representation is to rely heavily on exact object localization, without which the tracking performance can be greatly degraded because of the sub-optimal training sample selection. To address this issue, MIL is introduced to visual object tracking. It takes into account of the inherent ambiguity of object localization, representing an object by a set of image patches around the tracker location. As a result, the MIL-based tracking algorithms can achieve robust tracking results, but may lose accuracy if the image patches do not precisely capture the object appearance information.

However, all BDAMs need to construct a huge local feature pool for feature selection, leading to a low computational speed. Additionally, they usually obtain a local optimal solution to object tracking because of their focus on local features rather than global features.

4.5. SVM-based discriminative appearance models (SDAMs)

SDAMs aim to learn margin-based discriminative SVM classifiers for maximizing inter-class separability. SDAMs are able to discover and remember informative samples as support vectors for object/non-object classification, resulting in a strong discriminative power. Effective kernel selection and efficient kernel computation play an importance role in designing robust SDAMs. According to the used learning mechanisms, SDAMs are typically based on self-learning SDAMs and co-learning SDAMs.

— Self-learning SDAMs. In principle, the self-learning SDAMs are to construct SVM classifiers for object/non-object classification in a self-learning fashion. For example, Avidan [2004] proposes an offline SDAM for distinguishing a target vehicle from a background. Since the SDAM needs substantial prior training data in advance, extending the algorithm to general object tracking is a difficult task. Following the work in [Avidan 2004], Williams et al. [2005] propose a probabilistic formulation-based SDAM, which allows for propagating observation distributions over time. Despite its robustness, the proposed SDAM needs to fully encode the appearance variation information, which is impractical in the tracking process. Tian et al. [2007] utilize an ensemble of linear SVM classifiers to construct a SDAM. These classifiers can be adaptively weighted according to their discriminative abilities during different periods, resulting in the robustness to large appearance variations. The above SDAMs need to heuristically select positive and negative samples surrounding the current tracker location to update the object/non-object SVM classifier.
  To avoid the heuristic and unreliable step of training sample selection (usually requiring accurate estimation of object location), two strategies are adopted in the literature. One is based on structured output support vector machine (SVM) [Hare et al. 2011; Yao et al. 2012], and the other is based on ranking SVM [Bai and Tang 2012]. The key idea of these two strategies is to integrate the structured constraints (e.g., relative ranking or VOC overlap ratio between samples) into the max-margin optimization problem. For instance, Hare et al. [2011] propose a SDAM based on a kernelized structured SVM, which involves an infinite number of structured loss (i.e., VOC overlap ratio) based constraints in the structured output spaces. In addition, Bai and Tang [2012] therefore pose visual object tracking as a weakly supervised ranking problem, which captures the relative proximity relationships between samples towards the true target samples.
— Co-learning SDAMs. In general, the co-learning SDAMs rely on semi-supervised/multi-kernel learning to construct SVM classifiers for object/non-object classification. For instance, Tang et al. [2007] adopt the co-training SVM technique to design a semi-supervised tracker. The disadvantage of this tracker is that it requires several initial frames to generate adequate labeled samples, resulting in the inflexibility in practice. Lu et al. [2010] and Yang et al. [2010] design SVM classifiers using multi-kernel learning (MKL) for visual object tracking. MKL aims to learn an optimal linear combination of different kernels based on different features, including the color information and spatial pyramid histogram of visual words.

4.5.1. Discussion. With the power of max-margin learning, the SDAMs have a good generalization capability of distinguishing foreground and background, resulting in an effective SVM classifier for object localization. However, the process of constructing the SDAMs requires a set of reliable labeled training samples, which is a difficult task due to the influence of some complicated factors such as noisy corruptions, occlusions, illumination changes, etc. Therefore, most existing SDAMs take a heuristic strategy for training sample collection (e.g., spatial distance based or classification score based), which may lead to the instability or even "drift" of the tracking process. To address this issue, the structured SVM is applied to model the structural relationships (i.e., VOC overlap ratio) between samples, resulting in a good tracking performance in terms of generalization and robustness to noise. During tracking, a hard assignment of a sample to a class label usually leads to the classification error accumulation. To alleviate the issue, the ranking SVM (a weakly supervised learning method)

is also introduced into the tracking process, where the relative ranking information between samples is incorporated into the constraints of max-margin learning.

The common point of the above SDAMs is to take a self-learning strategy for object/non-object classification without considering the discriminative information from unlabeled data or multiple information sources. Motivated by this, the co-learning SDAMs are developed to integrate such discriminative information into the SVM learning process by semi-supervised/multi-kernel learning.

the co-learning SDAMs emerge

4.6. Randomized learning-based discriminative appearance models (RLDAMs)

More recently, randomized learning techniques (e.g., Random Forest [Breiman 2001; Shotton et al. 2008; Lepetit and Fua 2006] and Ferns [Özuysal et al. 2009]) have been successfully introduced into the vision community. In principle, randomized learning techniques can build a diverse classifier ensemble by performing random input selection and random feature selection. In contrast to boosting and SVM, they are more computationally efficient, and easier to be extended for handling multi-class learning problems. In particular, they can be parallelized so that multi-core and GPU implementations (e.g., [Sharp 2008]) can be performed to greatly reduce the run time. However, their tracking performance is unstable for different scenes because of their random feature selection.

Inspired by randomized learning, a variety of RLDAMs are proposed in the field of visual object tracking, including online random forests [Saffari et al. 2009; Santner et al. 2010], random naive Bayes classifiers [Godec et al. 2010], and MIForests [Leistner et al. 2010]. For instance, Godec et al. [2010] develop a visual object tracking algorithm based on online random naive Bayes classifiers. Due to the low computational and memory costs of Random Naive Bayes classifiers, the developed tracking algorithm has a powerful real-time capability for processing long-duration video sequences. In contrast to online Random Forests [Saffari et al. 2009], the random Naive Bayes classifiers have a higher computational efficiency and faster convergence in the training phase. Moreover, Leistner et al. [2010] present a RLDAM named MIForests, which uses multiple instance learning to construct randomized trees and represents the hidden class labels inside target bags as random variables.

4.7. Discriminant analysis-based discriminative appearance models (DADAMs)

Discriminant analysis is a powerful tool for supervised subspace learning. In principle, its goal is to find a low-dimensional subspace with a high inter-class separability. According to the learning schemes used, it can be split into two branches: conventional discriminant analysis and graph-driven discriminant analysis. In general, conventional DADAMs are formulated in a vector space while graph-driven DADAMs utilize graphs for supervised subspace learning.

4.7.1. Conventional DADAMs. Typically, conventional discriminant analysis techniques can be divided into one of the following two main branches.

— Uni-modal DADAMs. In principle, uni-modal DADAMs have a potential assumption that the data for the object class follow a uni-modal Gaussian distribution. For instance, Lin et al. [2004] build a DADAM based on incremental Fisher linear discriminant analysis (IFLDA). This DADAM models the object class as a single Gaussian distribution, and models the background class as a mixture of Gaussian distributions. In [Nguyen and Smeulders 2006], linear discriminant analysis (LDA) is used for discriminative learning in the local texture feature space obtained by Gabor filtering. However, there is a potential assumption that the distributions of the object and the background classes are approximately Gaussian ones with an equal covariance. Li et al. [2008] construct a DADAM using the incremental 2DLDA on the 2D image matrices. Since matrix operations are directly made on these 2D matrices, the DADAM is computationally efficient. Moreover, another way of constructing uni-modal DADAMs is by discriminant metric learning, which aims to linearly map the original feature space to a new metric space by a linear projection [Wang et al. 2010; Jiang et al. 2011; Jiang et al. 2012]. After discriminant metric learning, the similarity between intra-class samples are minimized while the distance between inter-class samples are maximized, resulting in an effective similarity measure for robust object tracking. Note that the above DADAMs are incapable of dealing well with the object and background classes having multi-modal distributions.

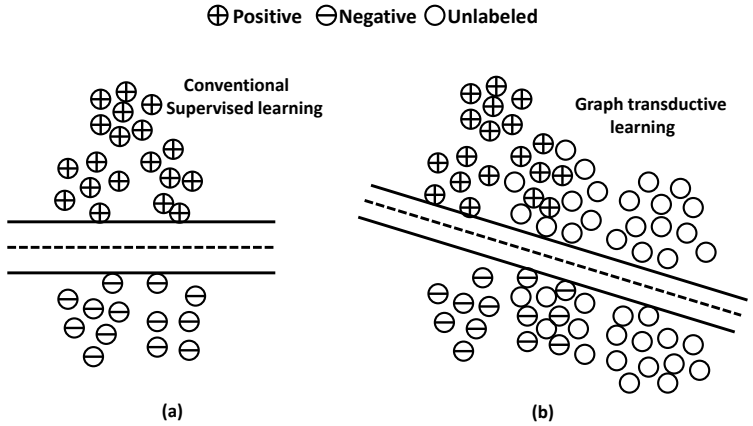

Fig. 17: Illustration of transductive learning. (a) shows the decision hyperplane obtained by the conventional supervised learning; and (b) displays the decision hyperplane (further adjusted by the unlabeled samples) of transductive learning.

— Multi-modal DADAMs. In essence, multi-modal DADAMs model the object class and the background class as a mixture of Gaussian distributions. For example, Xu et al. [2008] take advantage of adaptive subclass discriminant analysis (SDA) (i.e., an extension to the basic SDA [Zhu and Martinez 2006]) for object tracking. The adaptive SDA first partitions data samples into several subclasses by a nearest neighbor clustering, and then runs the traditional LDA for each subclass.

4.7.2. Graph-driven DADAMs. Researchers utilize the generalized graph-based discriminative learning (i.e., graph embedding and graph transductive learning) to construct a set of DADAMs for visual object tracking. Typically, these DADAMs mainly have the following two branches:

— Graph embedding based DADAMs. In principle, the goal of graph embedding based DADAMs is to set up a graph-based discriminative model, which utilizes the graph-based techniques to embed the high-dimensional samples into a discriminative low-dimensional space for the object/non-object classification. For instance, Zhang et al. [2007] design a DADAM based on graph embedding-based LDA, which makes a basic assumption that the background class is irregularly distributed with multiple modalities while the object class follows a single Gaussian distribution. However, this basic assumption does not hold true in the case of complex intrinsic and extrinsic object appearance changes.
— Graph transductive learning based DADAMs. In general, graph transductive learning based DADAMs aim to utilize the power of graph-based semi-supervised transductive learning for the likelihood evaluation of the candidate samples belonging to the object class. They make use of the intrinsic topological information between the labeled and unlabeled samples to discover an appropriate decision hyperplane for object/non-object classification, as shown in Fig. 17. For instance, Zha et al. [2010] develop a tracker based on graph-based transductive learning. The tracker utilizes the labeled samples to maximize the inter-class separability, and the unlabeled samples to capture the underlying geometric structure of the samples.

4.7.3. Discussion. The goal of DADAMs is to learn a decision hyperplane to separate the object class from the background class. However, the traditional DADAMs perform poorly when both the object class and the background class have multi-modal statistical distributions. To overcome this limitation, multi-modal discriminant analysis is adopted to explore the training data distributions by data clustering. To make a non-linear extension to the conventional DADAMs, graph-based DADAMs are proposed. These DADAMs try to formulate the problem of discriminant analysis as that of graph learning such as graph embedding and graph transductive learning. However, a draw-

Table V: Summary of representative tracking-by-detection methods using hybrid generative-discriminative learning techniques

Item No.	References	Single-layer combination	Multi-layer combination	Used learning techniques
1	[Kelm et al. 2006]	$\surd$	$\times$	Multi-conditional learning
2	[Lin et al. 2004]	$\surd$	$\times$	Combination of PCA and Fisher LDA
3	[Grabner et al. 2007]	$\surd$	$\times$	Combination of boosting and robust PCA
4	[Yang et al. 2009]	$\surd$	$\times$	Discriminative subspace learning using positive and negative data
5	[Everingham and Zisserman 2005]	$\times$	$\surd$	Combination of a tree-structured classifier and a Lambertian lighting model
6	[Shen et al. 2010]	$\times$	$\surd$	Combination of SVM learning and kernel density estimation
7	[Lei et al. 2008]	$\times$	$\surd$	Three-layer combination of relevance vector machine and GMM: learner combination (Layer 1) classifier combination (Layer 2) decision combination (Layer 3)
8	[Yu et al. 2008]	$\times$	$\surd$	Combination of the constellation model and fisher kernels

back is that these algorithms need to retain a large amount of labeled/unlabeled samples for graph learning, leading to their impracticality for real tracking applications.

4.8. Codebook learning-based discriminative appearance models (CLDAMs)

In principle, CLDAMs need to construct the foreground and background codebooks to adaptively capture the dynamic appearance information from the foreground and background. Recently, Yang et al. [2010a] construct two codebooks of image patches using two different features: RGB and LBP features, leading to the robustness in handling occlusion, scaling, and rotation. To capture more discriminative information, an adaptive class-specific codebook [Gall et al. 2010] is built for instance tracking. The codebook encodes the information on spatial distribution and appearance of object parts, and can be converted to a more instance-specific codebook in a probabilistic way (i.e., probabilistic votes for the object instance). Inspired by the tracking-by-detection idea, Andriluka et al. [2008] establish object-specific codebooks, which are constructed by clustering local features (i.e., shape context feature descriptors and Hessian-Laplace interest points) extracted from a set of training images. These codebooks are then embedded into a part-based model for pedestrian detection.

Therefore, CLDAMs often consider the discriminative information not only from the background but also from other object instances. However, it is very difficult to construct a universal codebook for different scenes or objects. As a result, it is necessary to collect different training samples for different scenes or objects, leading to inflexibility in practice. In addition, determining the codebook size is a difficult task in practice.

4.9. Hybrid generative-discriminative appearance models (HGDAMs)

As discussed in [Ulusoy and Bishop 2005], the generative and the discriminative models have their own advantages and disadvantages, and are complementary to each other to some extent. Consequently, much effort has been made to propose a variety of hybrid generative-discriminative models for combining the benefits of both the generative and the discriminative models in visual object tracking. These hybrid generative-discriminative models aim to combine the generative and the discriminative models in a single-layer or multi-layer manner.

4.9.1. HGDAMs via single-layer combination. HGDAMs via single-layer combination aim to fuse the generative and the discriminative models at the same layer. They attempt to fuse the confidence scores of the generative and the discriminative models to generate better tracking results than just using them individually. Typically, they have two kinds of combination mechanisms: decision-level combination and intermediate-level combination.

— HGDAMs via decision-level combination. In principle, such HGDAMs focus on how to effectively fuse the confidence scores from the generative and the discriminative models. For instance, a linear fusion strategy [Kelm et al. 2006] is taken to combine the log-likelihood of discriminative and generative models for pixel classification. It is pointed out in [Kelm et al. 2006] that the performance of the combined generative-discriminative models is associated with a balance between the purely generative and purely discriminative ones. In addition, Lin et al. [Lin et al. 2004] propose a HGDAM that is a generalized version of the Fisher Linear Discriminant Analysis. This HGDAM consists of two components: the observation sub-model and the discriminative sub-model.
— HGDAMs via intermediate-level combination. In principle, the HGDAMs via intermediate-level combination aim to simultaneously utilize both low-level features and high-level confidence scores from the generative and the discriminative models. For instance, Yang et al. [2009] impose three data-driven constraints on the proposed object appearance model: (1) negative data; (2) bottom-up pair-wise data constraints; and (3) adaptation dynamics. As a result, the object appearance model can greatly ameliorate the problem of adaptation drift and can achieve good tracking performances in various non-stationary scenes. Furthermore, Grabner et al. [2007] propose a HGDAM based on a boosting algorithm called Eigenboosting, which requires visual features to be discriminative with reconstructive abilities at the same time. In principle, eigenboosting aims to minimize a modified boosting error-function in which the generative information (i.e., eigenimages generated from Haar-like binary basis-functions using robust PCA) is integrated as a multiplicative prior.

4.9.2. HGDAMs via multi-layer combination. In principle, the goal of the HGDAMs via multi-layer combination is to combine the information from the generative and discriminative models at multiple layers. In general, such HGDAMs can be divided into two classes: HGDAMs via sequential combination and HGDAMs via interleaving combination.

— HGDAMs via sequential combination. In principle, the HGDAMs via sequential combination aim to fuse the benefits of the generative and discriminative models in a sequential manner. Namely, they use the decision output of one model as the input of the other model. For example, Everingham and Zisserman [Everingham and Zisserman 2005] combine generative and discriminative head models. A discriminative tree-structured classifier is trained to make efficient detection and pose estimation over a large pose space with three degrees of freedom. Subsequently, a generative head model is used for the identity verification. Moreover, Shen et al. [2010] develop a generalized kernel-based HGDAM which learns a dynamic visual representation by online SVM learning. Subsequently, the learned visual representation is incorporated into the standard MS tracking procedure. Furthermore, Lei et al. [2008] propose a HGDAM using sequential Bayesian learning. The proposed tracking algorithm consists of three modules. In the first module, a fast relevance vector machine algorithm is used to learn a discriminative classifier. In the second module, a sequential Gaussian mixture model is learned for visual representation. In the third module, a model combination mechanism with a three-level hierarchy is discussed, including the learner combination (at level one), classifier combination (at level two), and decision combination (at level three).
— HGDAMs via interleaving combination. In principle, the goal of the HGDAMs via interleaving combination is to combine the discriminative-generative information in a multi-layer interleaving manner. Namely, the decision output of one model is used to guide the learning task of the other model and vice versa. For instance, Yu et al. [2008] utilize a co-training strategy to combine the information from a SVM classifier and a generative multi-subspace model [Lee and Kriegman 2005] in a multi-layer interleaving manner.

5. BENCHMARK RESOURCES FOR VISUAL OBJECT TRACKING

To evaluate the performance of various tracking algorithms, one needs the same test video dataset, the ground truth, and the implementation of the competing tracking algorithms. Tab. VI lists the current major resources available to the public.

Another important issue is how to evaluate tracking algorithms in a qualitative or quantitative manner. Typically, qualitative evaluation is based on intuitive perception by human. Namely, if the

Table VI: Summary of the publicly available tracking resources

Item No.	Name	Dataset	Ground truth	Source code	Web link
1	Head track [Birchfield 1998]	√	×	√	www.ces.clemson.edu/∼stb/research/headtracker/seq/
2	Fragment tracker [Adam et al. 2006]	√	√	√	www.cs.technion.ac.il/∼amita/fragtrack/fragtrack.htm
3	Adaptive tracker [Jepson et al. 2003]	√	×	×	www.cs.toronto.edu/vis/projects/adaptiveAppearance.html
4	PCA tracker [Ross et al. 2008]	√	√	√	www.cs.utoronto.ca/∼dross/ivt/
5	KPCA tracker [Chin and Suter 2007]	×	×	√	cs.adelaide.edu.au/∼tjchin/
6	ℓ_1 tracker [Mei and Ling 2009]	×	×	√	www.ist.temple.edu/∼hbling/code_data.htm
7	Kernel-based tracker [Shen et al. 2010]	×	×	√	code.google.com/p/detect/
8	Boosting tracker [Grabner and Bischof 2006]	√	×	√	www.vision.ee.ethz.ch/boostingTrackers/
9	MIL tracker [Babenko et al. 2009]	√	√	√	vision.ucsd.edu/∼bbabenko/project_miltrack.shtml
10	MIForests tracker [Leistner et al. 2010]	√	√	√	www.ymer.org/amir/software/milforests/
11	Boosting+ICA tracker [Yang et al. 2010b]	×	×	√	ice.dlut.edu.cn/lu/publications.html
12	Appearance-adaptive tracker [Zhou et al. 2004]	×	×	√	www.umiacs.umd.edu/∼shaohua/sourcecodes.html
13	Tracking with histograms and articulating blocks [Nejhum et al. 2010]	√	√	√	www.cise.ufl.edu/∼smshahed/tracking.htm
14	Visual tracking decomposition [Kwon and Lee 2010]	√	√	√	cv.snu.ac.kr/research/∼vtd/
15	Structural SVM tracker [Hare et al. 2011]	×	×	√	www.samhare.net/research/struck
16	PROST tracker [Santner et al. 2010]	√	√	√	gpu4vision.icg.tugraz.at/index.php?content=subsites/prost/prost.php
17	Superpixel tracker [Wang et al. 2011]	√	√	√	faculty.ucmerced.edu/mhyang/papers/iccv11a.html
18	KLT feature tracker [Lucas and Kanade 1981]	×	×	√	www.ces.clemson.edu/∼stb/klt/
19	Deformable contour tracker [Vaswani et al. 2008]	√	×	√	home.engineering.iastate.edu/∼namrata/research/ContourTrack.html#code
20	Condensation tracker [Isard and Blake 1998]	√	×	√	www.robots.ox.ac.uk/∼misard/condensation.html
21	Motion tracking [Stauffer and Grimson 2000]	√	×	√	www.cs.berkeley.edu/∼flw/tracker/
22	Mean shift tracker	×	×	√	www.cs.bilkent.edu.tr/∼ismaila/MUSCLE/MSTracker.htm
23	Tracking-Learning-Detection Tracker	√	×	√	info.ee.surrey.ac.uk/Personal/Z.Kalal/tld.html
24	CAVIAR sequences	√	√	×	homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/
25	PETS sequences	√	√	×	www.hitech-projects.com/euprojects/cantata/datasets_cantata/dataset.html
26	SURF	×	×	√	people.ee.ethz.ch/∼surf/download_ac.html
27	XVision visual tracking	×	×	√	peipa.essex.ac.uk/info/software.html
28	The Machine Perception Toolbox	×	×	√	mplab.ucsd.edu/grants/project1/free-software/MPTWebSite/introduction.html
29	Compressive Tracker [Zhang et al. 2012]	√	√	√	www4.comp.polyu.edu.hk/∼cslzhang/CT/CT.htm
30	Structural local sparse tracker [Jia et al. 2012]	√	√	√	ice.dlut.edu.cn/lu/Project/cvpr12_jia_project/cvpr12_jia_project.htm
31	Sparsity-based collaborative tracker [Zhong et al. 2012]	√	√	√	ice.dlut.edu.cn/lu/Project/cvpr12_scm/cvpr12_scm.htm
32	Multi-task sparse tracker [Zhang et al. 2012]	×	×	√	sites.google.com/site/zhangtianzhu2012/publications
33	APG ℓ_1 tracker [Bao et al. 2012]	√	√	√	www.dabi.temple.edu/∼hbling/code_data.htm#L1_Tracker
34	Structured keypoint tracker [Hare et al. 2012]	√	√	√	www.samhare.net/research/keypoints
35	Spatial-weighted MIL tracker [Zhang and Song 2012]	×	×	√	code.google.com/p/online-weighted-miltracker/

calculated target regions cover more true object regions and contain fewer non-object pixels, the tracking algorithms are considered to achieve better tracking performances; otherwise, the tracking algorithms perform worse. For a clear illustration, a qualitative comparison of several representative visual representations is provided in Tab. VII in terms of computational speed as well as handling occlusion, illumination variation, and shape deformation capabilities. Moreover, Tab. VIII provides a qualitative comparison of several representative statistical modeling-based appearance models in terms of computational speed, memory usage, online adaptability, and discriminability.

In contrast, a quantitative evaluation relies heavily on the ground truth annotation. If objects of interest are annotated with bounding boxes, a quantitative evaluation is performed by computing the positional errors of four corners between the tracked bounding boxes and the ground truth. Alternatively, the overlapping ratio between the tracked bounding boxes (or ellipses) and the ground truth can be calculated for the quantitative evaluation: $r = \frac{A_t \bigcap A_g}{A_t \bigcup A_g}$, where A_t is the tracked bounding box (or ellipse) and A_g is the ground truth. The task of ground truth annotation with bounding boxes or ellipses is difficult and time-consuming. Consequently, researchers take a point-based annotation

Table VII: Qualitative comparison of visual representations (Symbols $\surd$ and $\times$ mean that the visual representation can or cannot cope with the situations of occlusions, illumination changes, and shape deformations, respectively.)

Item No.	Visual representations	What to track	Speed	Occlusion	Illumination	Shape deformation
1	Vector-based raw pixel representation [Ross et al. 2008]	rectangle	high	$\times$	$\times$	$\times$
2	Matrix-based raw pixel representation [Li et al. 2007]	rectangle	high	$\times$	$\times$	$\times$
3	Multi-cue raw pixel representation (i.e., color, position, edge) [Wang et al. 2007]	rectangle	moderate	$\surd$	$\times$	$\times$
4	Multi-cue spatial-color histogram representation (i.e., joint histogram in (x, y, R, G, B)) [Georgescu and Meer 2004]	rectangle	high	$\times$	$\times$	$\surd$
5	Multi-cue spatial-color histogram representation (i.e., patch-division histogram) [Adam et al. 2006]	rectangle	high	$\surd$	$\times$	$\surd$
6	covariance representation [Porikli et al. 2006; Li et al. 2008] [Hu et al. 2012; Wu et al. 2012]	rectangle	moderate	$\times$	$\surd$	$\surd$
7	Wavelet filtering-based representation [Li et al. 2009]	rectangle	slow	$\surd$	$\surd$	$\surd$
8	[Cremers 2006; Sun et al. 2011] Active contour representation	contour	slow	$\surd$	$\times$	$\surd$
9	Local feature-based represnetation (local templates) [Lin et al. 2007]	rectangle	moderate	$\surd$	$\surd$	$\surd$
10	Local feature-based represnetation (MSER features) [Tran and Davis 2007]	irregular regions	slow	$\surd$	$\times$	$\surd$
11	[Zhou et al. 2009] Local feature-based represnetation (SIFT features)	interest points	slow	$\surd$	$\surd$	$\surd$
12	Local feature-based represnetation (SURF features) [He et al. 2009]	interest points	moderate	$\surd$	$\surd$	$\surd$
13	Local feature-based represnetation (Corner features) [Kim 2008]	interest points	moderate	$\surd$	$\surd$	$\surd$
14	Local feature-based represnetation (Saliency detection-based features) [Fan et al. 2010]	saliency patches	slow	$\surd$	$\surd$	$\surd$

Table VIII: Qualitative comparison of representative statistical modeling based appearance models.

Item No.	Statistical modeling-based appearance models	Domain	Speed	Memory usage	Online adaptability	Discriminability
1	Linear subspace models	manifold learning	fast	low	strong	weak
2	Nonlinear subspace models	manifold learning	slow	high	weak	moderate
3	Mixture models	Parametric density estimation	moderate	low	strong	moderate
4	Kernel-based models	Nonparametric density estimation	fast	low	weak	weak
5	Boosting-based appearance models	ensemble learning	moderate	low	strong	strong
6	SVM-based appearance models	Maximum margin learning	slow	high	strong	strong
7	Randomized learning based appearance models	classifier ensemble based on random input selection and random feature selection	fast	high	strong	weak
8	Discriminant analysis based appearance models	supervised subspace learning	fast	low	strong	weak
9	Codebook learning based appearance models	Vector quantization	slow	high	strong	strong

strategy for the quantitative evaluation. Specifically, they either record object center locations as the ground truth for simplicity and efficiency, or mark several points within the object regions by hand as the ground truth for accuracy (e.g., seven mark points are used in the dudek face sequence [Ross et al. 2008]). This way, we can compute the positional residuals between the tracking results and the ground truth for the quantitative evaluation.

6. CONCLUSION AND FUTURE DIRECTIONS

In this work, we have presented a survey of 2D appearance models for visual object tracking. The presented survey takes a module-based organization to review the literature of two important modules in 2D appearance models: visual representations and statistical modeling schemes for tracking-by-detection, as shown in Fig. 3. The visual representations focus more on how to robustly describe the spatio-temporal characteristics of object appearance, while the statistical modeling schemes for

tracking-by-detection put more emphasis on how to capture the generative/discriminative statistical information of the object regions. These two modules are closely related and interleaved with each other. In practice, powerful appearance models depend on not only effective visual representations but also robust statistical models.

In spite of a great progress in 2D appearance models in recent years, there are still several issues remaining to be addressed:

— Balance between tracking robustness and tracking accuracy. Existing appearance models are incapable of simultaneously guaranteeing tracking robustness and tracking accuracy. To improve the tracking accuracy, more visual features and geometric constraints are incorporated into the appearance models, resulting in a precise object localization in the situations of particular appearance variations. However, these visual features and geometric constraints can also lower the generalization capabilities of the appearance models in the aspect of undergoing other appearance variations. On the other hand, to improve the tracking robustness, the appearance models relax some constraints on a precise object localization, and thus allow for more ambiguity of the object localization. Thus, balancing tracking robustness and tracking accuracy is an interesting research topic.
— Balance between simple and robust visual features. In computer vision, designing both simple and robust visual features is one of the most fundamental and important problems. In general, simple visual features have a small number of components. As a result, they are computationally efficient, but have a low discriminability. In contrast, robust visual features often have a large number of components. Consequently, they are computationally expensive, and have sophisticated parameter settings. Therefore, how to keep a good balance between simplicity and robustness plays an important role in visual object tracking.
— 2D and 3D information fusion. 2D appearance models are computationally efficient and simple to implement. Due to the information loss of 3D-to-2D projections, 2D appearance models cannot accurately estimate the poses of the tracked objects, leading to the sensitivity to occlusion and out-of-plane rotation. In contrast, 3D appearance models are capable of precisely characterizing the 3D pose of the tracked objects, resulting in the robustness to occlusion and out-of-plane rotation. However, 3D appearance models require a large parameter-search space for 3D pose estimation, resulting in expensive computational costs. Therefore, combining the advantages of 2D and 3D appearance models is a challenging research topic. To accelerate the pose estimation process of the 3D appearance models, a possible solution is to use the tracking results of the 2D appearance models as the initialization of the 3D appearance models. However, how to effectively transfer from 2D tracking to 3D tracking is still an unsolved problem.
— Intelligent vision models. Inspired by the biological vision, a number of high-level salient region features are proposed to capture the salient semantic information of an input image. These salient region features are relatively stable during the process of tracking, while they rely heavily on salient region detection which may be affected by noise or drastic illumination variation. Unreliable saliency detection leads to many feature mismatches across frames. Consequently, it is necessary to build an intelligent vision model that can robustly track these salient region features across frames like what human vision offers.
— Camera network tracking. Typically, the appearance models are based on a single camera, which provides a very limited visual information of the tracked objects. In recent years, several appearance models using multiple overlapping cameras are proposed to fuse different visual information from different viewpoints. These appearance models usually deal with the problem of object tracking in the same scene monitored by different cameras. Often they cannot complete the tracking task of the same object in different but adjacent scenes independently. In this case, tracking in a large camera network needs to be established for a long-term monitoring of the objects of interest. However, how to transfer the target information from one camera sub-network to another is a crucial issue that remains to be solved.
— Low-frame-rate tracking. Due to the hardware limits of processing speed and memory usage, mobile devices or micro embedded systems usually produces the video data with a low frame rate (e.g., abrupt object motion), which makes the tracking job challenging. In this situation, the appearance

models needs to have a good generalization and adaptation capability of online coping with the object appearance variations during tracking. Therefore, it is crucial to construct a robust appearance model with efficient visual modeling and effective statistical modeling for real-time applications.

REFERENCES

ADAM, A., RIVLIN, E., AND SHIMSHONI, I. 2006. Robust fragments-based tracking using the integral histogram. *IEEE International Conference on Computer Vision and Pattern Recognition*, 798–805.

ALLILI, M. S. AND ZIOU, D. 2007. Object of interest segmentation and tracking by using feature selection and active contours. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–8.

ANDRILUKA, M., ROTH, S., AND SCHIELE, B. 2008. People-tracking-by-detection and people-detection-by-tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–8.

ARSIGNY, V., FILLARD, P., PENNEC, X., AND AYACHE, N. 2006. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM Journal on Matrix Analysis and Applications 29,* 1, 328–347.

ARULAMPALAM, M. S., MASKELL, S., GORDON, N., AND CLAPP, T. 2002. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Trans. on Signal Processing 50,* 2, 174–188.

AUSTVOLL, I. AND KWOLEK, B. 2010. Region covariance matrix-based object tracking with occlusions handling. *Computer Vision and Graphics 6374/2010*, 201–208.

AVIDAN, S. 2004. Support vector tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence 26,* 8, 1064–1072.

AVIDAN, S. 2007. Ensemble tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence 29,* 2, 261–271.

BABENKO, B., YANG, M., AND BELONGIE, S. 2009. Visual tracking with online multiple instance learning. *IEEE International Conference on Computer Vision and Pattern Recognition*, 983–990.

BAI, Y. AND TANG, M. 2012. Robust tracking via weakly supervised ranking svm. *IEEE Conference on Computer Vision and Pattern Recognition*, 1854–1861.

BAO, C., WU, Y., LING, H., AND JI, H. 2012. Real time robust l1 tracker using accelerated proximal gradient approach. *IEEE Conference on Computer Vision and Pattern Recognition*, 1830–1837.

BAY, H., TUYTELAARS, T., AND GOOL, L. V. 2006. Surf: Speeded up robust features. *European Conference on Computer Vision*, 404–417.

BERGEN, J., BURT, P., HINGORANI, R., AND PELEG, S. 1992. A three-frame algorithm for estimating two-component image motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence 14,* 9, 886–896.

BIRCHFIELD, S. 1998. Elliptical head tracking using intensity gradients and color histograms. *IEEE International Conference on Computer Vision and Pattern Recognition*, 232–237.

BIRCHFIELD, S. AND RANGARAJAN, S. 2005. Spatiograms vs. histograms for region based tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1158–1163.

BLACK, M. AND ANANDAN, P. 1996. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding 63*, 75–104.

BLACK, M. J. AND JEPSON, A. D. 1996. Eigentracking: Robust matching and tracking of articulated objects using view-based representation. *European Conference on Computer Vision*, 329–342.

BRADSKI, G. 1998. Real time face and object tracking as a component of a perceptual user interface. *IEEE Workshop on Applications of Computer Vision*, 214–219.

BRAND, M. 2002. Incremental singular value decomposition of uncertain data with missing values. *European Conference on Computer Vision*, 707–720.

BREIMAN, L. 2001. Random forests. *Machine Learning 45,* 1, 5–32.

CANDAMO, J., SHREVE, M., GOLDGOF, D. B., SAPPER, D. B., AND KASTURI, R. 2010. Understanding transit scenes: A survey on human behavior-recognition algorithms. *IEEE Trans. Intelligent Transportation Systems 11,* 1, 206–224.

CANNONS, K. 2008. A review of visual tracking. Technical report, York University CSE-2008-07.

CHIN, T. J. AND SUTER, D. 2007. Incremental kernel principal component analysis. *IEEE Trans. on Image Processing 16,* 6, 1662–1674.

COIFMAN, B., BEYMER, D., MCLAUCHLAN, P., AND MALIK, J. 1998. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C 6,* 4, 271–288.

COLLINS, R. 2003. Mean-shift blob tracking through scale space. *IEEE International Conference on Computer Vision and Pattern Recognition 2*, 234–240.

COLLINS, R. T., LIPTON, A. J., KANADE, T., FUJIYOSHI, H., DUGGINS, D., TSIN, Y., TOLLIVER, D., ENOMOTO, N., HASEGAWA, O., BURT, P., AND WIXSON, L. 2000. A system for video surveillance and monitoring. Technical report cmu-ri-tr-00-12, VSAM final report, Carnegie Mellon University.

COLLINS, R. T., LIU, Y., AND LEORDEANU, M. 2005. Online selection of discriminative tracking features. *IEEE Trans. on Pattern Analysis and Machine Intelligence 27,* 10, 1631–1643.

COMANICIU, D., RAMESH, V., AND MEER, P. 2003. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence 25,* 5, 564–577.

CREMERS, D. 2006. Dynamical statistical shape priors for level set based tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence 28*, 1262–1273.

DALAL, N. AND TRIGGS, B. 2005. Histograms of oriented gradients for human detection. *IEEE International Conference on Computer Vision and Pattern Recognition 1*, 886–893.

DONOSER, M. AND BISCHOF, H. 2006. Efficient maximally stable extremal region (mser) tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, 553–560.

ELLIS, L., DOWSON, N., MATAS, J., AND BOWDEN, R. 2010. *Linear Regression and Adaptive Appearance Models for Fast Simultaneous Modelling and Tracking*. International Journal of Computer Vision.

EVERINGHAM, M. AND ZISSERMAN, A. 2005. Identifying individuals in video by combining 'generative' and discriminative head models. *IEEE International Conference on Computer Vision*, 1103–1110.

FAN, J., WU, Y., AND DAI, S. 2010. Discriminative spatial attention for robust tracking. *European Conference on Computer Vision*, 480–493.

FAN, J., YANG, M., AND WU, Y. 2008. A bi-subspace model for robust visual tracking. *International Conference on Image Processing*, 2260–2663.

FORSYTH, D. A., ARIKAN, O., IKEMOTO, L., O'BRIEN, J., AND RAMANAN, D. 2006. Computational studies of human motion: Part 1, tracking and motion synthesis. *Found. Trends Comput. Graph. Vis 1,* 2, 77–254.

FRIEDMAN, J. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics 29,* 5, 1189–1232.

GALL, J., RAZAVI, N., AND GOOL, L. V. 2010. On-line adaption of class-specific codebooks for instance tracking. *British Machine Vision Conference*.

GELZINIS, A., VERIKAS, A., AND BACAUSKIENE, M. 2007. Increasing the discrimination power of the co-occurrence matrix-based features. *Pattern Recognition 40,* 9, 2367–2372.

GEORGESCU, B. AND MEER, P. 2004. Point matching under large image deformations and illumination changes. *IEEE Trans. on Pattern Analysis and Machine Intelligence 26*, 674–689.

GERÓNIMO, D., LÓPEZ, A. M., SAPPA, A. D., AND GRAF, T. 2010. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence 32,* 7, 1239–1258.

GODEC, M., LEISTNER, C., SAFFARI, A., AND BISCHOF, H. 2010. On-line random naive bayes for tracking. *International Conference on Pattern Recognition*, 3545–3548.

GRABNER, H. AND BISCHOF, H. 2006. On-line boosting and vision. *IEEE International Conference on Computer Vision and Pattern Recognition*, 260–267.

GRABNER, H., GRABNER, M., AND BISCHOF, H. 2006. Real-time tracking via on-line boosting. *British Machine Vision Conference*, 47–56.

GRABNER, H., LEISTNER, C., AND BISCHOF, H. 2008. *Semi-supervised On-line Boosting for Robust Tracking*. Vol. 5302/2008.

GRABNER, H., ROTH, P. M., AND BISCHOF, H. 2007. Eigenboosting: Combining discriminative and generative information. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–8.

GRABNER, M., GRABNER, H., AND BISCHOF, H. 2007. Learning features for tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–8.

HAGER, G. AND BELHUMEUR, P. 1998. Efficient region tracking with paramteric models of geometry and illumination. *IEEE Trans. on Pattern Analysis and Machine Intelligence 20,* 10, 1125–1139.

HAN, B., COMANICIU, D., ZHU, Y., AND DAVIS, L. S. 2008. Sequential kernel density approximation and its application to real-time visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence 30,* 7, 1186–1197.

HAN, B. AND DAVIS, L. 2005. On-line density-based appearance modeling for object tracking. *IEEE International Conference on Computer Vision*, 1492–1499.

HARALICK, R., SHANMUGAM, K., AND DINSTEIN, I. 1973. Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics 3,* 6, 610–621.

HARE, S., SAFFARI, A., AND TORR, P. 2011. Struck: Structured output tracking with kernels. *IEEE International Conference on Computer Vision*, 263–270.

HARE, S., SAFFARI, A., AND TORR, P. 2012. Efficient online structured output learning for keypoint-based object tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, 1894–1901.

HARITAOGLU, I. AND FLICKNER, M. 2001. Detection and tracking of shopping groups in stores. *IEEE International Conference on Computer Vision and Pattern Recognition 1*, 431–438.

HARITAOGLU, I., HARWOOD, D., AND DAVIS, L. 2000. W4: Real-time surveillance of people and their activities. *IEEE Trans. on Pattern Analysis and Machine Intelligence 22,* 8, 809–830.

HE, C., ZHENG, Y., AND AHALT, S. 2002. Object tracking using the gabor wavelet transform and the golden section algorithm. *IEEE Transactions on Multimedia 4,* 4, 528–538.

HE, W., YAMASHITA, T., LU, H., AND LAO, S. 2009. Surf tracking. *IEEE International Conference on Computer Vision*, 1586–1592.

HO, J., LEE, K., YANG, M., AND KRIEGMAN, D. 2004. Visual tracking using learned linear subspaces. *IEEE International Conference on Computer Vision and Pattern Recognition*, 782–789.

HONG, X., CHANG, H., SHAN, S., ZHONG, B., CHEN, X., AND GAO, W. 2010. Sigma set based implicit online learning for object tracking. *IEEE Signal Processing Letters 17,* 9, 807–810.

HORN, B. K. P. AND SCHUNCK, B. G. 1981. Determining optical flow. *Artificial Intelligence 17*, 185–203.

HOU, X., LI, S., ZHANG, H., AND CHENG, Q. 2001. Direct appearance models. *IEEE International Conference on Computer Vision and Pattern Recognition 1*, 828–833.

HU, W., LI, X., LUO, W., ZHANG, X., MAYBANK, S., AND ZHANG, Z. 2012. Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. *IEEE Trans. on Pattern Analysis and Machine Intelligence*.

HU, W., LI, X., ZHANG, X., SHI, X., MAYBANK, S., AND ZHANG, Z. 2010. Incremental tensor subspace learning and its applications to foreground segmentation and tracking. *International Journal of Computer Vision 91,* 3, 303–327.

HU, W., TAN, T., WANG, L., AND MAYBANK, S. 2004. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. on Syst., Man, Cybern. C, Appl. Rev 34,* 3, 334–352.

IRANI, M. 1999. Multi-frame optical flow estimation using subspace constraints. *IEEE International Conference on Computer Vision*, 626–633.

ISARD, M. AND BLAKE, A. 1998. Condensation-conditional density propagation for tracking. *International Journal of Computer Vision 29,* 1, 2–28.

JAVED, O., SHAFIQUE, K., RASHEED, Z., AND SHAH, M. 2008. Modeling inter-camera space–time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding 109,* 2, 146–162.

JEPSON, A. D., FLEET, D. J., AND EL-MARAGHI, T. F. 2003. Robust online appearance models for visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence 25,* 10, 1296–1311.

JIA, X., LU, H., AND YANG, M. 2012. Visual tracking via adaptive structural local sparse appearance model. *IEEE Conference on Computer Vision and Pattern Recognition*, 1822–1829.

JIANG, N., LIU, W., AND WU, Y. 2012. Order determination and sparsity-regularized metric learning adaptive visual tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, 1956–1963.

JIANG, N., SU, H., LIU, W., AND WU, Y. 2011. Tracking low resolution objects by metric preservation. *IEEE Conference on Computer Vision and Pattern Recognition*, 1329–1336.

KALMAN, R. 1960. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering 82,* 1, 35–45.

KANG, W. AND DENG, F. 2007. Research on intelligent visual surveillance for public security. *IEEE/ACIS Int. Conf. Comput. Inf. Sci*, 824–829.

KELM, B. M., PAL, C., AND MCCALLUM, A. 2006. Combining generative and discriminative methods for pixel classification with multi-conditional learning. *International Conference on Pattern Recognition*, 828–832.

KIM, Z. 2008. Real time object tracking based on dynamic feature grouping with background subtraction. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–8.

KWON, J. AND LEE, K. M. 2010. Visual tracking decomposition. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1269–1276.

LEE, K. AND KRIEGMAN, D. 2005. Online learning of probabilistic appearance manifolds for video-based recognition and tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, 852–859.

LEI, Y., DING, X., AND WANG, S. 2008. Visual tracker using sequential bayesian learning: Discriminative, generative, and hybrid. *IEEE Trans. on System, Man, and Cybernetics-Part B: Cybernetics 38,* 6, 1578–1591.

LEICHTER, I., LINDENBAUM, M., AND RIVLIN, E. 2009. Tracking by affine kernel transformations using color and boundary cues. *IEEE Trans. on Pattern Analysis and Machine Intelligence 31,* 1, 164–171.

LEICHTER, I., LINDENBAUM, M., AND RIVLIN, E. 2010. Mean shift tracking with multiple reference color histograms. *114*.

LEISTNER, C., SAFFARI, A., AND BISCHOF, H. 2010. Multiple-instance learning with randomized trees. *European Conference on Computer Vision*, 29–42.

LEISTNER, C., SAFFARI, A., ROTH, P. M., AND BISCHOF, H. 2009. On robustness of on-line boosting - a competitive study. *International Conference on Computer Vision Workshops*, 1362–1369.

LEPETIT, V. AND FUA, P. 2006. Keypoint recognition using randomized trees. *IEEE Trans. on Pattern Analysis and Machine Intelligence 28,* 9, 1465–1479.

LEVIN, A., VIOLA, P., AND FREUND, Y. 2007. Unsupervised improvement of visual detectors using cotraining. *IEEE International Conference on Computer Vision*, 626–633.

LEVY, A. AND LINDENBAUM, M. 2000. Sequential karhunen-loeve basis extraction and its application to images.

LI, G., HUANG, Q., QIN, L., AND JIANG, S. 2013. Ssocbt: A robust semi-supervised online covboost tracker by using samples differently. *IEEE Trans. on Circuits and Systems and Video Technology*.

LI, G., LIANG, D., HUANG, Q., JIANG, S., AND GAO, W. 2008. Object tracking using incremental 2d-lda learning and bayes inference. *International Conference on Image Processing*, 1568–1571.

LI, H., SHEN, C., AND SHI, Q. 2011. Real-time visual tracking with compressed sensing. *IEEE International Conference on Computer Vision and Pattern Recognition*.

LI, M., KWOK, J. T., AND LU, B.-L. 2010. Online multiple instance learning with no regret. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1395–1401.

LI, M., ZHANG, Z., HUANG, K., AND TAN, T. 2009. Robust visual tracking based on simplified biologically inspired features. *International Conference on Image Processing*, 4113–4116.

LI, X., DICK, A., SHEN, C., VAN DEN HENGEL, A., AND WANG, H. 2013. Incremental learning of 3d-dct compact representations for robust visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*.

LI, X., HU, W., ZHANG, Z., ZHANG, X., AND LUO, G. 2007. Robust visual tracking based on incremental tensor subspace learning. *IEEE International Conference on Computer Vision*, 1–8.

LI, X., HU, W., ZHANG, Z., ZHANG, X., ZHU, M., AND CHENG, J. 2008. Visual tracking via incremental log-euclidean riemannian subspace learning. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–8.

LI, X., SHEN, C., SHI, Q., DICK, A., AND VAN DEN HENGEL, A. 2012. Non-sparse linear representations for visual tracking with online reservoir metric learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 1760–1767.

LI, Y., XU, L., MORPHETT, J., AND JACOBS, R. 2004. On incremental and robust subspace learning. *Pattern Recognition 37,* 7, 1509–1518.

LIM, H., MORARIU, V. I., CAMPS, O. I., AND SZNAIER, M. 2006. Dynamic appearance modeling for human tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, 751–757.

LIN, R., ROSS, D., LIM, J., AND YANG, M.-H. 2004. Adaptive discriminative generative model and its applications. *IEEE International Conference on Computer Vision and Pattern Recognition*, 801–808.

LIN, R.-S., YANG, M.-H., AND LEVINSON, S. E. 2004. Object tracking using incremental fisher discriminant analysis. *International Conference on Pattern Recognition*, 757–760.

LIN, Z., DAVIS, L., DOERMANN, D., AND DEMENTHON, D. 2007. Hierarchical part-template matching for human detection and segmentation. *IEEE International Conference on Computer Vision*, 1–8.

LIU, R., CHENG, J., AND LU, H. 2009. A robust boosting tracker with minimum error bound in a co-training framework. *IEEE International Conference on Computer Vision*, 1459–1466.

LIU, X. AND YU, T. 2007. Gradient feature selection for online boosting. *IEEE International Conference on Computer Vision*, 1–8.

LU, H., ZHANG, W., AND CHEN, Y. 2010. On feature combination and multiple kernel learning for object tracking. *Asian Conference on Computer Vision*.

LUCAS, B. D. AND KANADE, T. 1981. An iterative image registration technique with an application in stereo vision. *International Joint Conferences on Artificial Intelligence*, 674–679.

LUO, W., LI, X., LI, W., AND HU, W. 2011. Robust visual tracking via transfer learning. *IEEE International Conference on Image Processing*, 485–488.

MAHADEVAN, V. AND VASCONCELOS, N. 2009. Saliency-based discriminant tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1007–1013.

MASON, L., BAXTER, J., BARTLETT, P., AND FREAN, M. 1999. *chapter "Functional Gradient Techniques for Combining Hypotheses"*. MIT Press, Cambridge, MA. 221-247.

MASOUD, O. AND PAPANIKOLOPOULOS, N. P. 2001. A novel method for tracking and counting pedestrians in real-time using a single camera. *IEEE Trans. on Vehicular Technology 50,* 5, 1267–1278.

MATTHEWS, I. AND BAKER, S. 2004. Active appearance models revisited. *International Journal of Computer Vision 60,* 2, 135–164.

MCKENNA, S., RAJA, Y., AND GONG, S. 1999. Tracking colour objects using adaptive mixture models. *Image and Vision Computing 17*, 223–229.

MEI, X. AND LING, H. 2009. Robust visual tracking using ℓ_1 minimization. *IEEE International Conference on Computer Vision*, 1436–1443.

MEI, X., LING, H., WU, Y., BLASCH, E., AND BAI, L. 2011. Minimum error bounded efficient l1 tracker with occlusion detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 1257–1264.

NEAL, R. 2001. Annealed importance sampling. *Statistics and Computing 11,* 2, 125–139.

NEJHUM, S. M. S., HO, J., AND YANG, M. H. 2010. *Online Visual Tracking with Histograms and Articulating Blocks*. Computer Vision and Image Understanding.

NGUYEN, H. T. AND SMEULDERS, A. W. M. 2004. Tracking aspects of the foreground against the background. *European Conference on Computer Vision*, 446–456.

NGUYEN, H. T. AND SMEULDERS, A. W. M. 2006. Robust tracking using foreground-background texture discrimination. *International Journal of Computer Vision 69,* 3, 277–293.

NGUYEN, Q. A., ROBLES-KELLY, A., AND SHEN, C. 2007. Kernel-based tracking from a probabilistic viewpoint. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–8.

NING, J., ZHANG, L., ZHANG, D., AND WU, C. 2009. Robust object tracking using joint color-texture histogram. *International Journal of Pattern Recognition and Artificial Intelligence 23,* 7, 1245–1263.

OJALA, T., PIETIKÄINEN, M., AND MÄENPÄÄ, T. 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence 24,* 7, 971–987.

OKUMA, K., TALEGHANI, A., FREITAS, N. D., LITTLE, J. J., AND LOWE, D. 2004. A boosted particle filter: Multitarget detection and tracking. *European Conference on Computer Vision*, 28–39.

OZA, N. AND RUSSELL, S. 2001. Online bagging and boosting. *Artificial Intelligence and Statistics*, 105–112.

ÖZUYSAL, M., CALONDER, M., LEPETIT, V., AND FUA, P. 2009. Fast keypoint recognition using random ferns. *IEEE Trans. on Pattern Analysis and Machine Intelligence 32,* 3, 448–461.

PALMER, S. E. 1999. *Vision Science: Photons to Phenomenology*. The MIT Press.

PARAG, T., PORIKLI, F., AND ELGAMMAL, A. 2008. Boosting adaptive linear weak classifiers for online learning and tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*.

PARAGIOS, N. AND DERICHE, R. 2000. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence 22*, 266–280.

PASCHALAKIS, S. AND BOBER, M. 2004. Real-time face detection and tracking for mobile videoconferencing. *Real-Time Imaging 10,* 2, 81–94.

PAVLOVIE, V. I., SHARMA, R., AND HUANG, T. 1997. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence 19,* 7, 677–695.

PORIKLI, F. 2005. Integral histogram: A fast way to extract histograms in cartesian spaces. *IEEE International Conference on Computer Vision and Pattern Recognition 1*, 829–836.

PORIKLI, F., TUZEL, O., AND MEER, P. 2006. Covariance tracking using model update based on lie algebra. *IEEE International Conference on Computer Vision and Pattern Recognition*, 728–735.

REN, X. AND MALIK, J. 2007. Tracking as repeated figure/ground segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8.

ROSS, D., LIM, J., LIN, R., AND YANG, M. 2008. Incremental learning for robust visual tracking. *International Journal of Computer Vision 77,* 1-3, 125–141.

SAFFARI, A., LEISTNER, C., SANTNER, J., GODEC, M., AND BISCHOF, H. 2009. On-line random forests. *IEEE International Conference on Computer Vision Workshops*, 1393–1400.

SALARI, V. AND SETHI, I. K. 1990. Feature point correspondence in the presence of occlusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence 12,* 1, 87–91.

SANTNER, J., LEISTNER, C., SAFFARI, A., POCK, T., AND BISCHOF, H. 2010. Prost: Parallel robust online simple tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, 723–730.

SAWHNEY, H. AND AYER, S. 1996. Compact representations of videos through dominant and multiple motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence 18*, 814–830.

SCLAROFF, S. AND ISIDORO, J. 2003. Active blobs: region-based, deformable appearance models. *Computer Vision and Image Understanding 89,* 2, 197–225.

SETHI, I. K. AND JAIN, R. 1987. Finding trajectories of feature points in monocular image sequence. *IEEE Trans. on Pattern Analysis and Machine Intelligence 9,* 1, 56–73.

SHARP, T. 2008. Implementing decision trees and forests on a gpu. *European Conference on Computer Vision*, 595–608.

SHEN, C., BROOKS, M. J., AND VAN DEN HENGEL, A. 2007. Fast global kernel density mode seeking: Applications to localization and tracking. *IEEE Trans. on Image Processing 16,* 5, 1457–1469.

SHEN, C., KIM, J., AND WANG, H. 2010. Generalized kernel-based visual tracking. *IEEE Trans. Circuits and Systems for Video Technology*, 119–130.

SHOTTON, J., JOHNSON, M., AND CIPOLLA, R. 2008. Semantic texton forests for image categorization and segmentation. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–8.

SIKORA, T. 1997. The mpeg-4 video standard verification model. *IEEE Trans. on Circuits and Systems for Video Technology 7,* 1, 19–31.

SILVEIRA, G. AND MALIS, E. 2007. Real-time visual tracking under arbitrary illumination changes. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–6.

SIVIC, J., SCHAFFALITZKY, F., AND ZISSERMAN, A. 2006. Object level grouping for video shots. *International Journal of Computer Vision 67,* 2, 189–210.

SKOCAJ, D. AND LEONARDIS, A. 2003. Weighted and robust incremental method for subspace learning.

STAUFFER, C. AND GRIMSON, W. 2000. Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence 22,* 8, 747–757.

SUN, X., YAO, H., AND ZHANG, S. 2011. A novel supervised level set method for non-rigid object tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, 3393–3400.

SUN, Z., BEBIS, G., AND MILLER, R. 2006. On-road vehicle detection: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence 28,* 5, 694–711.

TAI, J., TSANG, S., LIN, C., AND SONG, K. 2004. Real-time image tracking for automatic traffic monitoring and enforcement application. *Image and Vision Computing 22,* 6, 485–501.

TANG, F., BRENNAN, S., ZHAO, Q., AND TAO, H. 2007. Co-tracking using semi-supervised support vector machines. 1–8.

TANG, F. AND TAO, H. 2008. Probabilistic object tracking with dynamic attributed relational feature graph. *IEEE Trans. on Circuits and Systems for Video Technology 18,* 8, 1064–1074.

TIAN, M., ZHANG, W., AND LIU, F. 2007. On-line ensemble svm for robust object tracking. *Asian Conference on Computer Vision*, 355–364.

TOYAMA, K. AND HAGER, G. D. 1996. Incremental focus of attention for robust visual tracking. *International Journal of Computer Vision*, 189–195.

TRAN, S. AND DAVIS, L. 2007. Robust object tracking with regional affine invariant features. *IEEE International Conference on Computer Vision*, 1–8.

TUZEL, O., PORIKLI, F., AND MEER, P. 2006. Region covariance: A fast descriptor for detection and classification. *European Conference on Computer Vision*, 589–600.

ULUSOY, I. AND BISHOP, C. 2005. Generative versus discriminative methods for object recognition. *IEEE International Conference on Computer Vision and Pattern Recognition*, 258–265.

VASWANI, N., RATHI, Y., YEZZI, A., AND TANNENBAUM, A. 2008. Pf-mt with an interpolation effective basis for tracking local contour deformations. *IEEE Trans. on Image Processing 19,* 4, 841–857.

VIOLA, P. AND JONES, M. 2002. Robust real-time object detection. *International Journal of Computer Vision 57,* 2, 137–154.

VISENTINI, I., SNIDARO, L., AND FORESTI, G. L. 2008. Dynamic ensemble for target tracking. *International Workshop on Visual Surveillance*.

WANG, H., SUTER, D., SCHINDLER, K., AND SHEN, C. 2007. Adaptive object tracking based on an effective appearance filter. *IEEE Trans. on Pattern Analysis and Machine Intelligence 29,* 9, 1661–1667.

WANG, J., CHEN, X., AND GAO, W. 2005. Online selecting discriminative tracking features using particle filter. *IEEE International Conference on Computer Vision and Pattern Recognition 2*, 1037–1042.

WANG, J. AND YAGI, Y. 2008. Integrating color and shape-texture features for adaptive real-time object tracking. *IEEE Trans. Image Processing 17,* 2, 235–240.

WANG, Q., CHEN, F., XU, W., AND YANG, M. 2012. Object tracking via partial least squares analysis. *IEEE Transactions on Image Processing 21,* 10, 4454–4465.

WANG, S., LU, H., YANG, F., AND YANG, M. 2011. Superpixel tracking. *IEEE International Conference on Computer Vision*.

WANG, T., GU, I. Y. H., AND SHI, P. 2007. Object tracking using incremental 2d-pca learning and ml estimation. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 933–936.

WANG, X., HUA, G., AND HAN, T. 2010. Discriminative tracking by metric learning. *European Conference on Computer Vision*, 200–214.

WEN, J., GAO, X., LI, X., AND TAO, D. 2009. Incremental learning of weighted tensor subspace for visual tracking. *IEEE International Conference on Systems, Man, and Cybernetics*, 3788–3793.

WEN, L., CAI, Z., LEI, Z., YI, D., AND LI, S. 2012. Online spatio-temporal structural context learning for visual tracking. *European Conference on Computer Vision*, 716–729.

WERLBERGER, M., TROBIN, W., POCK, T., WEDEL, A., CREMERS, D., AND BISCHOF, H. 2009. Anisotropic huber-l_1 optical flow. *British Machine Vision Conference*.

WILLIAMS, O., BLAKE, A., AND CIPOLLA, R. 2005. Sparse bayesian learning for efficient visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence 27*, 1292–1304.

WOLFE, J. M. 1994. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin & Review 1,* 2, 202–238.

WU, S., ZHU, Y., AND ZHANG, Q. 2012. A new robust visual tracking algorithm based on transfer adaptive boosting. *Mathematical Methods in the Applied Sciences*.

WU, Y., CHENG, J., WANG, J., AND LU, H. 2009. Real-time visual tracking via incremental covariance tensor learning. *IEEE International Conference on Computer Vision*, 1631–1638.

WU, Y., CHENG, J., WANG, J., LU, H., WANG, J., LING, H., BLASCH, E., AND BAI, L. 2012. Real-time probabilistic covariance tracking with efficient model update. *IEEE Transactions on Image Processing 21,* 5, 2824–2837.

WU, Y. AND FAN, J. 2009. Contextual flow. *IEEE International Conference on Computer Vision and Pattern Recognition*, 33–40.

WU, Y., WU, B., LIU, J., AND LU, H. 2008. Probabilistic tracking on riemannian manifolds. *International Conference on Pattern Recognition*, 1–4.

XU, Z., SHI, P., AND XU, X. 2008. Adaptive subclass discriminant analysis color space learning for visual tracking. *The 9th Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing*, 902–905.

YANG, C., DURAISWAMI, R., AND DAVIS, L. 2005. Efficient mean-shift tracking via a new similarity measure. *IEEE International Conference on Computer Vision and Pattern Recognition*, 176–183.

YANG, F., LU, H., AND CHEN, Y. 2010a. Bag of features tracking. *International Conference on Pattern Recognition*.

YANG, F., LU, H., AND CHEN, Y. 2010b. Robust tracking based on boosted color soft segmentation and ica-r. *International Conference on Image Processing*.

YANG, F., LU, H., AND WEI CHEN, Y. 2010. Human tracking by multiple kernel boosting with locality affinity constraints. *Asian Conference on Computer Vision*.

YANG, M., FAN, Z., FAN, J., AND WU, Y. 2009. Tracking nonstationary visual appearances by data-driven adaptation. *IEEE Trans. on Image Processing 18,* 7, 1633–1644.

YANG, M., YUAN, J., AND WU, Y. 2007. Spatial selection for attentional visual tracking. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–7.

YAO, R., SHI, Q., SHEN, C., ZHANG, Y., AND VAN DEN HENGEL, A. 2012. Robust tracking with weighted online structured learning. *European Conference on Computer Vision*.

YILMAZ, A. 2007. Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1–6.

YILMAZ, A., JAVED, O., AND SHAH, M. 2006. Object tracking: A survey. *ACM Computing Survey 38,* 4, 1–45.

YU, Q., DINH, T. B., AND MEDIONI, G. 2008. Online tracking and reacquisition using co-trained generative and discriminative trackers. *European Conference on Computer Vision*, 678–691.

YU, T. AND WU, Y. 2006. Differential tracking based on spatial-appearance model(sam). *IEEE International Conference on Computer Vision and Pattern Recognition*, 720–727.

ZEISL, B., LEISTNER, C., SAFFARI, A., AND BISCHOF, H. 2010. On-line semi-supervised multiple-instance boosting. *IEEE International Conference on Computer Vision and Pattern Recognition*, 1879–1886.

ZHA, Y., YANG, Y., AND BI, D. 2010. Graph-based transductive learning for robust visual tracking. *Pattern Recognition 43*, 187–196.

ZHAN, B., MONEKOSSO, N. D., REMAGNINO, P., VELASTIN, S. A., AND XU, L.-Q. 2008. Crowd analysis: A survey. *Mach. Vis. Appl 19,* 5, 345–357.

ZHANG, K. AND SONG, H. 2012. Real-time visual tracking via online weighted multiple instance learning. *Pattern Recognition*.

ZHANG, K., ZHANG, L., AND YANG, M. 2012. Real-time compressive tracking. *European Conference on Computer Vision*.

ZHANG, T., GHANEM, B., LIU, S., AND AHUJA, N. 2012. Robust visual tracking via multi-task sparse learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 2042–2049.

ZHANG, X., HU, W., MAYBANK, S., AND LI, X. 2007. Graph-based discriminative learning for robust and efficient object tracking. *IEEE International Conference on Computer Vision*, 1–8.

ZHAO, Q., YANG, Z., AND TAO, H. 2010. Differential earth mover's distance with its applications to visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence 32,* 2, 274–287.

ZHONG, W., LU, H., AND YANG, M. 2012. Robust object tracking via sparsity-based collaborative model. *IEEE Conference on Computer Vision and Pattern Recognition*, 1838–1845.

ZHOU, H., YUAN, Y., AND SHI, C. 2009. Object tracking using sift features and mean shift. *Computer Vision and Image Understanding*, 345–352.

ZHOU, S. K., CHELLAPPA, R., AND MOGHADDAM, B. 2004. Visual tracking and recognition using appearance-adaptive models in particle filters. *IEEE Trans. on Image Processing 13*, 1491–1506.

ZHU, M. AND MARTINEZ, A. M. 2006. Subclass discriminant analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence 28*, 1274–1286.

ZHU, X. 2005. Semi-supervised learning literature survey. *Technical Report, Computer Sciences, University of Wisconsin-Madison*.