

Toward situation awareness: a survey on adaptive learning for model-free tracking

Xinpeng L. Liao¹ · Chengcui Zhang¹

Received: 1 January 2016 / Revised: 21 September 2016 / Accepted: 23 September 2016 /
Published online: 22 October 2016
© Springer Science+Business Media New York 2016

Abstract Visual tracking estimates the trajectory of an object of interest in non-stationary image streams that change over time. Recently, approaches for model-free tracking have received increased interest since manually annotating sufficient examples of all objects in the world is prohibitively expensive. By definition, a model-free tracker has only one labeled instance in the form of an identified object in the first frame. In the subsequent frames, it has to learn variations of the tracked object with only unlabeled data available. There exists a dilemma for model-free trackers, i.e., whether the tracker would shift the focus to clutters (i.e., adaptivity) or result in very short tracks (i.e., stability) largely depends on how sensitive the appearance model is. In contrast to recent survey efforts with data-driven approaches focusing on the performance on benchmarks, this article aims to provide an in-depth survey on solutions to the dilemma between adaptivity and stability in model-free tracking focusing on the ability of achieving situation awareness, i.e., learning the object appearance adaptively in a non-stationary environment. The survey results show that, regardless of visual representations and statistical models involved, the way of exploiting unlabeled data in the changing environment and the extent of how rapidly the appearance model need be updated accordingly with selected example(s) of estimated labels are the key to many, if not all, evaluation measures for tracking. Such conceptual consensuses, despite the diversity of approaches in this field, for the first time capture the essence of model-free tracking and facilitate the design of visual tracking systems.

Keywords Computer vision · Machine learning · Model-free tracking · Semi-supervised online learning · Video surveillance

✉ Xinpeng L. Liao
xinpeng@uab.edu

Chengcui Zhang
czhang02@uab.edu

¹ Department of Computer and Information Sciences, The University of Alabama at Birmingham, 127 Campbell Hall, 1300 University Boulevard, Birmingham AL 35294-1170, USA

1 Introduction

Visual tracking refers to automatic estimation of trajectory of an object as it moves around in a video, which plays a role in almost every video analysis task, e.g., motion analysis, event detection and activity understanding. In recent years, the development of visual tracking algorithms has enjoyed rapid progress in terms of methodology and applications. In methodology, visual tracking receives huge attention from researchers not only because it inherently needs a wide range of computational mathematics tools, e.g., statistical theory, optimization, and numerical analysis, but also resides in the core intersection of computer vision, robotics, machine learning, intelligent systems and related fields. As a middle-level vision problem, it also has many applications, including video surveillance for security and forensics, human-computer interaction, intelligent transportation system, medical imaging, mobile robotics, film post-production and sports video analysis, etc. Although existing techniques may offer satisfactory solutions to this problem in well-controlled environments, designing robust tracking methods is still an open issue in many practical applications due to factors such as partial occlusion, clutter background, fast and abrupt motion, dramatic illumination changes and large variations in viewpoint and pose. The activity in this field is reflected in abundance of new tracking algorithms presented in journals, e.g., *IJCV*, and especially at high-profile conferences, such as, *ICCV*, *CVPR*, and *ECCV*. Developing robust visual tracking algorithms to solve a large range of practical applications will remain an active research topic in a foreseeable future.

1.1 Overview of visual tracking

Despite the abundance of various tracking algorithms, the most widely accepted framework [102] for a visual tracking system usually comprises three main components, which had recently been decomposed into five major components (Fig. 1) in a finer granularity [83]:

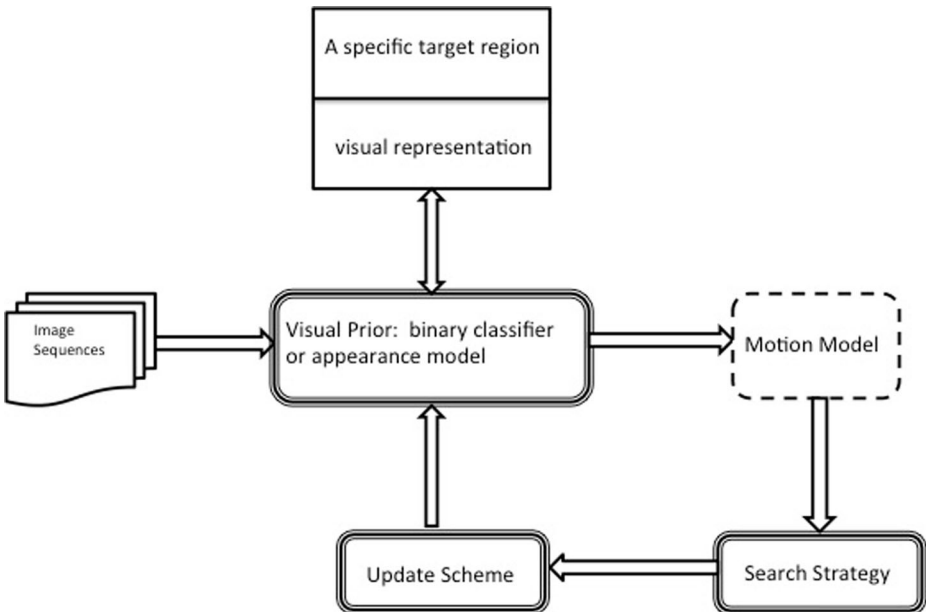


Fig. 1 The general framework of a visual tracking system

- A specific target region, which is often represented by a target bounding box, an ellipse, a contour of the target, a blob, a patch or interest points;
- A chosen representation of local appearance or an appearance model that captures the likelihood that the object is present at a particular location;
- A representation of motion or a location model that predicts the prior probability that the object is present at a particular location;
- A search strategy for finding the maximum a posteriori location of the object;
- An updating scheme for the target’s model in order to deal with the variation in the appearance.

Tracking algorithms were summarized in many survey papers in the last 15 years [33, 34, 43, 61, 68, 69, 102]. The most influential tracking survey is the work of 2006 [102] that describes methodologies on tracking, features and data association for general purposes. Another good work [61] provides a detailed review of the existing 2D appearance models focusing on decomposing the problem of appearance modeling into two different processing stages: visual representation and statistical modeling. In terms of statistical modeling, most existing trackers adopt either the generative or discriminative approach [61]. Generative trackers, like other generative models in machine learning, assume that certain generative process can describe the object being tracked and hence tracking corresponds to finding the most probable candidate among possibly infinitely many. On the other hand, the discriminative approach treats tracking as a binary classification problem, which learns to explicitly distinguish the object being tracked from its background [92]. In particular, the tracking problem is to model $P(X_t|Z_{1:t})$ — the posterior probability over the current joint configuration of the targets X_t at the current time step t , given all the observations Z_1, Z_2, \dots, Z_t up to that time instant. Assume the independence of conditions on Z_1, Z_2, \dots, Z_t , we have $P(X_t|Z_{1:t}) = P(X_t|Z_t)$. Figure 2 shows the difference between a typical generative model and a discriminative model for tracking. It all boils down to whether a generative process is needed. Figure 2a is a fragment of graphical model representation of generative tracker, in which the formulation of $P(X_t|Z_t)$ is derived from $P(X_t|X_{t-1})$ and $P(Z_t|X_t)$ via the application of Bayes rule. Figure 2b, however, directly models $P(X_t|Z_t)$ as a simple binary

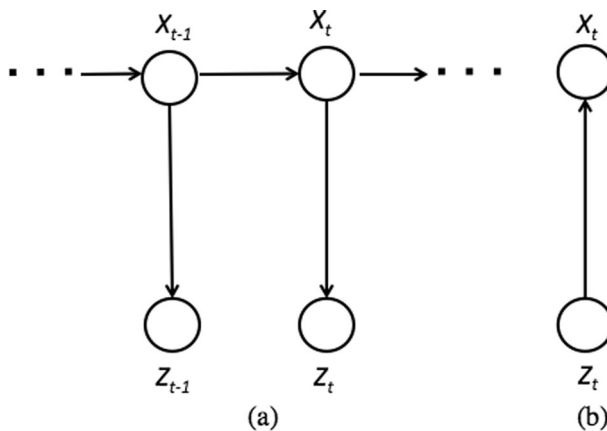


Fig. 2 (a) Generative approach adopts a generative process, e.g., a Bayesian recursive formulation, to model $P(X_t|Z_t)$, while (b) discriminative approach models the posterior probability $P(X_t|Z_t)$ directly

classification problem predicting the binary label X_t with Z_t as the features extracted in a video frame at time instant t .

1.2 Motivation: data-driven approach vs. conceptual consensus

The computer vision community recently has shifted its focus to benchmarking (Table 1) since this field suffers from a lack of established methodology for objective comparison. An important paper [98] initiated this shift by performing a large-scale benchmark of several trackers and developed an evaluation kit that allows integration of other trackers as well, which was later criticized by the computer vision community for lack of standardization of the input/output communication between the trackers and the evaluation kit. To address this issue, the Visual Object Tracking (VOT) workshop has been organized in conjunction with ICCV each year since 2013. The VOT2013, VOT2014 and VOT2015 challenges, which aimed at single-object visual trackers that do not apply pre-learned models of object appearance (model-free), have received many submissions from researchers both in academia and industry. For the competition results, interested readers can refer to these papers from [51–54].

However, little efforts (Fig. 3) were made to sort through the gigantic literature with conceptual consensus since Yilmaz's work in 2006 [102]. Although a general framework was widely adopted, component-wise approaches are very diversified and specifically tailored to limited video sequences at hand in each paper. More recent advances are even outside of this framework, being completely a game changer, e.g., circulant trackers. The lack of conceptual consensus of the diversified approaches in this field is a strong indicator that visual tracking, other than improving performance on specific video sequences with tailored approaches, is still an open issue in general. The importance of conceptual consensus of solutions to a research problem does not merely echo the craving for truth underlying the physical world, but also provides guidance to system design of visual tracking in real-world, thereby contributing to the advancement of this field. Due to the rapid growth of papers published in this field, more recent surveys seem to have already given up on finding such conceptual consensus since many different and varying circumstances need to be reconciled in one algorithm. For example, in 2013, Li's survey [61] focused on only two components in a visual tracking system with the other three components untouched. Their survey decomposed the problem into two separate modules: visual representation and statistical modeling. However, it is already difficult, if not entirely impossible, to enumerate

Table 1 Summary of common benchmarking datasets

No.	Datasets	Sources
1	PETS 2015	http://pets2015.net/
2	CAVIAR	http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1
3	i-LIDs	http://www.homeoffice.gov.uk/science-research/hosdb/i-lids
4	ETISEO	http://www-sop.inria.fr/orion/ETISEO
5	CVBASE	http://vision.fe.uni-lj.si/cvbase06/
6	VIVID	http://vision.cse.psu.edu/data/vividEval/main.html
7	OTB	http://cvlab.hanyang.ac.kr/tracker_benchmark/index.html
8	ALOV	http://www.alov300.org/
9	VOT	http://www.votchallenge.net/

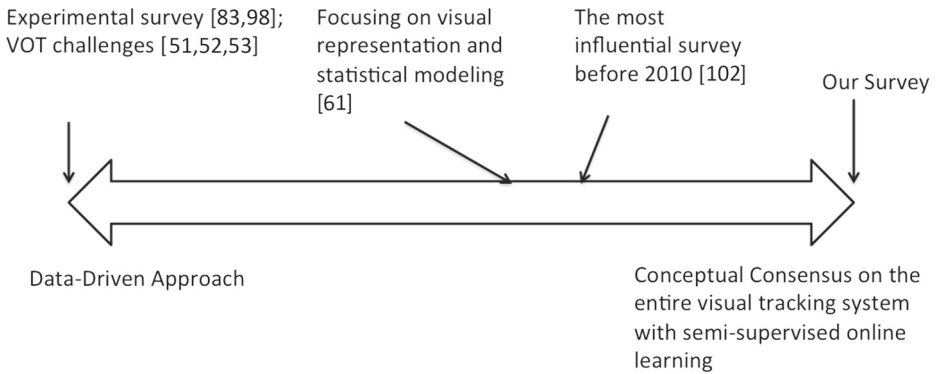


Fig. 3 The relationship between this survey and currently existing surveys

different combinations of visual representation and statistical models that lead to different tracker performance, let alone adding the other additional components into the combination. In 2014–2015, an experimental survey [83] and ICCV VOT challenges completely went for the other extreme of the spectrum, using a data-driven approach in the comparison work. In our article, however, the goal of investigation is to discover such conceptual consensus of solutions regardless of the data difficulty encountered.

1.3 Survey scope and methodology

Recently, approaches for model-free tracking have received increased interest since manually annotating sufficient examples of all objects in the world is prohibitively expensive and time-consuming. In this survey, we focus on model-free trackers without considering other tracking methods, e.g., pre-trained trackers or off-line trackers. By definition, model-free tracking has only one labeled instance in the form of an identified object in the first video frame. In the subsequent frames, the tracker has to learn variations of the tracked object with only unlabeled data available. It should be noted that there exists a dilemma for model-free trackers: 1) if the appearance model is more sensitive to the tracking environment than it is supposed to be, the tracker would drift off the target or shift the focus to an object that is not even a target; 2) if the appearance model is too stubborn to embrace the rapidly changing environment, the tracker would lose the track too easily. Therefore, the way of detecting the changing environment and the extent of how rapidly the appearance model need be updated accordingly are the key to many, if not all, evaluation measures for tracking. The fundamental problem is to robustly integrate data derived during tracking into the model without drifting. The essence in this context is actually a semi-supervised online learning problem, which would affect all the five components of visual tracking systems.

The survey methodology is mainly motivated by the following two observations. First, the way of exploiting unlabeled data and updating the appearance model with selected example(s) of estimated labels for model-free tracker is critical, irrespective of what visual representations and statistical modeling techniques involved. Previously most trackers were evaluated on a limited few video sequences of the author's choice; therefore most papers on visual tracking more or less biased their results toward the selected scenarios with the selected visual representations and statistical models. With the advent of more objective benchmarks since 2013 [53], it is imperative to re-examine all the trackers in the literature.

Based on the recent VOT challenge results as in Table 2, the trackers with top performance literally have no consensus on the visual representations or learning machines, but share similar training sample selection strategies. This is the major observation that was missed by other existing surveys [61, 83, 102]. Given the continued growth of literature in tracking, it is highly desirable to share among the visual tracking community the summarization and taxonomy of recent advances from the perspective of semi-supervised online learning, in particular, the way of exploring unlabeled data and model update scheme. To our best knowledge, this is the first time visual tracking is thoroughly analyzed in this new dimension. Second, some misconceptions exist and new insights are needed based on the rapid progress in this field. For instance, in terms of statistical modeling, Li's survey [61] employed "tracking-by-detection" as an umbrella concept to accommodate all the variants both in generative and discriminative trackers. However, this is not appropriate because "tracking-by-detection" usually makes use of an object detector, i.e., a binary classifier, to detect objects in a video frame and makes association between subsequent frames with spatial-temporal constraints, which typically falls into the category of discriminative trackers. This article is devoted to reviewing the literature from the perspective of semi-supervised online learning to complement existing surveys and help the reader swiftly learn the state-of-the-art in this field.

1.4 The goals of investigation and achieved survey results

In order to cope with variations of the object that are not known a priori, visual tracking in general can be formulated as a semi-supervised online learning problem in that 1) it starts

Table 2 Categorizing 10 Top-performing trackers out of 37 in the most recent VOT challenges prior to 2015

Tracker	Framework	Tracking method	Sample selection
DSST	Self-Learning	Discriminative (Circulant tracker, scale adaptive)	Dense sampling
SAMF	Self-Learning	Discriminative (Circulant tracker, scale adaptive)	Dense sampling
KCF	Self-Learning	Discriminative (Circulant tracker, not scale adaptive)	Dense sampling
DGT	Self-Learning	Hybrid (discriminative SVM and generative color histogram)	Structural constraints (graph-based representation)
eASMS	Self-Learning	Generative (Scale adaptive Mean Shift)	Target representation with background weighting
MCT	Co-Training	Hybrid (discriminative model integrated with particle filter)	Structural constraints (Sampling from contextual motion density)
qwsEDFT	Self-Learning	Generative (Enhanced version of DF tracker)	Distribution field (DF) selection and updating
ACT	Self-Learning	Discriminative	Low-dimensional color attribute selection and updating
LGT	Co-Training	Hybrid	Structural constraints (part-based representation)
Struck	Self-Learning	Discriminative (Structured output SVM)	Structural constraints (No sampling, avoided by estimating the transformation between frames)

by training an initial classifier from a labeled training set with discriminative trackers, or by initializing/training an appearance model based on a labeled training set with generative trackers; 2) then the classifier is evaluated on the upcoming unlabeled data to decide whether it is an object or not in the case of discriminative trackers, while with generative trackers the appearance model is used to predict the likelihood that an object is at present in a particular location in the next video frame; 3) The classifier or the appearance model needs be updated accordingly as new visual information arrived. In both scenarios, the semi-supervised online learning serves as the conceptual consensus regardless of visual representations and the selected learning models. Some researchers [36, 47, 49, 84, 104] exploited visual tracking in this way explicitly only for “tracking-by-detection” methods, while most other trackers, irrespective of being discriminative or generative, focused on the design of visual representation, appearance modeling, motion prior, and the associated optimization problems, e.g., the maximum likelihood estimation for appearance model or the computation of optimal decision boundary for a binary classifier. In this survey, we argue that beyond “tracking-by-detection” methods, almost all the existing model-free trackers can roughly, if not perfectly, be interpreted in the sense of semi-supervised online learning. In order to provide conceptual consensus to this field, this survey is attempting to answer the following questions:

- What are the existing techniques for semi-supervised online trackers in terms of the way of exploiting unlabeled data?
- Why the semi-supervised online learning was primarily investigated in the case of discriminative trackers? Is it possible to relate this conceptual scheme with generative trackers and how can we find the common ground between the two categories?
- How the semi-supervised online learning affects the design of visual representation, the appearance model/object detector, and the motion prior? What components in a visual tracking system enhance the semi-supervised online learning by striking a balance between adaptivity and stability?
- What are the advantages of finding the conceptual consensus on semi-supervised online learning over the data-driven approaches for visual tracking? Would this conceptual consensus really help the tracker achieve situation awareness? What are the future directions for this open problem?

As shown in Fig. 4, the contribution of this survey is as follows: 1) We categorized and connected the literature body roughly based on self-learning and co-training — the two typical modes for semi-supervised online learning; since the work on co-training is relatively few, we naturally focus more on the sub-categories of self-learning; 2) For self-learning discriminative trackers, the criterion to differentiate these approaches is via sample selection strategies before those samples are collected for model update, i.e., random sampling, sampling within some structural constraints and dense sampling, leading to a better taxonomy that makes sense from the perspective of semi-supervised online learning; 3) For self-learning generative trackers, we summarized the two major mechanisms: predict-update and direct optimization, which correspond to two large categories — probabilistic trackers and kernel-based trackers. However, a large amount of these generative trackers are not adaptive due to fixed appearance models that are not naturally designed for updating. Toward this end, the appearance models, irrespective of probabilistic trackers or kernel-based trackers, need be designed using specific visual representations, e.g., sparse subspace representation and distribution field; 4) Despite the discrepancy of terminologies between discriminative and generative trackers, we attempted to find the conceptual consensus, e.g., subspace basis selection or distribution field selection for generative trackers may share some

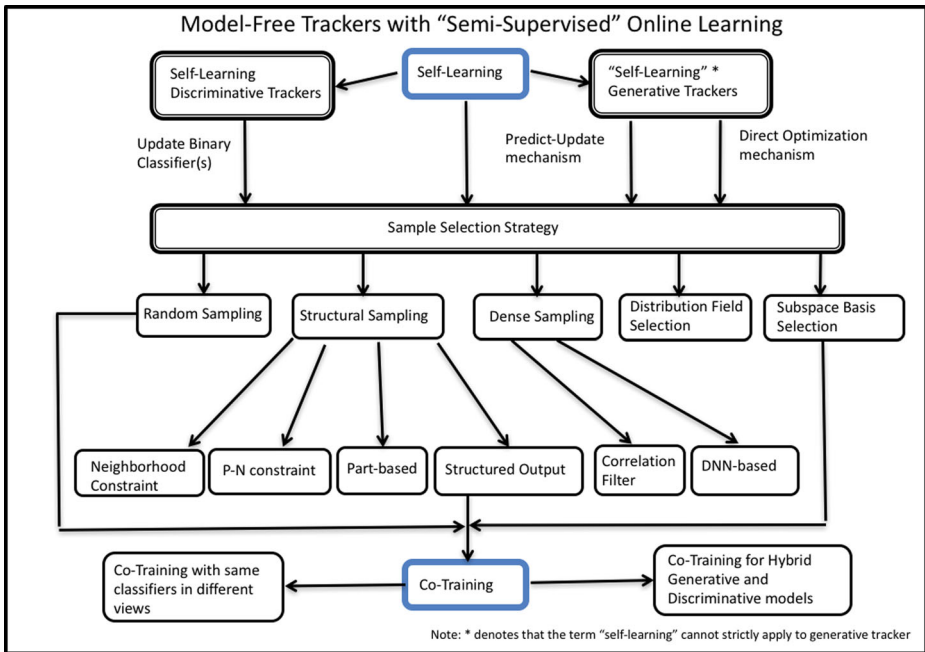


Fig. 4 The organization of this survey

similarity with the sample selection strategy for discriminative trackers in spirit. Since, these approaches are not all mutually exclusive, this conceptual consensus is rewarding for us to discuss the design considerations for a typical online adaptive tracking system, and may help researchers advance this field; 5) We summarized the two categories of co-training, i.e., co-training with the same classifiers and co-training for hybrid generative and discriminative models. The main constituents of such hybrid models usually have overlaps with subspace basis selection based trackers, random sampling based trackers and part-based trackers (see Fig. 4); 6) We summarized typical time complexity analysis, the major applications, design considerations and the open issues of model-free tracking. We also pointed out future directions before concluding this survey.

2 Existing categorization

Other than the distinction between discriminative and generative trackers, existing categorization also encompasses self-learning discriminative trackers, co-training trackers and two typical mechanisms for generative trackers.

2.1 Self-learning discriminative trackers

As the oldest approach for semi-supervised learning [15], self-learning assumes pseudo-labels as true labels and re-trains the model. In the context of visual tracking, it starts by training an initial object detector - a binary classifier, with labeled training examples from the first frame, then the classifier is evaluated on unlabeled data from upcoming video

frames. The most confident examples are added along with the estimated labels, by the model itself, into the training set and the classifier is updated accordingly. This is a wrapper algorithm with an iterative process. An unsatisfactory aspect of self-learning is that the wrapper depends on the supervised method used inside it. Due to this fact, a large amount of supervised tracking algorithms can literally have their own semi-supervised version.

One of the earliest influential and highly cited works for discriminative tracker is the Support Vector Tracking [2], which combines the computational efficiency of an optical-flow-based tracker with the power of a general Support Vector Machine (SVM) classifier. The detection and tracking modules of SVT cooperated in tandem. The major limitation of this framework is that the SVM classifier needs substantial prior training data in advance, thereby making it computationally unaffordable to explicitly involve an classifier update scheme. To accommodate appearance changes, SVT combines with optical-flow-based tracking to balance between maximizing the SVM classification score and the similarity to previous frame (as is done in previous optical-flow-tracker). Such integration is not semi-supervised learning although it also balances the stability and adaptivity to certain degree. As we will introduce later, a large amount of trackers adopted this strategy to balance between the fixed prior (a trained classifier or initialized statistical model) and the similarity to previous/recent frames. After the work of SVT, Williams et al. [96] proposed a tracker using sparse probabilistic regression by Relevance Vector Machines (RVMs), with a temporal fusion for high efficacy and robustness. Although tracking is efficient – better than real time (i.e., leaving processor cycles free for other processes) and the tracker is trained online from labeled images, it is impractical to have a large amount of labeled samples in the scenario of visual tracking. An ensemble SVM tracker, proposed by Tian et al. [87], was proved to be especially strong in selecting and recording the key frames of the objects as support vectors. By online adjusting the weight of each SVM classifier and integrating historical information, the ensemble classifier was claimed to be strongly discriminative between object and background and be able to accommodate large appearance variations. The robustness was achieved by region-based patterns/features instead of pixel-based features in Ensemble Tracking [3].

2.2 Co-Training trackers

Self-learning is able to adapt the tracker to new appearances and background, but breaks down as soon as the tracker makes a mistake. This problem can be addressed by co-training in the context of tracking. Contrary to self-learning, the idea of co-training is to make use of e.g., two different views on the objects to be classified. The basic intuition is that sometimes features describing the data are over-complete and could be split into two sets, each of which on its own is sufficient for correct classification. As shown in Fig. 5, the training is initialized by training a separate classifier on each view. Both classifiers are then evaluated on unlabeled data. The confidently labeled samples from the first classifier are used to augment the training set of the second classifier and vice versa in an iterative process [47]. The underlying assumption of co-training is that the two views are statistically independent. This assumption is satisfied in problems with two modalities, e.g., text classification (text and hyperlinks) and biometric recognition systems (appearance and voice). In visual object detection, co-training has been applied to car detection in surveillance or moving object recognition. Since the examples (image patches) are sampled from a single modality, Kalal et al. argued that co-training is not a good choice for object detections for the following two reasons: 1) features extracted from a single modality may be dependent and therefore violate the assumptions of co-training; 2) another disadvantage of co-training is that it cannot

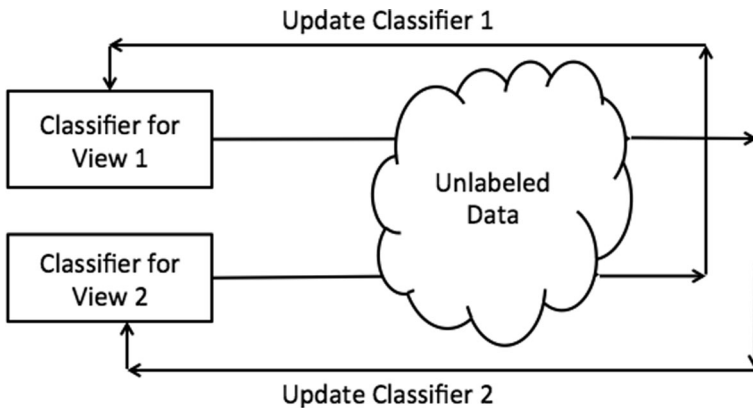


Fig. 5 Co-Training with two statistically independent views: confidently labeled samples from the first classifier are used to augment the training set of the second classifier and vice versa in an iterative process

exploit the data structure as each example is considered to be independent. Nevertheless, several works along this research line were reviewed in this survey.

2.2.1 Co-Training with Same Classifiers

Classifiers trained in different views can be uniformly the same. Tang et al. [86] proposed a semi-supervised learning method, which uses multiple independent features (i.e., color histogram and Histogram of Gradient) for training a set of SVM classifiers online. The classifiers collaboratively classify the unlabeled data and use these newly labeled data to update each other. Here, the two features used played a complementary role although extracted from the same modality (image patches).

2.2.2 Co-Training for hybrid generative and discriminative model

Another typical co-training framework is to integrate generative and discriminative model from two views. For example, Yu et al. proposed such a co-training framework to combine one global generative tracker and one local discriminative tracker [103]. The generative tracker builds a compact representation of the complete appearance of an object by online learning a number of local linear subspaces. The discriminative tracker adopts the online SVM to focus on the local appearance. By co-training, the two trackers can train each other on-the-fly with limited initialization. In contrast, Zhong et al. [110] proposed a sparsity-based collaborative model where tracking is based on the collaboration of local generative and global discriminative modules. In this tracker, holistic templates are incorporated to construct a discriminative classifier that can effectively deal with cluttered and complex background. Local representations are adopted to form a robust histogram that considers the spatial information among local patches with an occlusion-handling module, which enables the tracker to better handle heavy occlusion. The contributions of these holistic discriminative and local generative modules are integrated in a unified manner. Moreover, the online update scheme reduces drifts and enhances the proposed method to adaptively account for

appearance change in dynamic scenes. Duffner et al. [29] presented another novel algorithm for fast tracking of generic objects in videos using two components: a detector that makes use of the generalized Hough transform with pixel-based descriptors, and a probabilistic segmentation method based on global models for foreground and background. These components are used for tracking in a combined way, and they adapt each other in a co-training manner. Through effective model adaptation and segmentation, the algorithm is able to track objects that undergo rigid and non-rigid deformations and considerable shape and appearance variations.

2.3 Two mechanisms for generative trackers

2.3.1 Predict-update

The earliest probabilistic trackers dated back to the age of Kalman filter used in radar target tracking. The limitation of Kalman filter is its assumption of likelihood being Gaussian and linear, which motivated the widely accepted proposal of particle filtering [13, 44] to handle non-Gaussian and non-linear case. Particle filtering is basically an approximation method to Bayesian recursive formulation using Importance Sampling – one Monte Carlo method. The goal of such a Bayesian recursive formulation is to determine the posterior distribution $P(X_t|Z_{1:t})$, over the current joint configuration of the targets X_t at current time step t , given all the observations $Z_{1:t} = Z_1, \dots, Z_t$ up to that time instant. This recursive formulation is shown in (1) with appearance model $P(Z_t|X_t)$ and motion model $P(X_t|X_{t-1})$ predefined in advance. As a note, in contrast, the posterior distribution is modeled directly if using discriminative model without any motion estimation involved, which explains the name “tracking-by-detection”.

$$P(X_t|Z_{1:t}) = cP(Z_t|X_t) \int P(X_t|X_{t-1})P(X_{t-1}|Z_{1:t-1})dX_{t-1} \quad (1)$$

The process of particle filtering is a self-contained iteration of factored sampling with fixed particle-set size at each time step. The first layer of weighted particles altogether approximates the prior density in the previous state since this density is multimodal and there is no functional representation of it available. To infer the posterior distribution in the current state, each particle in the first layer is drifted to a new position by the designed motion model resulting the second layer of un-weighted particles. After introducing randomness by diffusion, the third layer of particles reflects the predicted step. Particles with higher weight may be chosen more frequently while particles with lower weight may not be chosen at all. In the end, this particle-set is updated by the observation density (i.e., the assumption about appearance/likelihood model) leading to a new particle-set that approximates the current posterior distribution. This new posterior estimate can serve as the prior for the next state and this tracking/filtering loop continues.

Vast majority of trackers adopted this Sequential Monte Carlo framework [13, 17, 26, 27, 44, 55, 71, 75–77, 88, 100]. The literature has seen increasing complex design of appearance and motion models that usually are predefined for generative trackers, in the hope that the more complex the model is, the better capability it gains to handle various scenarios in visual tracking. However, not all of them perform model update and the trackers with this ability employed specific visual representation, which will be detailed in later sections. This predict-update mechanism introduced by particle filtering plays a guidance role for most

trackers; another common mechanism is via direct optimization techniques, e.g., Mean-Shift theory in kernel trackers.

2.3.2 Direct optimization

As a non-parametric mode-seeking method for density functions, the Mean-Shift (MS) algorithm was introduced by Comaniciu et al. [19, 20] who proposed its use for object tracking. The MS algorithm tracks by minimizing a distance between two probability density functions (pdfs) represented by a reference and a candidate histogram. Minimizing the distance is equivalent to maximizing the Bhattacharyya coefficient, which has the meaning of correlation score. In other words, by spatially masking the target with an isotropic kernel, a spatially smooth similarity function can be defined. This function plays the role of a likelihood and its local maxima in the image indicate the presence of objects in the second frame having representations similar to the reference histogram defined in the first frame. The target localization problem is then reduced to a search in the basin of attraction of this function. Since the histogram distance does not depend on spatial structure of the search window, the method is suitable for deformable and articulated objects.

One problem the Mean-Shift algorithm [18–20, 30, 91] suffers from is a fixed search window (i.e., a fixed object scale). When an object becomes larger, the localization becomes poor since not all pixels belonging to the object are included in the search window and the similarity function has local maxima on parts of the object. If the object becomes smaller, the kernel window includes background clutter, which often leads to tracking failure. Vojir et al. [91] proposed a robust scale-adaptive mean-shift (eASMS) which was ranked No.7 in the VOT 2014 competition.

3 Sample selection strategies for model-free trackers

The key idea of semi-supervised learning, specifically semi-supervised classification, is to exploit both the labeled and unlabeled data to learn a classification model. For model-free tracking, there is an immense need for algorithms that can utilize the small amount of labeled data in the first video frame, combined with large amount of unlabeled data in the remaining frames in the video. In particular, for discriminative trackers where a binary classifier differentiating the foreground and background is involved, it is natural to integrate semi-supervised classification into the tracking framework. The way of exploiting unlabeled data is the defining factor for different categories: random sampling, sampling with structural constraints, and dense sampling, etc. Most discriminative trackers explore the unlabeled data explicitly in the ways aforementioned.

Generative trackers, in the literature, share no terminology with semi-supervised learning since no binary classifier is involved. Sample selection or labeling issue for updating binary classifier(s) is unique to online discriminative trackers. As one of the two major modeling techniques, however, generative models also need to be learned adaptively and incrementally to capture the appearance changes, which can be interpreted in the sense of semi-supervised learning. In particular, it starts by initializing/training an appearance/likelihood model based on a labeled training set; then the appearance/likelihood model is used to predict the likelihood, from which the maximum value of such likelihood indicates the object location in the next video frame. In probabilistic trackers, appearance model is equivalent to likelihood model. Without loss of generality, in this survey, we use appearance model only. In order to fit this conceptual consensus of semi-supervised online learning,

the appearance model is updated somehow, e.g., via subspace updating. One key difference from the model update in discriminative trackers is that the updating mechanisms for generative trackers usually are correlated with the formulation of appearance models. Efforts have been made in different directions, e.g., sparse subspace representation, and, more recently, distribution field.

To highlight the defining dimension we introduced in this survey, Table 3 summarizes all the five categories of sample selection strategies, i.e., random sampling, sampling with structural constraints, dense sampling, subspace basis selection, and distribution field selection.

3.1 Random sampling

The most intuitive way of exploiting unlabeled data is to randomly select image patches online to update the model. To alleviate the drift problem in on-line adaptation, ensemble

Table 3 Classification of State-of-the-art and Baseline Trackers based on Sample Selection Strategy Prior to 2015

Sample selection	Rep. Trackers	Learning models	Visual features
Random sampling	Ensemble learning trackers. e.g., OAB, SemiBoost tracker	Online adaboost	Haar-like features, orientation histograms, local binary patterns
Sampling with structural constraints	Spatial constraints by neighborhood, e.g., multiple instance learning (MIL) Tracker	Online adaboost	Bags of Haar-like features computed from integral image
	Part based trackers with predefined spatial configurations, e.g., LGT, DGT	Adaptive coupled-layer visual model; dynamic Target graph, discriminative SVM and generative model for color histogram,	Local color, shape and motion features; color histogram
	PN learning with Positive and Negative structural constraints, e.g., TLD	Random forrest classifier	2bit binary patterns
	Structured output tracking with kernels, e.g., struck	Structured Output SVM	Great flexibility in visual representation with a restriction kernel
Dense sampling	Correlation filter based trackers, e.g., DSST, SMAF, KCF	Learning discriminative correlation filters; Fast kernel regression	Histogram of gradient (HOG); PCA-HOG; color-naming
Subspace basis selection	Sparse subspace representation trackers, e.g., IVT, L1APG	incremental learning a low-Dimensional subspace representation	Sparse linear combination of eigenvectors or templates
Distribution field selection	Distribution field (DF) tracker	Learning the statistics of the appearance by combining the DF of initial model and the DFs of new observations	Convex combination of DFs

learning and its online variants [3, 5, 35, 36, 105] are probably the most widely used techniques in the last decade. Other than Avidan’s Ensemble Tracking, Grabner et al. pioneered this online-boosting idea, which is to formulate the update process in a semi-supervised fashion as combined decision of a given prior and an online classifier. Specifically, as shown in Fig. 6, given a fixed prior and an initial position of the object in time t , the classifier is evaluated at many possible positions in a surrounding search region in frame $t + 1$. The obtained confidence map is analyzed in order to estimate the most probable position and finally the tracker (classifier) is updated in an unsupervised manner, using randomly selected patches [36].

To date, most previous efforts have focused on adapting offline ensemble algorithms into online mode. This strategy, despite its success in many online visual learning tasks, has limitations in the visual tracking domain [5]. First, the common assumption of the observed data, examples and labels, have an unknown but fixed joint distribution that does not apply to visual tracking scenarios where the object of interest may undergo such significant appearance change that a negative example in the current frame looks more similar to the positive example identified in the past. Second, many online self-learning methods update the weights of their classifiers by first computing the importance weights of the incoming data. As noted by Grabner and Bischof, however, there are difficulties in computing these weights. This becomes even more challenging when it is recognized that the distribution that generated the data is non-stationary. Bai et al. [5] suggested that this is an inherent challenge for online self-learning methods and propose an approach for estimating the ensemble weights that is Bayesian and ensures that the update of the ensemble weights is smooth. In the context of “tracking-by-detection”, they are, as claimed, the first to present such an online learning scheme that characterizes the uncertainty of a self-learning algorithm and enables a Bayesian update of the classifier.

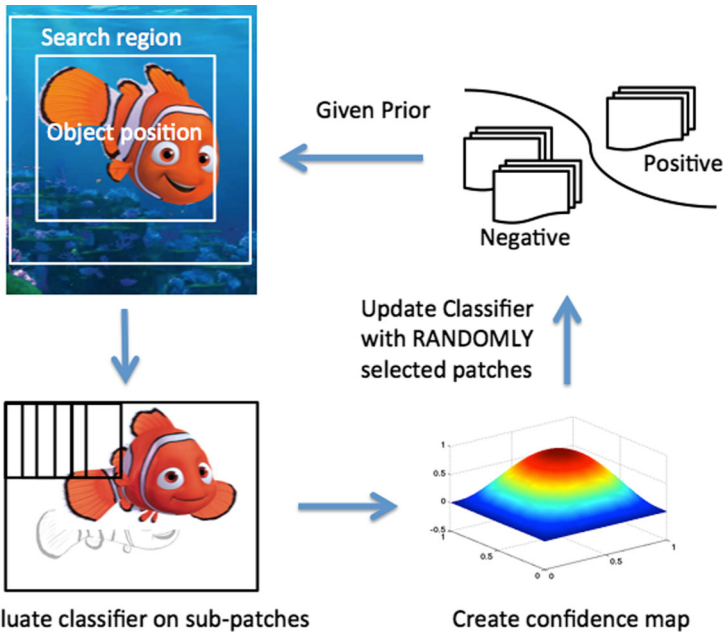


Fig. 6 Semi-Supervised Online-Boosting for Robust Tracking [36]

3.2 Sampling with structural constraints

Instead of using independent training examples, learning that exploits the structural constraint on the data is an alternative to adaptive object tracking. In computer vision, data are rarely independent since their labeling is related to spatial-temporal dependency [47]. The object to be tracked can be viewed as a single labeled example and the video as unlabeled data. Slight inaccuracies accumulated in the tracker can therefore lead to incorrectly labeled training examples, which degrades the classifier and can cause further drift. Self-learned and co-trained trackers, in general, assume that the unlabeled examples are independent. Therefore, such algorithms do not enable to exploit dependencies between unlabeled examples, which might represent a substantial amount of information. As a motivation for P-N learning, Kalal et al. [47] introduced the idea that the data with dependent labels are structured. For example, the trajectory represents a structure of labeling of video sequences, i.e., patches close to the trajectory are positive, patches far from the trajectory are negative. Since the trajectory is unknown, it would lessen the effect of labeling errors by partly recovering it using validated adaptive tracker with structural constraints. In this survey, we generalize this concept to a broader range of structured learning techniques, which are intended to address the labeling errors during tracking. Recently several research lines with different structural constraints fan out in this direction.

3.2.1 Spatial constraints by neighborhood

In the literature, Multiple Instance Learning (MIL) *per se* generally is not regarded as structured learning. However, training examples are bundled together with a bag label by spatially related units in MIL, rather than independent examples, as demonstrated by Fig. 7. We argue that this spatial information serves as spatial constraints, which leads MIL to the regime of structured learning. To avoid labeling errors during tracking, an online MIL boosting framework (MILTrack) [4] renders the capability to update the discriminative appearance model with a set of image patches cropped automatically based on the previous position of object, even though it is not known which image patch precisely captures the object of interest. This work shows MILTrack, in their experimental setting, outperforms both versions of Online Adaboost [35] and SemiBoost trackers [64]. Semi-supervised learning allows for incorporating priors and is more robust in case of occlusions while multiple-instance learning resolves the uncertainties where to take positive updates during tracking. Following MILTrack, Zeisl et al. [105] proposed an on-line semi-supervised multiple instance learning algorithm which is able to combine both of these approaches into a coherent framework. This leads to more robust results than applying both approaches separately.

3.2.2 Part-based trackers with predefined/learned spatial configuration

Some other examples of structured data are detection of object parts or multi-class recognition in a scene irrespective of generative or discriminative approach. Since the template object is represented by multiple image patches, not only can they be represented by region-based descriptor (e.g., histogram) and other non-parametric descriptors (e.g., kernel density estimate) in generative approach, but also be modeled as part detectors in discriminative approach. One highly cited paper is FragTrack [1], in which every patch votes on the possible positions and scales of the object in the current frame by histogram comparison. Minimizing a robust statistic combines the vote maps of multiple patches. Although

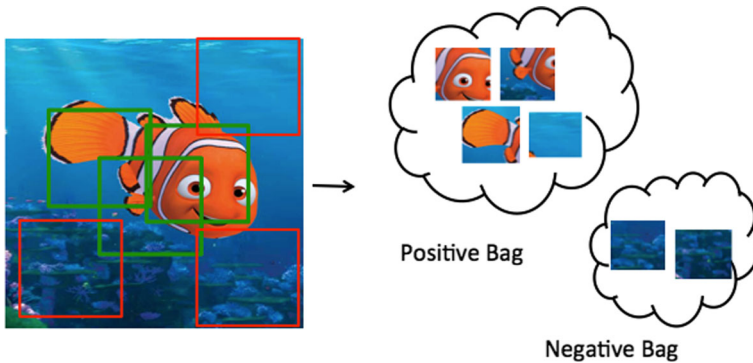


Fig. 7 Multiple Instance Learning Tracker. Training examples are bundled together with a bag label by spatially related units. A bag is labeled as positive as long as there is one positive instance inside [4]

this algorithm overcomes several difficulties, e.g., partial occlusions and pose change, by maintaining the geometric relations between templates patches, it does not incorporate any template updating.

Another part-based tracker is Hough tracker. Online learning has shown to be successful in tracking but always encounter drifting problem due to tracking inaccuracy. Such inaccuracy of position is limited to bounding-box representation with a fixed aspect ratio, which renders a less accurate discriminative capability and cannot handle highly non-rigid and articulated objects. One direction to reduce noise during online adaptive tracking is to exploit the General Hough Transform (GHT) strategy as it has been validated in object detection with arbitrary shape. Visual context has been successfully used in object detection tasks; however, it is often ignored in object tracking. Grabner et al. [37] proposed a method to temporally learn supporters that are useful for determining the position of the object of interest. This approach exploits the GHT strategy, which couples the supporters with the target and naturally distinguishes between strongly and weakly coupled motions. By this, the position of an object can be estimated even when it is not seen directly or when it changes its appearance quickly and significantly. Godec et al. [38] presented another novel “tracking-by-detection” approach to overcome this limitation based on GHT. They extended the idea of Hough Forests to the online domain and coupled the voting-based detection and back-projection with a rough segmentation based on Graph-Cut. This significantly reduces the amount of noisy training samples during online learning and thus effectively prevents the tracker from drifting.

More recent works along this line are Local-Global Tracker (LGT) [13] and Graph-based Tracker (DGT) [11], which specifically targeted at addressing tracking problem that undergoes rapid and significant appearance change or deformation. Cehovin et al. [13] proposed an adaptive coupled-layer visual model that combines the object’s global and local appearance by interlacing two layers. The local layer in this model is a set of patches, as shown in Fig. 8a, which probabilistically adapt to the target’s geometric deformation, and the structure is updated by removing and adding the local patches. The addition of these patches is constrained by the global layer that probabilistically models the target’s global visual properties, such as color, shape, and apparent local motion. The global visual properties are updated during tracking using the stable patches from the local layer. A more robust tracking is achieved by this coupled constraint paradigm between the adaptation of the global and the local layers. In a similar spirit, Cai et al. [11] approached this problem with a dynamic

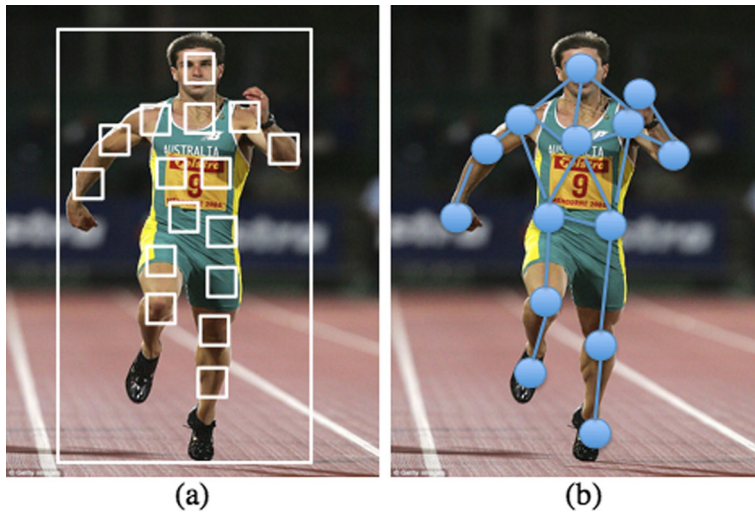


Fig. 8 (a): The target is represented by a set of local patches in a Local-Global Tracker; (b): The target is represented by a dynamic graph in a Graph-based Tracker [11, 13]

graph-based tracker. In the dynamic target graph, as shown in the Fig. 8b, nodes are the target local parts encoding appearance information, and edges are the interactions between nodes encoding the inner geometric structure. The target tracking is then formulated as tracking this undirected graph, which is also a graph-matching problem between the target graph and the candidate graph. As in LGT appearance changes are updated with addition or deletion of local patches, DGT enables model update by adapting to variations of target structure using the graph representation. DGT's performance was ranked No.4 on the newly created benchmark in the VOT2014 challenge.

3.2.3 P-N learning with positive and negative structural constraints

The work of Kalal et al. [47] on P-N Learning or Tracking-Learning-Detection (TLD) [49] is a highly influential one, which first introduced the learning of structured unlabeled data and partly inspired the formation of this survey. The structure of the data is named positive and negative structural constraints, which rules the certain labeling of unlabeled data. Positive and negative constraints specify the acceptable patterns of positive and negative patterns, i.e., patches close to the trajectory are positive while patches farther from the trajectory are negative, as has been shown in Fig. 9. P-N learning is essentially a bootstrapping process. An initial classifier is trained using labeled data with structural constraints predefined by the labeled samples as well; the classifier is then evaluated on the unlabeled data with examples identified as contradicted to structural constraints; these examples are corrected by structural constraints, added into training set and used to retrain the classifier.

As formulated by Kalal et al., a constraint can be a function that accepts a set of examples with labels given by the classifier and outputs a subset of examples with changed labels. P-N learning enables to use arbitrary number of such constraints. In contrast to the randomness of the spatial constraint in a bag of multiple instance with MIL trackers and the predefined spatial configuration in template object with part-based trackers, the formalized P-N learning enables the guidance to the design of structural constraints, the quality of which highly

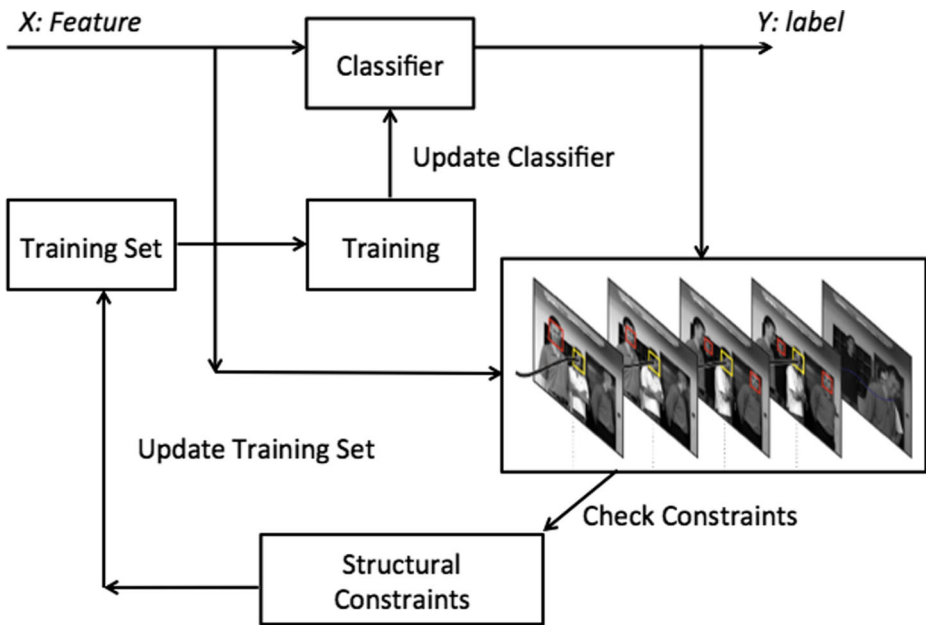


Fig. 9 P-N Learning framework with trajectory structural constraints [47]

impacts the classifier performance. The more sophisticated design of structural constraints that satisfy the requirement of learning stability would probably be a promising direction.

3.2.4 Structured output tracking with kernels

This work [39] has gained great popularity due to its novelty. For online adaptive tracking, one needs to convert the estimated object position into a set of examples to be labeled, but it is unclear how to perform this intermediate step, let alone the accumulated inaccuracy of trackers. The motivation of Struck is to explicitly allow the output space to express the needs of the trackers to avoid the intermediate classification step. Traditional algorithms separate the adaptation phase of the tracker into three distinct parts: i) the sampling of unlabeled examples that are to be labeled – the sampler; ii) the generation of labeling of samples – the labeller; iii) the updating of the classifier – the learner. Rather than using the tracker position to generate binary examples to learn the classifier, the method used here, as in Fig. 10, is learning a discriminant function to directly estimate the object transformation between frames. Thus the output space is the space of all transformation instead of the binary labels.

In this approach, a labeled example is a pair (x, y) where x is the feature extracted from an image patch within the bounding box and y is the desired transformation of the target. The discriminant function is formulated as a kernelized structured output Support Vector Machine framework. An online maximization step with Sequential Minimal Optimization (SMO) style is performed to predict the object transformation, and the label is output explicitly as the best transformation between frames thus making it viable to update the learner directly. One limitation of this online optimization is that the number of support vectors is unbounded and in general will increase over time. Incorporating a budget mechanism with GPU implementation was claimed to alleviate this problem. In contrast to the structured

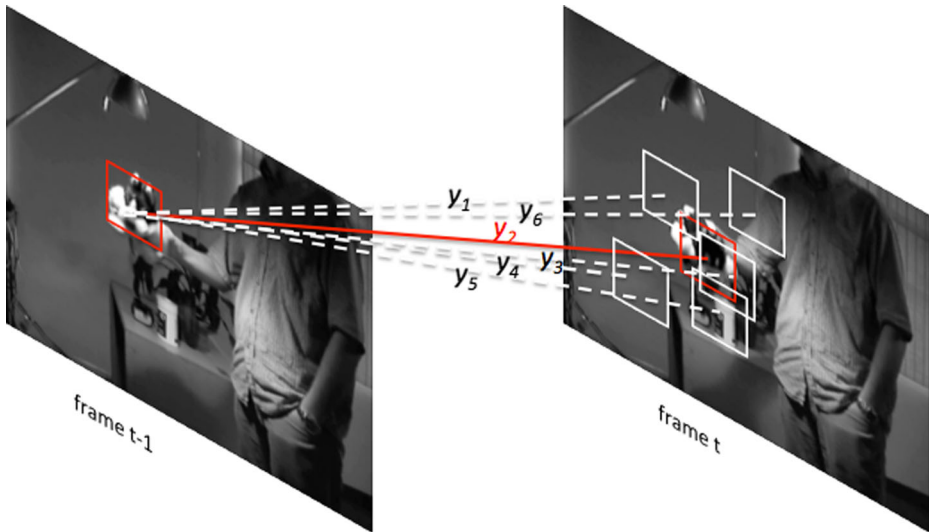


Fig. 10 Conceptual illustration of Struck — Structured Output SVM: y_1, y_2, \dots, y_6 are 3D affine transformations (e.g., 2D translation in most “tracking-by-detection” methods) between image patch pairs. y_2 should be the output of classifier since it gives the highest score. Contrary to other discriminative methods with an artificial binarization step, sample selection in Struck is fully controlled by the structural learner itself, and the relationships between samples such as their relative similarity are taken into account during learning [39]

constraints in the abovementioned categories, Struck and its variants opened a new direction to tackle the problem of sample selection and labeling errors during online adaptive tracking. In addition, structural constraints persevering the spatial relationships between multiple objects can also be learned in an online structured SVM framework [106].

3.3 Dense sampling

Dense sampling is the third approach for sample selection. Almost all the above trackers have one thing in common: sparse random sampling strategy. In each frame, several samples are collected (see Fig. 11a) in the target’s neighborhood, where each sample is represented by a bounding box with the same size of the target. Such sets of sparsely sampled patches are riddled with redundancy – any overlapping pixels are constraint to be the same, which means that we are probably not exploiting its structure efficiently. Recall that in the first step of the core component of most modern discriminative object detectors, the classifier is typically evaluated with translated and scaled sample patches, namely dense sampling. In a similar spirit of dense sampling for object detector yet in a more delicate fashion (see Fig. 11b), circulant trackers or correlation filter-based trackers [9, 21, 40, 41, 59] have demonstrated that the visual tracking problem, although traditionally solved using weighted classifiers, complex appearance models and stochastic search strategies, can be replaced by efficient and faster correlation filters that are simpler to implement.

The theoretical framework behind correlation filters is that the process of taking sub-windows, i.e., the translation operation, exhibits circular structure, from which a link to Fourier analysis can be established to allow Discrete Fourier Transform (DFT) to quickly incorporate the information from all sub-windows, without iterating over them, reducing

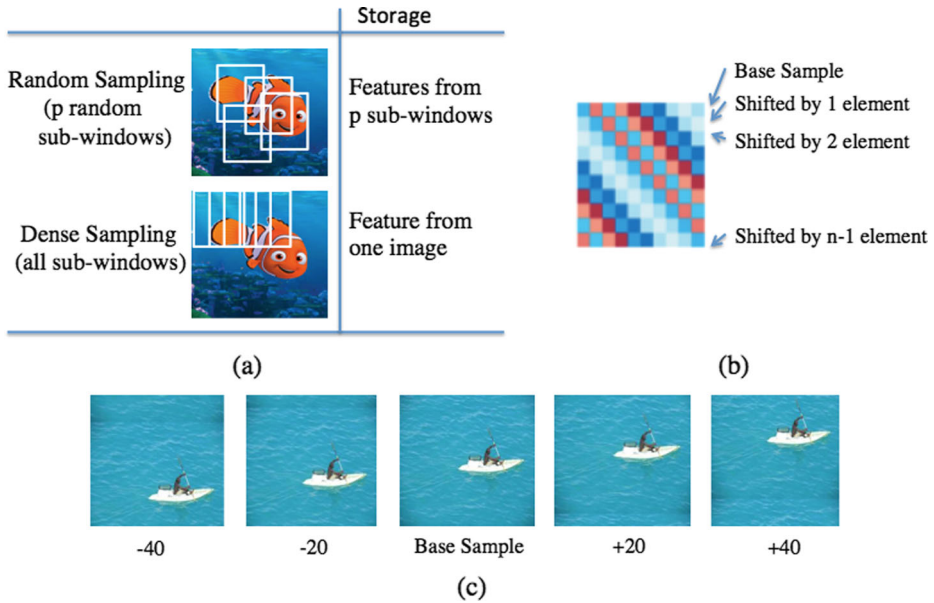


Fig. 11 (a) Random Sampling in all other trackers vs. Dense Sampling in ciculant trackers; (b) Circular Structure represented by a circulant matrix for correlation filter; (c) Vertically cyclic shifts of a base example [40, 41]

both storage and computation by several orders of magnitude. In particular, consider a vector representing a patch with object of interest, which is the base sample. Figure 11 b and c show the vertically cyclic shifted samples of a base sample. The Fourier domain formulation with circular structure allows us to train a tracker with all possible cyclic shifts of a base example, both vertically and horizontally, without iterating over them. The learning problem here is formulated as a ridge regression problem due to its closed-form solution that can achieve performance which is close to more sophisticated methods, e.g., SVM. When putting everything together with the classic Fourier analysis, it produces a nice formula with great computation efficiency.

The pipeline for these trackers is intentionally simple, and does not include any heuristics for failure detection or motion modeling. For example [41], in the first frame, they train a model with the image patch at the initial position of the target. This patch is larger than the target, to provide some context. For each new frame, they detect over the patch at the previous position, and the target position is updated to the one that yielded the maximum value. Finally, a new model is trained at the new position. It should be emphasized again that, in this model-updating scheme, sample selection is dense sampling instead of sparse random sampling. Another success story along this research line is that the top-3 best performing trackers – DSST [21], SAMF [59], KCF [41], all make use of this circular structure property.

3.4 Subspace basis selection

Despite the existence of a survey [61] specifically focusing on the appearance models, most of them are fixed models that need to be trained before tracking starts and hence in practice

limit the range of appearances that are modeled, ignoring large volume of information available during tracking. In contrast to discriminative trackers where binary classifier(s) need be updated via sample selection and labeling, the update of appearance models usually is as diversified as the various visual representation tools, e.g., sparse subspace representation, and distribution field.

Sparse representation has proven to be an extremely powerful tool for acquiring, representing, and compressing high-dimensional signals largely due to the fact that images have natural sparse representation with respect to fixed basis (i.e., Fourier, Wavelet), or concatenations of such basis. Although the images (or their features) are naturally very high dimensional, in many application images belonging to the same class exhibit degenerate structure, i.e., they lie on or near low-dimensional subspaces, sub-manifolds, or stratifications [97]. Such sparse representation, if computed correctly, could naturally encode the semantic information and thus exhibit capability to handle appearance changes when successfully integrated in the online adaptation of visual tracking [6, 8, 45, 60, 66, 67, 79, 99, 109].

A classic example of such tracker is IVT [79] (Incrementally learning for Visual Tracking), which is motivated by the prowess of subspace representation as appearance models, the effectiveness of particle filters, and the adaptability of online-update scheme. In contrast to the Eigentracking algorithm [8], IVT does not require a training phase but learns the eigenbasis online during the object tracking process. The model update, based on incremental algorithms for principal component analysis (PCA), includes two features: a method for correctly updating the sample mean, and a forgetting factor to ensure less modeling power is expended fitting older observations. Both of these features contribute measurably to improving overall tracking performance. This combination resembles the semi-supervised online boosting [36], which is a combined decision of a given prior and an online classifier; while IVT combines the given eigenbasis and an incremental learning mechanism to update them.

The global appearance of one object under different illumination and viewpoint conditions is known to lie approximately in a low dimensional subspace. Mei et al. casted tracking as a sparse approximation problem in a particle filtering framework [66, 67]. They proposed handling occlusion using trivial templates, such that each trivial template has only one non-zero element (see Fig. 12). Then, during tracking, a target candidate is represented as a linear combination of the template set composed of both target templates T (obtained from previous frames) and trivial templates. The number of target templates are far fewer than the number of trivial templates. Intuitively, a good target candidate can be efficiently represented by the target templates. This leads to a sparse coefficient vector, since coefficients corresponding to trivial templates (named trivial coefficients) tend to be zeros. In the case of

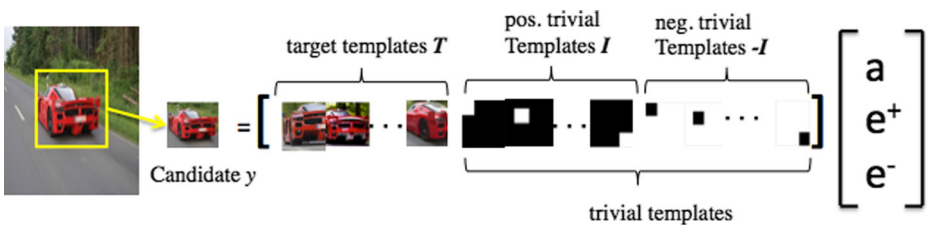


Fig. 12 Sparse representation used in $L1$ tracker: a target candidate is represented as a linear combination of the template set composed of both target templates T (obtained from previous frames) and trivial templates [66, 67]

occlusion (and/or other unpleasant issues such as noise corruption or background clutter), a limited number of trivial coefficients will be activated, but the whole coefficient vector remains sparse. A bad target candidate, on the contrary, often leads to a dense representation. The sparse representation is achieved through solving an $L1$ -regularized least squares problem, which can be done efficiently through convex optimization. Then the candidate with the smallest target template projection error is chosen as the tracking result. Following such $L1$ tracker, a few works further advanced this research line [6, 45, 99, 109].

3.5 Distribution field selection

As a generalization of many previous image representation used for different purposes, Distribution Field (DF) [31, 74, 80] is a descriptor that was proposed for representing object appearance in the tracking scenarios by Selilla et al. for the first time. Recall that, other than the predict-update mechanism provided by particle filter, direct optimization is another common framework for generative tracking, e.g., Mean-Shift. Visual tracking using such optimization framework relies on the assumption that gradient descent of the alignment function will reach the global optimum. The traditional blurring technique to smooth the function would destroy the image information. To address this problem, Selilla et al. [80] successfully validated, by representing object appearance with DF, the superiority of the width of the basin of attraction around the global optimum over other descriptors. They argue that this algorithm for tracking with DFs is able to avoid the drifting problem naturally by keeping a model of the target that is flexible enough to account for changes in appearance but allows a certain memory on the appearance model.

A DF is represented as a matrix d with $(2 + N)$ dimensions, where the first two dimensions are the width and height of the image, and the other N dimensions index the feature space that we choose [80]. Figure 13a shows the results of computing this DF for the well-known lena image. At this point, the DF representation contains exactly the same information as the original representation, albeit in a larger representation. Figure 13b shows a smoothed version of the DF on the left. The 3D DF has simply been convolved with a 2D Gaussian filter which spreads out in the x and y dimensions, but not in the feature dimension.

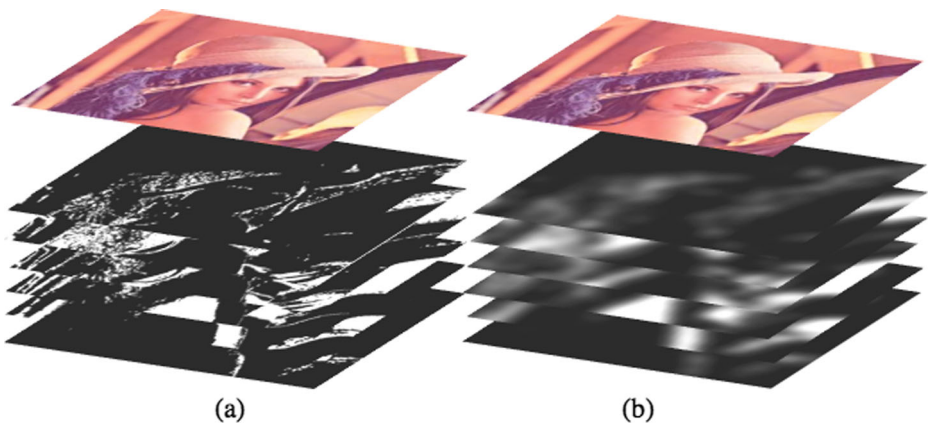


Fig. 13 (a): DF after exploding the lena image; (b): The same DF after smoothing in the dimension of the original image [80]

In summary, exploding an image into a DF and smoothing it can be viewed as introducing uncertainty about the object appearance [80]. A DF is then a compact representation of the image itself and a set of its neighboring images. These images are the result of transforming the original image with small changes in appearance and in location. Combining the information of several DFs can also be useful. In tracking, Selilla et al. combine the DF of initial model and the DFs of new observations using a component-wise convex combination of them. By combining DFs of different instances of the same object, Selilla et al. build a non-parametric data-driven model of the distribution at each pixel. This is useful for learning the statistics of the appearance of the object during tracking.

4 Compare and contrast time complexity between various visual tracking algorithms

It is hardly possible to summarize the state-of-the-art concerning time complexity for the following reasons:

- Very few papers literally discussed time complexity issue in this field. Instead, most papers reported the tracker speed in term of frames per second (FPS) to indicate the real-time performance.
- Given the diversity of visual tracking systems, no existing papers compare and contrast the time complexity due to incomparability. For example, the updating for weak classifiers is the most time-consuming part in ensemble learning based trackers, while the iteration for searching the basin of attraction dominates the computational load in kernel based trackers.
- Even for trackers in similar spirit, different design in details would lead to large disparity. For example, there exists a large body of generative trackers using particle filtering, which is also known as CONDENSATION algorithm [44]. Formally the complexity is $O(N \log N)$ with respect to the number of particles N since the construction of each particle involves a binary search. However, the time complexity varies greatly among these trackers using Bayesian recursive estimation of a time-evolving posterior due to the specific design of likelihood model, motion prior, and update scheme. Therefore, it would be impossible to do an exhaustive comparison beyond the basic particle filtering framework.

Despite the aforementioned difficulties, a summarization (Table 4) is made in attempt based on the taxonomy introduced in our survey (Note that we only discuss the complexity of learning, excluding the computational load of testing and other peripheral operations, e.g., feature extraction):

Random sampling Many trackers fall into this categories as long as the image patches used for learning/updating the model are randomly selected in an online fashion. One typical example is ensemble learning based trackers, which include Avidans Ensemble tracking and Garners variants of online-boosting tracking. Despite a lack of detailed discussion of time complexity for these algorithms in the literature, the most time-consuming part in common is the involvement of AdaBoost [32]. The computational complexity for boosting framework [89] is $O(MNK)$, where M is the number of rounds, N is the numebr of samples, and K is the number of features. Therefore, most ensemble learning based trackers should start off with this range of complexity. Anything beyond this needs be analyzed case by case.

Table 4 Time complexity of state-of-the-art and baseline trackers based on sample selection strategy

Sample selection	Rep. Trackers	Estimated time complexity
Random sampling	Ensemble learning trackers. e.g., OAB, SemiBoost Tracker	At Least $\mathbf{O}(MNK)$, where M is the number of rounds, N is the number of samples, and K is the number of features
Sampling with structural constraints	Spatial constraints by neighborhood, e.g., multiple Instance learning (MIL) tracker	Should be Similar to $\mathbf{O}(MNK)$
	Part based trackers with predefined spatial configurations, e.g., LGT, DGT	Difficult to generalize. For DGT, graph-cut ($\mathbf{O}(mn^2 C)$ where m = edges, n = nodes, $ C $ = capacity of minimum cut) and updating SVM ($\mathbf{O}(\max(n, d), \min(n, d)^2)$ where n is the number of feature points, d is the number of feature dimensions) dominate the computational load
	PN learning with positive and negative structural constraints, e.g., TLD	The worst time complexity for building forests of M randomized trees is $\mathbf{O}(MKN^2 \log N)$, where N denotes number of samples times 0.63, and K denotes the number of variables randomly drawn at each node
	Structured output tracking with kernels, e.g., struck	Similar complexity with conventional SVM ($\mathbf{O}(\max(n, d), \min(n, d)^2)$ where n is the number of feature points, d is the number of feature dimensions)
Dense sampling	Correlation filter based trackers, e.g., DSST, SMAF, KCF	$\mathbf{O}(N \log N)$ w.r.t the feature dimension N since all operations are element-wise
Subspace basis selection	Sparse subspace representation Trackers, e.g., IVT, L1APG	Difficult to generalized; For $L1$ -tracker, the complexity is $\mathbf{O}(MK \log K)$, where M is the total number of PCG steps, and K is the dimension of unknown vector X in a $L1$ regularized least square problem)
Distribution field selection	Distribution field (DF) trackers	The complexity is $\mathbf{O}(b * (T + MNR^2))$, where T refers to the cost of template matching that highly depends on the specific method used, M and N is the image size of each layer in DF, R denotes the radius of Gaussian kernel, and b denotes the number of bins used in the DF representation

Sampling with structural constraints Four subcategories were introduced in our survey.

- First, multiple instance learning based tracker, in particular, MILTrack is actually an online boosting framework extended into a multi-instance case. The time complexity should reside in similar range of $\mathbf{O}(MNK)$ as discussed above.
- Second, it is extremely hard to summarize the common complexity for part-based trackers due to great variety. One example could be a graph-based tracker (DGT) that can be formulated as a graph matching problem. The computation involves candidate graph construction, the minimization of a Markov Random Field energy function using graph-cut [10, 82] ($\mathbf{O}(mn^2|C|)$ where m = edges, n = nodes, $|C|$ = capacity of minimum cut), local part correspondence with spectral matching in which the optimization problem used a simple greedy approach, and updating a SVM classifier [16, 78] (i.e., quadratic optimization can be computationally expensive, and the complexity highly depends on the optimizer. Oliver gave a reasonable analysis $\mathbf{O}(\max(n, d), \min(n, d)^2)$ where n is the number of feature points, d is the number of feature dimensions). Although it is hard to assess the complexity of this DGT tracker without explicit analysis by the authors, it would be reasonable to expect graph-cut and updating SVM dominate the computational load if the greedy approach was as simple as claimed.
- Third, the implementation of P-N learning based tracker is to learn a randomized forest classifier and bootstrap its performance using unlabeled data. Random forest essentially is an ensemble model of decision trees. The worst time complexity [62] for building forests of M randomized trees is $\mathbf{O}(MKN^2 \log N)$, where N denotes number of samples times 0.632 due to the fact that bootstrap samples draw, on average, 63.2 % of unique samples, and K denotes the number of variables randomly drawn at each node.
- Fourth, structured output tracking involves the learning of a discriminant function to directly estimate the transformation between frames. This structured output SVM framework used SMO solver for the quadratic optimization problem in the Struck paper. Again this paper did not explicitly analyze time complexity, but Thorsten Joachims [46] claimed that the formulation of structured output SVM is equivalent to the conventional SVM optimization problem, which indicates similar complexity with conventional SVM. Thorsten Joachims [46] also proposed a cutting-plane algorithm that is several orders magnitude faster than the decomposition methods, such as SMO. However, this linear time training algorithm for SVMs has not been adopted in Struck tracker.

Dense sampling Normally, ridge regression has a cost of $\mathbf{O}(N^3)$ with respect to the feature dimension N , bounded by the matrix inversion and products in the closed-form solution. However, in circulant trackers, all operations are element-wise ($\mathbf{O}(N)$), except for Discrete Fourier Transform, which bounds the cost at a nearly linear $\mathbf{O}(N \log N)$ [41]. The rationale behind is that the process of taking sub-windows includes circular structure, from which a link to Fourier analysis can be established to allow FFT to quickly incorporate the information from all sub-windows, without iterating over them, reducing both storage and computation by several orders of magnitude.

Subspace Basis Selection A typical example of such kind is $L1$ -tracker [66]. This generative tracker adopts a particle filtering framework, therefore the $\mathbf{O}(N \log N)$ complexity with respect to the number of particles N applies here. In addition, the computation of the best likelihood between a target candidate and the target template can be formulated as an

$L1$ -regularized least square problem. The sparseness is achieved by $L1$ minimization based on the implementation of an interior-point method [50]. The method uses the preconditioned conjugate gradients (PCG) algorithm to compute the search direction and the run time is determined by the product of the total number (M) of PCG steps required over all iterations and the cost of a PCG step ($O(K \log K)$, where K is the dimension of unknown vector X in a $L1$ regularized least square problem).

Distribution Field Selection The computational cost of distribution field tracker depends on the number of bins (b) used in the DF representation [80]. The running time is b times that of the template matching with a Gaussian pyramid. Extra computation is the convolution of each layer of the DF which has complexity of $O(MNR^2)$, where M and N is the image size of each layer in DF, and R denotes the radius of Gaussian kernel. Therefore, the complexity is $O(b*(T + MNR^2))$, where T refers to the cost of template matching that highly depends on the specific method used.

5 State-of-the-art of major applications

Visual tracking resides in the middle level of almost all video analysis tasks. For example, video event detection highly relies on the quality of tracks. It has many applications in video surveillance for security and forensics, human-computer interaction, intelligent transportation system, medical imaging, mobile robotics, and sports video analysis, etc. Some contemporary vision systems of such kind are shown in Table 5.

One has to admit that there exists a gap between industrial interests and pure intellectual curiosity due to the fact that there exists many visual tracking commercial systems but rarely are model-free.

In industry, most visual tracking applications are dealing with short term tracking in a well-controlled environment. For example, tracking face in human-computer interaction games usually applies in an indoor environment where luminance remains constant and the face does not encounter much change in such short tracks. Another example is the vision system for traffic management where the surveillance zone is fixed and only the upcoming cars entering the surveillance zone would possibly be tracked. Such short tracks then provide information for further decision of traffic management. Similar observations can be found in tracking people in stores for security or marketing, tracking eye gaze in a in-car environment for accident prevention systems of cars, or even tracking the high-speed ball of tennis for refereeing and analysis, etc. Although existing techniques may offer satisfactory solutions to this problem in well-controlled environments, it is still challenging especially for long term tracking due to factors such as partial occlusion, clutter background, fast and abrupt motion, dramatic illumination changes, and large variations in viewpoint and pose. With such limitations in practice, there does not yet exist a generic model-free tracker that can handle all circumstances for long-term tracking.

In academia, however, pure intellectual curiosity tends to lead to the development of an algorithm that can track an object with appearance variations for longer time in any dynamic changing environment once the model is set up well initially. Therefore, researchers have gone great length to restrict the assumption as such. First, for generative trackers, the visual prior is a predefined appearance model (e.g., a histogram to represent an image patch). It is natural to predefine it in the first frame while updating this model in the subsequent frames. Second, for discriminative trackers where a binary classifier is involved, such a restriction is

Table 5 Contemporary vision systems for tracking [63]

Coporation	Location	Product
TrafficVision	(Pendleton, South Carolina)	Real-time traffic management using computer vision.
Image sensing systems	(St. Paul, Minnesota)	Real-time traffic management using roadside cameras. License plate recognition systems.
Iteris	(Santa Ana, California)	Real-time traffic management and signaling using video detection.
MobilEye	(Jerusalem, Israel)	Vision systems that warn automobile drivers of danger, provide adaptive cruise control, and give driver assistance such as active braking. Systems are currently available in certain cars from BMW, Volvo, GM, and others.
Mirametrix	(Montreal, Canada)	Free-head eye-tracker.
Gazepoint	(Vancouver, Canada)	Low-cost eye-trackers for consumer and research applications
Smart Eye	(Gteborg, Sweden)	Systems to track eye and gaze position.
SMI	(Berlin, Germany)	Eye and gaze tracking systems.
Hawkeye	(Winchester, UK)	Uses multiple high-speed cameras to provide precise tracking of the ball in tennis, cricket, and other sports for refereeing and analysis.
PlayfulVision	(Lausanne, Switzerland)	Provides real time, automatic video analytics and statistics for team-sports.
QuesTec	(Deer Park, New York)	Systems for tracking sports action to provide enhanced broadcasts.
2d3	(Oxford, UK)	Systems for tracking objects in video or film and solving for 3D motion to allow for precise augmentation with 3D computer graphics.
Image metrics	(Manchester, England)	A markerless tracking system for the human face that can be used to map detailed motion and facial expressions to synthetic characters.
MirriAd	(London, UK)	Uses computer vision methods to track consistent regions in video and insert virtual advertising.
GestureTek	(Toronto, Canada)	Tracks human gestures for playing games or interacting with computers.
Seeing machines	(Canberra, Australia)	Systems for tracking faces and eye gaze direction for human-computer interaction.

Table 5 (continued)

Coporation	Location	Product
Claron technology	(Toronto, Canada)	Uses real-time stereo vision to detect and track the pose of markers for surgical applications.
Brickstream	(Atlanta, GA)	Tracking people within stores for sales, marketing, and security.
Reveal	(Auckland, New Zealand)	Systems for counting and tracking pedestrians using overhead cameras.
VideoMining	(State College, PA)	Tracking people in stores to improve marketing and service.

intended for reducing the online computational load. In this case, “using labeled instances only in the 1st frame” means that, the classifier can only be trained offline in the 1st frame while being updated by selected image patches in an unsupervised manner for the rest of the video sequence. The state-of-the-art of academic research on model-free tracking that can handle all aspects of tracking performances is still on its preliminary stage due to the open issues that will be summarized in Section 7.

6 Design considerations for semi-supervised online trackers

Designing a robust visual tracker to handle various scenarios is a very difficult open-ended problem, which is why this has been an intensely researched field with abundance of various tracking algorithms. Despite the vast majority of trackers we touched so far (e.g., self-learning discriminative trackers, online adaptive generative trackers and co-training trackers), it should be noted that, from the perspective of machine learning, it is nearly impossible to enumerate all types of trackers since many of them literally are more emphasizing a vision-related aspect in the tracking system. For example, the discussion of keypoints-based trackers [74], optical flow trackers [48, 72], and context-aware trackers [95, 101], is beyond the scope of this paper. As aforementioned in this survey, the way of detecting the changing environment in a new frame (i.e., the sample selection for discriminative trackers, or searching the maximum likelihood score for generative trackers) and the extent of how rapidly the classifier(s)/appearance model need be updated accordingly are the key to many, if not all, evaluation measures for tracking. The fundamental problem is to robustly integrate data derived during tracking into the model without drifting. The essence in this context is actually an online semi-supervised learning problem, which would affect all the five components of visual tracking systems. In this section, we discuss how the semi-supervised online learning affects the design of visual representation, the appearance model or object detector, and the motion model, etc. We are also attempting to answer the question what components in a visual tracking system ameliorate the semi-supervised learning to strike a balance between adaptivity and stability.

6.1 Good features for online tracking

Feature engineering is critical in machine learning and data mining tasks. However, there has been no consensus what features are good without a specific task given. In the context of tracking, an earlier work [81] attempted to tackle this problem. In a closely related

context, i.e., rapid object detection [89], a boosted cascade framework for selecting simple features was proposed, which was later successfully extended to online AdaBoost feature selection for tracking [35]. To the best of our knowledge, no work till now was particularly addressing this problem in the setting of visual tracking. Most features are associated with the chosen learning models (classifiers or appearance models), and in most tracking papers features were utilized favoring toward the better results on the chosen testing video sequences. Novel feature selection for generic visual tracking would be expected to appear in the future if the newly created benchmark in VOT competition became more robust and mature. Employing deep features is a good attempt for learning generic visual representation as the top two performing trackers in VOT2015, but the state-of-the-art has not rendered good practicality with the involvement of deep learning. Despite all that, based on our experiences, two basic requirements for good features in visual tracking are as follows: (1) These features should be fast computable (e.g., Haar-like wavelets, orientation histograms, local binary patterns) so that the tracking algorithm runs in real-time; (2) These features should be seamlessly integrated with the selected learning models in a fast implementation of model update, where the update rate is controllable.

6.2 Common motion models for online tracking

Motion models perhaps are the component of visual tracking where the least efforts have been invested because any predefined motion prior has its limitation on random movement in video sequences. Motion models usually exist in stage of prediction in probabilistic trackers due to the Bayesian recursive formulation, which dates back to the radar target tracking where motions can be better captured by an autoregressive model. There are no motion models involved in optimization based generative trackers. In addition, discriminative trackers coined as “tracking-by-detection” methods also discard motion estimation to avoid the difficulty of abrupt motion change. Therefore, design consideration in this respect is straightforward.

6.3 Common statistical models for appearances during online tracking

The statistical models for appearances, as have been discussed throughout this survey, have two categories:

- Statistical models or classifiers that differentiate the object and background (e.g., SVMs, random forests, structured output SVM, and correlation-filter based model);
- Statistical models that only capture the characteristics of the object of interest (e.g., sparse subspace representation, distribution field).

As the most critical part of visual tracking, conceptually these models serve both as the visual prior in the first frame and the appearance changes on the fly in the sense of semi-supervised (cannot strictly apply this term on generative models) online learning. However, design of these models is nontrivial (Table 6).

Most discriminative trackers somehow exploited SVM or its variants as the classifier with most in the form of ensemble learning. The introduction of structured output SVM initialized a more generalized framework without the need to deal with labeling errors and its consequences. More recent advances with discriminative trackers are the top-performing circulant trackers, which rejuvenated the usage of the classic Fourier analysis to formulate a ridge regression with closed form solutions. Circulant trackers make the tracking

Table 6 Design considerations for semi-supervised online trackers

Components in visual tracking	Common types	Design requirements
Visual features	Haar-like features, orientation histogram, local binary patterns, histogram of gradient, deep features, etc.	(1) Should be fast computable so that the tracking algorithm runs in real-time; (2) Should be seamlessly integrated with the selected learning models in a fast implementation of model updating
Motion model	Types highly depend on the prior knowledge of motion: autoregressive model, or No motion priors	(1) Usually an autoregressive model for probabilistic trackers; (2) No motion models involved in discriminative trackers, and optimization based generative trackers
Statistical models for appearances	Discriminative models, e.g., SVM, Random forest, structured output SVM, correlation filter; or generative models, e.g., meanshift, and sparse subspace learning	(1) Requires that statistical models or classifiers can differentiate the object from the background; (2) Requires that statistical models that only capture the characteristics of the object of interest; (3) Requires that statistical models be light-weight and be able to self-update with newly arrived information
Sample selection strategy	Random sampling; sampling with Structural constraints; dense Sampling; subspace basis selection; Distribution field selection	(1) Requires that the situation encountered by the tracker be actively assessed; (2) Require fast implementation with sampling ; (3) More information is usually more helpful via either structural constraints or dense sampling, e.g., the top-performing trackers in VOT2014, and many of those above the average state-of-the-art performance in VOT2015 are dense sampling related
Updating scheme	Self-learning/Self-updating/Bootstrapping with different sample selection strategies; or co-training	(1) Requires that the tracker strike a balance between the prior and newly arrived information (2) The update rate is controllable

framework simpler than ever before. On the other hand, appearance models in online adaptive generative trackers need compact yet flexible representations to facilitate the model update.

6.4 Common approaches for selecting training samples for online tracking

The training sample selection approaches only apply to discriminative trackers with the following categories:

- Random selection around the object position, labeling the closer patches positive while farther ones negative;
- Sample selection with structural constraints, e.g., multiple patches with only a bag label, patches labeled as positive when satisfying a spatial constraint defined in object

- template, patches labeled as positive if close to trajectory while negative otherwise, or avoidance of labeling samples altogether as in structured output SVM;
- Dense sampling achieved by the circulant structure with cyclic shifts as in correlation-filter based trackers.

On the other hand, there is no explicit sample selection in most generative trackers. However, in analogy, basis selection or distribution field selection may be close to it in spirit. For example, the sparse subspace representation in $L1$ tracker is a linear combination of target templates and trivial templates; a Distribution Field (DF) is a compact representation of the image itself and a set of its neighboring images in DF tracker. Here, the trivial templates in $L1$ tracker handle occlusions and need be optimally selected in order to achieve sparsity, which then gives the estimated object position. In DF tracker neighboring images that deal with uncertainty in object appearance need also be selected to incorporate new DF information.

6.5 Common updating schemes for online tracking

Since there exists a dilemma between the fixed part and the flexible part for online trackers, a simple strategy that has been extensively used in online tracking is to strike a balance between these two parts. The fixed part – an object detector or an appearance model, refers to a given prior that can be predefined or learned. The flexible part usually follows the new information collection procedure – sample selection, or roughly speaking, basis/DF selection. The model generally gets updated in the fashion of self-learning or co-training irrespective of discriminative or generative models. The above description helps all the surveyed trackers reach the conceptual consensus in the sense of semi-supervised online learning.

7 Summarization of open issues

The state-of-the-art of visual tracking becomes increasingly difficult to be summarized due to the fact that many different and varying circumstances need to be reconciled in one algorithm [83]. For example, a tracker may be good at handling variations in illumination, but has difficulty in coping with appearance changes of the object due to variations in viewpoint. A tracker may predict the next position of the moving object using a second order autoregressive model, but then may have difficulty in following bouncing objects that can only be captured by activity-specific models. A tracker may make a specific assumption of the appearance of a rigid object, but then may fail on articulated objects that have components attached via joints and can move with respect to one another. Designing a generic tracker (a single-object model-free tracker in this survey) that can handle all these circumstances is still an open problem.

Much of the advances in computer vision fields, e.g., object detection, can be attributed to a ubiquitous access to standard datasets and evaluation protocols. However, the field of visual tracking still suffers from the lack of established methodology for objective comparison since collecting larger benchmark datasets entails significant efforts from the entire computer vision community. Prior to 2013, papers appearing in top conferences of this field only include an average of 5 to 10 video sequences, in which few of the varying circumstances were adequately tested. Despite several initiatives (e.g., 1-6 in Table 1), many of these datasets lack a standard ground truth labeling, which makes comparison of algorithms unfair. After 2013, the computer vision community shifts its focus to developing a more

objective benchmarking dataset with good quality of its construction and annotation. As of the time of writing, currently the most widely used benchmarking datasets include Online Tracking Benchmark (OTB) [98], Amsterdam Library of Original Videos (ALOV) [83], and Visual Object Tracking challenges (VOT) [51–54]. This trend culminates at the VOT2015 challenge where a new dataset is constructed by combining 51 sequences of OTB, 315 sequences of ALOV, and over 30 additional sequences from other resources. After removal of duplicates, the sequences are automatically selected to obtain a dataset in which the following five visual attributes are sufficiently well represented, i.e., occlusion, illumination change, motion change, size change and camera motion. To achieve this, these sequences were automatically clustered according to their similarity in 11 different features (i.e., illumination change, object size change, object motion, clutter, camera motion, blur, aspect-ratio change, object color change, deformation, scene complexity and absolute motion). As for the annotation, researchers have manually or semi-automatically annotated each frame in each selected sequence with those five visual attributes. This new fully annotated dataset in VOT2015 doubles the number of sequences compared to VOT2014. Interested readers may refer to this paper [54] for details.

On the other hand, a general evaluation protocol for all visual tracking algorithms, although of great interest, remains a challenge due to the hardly avoidable biases. These biases arise from many sources, such as tracker parameters (e.g., the number of particles in probabilistic trackers), initialization, and sequences used. In Table 7, we tabulated 16 existing evaluation measures with different tuning parameters in different papers, as well as their respective limitations. While some authors choose several basic measures (e.g., center error, region overlap, tracking length, failure rate) to evaluate their trackers, a recent study [53] has shown that many of these measures are correlated and do not reflect diverse aspects of tracking performance. In this respect, choosing a large number of measures may in fact again bias results toward some particular aspects of tracking performance. Based on the recent analysis of widely used performance measures, two weakly correlated and easily interpretable measures were chosen: accuracy and robustness. Thus, the accuracy-robustness (AR) plots served as the standard evaluation measures in VOT2014. The accuracy measures how well the bounding box predicted by the tracker overlaps with the ground-truth bounding box. The robustness measures how many times the tracker loses tracks of the target. As in any competition, ranking is of great concern. To account for statistical and practical differences, the average of ranked lists with respect to each measure was used in VOT2014, but it cannot be interpreted well in terms of a concrete tracking application. Thus VOT2015 introduced a new measure (Expected Average Overlap measure) that combines the raw per-frame accuracies and failures in a principled manner. For result visualization, AR rank plots, AR raw plots and the Expected Average Overlap graph were employed in VOT2015. More specifics can be found in these papers [52–54].

As the benchmark datasets and the evaluation protocol continue evolving, the advance of tested state-of-the-art since 2014 is clear since VOT2014 winner is now ranked 38th among the 62 trackers in VOT2015 and almost half of the submitted trackers are above the average state-of-the-art performance in terms of Expected Average Overlap score. Despite such advances with data-driven approaches, there are still many open issues:

- The benchmark datasets and the evaluation protocol have not been standardized yet, given the fact the size of VOT2015 dataset doubled and another new measure was introduced with only one year apart. In addition, not all trackers in VOT2013 and VOT2014 were involved in the competition of VOT2015. Although VOT2015 is the most challenging dataset to date with various visual attributes well represented, the gap between

Table 7 Visual Tracking Performance Measures [12] (where τ denotes the threshold of overlap, P_τ denotes the percentage of correctly tracked frames, F_τ denotes the failure rate measure with an overlap-based failure criterion with threshold τ)

No.	Measures	Limitations
1	Average center error	Sensitive to subjective annotations, ignore target size and apparent tracking failure
2	Average normalized center error	Misleading as the center error is reduced proportionally to the estimated target size
3	Root-mean-square error	Sensitive to subjective annotations, ignore target size and apparent tracking failure
4	Average overlap	Not necessarily account for complete trajectory reconstruction which is an important aspect in many tracking applications
5	Percent of correct frames for $\tau = 0.1$, $P_{0.1}$	0.1 threshold may lead to false positives, do not necessarily account for complete trajectory reconstruction
6	Percent of correct frames for $\tau = 0.5$, $P_{0.5}$	0.5 threshold may lead to false negatives, it may be reached even for visually well overlapping rectangles, problematic for non-rigid articulated object, and do not necessarily account for complete trajectory reconstruction
7	Tracking length for threshold $\tau > 0.1$, $L_{0.1}$	Only uses the part of the video sequence up to the first tracking failure, difficult situation in beginning frames may lead to poor initialization
8	Tracking length for threshold $\tau > 0.5$, $L_{0.5}$	Only uses the part of the video sequence up to the first tracking failure, difficult situation in beginning frames may lead to poor initialization
9	Failure rate F_0	Not reflect the distribution of these failures across the sequence
10	Average center error for F_0	Combined drawbacks from 1 and 9
11	Average normalized center error F_0	Combined drawbacks from 2 and 9

Table 7 (continued)

No.	Measures	Limitations
12	Root-mean-square error for F_0	Combined drawbacks from 3 and 9
13	Percent of correct frames for $\tau = 0.1$, $P_{0.1}$ for F_0	Combined drawbacks from 5 and 9
14	Percent of correct frames for $\tau = 0.5$, $P_{0.5}$ for F_0	Combined drawbacks from 6 and 9
15	Average overlap in case of F_0	Combined drawbacks from 4 and 9
16	Combined tracking performance Score (CoTPS) measure	No justification, neither theoretical nor experimental, for such rather complicated fusion; Difficult to interpret this measure

academic research and industry application still exists as mentioned in Section 5. Without the standardization of this dataset and the evaluation protocol, researchers could only hypothesize the design choices of a generic model-free tracker based on their intuitions on the previous competition results while at the same time industry practitioners in this field would have a hard time deciding by which standard to evaluate those tracking algorithms and to select the suitable ones for their application needs.

- Aside from the standardization issue, the results in VOT2015 show that there is still a lot of space for improvement. This is evident from the fact that the VOT2015 winner only achieved 0.60 accuracy, 0.69 robustness, and 0.38 expected average overlap with a very slow tracking speed.
- From the results there is no clear indication as to which visual representation or statistical model is in general a good choice for top-performing trackers. The ranking of VOT2015 is significantly shuffled compared to VOT2014. For example, Struck was the 10th in VOT2014. Almost all top-9 trackers in VOT2014 now fall far behind Struck that is now the 9th in VOT2015. In the current top-10 or top-20 ranking list, little consensus can be found with the exception that many of those above the average state-of-the-art performance are based on correlation filters. However, this might be an illusive consensus because 20 out of 62 trackers are literally extended from correlation filters that were the top-3 in VOT2014 (Table 8). It would be much safer to agree if equal amount of efforts had been invested on other trackers.
- There is a tendency to focus on the performance competition *per se* without considering the design issues summarized in Table 6. For example, the top-2 performing trackers in

Table 8 Top performing trackers with five visual attributes in benchmarking datasets of VOT2014

Visual attributes	Top-performing trackers in AR plot
Camera motion	KCF [41], SMAF [59], DSST[21], DGT [11], PLT_14 [52]
Illumination change	DSST [21], KCF [41], SAMF [59], ACAT [22]
Occlusion	DSST [21], PLT_14 [52], PLT_13 [51]
Size change	DGT [11], eASMS [91], PLT_14 [52]
Motion change	KCF [41], SAMF [59], DSST [21], DGT [11], eASMS [91]

Table 9 Top performing trackers with five visual attributes in benchmarking datasets of VOT2015

Visual attributes	Top-performing trackers w.r.t robustness
Camera motion	MDNet [70], EBT [111], DeepSRDCF [23]
Illumination change	MDNet [70], EBT [111], DeepSRDCF [23]
Occlusion	DeepSRDCF [23], MDNet [70], EBT [111]
Size change	MDNet [70], EBT [111], DeepSRDCF [23]
Motion change	FragTrack [1], MDNet [70], NSAMF [59]

It should be noted that AR plots in VOT2015 were normalized among different visual attributes, thus it is impossible to present results per attribute w.r.t AR plots here

VOT2015 (Table 9) are MDNet and DeepSRDCF, which both utilized Convolutional Neural Network (CNN) for training visual priors. The involvement of CNN significantly slows down the tracking speed, and these two trackers are indeed the slowest among the 62 trackers. Deep learning is almost about scalability, and its recent success in areas, such as object detection, is mostly due to the factors of more data, bigger model and more computation. Although it was a good attempt to learn generic visual representation for model-free tracking, it is more desirable to see in the future that similar results could be achieved with less data, smaller model and less computation so that it could be more practical in real scenarios.

As can be seen from the state-of-the-art in this field, most recent survey efforts were focused on data-driven approaches due to little conceptual consensus in diversified approaches. In our survey, we introduced a new dimension on perspective to survey single object model-free tracking. In particular, this paper formulated model-free tracking as a semi-supervised online learning problem with the sample selection strategy as the key to the categorization, in the hope that researchers in the following VOT competitions as well as in this field could use this new dimension to sort through the existing approaches instead of focusing merely on improving performances on particular datasets yet to be standardized. Although only a subset of representative trackers was compared and discussed in our survey, almost all the trackers in the most up-to-date VOT2015 can be easily classified with this new dimension (Table 10). For example, 16 trackers were based on correlation filters that can be classified into dense sampling. Three deep learning based trackers also belong to dense sampling due to the nature of convolutions in deep neural networks. A few Struck related variants and 14 part-based trackers resonates with the sampling with structural constraints. Some online Adaboost trackers are clearly in the regimen of random sampling. In addition, most of the baseline trackers are already included in Tables 3 and 4. Two key observations in Table 10 are as follows. First, most of the state-of-the-art trackers are related with structural sampling, dense sampling, or both. Second, an emerging trend, other than deep learning based dense sampling, is to generate or select object proposals for candidate locations for detection and tracking.

8 Transductive vs. inductive learning for model-free tracking

Curious minds may wonder how learning machines tackle situation awareness philosophically. In the machine learning field, two major approaches are used: induction and

Table 10 Performance comparison of state-of-the-art and baseline trackers based on sample selection strategy in VOT2015 (A: Accuracy; R: Robustness; overlap: Expected average overlap score)

Sample selection	Trackers	A	R	Overlap	Speed
Random sampling	OAB[35]	0.45	4.19	0.13	8.00
	MCT [28]	0.47	1.76	0.22	2.77
Structural sampling	LDP	0.51	1.84	0.28	4.36
	MIL [4]	0.42	3.11	0.17	5.99
	CMIL	0.43	2.47	0.19	5.14
	TRIC-track [94]	0.46	2.34	0.21	0.03
	G2T	0.45	2.13	0.20	0.43
	AOG-tracker	0.51	1.67	0.21	0.97
	LGT [13]	0.42	2.21	0.17	4.12
	Houghtrack [38]	0.42	3.61	0.15	0.87
	MatFlow	0.42	3.12	0.15	81.34
	CMT [73]	0.40	4.09	0.12	6.72
	LT-FLO [57]	0.44	4.44	0.13	1.83
	ZHANG	0.33	3.59	0.10	0.21
	FoT [90]	0.43	4.36	0.14	143.62
	BDF [65]	0.40	3.11	0.15	200.24
	FCT	0.43	3.34	0.15	83.37
	FragTrack [1]	0.43	4.85	0.11	2.08
	Struck [39]	0.47	1.61	0.25	2.44
	RobStruck	0.48	1.47	0.22	1.89
	SRAT	0.47	2.13	0.20	15.23
	EBT [112]	0.47	1.02	0.31	1.76
sPST [42]	0.55	1.48	0.28	1.01	
Dense sampling	KCFv2	0.48	1.95	0.19	10.90
	DSST[21]	0.54	2.56	0.17	3.29
	SAMF [59]	0.53	1.94	0.20	2.25
	SRDCF [24]	0.56	1.24	0.29	1.99
	PTZ-MOSSE	0.20	7.27	0.03	18.73
	NSAMF	0.53	1.29	0.25	5.47
	RAJSSC	0.57	1.63	0.24	2.12
	OACF [56]	0.58	1.81	0.22	2.00
	sKCF	0.48	2.68	0.16	66.22
	LOFT-Lite	0.34	6.35	0.08	0.75
	STC [108]	0.40	3.75	0.12	16.00
	MKCF+ [85]	0.52	1.83	0.21	1.23
	SME	0.55	1.98	0.21	4.09
	MvCFT	0.52	1.72	0.21	2.24
	MTSA-KCF	0.49	2.29	0.18	2.83
	MDNet [70]	0.60	0.69	0.38	0.87
DeepSRDCF [23]	0.56	1.05	0.32	0.38	
SO-DLT [93]	0.56	1.78	0.23	0.83	

Table 10 (continued)

Sample selection	Trackers	A	R	Overlap	Speed
SB selection	ASMS [91]	0.51	1.85	0.21	115.09
	SumShift [58]	0.52	1.68	0.23	16.78
	S3Tracker [58]	0.52	1.77	0.24	14.27
	PKLTF [81]	0.45	2.72	0.15	29.93
	IVT [79]	0.44	4.33	0.12	8.38
	CT [107]	0.39	4.09	0.11	12.90
	L1APG [6]	0.47	4.65	0.13	1.51
DF Selection	DFT [91]	0.46	4.32	0.14	3.33

transduction. An inductive algorithm can be used to predict the labels of samples that are unseen during training (irrespective of it being labeled or unlabeled). On the other hand, transductive algorithms are limited to predicting only the labels of unlabeled samples seen during training. In other words [14], if label predictions are only required for a given test set, transduction can be argued to be more straightforward than induction: while an inductive method infers a function $f : X \rightarrow Y$ on the entire space X , and afterward returns the evaluations $f(x_i)$ at the test points, transduction consists of directly estimating the finite set of test labels, i.e., a function $f : X_u \rightarrow Y$ only defined on the test set. Obviously, induction is attempting to address a more general and complex problem. Semi-supervised learning algorithms are not necessarily transductive or inductive. However, as we were exploring all the learning aspects for visual tracking, most learning models in this setting of model-free tracking employed the induction principle since the classifiers were trained in the first frame and then evaluated on the unlabeled data in subsequent frames. To train and update a universal learning machine to accommodate various appearance changes, induction seems the only option. Whether and how transduction can help produce a simpler solution in this setting is still a topic of debate.

9 Conclusion

In this article, we structured the literature body roughly based on self-learning and co-training — the two typical modes for semi-supervised online learning. For self-learning discriminative trackers, the criterion to differentiate these approaches is via sample selection strategies before samples are collected for model update, i.e., random sampling, sampling within structural constraints and dense sampling. For self-learning generative trackers, we summarized the two major mechanisms — predict-update and direct optimization, with which the appearance models need be designed using specific visual representations, e.g., sparse subspace representation and distribution field. Despite the discrepancy of terminologies between discriminative and generative trackers, we have found the conceptual consensus, i.e., basis selection or distribution field selection for generative trackers seem to share some similarity with the sample selection strategy for discriminative trackers in spirit. This conceptual consensus is rewarding for us to discuss the design considerations for a typical online adaptive tracking system, and could bring along insights and trigger more discussions in this field.

This survey results could potentially complement the currently existing data-driven experimental surveys and VOT challenge results because the increasingly emerging trackers

more often than not confuse researchers in terms of their performances and highly diversified approaches. The more diversified the approaches adopted, the more evident that online adaptive model-free tracking is an open-ended problem. On one hand, the computer vision community shifts its focus to more objective benchmarks and evaluation measures; we believe, on the other hand, a potentially meaningful guidance with conceptual consensus, i.e., semi-supervised online learning with various sample selection strategies, could shape a better big picture in this field.

References

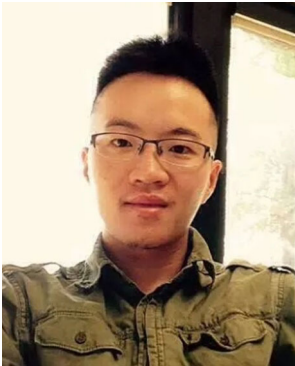
1. Adam A, Rivlin E, Shimshoni I (2006) Robust fragments-based tracking using the integral histogram. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol 1. IEEE, pp 798–805
2. Avidan S (2004) Support vector tracking. *IEEE Trans Pattern Anal Mach Intell* 26(8):1064–1072
3. Avidan S (2007) Ensemble tracking. *IEEE Trans Pattern Anal Mach Intell* 29(2):261–271
4. Babenko B, Yang M, Belongie S (2009) Visual tracking with online multiple instance learning. In: 2009. CVPR 2009. IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 983–990
5. Bai Q, Wu Z, Sclaroff S, Betke M, Monnier C (2013) Randomized ensemble tracking. In: 2013 IEEE International Conference on Computer Vision (ICCV). IEEE, pp 2040–2047
6. Bao C, Wu Y, Ling H., Ji H (2012) Real time robust L1 tracker using accelerated proximal gradient approach. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1830–1837
7. Black MJ, Jepson AD (1998) Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int J Comput Vis* 26(1):63–84
8. Black MJ, Jepson AD (1998) Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int J Comput Vis* 26(1):63–84
9. Bolme DS, Beveridge JR, Draper B, Lui YM (2010) Visual object tracking using adaptive correlation filters. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 2544–2550
10. Boykov Y, Kolmogorov V (2004) An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans Pattern Anal Mach Intell* 26(9):1124–1137
11. Cai Z, Wen L, Lei Z, Vasconcelos N, Li S (2014) Robust deformable and occluded object tracking with dynamic graph. *IEEE Trans Image Process* 23(12):5497–5509
12. Cehovin L, Leonardis A, Kristan M (2015) Visual object tracking performance measures revisited. arXiv:1502.05803
13. Cehovin L, Kristan M, Leonardis A (2013) Robust visual tracking using an adaptive coupled-layer visual model. *IEEE Trans Pattern Anal Mach Intell* 35(4):941–953
14. Chapelle, O, Zien, A, Scholkopf, B (Eds.) (2006). *Semi-supervised learning*. MIT Press
15. Chapelle, O, Zien, A, Scholkopf, B (Eds.). (2006c) *Semi-supervised learning*. MIT Press
16. Chapelle O (2007) Training a support vector machine in the primal. *Neural Comput* 19(5):1155–1178
17. Chen W, Cao L, Zhang J, Huang K (2013) An adaptive combination of multiple features for robust tracking in real scene. In: 2013 IEEE International Conference on Computer Vision Workshops (ICCVW). IEEE, pp 129–136
18. Collins RT (2003) Mean-shift blob tracking through scale space. In: 2003. Proceedings. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol 2. IEEE, pp II–234
19. Comaniciu D, Meer P (2002) Mean shift: A robust approach toward feature space analysis. *IEEE Trans Pattern Anal Mach Intell* 24(5):603–619
20. Comaniciu D, Ramesh V, Meer P (2003) Kernel-based object tracking. *IEEE Trans Pattern Anal Mach Intell* 25(5):564–577
21. Danelljan M, Häger G, Khan F, Felsberg M (2014) Accurate scale estimation for robust visual tracking. In: *British Machine Vision Conference*. BMVA Press, Nottingham, pp 1–5
22. Danelljan M, Khan F, Felsberg M, Weijer J (2014) Adaptive color attributes for real-time visual tracking. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp 1090–1097

23. Danelljan M, Hager G, Khan F, Felsberg M (2015) Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp 4310–4318
24. Danelljan M, Hager G, Shahbaz Khan F, Felsberg M (2015) Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp 4310–4318
25. Dinh TB, Vo N, Medioni G (2011) Context tracker: Exploring supporters and distracters in unconstrained environments. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1177–1184
26. Du W, Piater J (2008) A probabilistic approach to integrating multiple cues in visual tracking. In: Computer Vision ECCV 2008. Springer, Berlin Heidelberg, pp 225–238
27. Duffner S, Garcia C (2014) Exploiting contextual motion cues for visual object tracking. In: Computer Vision-ECCV 2014 Workshops. Springer International Publishing, pp 232–243
28. Duffner S, Garcia C (2014) Exploiting contextual motion cues for visual object tracking. In: European Conference on Computer Vision. Springer International Publishing, pp 232–243
29. Duffner S, Garcia C (2013) PixelTrack: a fast adaptive algorithm for tracking non-rigid objects. In: 2013 IEEE International Conference on Computer Vision (ICCV). IEEE, pp 2480–2487
30. Fan Z, Wu Y, Yang M (2005) Multiple collaborative kernel tracking. In: 2005. CVPR 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol 2. IEEE, pp 502–509
31. Felsberg M (2013) Enhanced distribution field tracking using channel representations. In: 2013 IEEE International Conference on Computer Vision Workshops (ICCVW). IEEE, pp 121–128
32. Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci* 55(1):119–139
33. Gavrilu DM (1999) The visual analysis of human movement: A survey. *Comput Vis Image Understand* 73(1):82–98
34. Gabriel PF, Verly JG, Piater J, Genon A (2003, September) The state of the art in multiple object tracking under occlusion in video sequences. In: Advanced Concepts for Intelligent Vision Systems, pp 166–173
35. Grabner H, Grabner M, Bischof H (2006, September) Real-time tracking via on-line boosting. In: BMVC, vol 1, p 6
36. Grabner H, Leistner C, Bischof H (2008) Semi-supervised on-line boosting for robust tracking. In: Computer Vision ECCV 2008. Springer Berlin Heidelberg, pp 234–247
37. Grabner H, Matas J, Van Gool L, Cattin P (2010) Tracking the invisible: Learning where the object might be. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1285–1292
38. Godec M, Roth PM, Bischof H (2013) Hough-based tracking of non-rigid objects. *Comput Vis Image Understand* 117(10):1245–1256
39. Hare S, Saffari A, Torr P (2011) Struck: Structured output tracking with kernels. In: 2011 IEEE International Conference on Computer Vision (ICCV). IEEE, pp 263–270
40. Henriques JF, Caseiro R, Martins P, Batista J (2012) Exploiting the circulant structure of tracking-by-detection with kernels. In: Computer Vision ECCV 2012. Springer, Berlin Heidelberg, pp 702–715
41. Henriques JF, Caseiro R, Martins P, Batista J (2015) High-speed tracking with kernelized correlation filters. *IEEE Trans Pattern Anal Mach Intell* 37(3):583–596
42. Hua Y, Alahari K, Schmid C (2015) Online object tracking with proposal selection. In: Proceedings of the IEEE International Conference on Computer Vision, pp 3092–3100
43. Hu W, Tan T, Wang L, Maybank S (2004) A survey on visual surveillance of object motion and behaviors. *IEEE Trans Syst, Man, Cybern, Part C: Appl Rev* 34(3):334–352
44. Isard M, Blake A (1998) Condensation conditional density propagation for visual tracking. *Int J Comput Vis* 29(1):5–28
45. Jia X, Lu H, Yang M (2012) Visual tracking via adaptive structural local sparse appearance model. In: 2012 IEEE Conference on Computer vision and pattern recognition (CVPR). IEEE, pp 1822–1829
46. Joachims T, Finley T, Yu CNJ (2009) Cutting-plane training of structural SVMs. *Mach Learn* 77(1):27–59
47. Kalal Z, Matas J, Mikolajczyk K (2010) Pn learning: Bootstrapping binary classifiers by structural constraints. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 49–56
48. Kalal Z, Mikolajczyk K, Matas J (2010) Forward-backward error: Automatic detection of tracking failures. In: 2010 20th International Conference on Pattern Recognition (ICPR). IEEE, pp 2756–2759

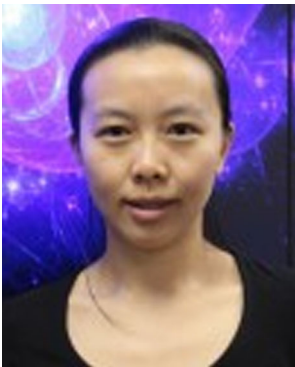
49. Kalal Z, Mikolajczyk K, Matas J (2012) Tracking-learning-detection. *IEEE Trans Pattern Anal Mach Intell* 34(7):1409–1422
50. Kim SJ, Koh K, Lustig M, Boyd S, Gorinevsky D (2007) An interior-point method for large-scale L1-regularized least squares. *IEEE J Select Top Signal Process* 1(4):606–617
51. Kristan M, Pflugfelder R, Leonardis A, Matas J, Porikli F, Cehovin L, Vojir T (2013) The Visual Object Tracking VOT2013 challenge results. *ICCV2013 Workshops*. In: Workshop on Visual Object Tracking Challenge
52. Kristan M, Pflugfelder R, Leonardis A, Matas J, Cehovin L, Nebehay G, Golodetz S (2014) The visual object tracking vot2014 challenge results. In: *Computer Vision-ECCV 2014 Workshops*. Springer International Publishing, pp 191–217
53. Kristan M, Matas J, Leonardis A, Vojir T, Pflugfelder R, Fernandez G (2015) A Novel Performance Evaluation Methodology for Single-Target Trackers. arXiv:1503.01313
54. Kristan M, Matas J, Leonardis A, Felsberg M, Cehovin L, Fernandez G, Vojir T, Hager G, Nebehay G, Pflugfelder R (2015) The visual object tracking VOT2015 challenge results. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp 1–23
55. Kwon J, Lee KM (2010) Visual tracking decomposition. In: *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp 1269–1276
56. L Bertinetto MO, J Valmadre GS, Torr P The importance of estimating object extent when tracking with correlation filters. Preprint, 2015
57. Lebeda K, Hadfield S, Matas J, Bowden R (2013) Long-term tracking through failure cases. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp 153–160
58. Lee JY, Yu W (2011) Visual tracking by partition-based histogram backprojection and maximum support criteria. In: *2011 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, pp 2860–2865
59. Li Y, Zhu J (2014) A scale adaptive kernel correlation filter tracker with feature integration. In: *Computer Vision-ECCV 2014 Workshops*. Springer International Publishing, pp 254–265
60. Liu B, Huang J, Yang L, Kulikowsk C (2011) Robust tracking using local sparse appearance model and k-selection. In: *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp 1313–1320
61. Li X, Hu W, Shen C, Zhang Z, Dick A, Hengel AVD (2013) A survey of appearance models in visual object tracking. *ACM Trans Intell Syst Technol (TIST)* 4(4):58
62. Louppe G (2014) Understanding Random Forests: From Theory to Practice. arXiv:1407.7502
63. Lowe, D <https://www.cs.ubc.ca/lowe/vision.html>
64. Mallapragada PK, Jin R, Jain AK, Liu Y (2009) Semiboost: Boosting for semi-supervised learning. *IEEE Trans Pattern Anal Mach Intell* 31(11):2000–2014
65. Maresca ME, Petrosino A (2014) Clustering local motion estimates for robust and efficient object tracking. In: *European Conference on Computer Vision*. Springer International Publishing, pp 244–253
66. Mei X, Ling H (2009) Robust visual tracking using L1 minimization. In: *2009 IEEE 12th International Conference on Computer Vision*. IEEE, pp 1436–1443
67. Mei X, Ling H (2011) Robust visual tracking and vehicle classification via sparse representation. *IEEE Trans Pattern Anal Mach Intell* 33(11):2259–2272
68. Moeslund TB, ranum E (2001) A survey of computer vision-based human motion capture. *Comput Vis Image Understand* 81(3):231–268
69. Moeslund TB, Hilton A, Krger V (2006) A survey of advances in vision-based human motion capture and analysis. *Comput Vis Image Understand* 104(2):90–126
70. Nam H, Han B (2015) Learning multi-domain convolutional neural networks for visual tracking. arXiv:1510.07945
71. Nam H, Hong S, Han B (2014) Online graph-based tracking. In: *Computer Vision ECCV 2014*. Springer International Publishing, pp 112–126
72. Nebehay G, Pflugfelder R (2014) Consensus-based matching and tracking of keypoints for object tracking. In: *2014 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp 862–869
73. Nebehay G, Pflugfelder R (2014) Consensus-based matching and tracking of keypoints for object tracking. In: *IEEE Winter Conference on Applications of Computer Vision*. IEEE, pp 862–869
74. Ofjall K, Felsberg M (2014) Weighted update and comparison for channel-based distribution field tracking. In: *Computer Vision-ECCV 2014 Workshops*. Springer International Publishing, pp 218–231
75. Okuma K, Taleghani A, De Freitas N, Little JJ, Lowe DG (2004) A boosted particle filter: Multitarget detection and tracking. In: *Computer Vision-ECCV 2004*. Springer, Berlin Heidelberg, pp 28–39
76. Oron S, Bar-Hillel A, Levi D, Avidan S (2012) Locally orderless tracking. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp 1940–1947

77. Pérez P, Hue C, Vermaak J, Gangnet M (2002) Color-based probabilistic tracking. In: Computer Vision ECCV 2002. Springer, Berlin Heidelberg, pp 661–675
78. Platt J (1999) Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods support vector learning*, 3
79. Ross DA, Lim J, Lin RS, Yang M (2008) Incremental learning for robust visual tracking. *Int J Comput Vis* 77(1-3):125–141
80. Sevilla-Lara L, Learned-Miller E (2012) Distribution fields for tracking. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1910–1917
81. Shi J, Tomasi C (1994) Good features to track. In: 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, pp 593–600
82. Sinha SN (2004) Graph Cut Algorithms in Vision, Graphics and Machine Learning An Integrative Paper. UNC Chapel Hill
83. Smeulders AW, Chu DM, Cucchiara R, Calderara S, Dehghan A, Shah M (2014) Visual tracking: an experimental survey. *IEEE Trans Pattern Anal Mach Intell* 36(7):1442–1468
84. Stalder S, Grabner H, Van Gool L (2009) Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In: 2009 IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops). IEEE, pp 1409–1416
85. Tang M, Feng J (2015) Multi-kernel correlation filter for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, pp 3038–3046
86. Tang F, Brennan S, Zhao Q, Tao H (2007) Co-tracking using semi-supervised support vector machines. In: ICCV 2007. IEEE 11th International Conference on Computer Vision, 2007. IEEE, pp 1–8
87. Tian M, Zhang W, Liu F (2007) On-line ensemble SVM for robust object tracking. In: Computer Vision/ACCV 2007. Springer, Berlin Heidelberg, pp 355–364
88. Vermaak J, Doucet A, Pérez P (2003) Maintaining multimodality through mixture tracking. In: 2003. Proceedings. Ninth IEEE International Conference on Computer Vision. IEEE, pp 1110–1116
89. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol 1. IEEE, pp I–511
90. Vojir T, Matas J (2014) The enhanced flock of trackers. In: Registration and Recognition in Images and Videos. Springer, Berlin Heidelberg, pp 113–136
91. Vojir T, Noskova J, Matas J (2013) Robust scale-adaptive mean-shift for tracking. In: Image Analysis. Springer, Berlin Heidelberg, pp 652–663
92. Tang N, Yeung DY (2013) Learning a deep compact image representation for visual tracking. In: Advances in Neural Information Processing Systems, pp 809–817
93. Wang N, Li S, Gupta A, Yeung DY (2015) Transferring rich feature hierarchies for robust visual tracking. arXiv:1501.04587
94. Wang X, Valstar M, Martínez B, Haris Khan M, Pridmore T (2015) Tric-track: Tracking by regression with incrementally learned cascades. In: Proceedings of the IEEE International Conference on Computer Vision, pp 4337–4345
95. Wendel A, Sternig S, Godec M (2011) Robustifying the flock of trackers. Citeseer, p 91
96. Williams O, Blake A, Cipolla R (2005) Sparse bayesian learning for efficient visual tracking. *IEEE Trans Pattern Anal Mach Intell* 27(8):1292–1304
97. Wright J, Ma Y, Mairal J, Sapiro G, Huang TS, Yan S (2010) Sparse representation for computer vision and pattern recognition. *Proc IEEE* 98(6):1031–1044
98. Wu Y, Lim J, Yang M (2013) Online object tracking: A benchmark. In: 2013 IEEE Conference on Computer vision and Pattern Recognition (CVPR). IEEE, pp 2411–2418
99. Wu Y, Shen B, Ling H (2012) Online robust image alignment via iterative convex optimization. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1808–1814
100. Xiao J, Stolk R, Leonardis A (2013) An enhanced adaptive coupled-layer LGTracker++. In: 2013 IEEE International Conference on Computer Vision Workshops (ICCVW). IEEE, pp 137–144
101. Yang M, Wu Y, Hua G (2009) Context-aware visual tracking. *IEEE Trans Pattern Anal Mach Intell* 31(7):1195–1209
102. Yilmaz A, Javed O, Shah M (2006) Object tracking: A survey. *Acm Comput Surv (CSUR)* 38(4):13
103. Yu Q, Dinh TB, Medioni G (2008) Online tracking and reacquisition using co-trained generative and discriminative trackers. In: Computer Vision ECCV 2008. Springer, Berlin Heidelberg, pp 678–691
104. Zeisl B, Leistner C, Saffari A, Bischof H (2010) On-line semi-supervised multiple-instance boosting. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1879–1879
105. Zeisl B, Leistner C, Saffari A, Bischof H (2010) On-line semi-supervised multiple-instance boosting. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 1879–1879
106. Zhang L, Van der Maaten LJ (2014) Preserving structure in model-free tracking. *IEEE Trans Pattern Anal Mach Intell* 36(4):756–769

107. Zhang K, Zhang L, Yang M (2012) Real-time compressive tracking. In: European Conference on Computer Vision. Springer, Berlin Heidelberg, pp 864–877
108. Zhang K, Zhang L, Liu Q, Zhang D, Yang M (2014) Fast visual tracking via dense spatio-temporal context learning. In: European Conference on Computer Vision. Springer International Publishing, pp 127–141
109. Zhang T, Ghanem B, Liu S, Ahuja N (2012) Low-rank sparse learning for robust visual tracking. In: Computer Vision ECCV 2012. Springer, Berlin Heidelberg, pp 470–484
110. Zhong W, Lu H, Yang M (2012) Robust object tracking via sparsity-based collaborative model. In: 2012 IEEE Conference on Computer vision and pattern recognition (CVPR). IEEE, pp 1838–1845
111. Zhu G, Porikli F, Li H Tracking randomly moving objects on edge box proposals. arXiv:[1507.08085](https://arxiv.org/abs/1507.08085).2015
112. Zhu G, Porikli F, Li H (2015) Tracking randomly moving objects on edge box proposals. arXiv:[1507.08085](https://arxiv.org/abs/1507.08085)



Xinpeng L. Liao received his B.S. degree in electrical engineering from the University of PLA Information Engineering, China and his M.S. degree in electrical engineering from a joint program between University of Science and Technology of China and the 57th Research Institute of PLA General Staff Headquarters, China. In Fall 2013 he joined the Knowledge Discovery and Data Mining Laboratory (KDDM) at the University of Alabama at Birmingham, under the supervision of Prof. Chengcui Zhang. Currently, he is pursuing the Ph.D. degree in computer and information sciences. His primary research interests include Image Retrieval, Online Learning for Model-free Tracking and Multimedia Forensics and Security. He has authored 2 journal and 5 conference papers.



Dr. Chengcui Zhang is a Professor in the Department of Computer and Information Sciences, the University of Alabama at Birmingham and an Associate Director of the Knowledge Discovery and Data Mining Laboratory. She has authored and coauthored over 140 research papers focusing on Multimedia Systems, Multimedia Data Mining, Multimedia Information Retrieval, Multimedia Databases, Bioinformatics, and Geographic Information Systems. She has served on more than 100 international conferences and workshops and taken

a leadership role at the 2014 IEEE International Conference on Multimedia and Expo (ICME), the 2012–2013 IEEE International Conference on Information Reuse and Integration (IRI), 2012 IEEE International Symposium on Multimedia (ISM), and 2010 IEEE International Conference on Multimedia and Ubiquitous Engineering (MUE). She is currently the Chair of IEEE Technical Committee on Semantic Computing and an Associate Editor of IEEE Transactions on Multimedia.