Chapter 8
Toward Robust Online Visual Tracking

Ming-Hsuan Yang and Jeffrey Ho

Abstract We pursue a research direction that will empower machines with simultaneous tracking and recognition capabilities similar to human cognition. Toward that, we develop algorithms that leverage prior knowledge/model obtained offline with information available online via novel learning algorithms. While humans can effortlessly locate moving objects in different environments, visual tracking remains one of the most important and challenging problems in computer vision. Robust cognitive visual tracking algorithms facilitate answering important questions regarding how objects move and interact in complex environments. They have broad applications including surveillance, navigation, human computer interfaces, object recognition, motion analysis and video indexing, to name a few.

Keywords Visual tracking · Object tracking · Online learning · Incremental learning

1 Introduction

While we have witnessed significant progress in visual tracking over the last decade [3, 7–9, 11, 12, 23, 28, 30, 59, 63], developing visual tracking systems that match human cognitive abilities is still a very challenging research problem. Existing visual tracking systems tend to perform well over short durations, and more importantly

M.-H. Yang (✉)
Electrical Engineering and Computer Science, University of California, Merced, CA 95344, USA
e-mail: mhyang@ucmerced.edu

J. Ho
Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32607, USA
e-mail: jho@cise.ufl.edu

B. Bhanu et al. (eds.), *Distributed Video Sensor Networks*,
DOI 10.1007/978-0-85729-127-1_8,

only when the target objects stay visible in camera view (i.e., not in and out of the scene). One main reason is that most existing algorithms employ a static representation of the target object and operate on the premise of constancy in appearance. In other words, most algorithms assume that the appearance of a target object does not change rapidly. For such algorithms to perform robustly, it is imperative to collect a large set of training images to account for all possible appearance variations caused by change of viewing angles and illumination. These models do not exploit rich and important information (e.g., most recent appearance and illumination condition) that becomes available online during tracking. More importantly, it is of great interest to develop algorithms that leverage prior knowledge and online learning to enhance the recognition and tracking capabilities. Another reason is that most existing algorithms are not able to detect and recover from drifts accumulated during tracking. Once the target position is initialized, most tracking algorithms operate as a series of predictions, and consequently accumulated drifts are inevitable unless they are able to reinitialize their positions periodically. Finally, tracking articulated objects poses additional difficulties due to high dimensionality of the state variables and partial occlusion.

The above-mentioned problems entail the need for learning robust appearance models adaptively which in turn facilitate the tracking processes as well as algorithms to detect and correct deviations from the true target locations. Specifically, a robust appearance model should constantly learn a compact notion of the "thing" being tracked rather than treating the target as a set of independent pixels, i.e., "stuff" [2]. For visual tracking, an appearance model needs to be learned efficiently and effectively to reflect the most recent appearance change of *any* target objects. We note that it is a daunting, if not impossible, task to collect a large set of data encompassing *all* appearance variation of a target object caused by change of pose, illumination, shape, and occlusion. Meanwhile, it is equally important to exploit prior knowledge or model, when available, within the online learning framework. Clearly, there is a need to develop robust algorithms that can learn to update appearance models online for *any* objects, and use these models to address drifting problems. We emphasize that the problems of detection, tracking, recognition and appearance models can be simultaneously addressed with online and prior learning. Here we present our works in addressing these problems.

2 Appearance Modeling for Visual Tracking

Visual tracking essentially deals with non-stationary data, both the target object and the background, that change over time. Most existing algorithms are able to track objects, either previously seen or not, in short durations and in well controlled environments. However, due to drastic change in the object's appearance or large lighting variation in its surroundings, these algorithms usually do not perform well after some period of time or have significant drifts. Although such problems can be ameliorated with recourse to richer representations and effective prediction schemes, most algorithms typically operate on the premise that the model of the target object

does not change drastically over time. These algorithms usually adopt a model of the target object first and then use it for tracking, without adapting the model to account for appearance change of the object due to variation of imaging conditions (e.g., viewing angles and illumination). Furthermore, it is usually assumed that all images are acquired with a stationary camera.

2.1 Learning Nonlinear Appearance Manifold

It is well known that images of an object taken under different lighting and pose can be well modeled with nonlinear manifold via a set of linear subspaces [6, 40, 43]. However, prior work has focused on learning such models in a batch mode offline fashion. Here we describe algorithms that use nonlinear appearance models learned offline and show how they facilitate tracking and recognition tasks.

We propose to learn nonlinear manifold with online update and clustering, as well as their underlying constraints. The nonlinear manifold is modeled with a set of submanifolds constructed in an online manner, where each submanifold is approximated with a PCA (Principal Component Analysis) subspace. It entails the need to efficiently process the incoming images into clusters from which submanifolds are constructed and updated. Each submanifold model is expected to capture certain appearance variation of the target object due to illumination and pose change. In addition, a nonlinear manifold provides a way to retain "long-term memory" of the target rather than to rely on one single subspace which has only "short-term memory."

Learning Nonlinear Manifold Online

The complex nonlinear appearance manifold of a target object k, M_k, is partitioned by a collection of submanifolds, $C^{k1}, C^{k2}, \ldots$, where each models the appearances of the target object under illumination and pose change. The submanifold is approximated by a low-dimensional linear subspace computed by PCA using images observed sequentially (see Fig. 1).

The tight coupling between the tracking and recognition components is achieved via the shared appearance models $M_1, \ldots, M_N$. Another difficulty is related to the definition of the ℓ_2-distance $d(I, M_k)$ between an image I and a manifold M_k in the image space. By definition, $d(I, M_k) = d(I, x^*)$ with x^* is a point on M_k having minimal ℓ_2-distance to I (see Fig. 1). Even if an analytic description of M_k were available, finding x^* is generally not an easy problem. In our case M_k is, at best, modeled by a modest number of images sampled from it; therefore, M_k is available to us only through a very coarse and sparse representation with many "gaps" in which we have inadequate or incompletely information. The main focus of our work is to provide an effective definition for $d(I, M_k)$ that works for a coarse representation of M_k.

Since the appearance manifold M_k is nonlinear, it is reasonable to decompose M_k into a collection of m simpler disjoint submanifolds, $M_k = C^{k1} \cup \cdots \cup C^{km}$, with C^{ki} denoting a submanifold in a decomposition of person k's appearance manifold. Each C^{ki} is assumed to be amenable to linear approximations by a low-dimensional linear subspace computed through Principal Component Analysis (i.e., a PCA plane). We define the conditional probability $p(C^{ki}|I)$ as the probability that C^{ki} contains a point x with minimal distance to I. With $p_{M_k}(x|I) = \sum_{i=1}^{m} p(C^{ki}|I) p_{C^{ki}}(x|I)$, we have,

$$d(I, M_k) = \int_{M_k} d(x, I) p_{M_k}(x|I)\, dx = \sum_{i=1}^{m} p\big(C^{ki}|I\big) \int_{C^{ki}} d(x, I) p_{C^{ki}}(x|I)\, dx$$
$$= \sum_{i=1}^{m} p\big(C^{ki}|I\big) d\big(I, C^{ki}\big). \tag{1}$$

The equation above shows that the expected distance $d(I, M_k)$ can be treated as the expected distance between I and each C^{ki}. In addition, this equation transforms the integral to a finite summation which is feasible to compute numerically. The details of this formation can be found in [25, 31].

For visual tracking and recognition in video sequences, we can exploit temporal coherence between consecutive image frames. As shown in the right panel of Fig. 1, the ℓ_2-distance may occasionally be misleading during tracking/recognition. But if we consider previous frames in an image sequence rather than just one, then the set of closest points x^* will trace a curve on a submanifold C^{ki}. In our framework, this is embodied by the term $p(C^{ki}|I)$ in (1). We apply Bayesian inference to incorporate temporal information to provide a better estimate of $p(C^{ki}|I)$ and thus $d(I, M_k)$; this will then yield better tracking/recognition performance. We show that

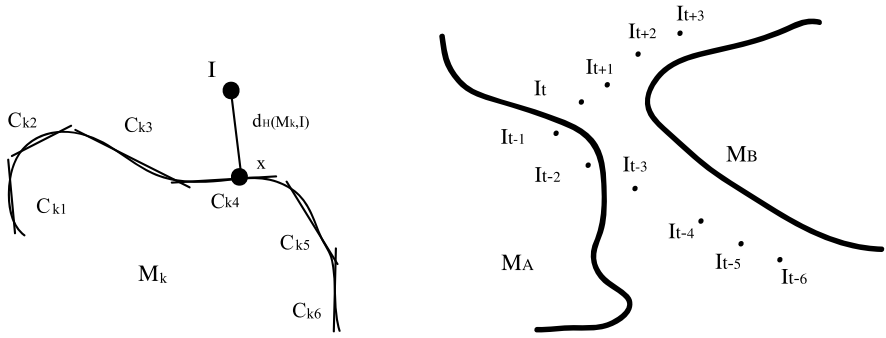

Fig. 1 (*Left*) Appearance manifold: A complex and nonlinear manifold M_k can be approximated as the union of several simpler submanifolds; here, each submanifold C^{ki} is represented by a PCA plane. (*Right*) Difficulty of frame-based tracking/recognition: The *two solid curves* denote two different appearance manifolds, M_A and M_B. It is difficult to reach a decision on the identity from frame I_{t-3} to frame I_t because these frames have smaller ℓ_2-distance to appearance manifolds M_A than M_B. However, by looking at the sequence of images $I_{t-6} \ldots I_{t+3}$, it is apparent that the sequence has most likely originated from appearance manifold M_B [25, 31]

it is a recursive formulation that depends on the generative model, $p(I|C^{ki})$, and the transition probability, $p(C_t^{ki}|C_{t-1}^{kj})$ [25, 31]. The connectivity between the submanifolds is modeled as transition probabilities, between pairs of submanifolds, and these are learned directly online via frequency estimation or simple counting. The integrated task of tracking and recognition is formulated as a maximum a posteriori estimation problem. Within our framework, the tracking and recognition modules are complementary to each other, and the capability and performance of one are enhanced by the other. Our approach contrasts sharply with more rigid conventional approaches in which these two modules work independently and in sequence.

Recent work on incremental clustering data streams [10, 29] has shown its promise for its applicability to numerous types of data, including web documents, routing packages, financial transactions, and telephone records. We may draw on such ideas and extend to learning image data online for learning a set of PCA subspaces. Specially, it is worthwhile exploiting the characteristics pertaining to 2D image data in developing new algorithms to handle image sequences. In the vision context, it is important to exploit the fact that the similarity measure can be better modeled with distance from an image to a subspace. For image sequences, one may compute the distance from an incoming image I to the submanifold, C^{ki} (see Fig. 1) rather than the ℓ_2-distance to other images. On the other hand, randomization and sampling schemes have shown much promise in fast approximation of clustering data streams. We will exploit both characteristics in developing online approximate algorithms that are able to assign each image I to one or a couple of submanifold C^{ki} for weighted update.

Online Update of Submanifold

For each submanifold modeled by a PCA subspace, we have developed an efficient online subspace update algorithm [35] for appearance model based on the R-SVD algorithm [21] and the sequential Karhunen–Loeve method [32]. The proposed method not only updates the orthonormal basis but also the subspace means, which is of great importance for certain applications. For example, it can be applied to adaptively update the between-class and within-class covariance matrices used in Fisher linear discriminant analysis [37]. Experimental results show that our subspace update algorithm is 20% more efficient than the most related work [24].

We develop robust tracking algorithms using online appearance model with subspace update [35, 47, 48]. In contrast to the eigentracking algorithm [7], our algorithm does not require a training phase but instead learns the eigenbases online during the object tracking process. Thus our appearance model can adapt to account for change in pose, view angle, and illumination which is not captured by the set of training images. Our appearance model provides a richer description than simple curves or splines as used in [28], and has a compact notion of the "thing" being tracked [2]. The learned representation can also be utilized for other tasks such as object recognition. Furthermore, our algorithm is able to simultaneously track and learn a compact representation of the target object even when the camera is moving.

Fig. 2 A person undergoing large pose, expression, appearance, and lighting change, as well as partial occlusions. The *red window* shows the maximum a posteriori estimate of the particle filter, and the green windows show the other particles with large weights. The *images in the second row* show the current sample mean, tracked region, reconstructed image, and the reconstruction error respectively. The *third row* shows the top 10 principal eigenvectors [35, 48]. The MATLAB code and data sets can be found at http://faculty.ucmerced.edu/mhyang

Our experiments [35, 48] show that robust tracking results can be obtained using this representation without employing more complicated wavelet features as in [30], although this elaboration is still possible and may lead to even better results. Figure 2 shows some experimental results using our algorithm. Note also that the view-based eigenbasis representation has demonstrated its ability to model the appearance of objects in different poses [40], and under different lighting conditions [6]. Consequently, the learned eigenbasis facilitates tracking objects undergoing illumination and pose change.

2.2 Leveraging Prior Knowledge with Online Learning

Cognitive psychologists have suggested computational models to explain human visual cognition in terms of long-term and short-term memories [26]. Numerous studies suggest that interplay between long-term and short-term memories explains how humans track and recognize objects. For vision problems such as object recognition, we have access to prior knowledge of the objects. One natural way is to exploit the prior knowledge obtained offline with the information obtained online, thereby simultaneously enhancing the abilities to recognize and track objects robustly. In such situations, the prior knowledge can be encoded as long-term visual memory via construction of nonlinear manifold offline while the proposed online update algorithm serves as short-term memory to account for the most recent appearance change.

Our algorithms [25, 31] facilitate the integration of long-term and short-term memories via the use of submanifold construction and update. The long-term memory provides rich prior information about the object appearance that helps in assigning one or more subspaces to account for appearance change when a new image arrives. In our study with one single submanifold, the newly arrived images are always added to the retained covariance matrix. It is of great interest to develop algorithms with soft assignments for multiple submanifolds to account for the appearance of a newly arrived image. In addition, algorithms may not take an irrevocable action when a data point arrives, and may modify the current model after

a group of image data arrives. Another direction to pursue is to periodically check the reconstruction errors after a group of points have been added. If the recognition error increases over time, it suggests the data points previously added should be discarded or downweighted. As each submanifold is modeled by a PCA subspace modeled with a covariance matrix, adding or deleting particular data points can be carried out efficiently with matrix update and downdate [21].

3 Learning Detectors Online for Visual Tracking

It has been shown that in many scenarios an adaptive appearance model, which evolves during the tracking process as the appearance of the object changes, is the key to good performance [30, 48]. Another choice in the design of appearance models is whether to model only the object [5, 48], or both the object and the background [3, 11, 22, 37]. Many of the latter approaches have shown that training a model to separate the object from the background via a discriminative classifier can often achieve superior results. In this case, the tracking problem becomes a detection one as the target is located by scanning through the image region as shown in the left panel of Fig. 3. In particular, the recent advances in face detection [61] have inspired some successful real-time tracking algorithms [22]. However, almost all the detectors are constructed offline for a *specific* object class (e.g., faces, cars, and pedestrians) which demand significant efforts in collecting data as well as training time [13, 38, 50, 51, 56, 61].

Another challenge that is often not discussed in the literature is how to choose positive and negative examples when updating the appearance model. Most commonly this is done by taking the current tracker location as one positive example, and

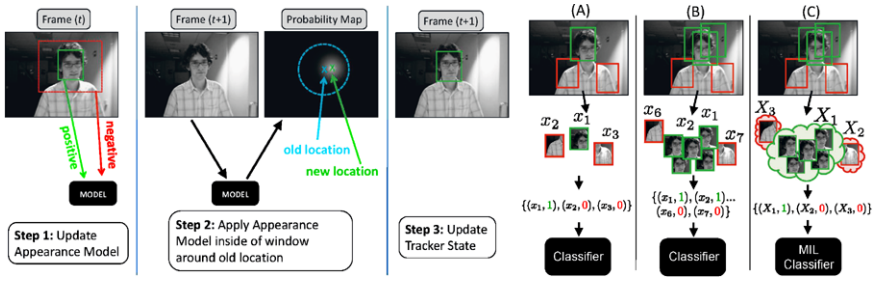

Fig. 3 (*Left*) Tracking by detection with a greedy motion model: Generally, the appearance model is a discriminative classifier that can be trained in an online manner. A greedy motion model is used to search for the most probable location of the object in a frame within some search window. An alternative is to use particle filter. (*Right*) Updating a discriminative appearance model: (**A**) Using a single positive image patch to update a traditional discriminative classifier. The positive image patch chosen does not capture the object perfectly. (**B**) Using several positive image patches to update a traditional discriminative classifier. This can confuse the classifier causing poor performance. (**C**) Using one positive bag consisting of several image patches to update a MIL classifier [4]. The C++ code and data sets can be found at http://faculty.ucmerced.edu/mhyang

sampling the neighborhood around the tracker location for negatives. If the tracker location is not precise, however, the appearance model ends up getting updated with a suboptimal positive example. Over time this can degrade the model, and can cause drift. On the other hand, if multiple positive examples are used (taken from a small neighborhood around the current tracker location), the model can become confused and its discriminative power can suffer as illustrated in the right panel of Fig. 3.

Similar problems are encountered in object detection because it is difficult for a human labeler to be consistent with respect to how the positive examples are cropped. In other words, the *exact* object locations are unknown. In fact, Viola et al. [62] argue that object detection has inherent ambiguities that make it more difficult to train a classifier using traditional methods. For this reason they suggest the use of a Multiple Instance Learning (MIL) [15] approach for training object detectors offline. The basic idea of this learning paradigm is that during training, examples are presented in sets (often called "bags"), and labels are provided for the bags rather than individual instances. If a bag is labeled positive it is assumed to contain at least one positive instance, otherwise the bag is negative. For example, in the context of object detection, a positive bag could contain a few possible bounding boxes around each labeled object (e.g., a human labeler clicks on the center of the object, and the algorithm crops several rectangles around that point). Therefore, the ambiguity is passed on to the learning algorithm, which now has to figure out which instance in each positive bag is the most "correct." Although one could argue that this learning problem is more difficult in the sense that less information is provided to the learner, it is actually easier in the sense that there is less risk of *correct* information being lost.

3.1 Multiple Instance Learning

We present an online learning algorithm that builds detectors *specific* to the target object *online* for robust visual tracking [4]. The basic flow of the tracking system is illustrated in the left panel of Fig. 3, and it contains three components: image representation, appearance model and motion model. Local feature-based or part-based representations have been demonstrated to perform well when the objects are partially occluded [38, 61]. Our image representation consists of a set of Haar-like features that are computed efficiently for each image patch [16, 61]. The appearance model is comprised of a discriminative classifier which is able to return $p(y = 1|x)$ (we will use $p(y|x)$ as shorthand), where x is an image patch (or the representation of an image patch in feature space) and y is a binary variable indicating the presence of the object of interest in that image patch. At every time step t, our tracker maintains the object location l_t^*. Let $l(x)$ denote the location of image patch x. For each new frame we crop out a set of image patches $X^s = \{x|s > \|l(x) - l_{t-1}^*\|\}$ that are within some search radius s of the current tracker location, and compute $p(y|x)$ for all $x \in X^s$. We update the tracker location with maximum likelihood (ML). In other words, we do not maintain a distribution of the target's location at every frame

and use a motion model where the location of the tracker at time t is equally likely to appear within a radius s of the tracker location at time $(t-1)$, although it could be extended with something more sophisticated, such as a particle filter, as is done in [48].

Once the tracker location is updated, we proceed to update the appearance model. We crop out a set of patches $X^r = \{x | r > \|l(x) - l_t^*\|\}$, where $r < s$ is the positive radius, and label this bag positive (recall that in MIL we train the algorithm with labeled bags). On the other hand, if a standard learning algorithm were used, there would be two options: set $r = 1$ and use this as a single positive instance, or set $r > 1$ and label all these instances positive. For negatives we crop out patches from an annular region $X^{r,\beta} = \{x | \beta > \|l(x) - l_t^*\| > r\}$, where r is same as before, and β is a scalar. Since this generates a potentially large set, we then take a random subset of these image patches and label them negative. We place each negative example into its own negative bag. Note that we could place all negative examples into a single negative bag. However, our intuition is that there is no ambiguity about negative examples, so placing them into separate bags makes more sense. Figure 3 (right panel) contains an illustration comparing appearance model updating using MIL and a standard learning algorithm.

Traditional discriminative learning algorithms for training a binary classifier that estimates $p(y|x)$ require a training data set of the form $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where x_i is an instance (in our case a feature vector computed for an image patch), and $y_i \in \{0, 1\}$ is a binary label. In the Multiple Instance Learning framework the training data has the form $\{(X_1, y_1), \ldots, (X_n, y_n)\}$ where a bag $X_i = \{x_{i1}, \ldots, x_{im}\}$ and y_i is a bag label. The bag labels are defined as:

$$y_i = \max_j(y_{ij}), \tag{2}$$

where y_{ij} are the instance labels, which are assumed to exist but are not known during training. Numerous algorithms have been proposed for solving the MIL problem [15, 62]. The algorithm that is most closely related to our work is the MILBoost algorithm [62], which uses the gradient boosting framework [19] to train a boosting classifier that maximizes the log likelihood of bags:

$$\log \mathcal{L} = \sum_i \big(\log p(y_i|X_i)\big). \tag{3}$$

Notice that the likelihood is defined over bags and not instances, because instance labels are unknown during training, and yet the goal is to train an instance classifier that estimates $p(y|x)$. We therefore need to express $p(y_i|X_i)$, the probability of a bag being positive, in terms of its instances. In [62] the Noisy-OR (NOR) model is adopted for doing this: $p(y_i|X_i) = 1 - \prod_j (1 - p(y_i|x_{ij}))$. This equation has the desired property that if one of the instances in a bag has a high probability, the bag probability will be high as well. However, the MILBoost algorithm is a batch algorithm and cannot be trained in an online manner as we need in our tracking application.

We propose an online MIL boosting algorithm to learn object specific detectors for visual tracking. The goal of boosting is to combine many weak classifiers $\mathbf{h}(x)$

(usually decision stumps) into an additive strong classifier: $\mathbf{H}(x) = \sum_{k=1}^{K} \alpha_k \mathbf{h}_t(x)$ where α_k are scalar weights. There have been many boosting algorithms proposed to learn this model in batch mode [18, 19], but generally this is done in a greedy manner where the weak classifiers are trained sequentially. After each weak classifier is trained, the training examples are re-weighted such that examples that were previously misclassified receive more weight. If each weak classifier is a decision stump, then it chooses one feature that has the most discriminative power for the entire training set. In this case boosting can be viewed as performing feature selection, choosing a total of K features, which is generally much smaller than the size of the entire feature set.

3.2 Learning Detectors with Online Multiple Instance Boosting

In [42], Oza develops an online variant of the discrete AdaBoost algorithm [18], which minimizes the exponential loss function. As such, this online algorithm is limited to classification problems. We take a statistical view of boosting similar to in [19] where the algorithm minimizes a generic loss function J. In this view, the weak classifiers are chosen sequentially to optimize the following criteria:

$$(\mathbf{h}_k, \alpha_k) = \underset{\mathbf{h}\in\mathcal{H},\alpha}{\operatorname{argmin}} J(\mathbf{H}_{k-1} + \alpha\mathbf{h}), \tag{4}$$

where $\mathbf{H}_{k-1}$ is the strong classifier made up of the first $(k-1)$ weak classifiers, and $\mathcal{H}$ is the set of all possible weak classifiers. In batch boosting algorithms, the loss function J is computed over the entire training data set.

In our case, for the current video frame we are given a training data set $\{(X_1, y_1), (X_2, y_2), \ldots\}$, where $X_i = \{x_{i1}, x_{i2}, \ldots\}$. We would like to update our estimate of $p(x|y)$ to minimize the negative log likelihood of these data (3). We model the instance probability as $p(y|x) = \sigma(\mathbf{H}(x))$ where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, and the bag probabilities $p(y|X)$ using the NOR model described above. To simplify the problem, we absorb the scalar weights α_t into the weak classifiers, by allowing them to return real values rather than binary. To perform online feature selection, our algorithm maintains a pool of $M > K$ candidate weak classifiers h. We update all of these weak classifiers in parallel, similar to [22]. Note that although examples are passed in bags, the weak classifiers in a MIL algorithm are instance classifiers, and therefore require instance labels y_{ij}. Since these are unavailable, we pass in the bag label y_i for all instances x_{ij} to the weak training procedure. We then choose K weak classifiers $\mathbf{h}$ from the candidate pool sequentially, using the following criteria:

$$\mathbf{h}_k = \underset{h\in\{h_1,\ldots,h_M\}}{\operatorname{argmin}} \log \mathcal{L}(\mathbf{H}_{k-1} + h). \tag{5}$$

We use classifier h_k that can be updated online, and each classifier is composed of a Haar-like feature f_k and modeled with univariate Gaussian distributions whose parameters are updated online. The classifiers return the log likelihood ratio based

on the estimated Gaussian distributions. When the weak classifier receives new data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ we use the weighted running mean and variance for update. To learn detectors online for real-time visual tracking, we represent each image patch as a vector of Haar-like features [61] where they are randomly generated, similar to [16]. The feature value is then a weighted sum of the pixels in all the rectangles, which can be computed efficiently using the integral image described in [61]. The proposed online visual tracking algorithm with Multiple Instance Learning is dubbed as MILTrack algorithm.

We evaluate the proposed MILTrack algorithm on several challenging video sequences, some of which are publicly available. For comparison, we implemented a tracker based on the Online-AdaBoost (OAB) algorithm described in [22]. We plug this learning algorithm into our system, and used the same features and motion model as for MILTrack. We demonstrate the merits of the proposed MILTrack algorithm with experiments where *all* algorithm parameters were fixed (i.e., no tuning for particular sequences). To further gauge performance we also compare our results to the recently proposed algorithms using online discrete AdaBoost [22] and local histograms [1]. For MILTrack we sample positives in each frame using a positive radius $r = 5$, which generates a total of 45 image patches composing one positive bag. For the OAB tracker we experiment with two variations. In the first variation we set $r = 1$ generating only one positive example per frame; in the second variation we set $r = 5$ as we do in MILTrack (although in this case each of the 45 image patches is labeled positive). The reason we experiment with these two versions was to show that the superior performance of MILTrack is not simply due to the fact that we extract multiple positive examples per frame. Some results are shown in Fig. 4 and Fig. 5. These sequences exhibit many occlusions, lighting and appearance variations, and fast motion which causes motion blur. For the "Occluded Face" sequence,

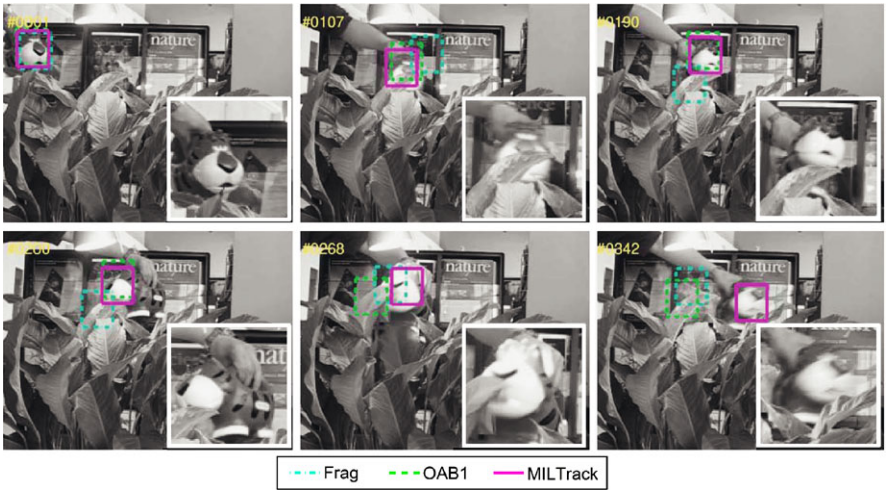

Fig. 4 Screen shots of tracking results with zoom-in images [4]. Videos and source code can be found at http://faculty.ucmerced.edu/mhyang

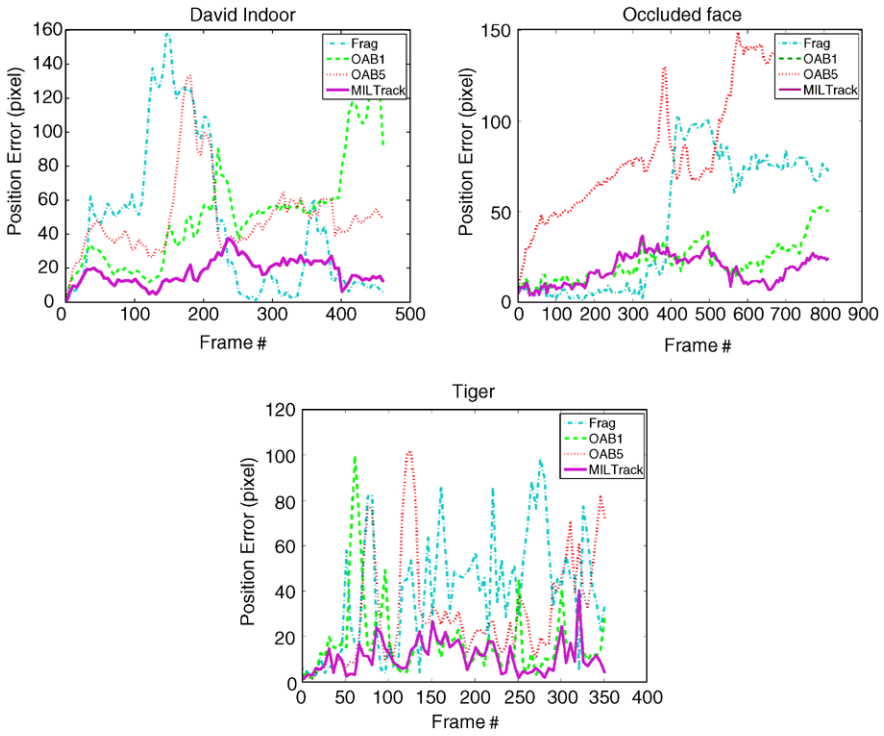

Fig. 5 Error plots for three test video clips. See [4] for details

FragTrack performs poorly because it cannot handle appearance changes well (e.g., when the subject puts a hat on, or turns his face).

In all cases our MILTrack algorithm outperforms both versions of the Online AdaBoost Tracker, and in most cases it outperforms the FragTrack algorithm as well. The reason for the superior performance is that the Online MILBoost algorithm is able to handle ambiguously labeled training examples, which are provided by the tracker itself. On the other hand, when the Online AdaBoost Tracker is updated with multiple positive examples it performs quite poorly because the classifier is unable to learn a good decision boundary between positives and negatives. We notice that even when MILTrack loses the target due to severe occlusions, it is able to recover quickly since the temporary distraction to the appearance model is not as significant.

The proposed online MILBoost algorithm can easily exploit the prior knowledge of the target object. From a set of training images, we can extract a set of Haar-like features that best model the target object before applying online MILBoost for visual tracking. In addition, the motion model we used here is fairly simple, and could be replaced with something more sophisticated, such as a particle filter as in [48] for additional gain in performance. We also plan to investigate the use of other part-based appearance models [1] with our algorithm and evaluate these alternative representation methods. The proposed algorithms provides the basic mechanism to

detect and recognize the objects when they come in and out of the camera view. One straightforward way is to set thresholds on the detector confidence. However, more sophisticated algorithms will be investigated to empower machines to mimic human-level visual cognition.

3.3 Articulated Objects

Tracking articulated objects is of great importance for motion analysis with broad impact. Once we are able to track single objects robustly with online algorithms, the next question is how we can extend these ideas to articulated objects.

A major difficulty in applying Bayesian tracking methods for tracking 3D human body pose is the high dimensionality of the state vector—typically 20–60 dimensions depending on the parameterization [14, 52, 55, 60]. Although the state vector is high dimensional, the pose parameters typically can be assumed to lie on a low-dimensional manifold embedded in the high-dimensional space. We propose to approximate the low-dimensional manifold so that the dimensionality of the state vector is reduced for efficient and effective Bayesian tracking [36]. To achieve this goal, a statistical model known as the globally coordinated mixture of factor analyzers (GCMFA) is learned from motion capture data. This model provides a global parametrization of the low-dimensional manifold. Each factor analyzer in the mixture is a "locally linear dimensionality reducer" that approximates a part of the manifold. The global parametrization of the manifold is obtained via aligning these locally linear pieces in a global coordinate system. The parameters of the GCMFA model for our application are learned from motion capture sequences. Since the GCMFA is effective in preserving important information during dimensionality reduction [49, 57], it can capture the key kinematic information with the use of motion capture sequences as training data. The global coordination of the local linear factor analyzers ensures that poses have a globally consistent parameterization in the latent space. The global coordination also preserves the continuity of the manifold as similar poses are mapped to the coordinates that are close to each other on the manifold. The density of high-dimensional pose data is approximated by the piecewise linear Gaussian factor analyzers in the low-dimensional space. By encouraging the internal coordinates of the factor analyzers to agree, a single, coherent low-dimensional coordinate system can be obtained for dimensionality reduction. The mixing and coordination of the linear factors provides nonlinear bidirectional mappings between the low-dimensional (latent) space and the pose space. Because the nonlinear mapping functions are broken down into linear factors, the learning algorithm is efficient and can handle large training data sets with grace.

Once the GCMFA model is learned, we demonstrate its use in a multiple hypothesis tracker with a dimensionality reduced state space for 3D human tracking. The performance of this tracker is currently being evaluated on the HumanEva benchmark data sets [54]. In experiments with real videos, the proposed system reliably tracks body motion during self-occlusions and in the presence of motion blur. Figure 6 shows some tracking results using the proposed algorithm, annealed particle

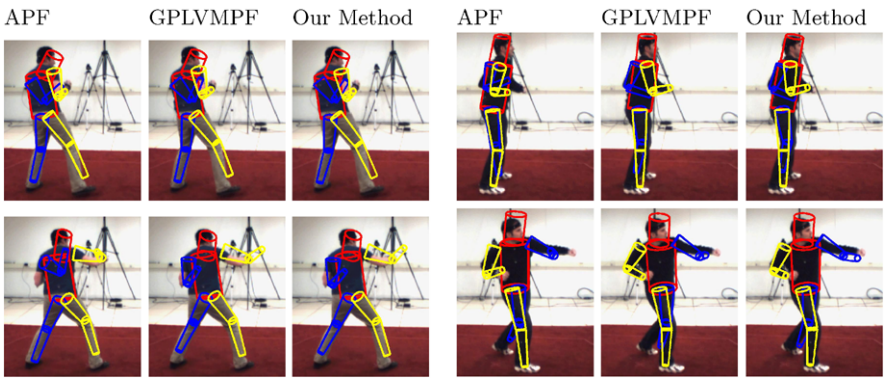

Fig. 6 Sample tracking results from the test video sequence of S2 (*Left*) and S3 (*Right*) performing boxing using the annealed particle filter [14] (APF), the GPLVM-based method [58, 60] (GPLVMPF), and the proposed method [33]

[14] (APF) and Gaussian Process Latent Variable Model (GPLVM) [58, 60]. The proposed algorithm is able to accurately track large movements of the human limbs in adjacent time steps by propagating each cluster's information over time in the multiple hypothesis tracker. Some quantitative evaluation [33] using the HumanEva benchmark data sets shows that our method produces more accurate 3D pose estimates than those obtained via two previously-proposed Bayesian tracking methods [14, 60].

Although the GCMFA framework has all the desirable properties of a dimensionality reduction algorithm for tracking, a main disadvantage is that one has to choose the optimal structure of the GCMFA model empirically or manually. We address this issue by proposing a variational Bayesian solution [34] for automatic selection of the optimal model structure in a way similar to [20]. In addition, we plan to learn part-based object detector online using the adaptive appearance models presented in Sect. 2.1 as well as the algorithm described in Sect. 3.2, and exploit the constraints enforced among them with dimensionality reduction techniques such as GCMFA.

Several methods have shown the potential of exploiting constraints among subspaces and parts in vision applications [53]. With the current algorithm, the dimensionality reducers are used mainly to map between the input and low-dimensional spaces. One way to extend the current algorithm is to further exploit the temporal and spatial constraints of the clusters in the low-dimensional space. The appearance model in our current algorithm can be improved with online update using the algorithms discussed in Sects. 2.1 and 3 to better account for change in illumination and shape. It is of great interest to extend the online algorithms discussed in Sect. 3.2 to learn detectors for parts of an articulated object. Different from the recent work that learn body parts offline [17, 27, 38, 39, 45, 46, 53], we aim to exploit the potential of learning detectors online with their constraints aside from relying on prior knowledge. As it involves learning multiple detectors simultaneously as well as their kinematic constraints, we expect to explore top-down and bottom-up approaches for efficient visual tracking.

One single model or algorithm is not expected to succeed in all tracking scenarios, and we will explore other representations such as integral histograms [44] for tracking articulated objects at a distance. Our recent results show that articulated objects at a distance can be well tracked by integrating online appearance models, object segmentation and spatial constraints of the articulating parts [41]. We plan to pursue this line of research to account for larger shape deformation and self occlusions.

4 Conclusions

The ultimate goal of our research focuses on developing efficient and effective algorithms that mimic human cognitive abilities for tracking as well as recognizing objects. Toward that, we have developed several algorithms that leverage online and offline information for robust tracking and recognition. As one single model or method is not expected to succeed in all tracking scenarios, we plan to exploit generative and discriminative algorithms for tracking objects in different scenarios. We also aim to further explore the interplay between online and offline learning for robust appearance models.

References

1. Adam, A., Rivlin, E., Shimshoni, I.: Robust fragments-based tracking using the integral histogram. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 798–805 (2006)
2. Adelson, E.H., Bergen, J.R.: The plenoptic function and the elements of early vision. In: Landy, M., Movshon, J.A. (eds.) Computational Models of Visual Processing, pp. 1–20. MIT Press, Cambridge (1991)
3. Avidan, S.: Ensemble tracking. IEEE Trans. Pattern Anal. Mach. Intell. **29**(2), 261–271 (2007)
4. Babenko, B., Yang, M.-H., Belongie, S.: Visual tracking with online multiple instance learning. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 983–990 (2009)
5. Balan, A.O., Black, M.J.: An adaptive appearance model approach for model-based articulated object tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 758–765 (2006)
6. Belhumeur, P., Kreigman, D.: What is the set of images of an object under all possible lighting conditions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 270–277 (1997)
7. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. Int. J. Comput. Vis. **26**(1), 63–84 (1998)
8. Bregler, C., Malik, J.: Tracking people with twists and exponential map. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 8–15 (1998)
9. Cham, T.J., Rehg, J.M.: A multiple hypothesis approach to figure tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 239–245 (1998)
10. Charikar, M., Chekuri, C., Feder, T., Motwani, R.: Incremental clustering and dynamic information retrieval. SIAM J. Comput. **33**(6), 1417–1440 (2004)
11. Collins, R.T., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Trans. Pattern Anal. Mach. Intell. **27**(10), 1631–1643 (2005).

12. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. **25**(5), 564–577 (2003)
13. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
14. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 126–133 (2000)
15. Dietterich, T.G., Lathrop, R.H., Perez, L.T.: Solving the multiple-instance problem with axis parallel rectangles. Artif. Intell. **89**(1–2), 31–71 (1997)
16. Dollár, P., Tu, Z., Tao, H., Belongie, S.: Feature mining for image classification. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007
17. Forsyth, D., Arikan, O., Ikemoto, L., O'Brien, J., Ramanan, D.: Computational Studies of Human Motion: Part 1, Tracking and Motion Synthesis. Now publishers, Hanover (2006)
18. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**, 119–139 (1997)
19. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. Ann. Stat. **29**(5), 1189–1232 (2001)
20. Ghahramani, Z., Beal, M.: Variational inference for Bayesian mixtures of factor analysers. In: Advances in Neural Information Processing Systems, pp. 449–455 (2000)
21. Golub, G.H., Van Loan, C.F.: Matrix Computations. The Johns Hopkins University Press, Baltimore (1996)
22. Grabner, H., Grabner, M., Bischof, H.: Real-time tracking via on-line boosting. In: Proceedings of British Machine Vision Conference, pp. 47–56 (2006)
23. Hager, G.D., Belhumeur, P.N.: Efficient region tracking with parametric models of geometry and illumination. IEEE Trans. Pattern Anal. Mach. Intell. **20**(10), 1025–1039 (1998)
24. Hall, P., Marshall, D., Martin, R.: Adding and subtracting eigenspaces with eigenvalue decomposition and singular value decomposition. Image Vis. Comput. **20**(13–14), 1009–1016 (2002)
25. Ho, J., Lee, K.-C., Yang, M.-H., Kriegman, D.: Visual tracking using learned linear subspaces. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 782–789 (2004)
26. Humphreys, G., Bruce, V.: Visual Cognition: Computational, Experimental and Neuropsychological Perspectives. Psychology Press, London (1989)
27. Ioffe, S., Forsyth, D.: Probabilistic methods for finding people. Int. J. Comput. Vis. **43**(1), 45–68 (2001)
28. Isard, M., Blake, A.: CONDENSATION—conditional density propagation for visual tracking. Int. J. Comput. Vis. **29**(1), 5–28 (1998)
29. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: A review. ACM Comput. Surv. **31**(3), 264–323 (1999)
30. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. IEEE Trans. Pattern Anal. Mach. Intell. **25**(10), 1296–1311 (2003)
31. Lee, K.-C., Ho, J., Yang, M.-H., Kriegman, D.: Visual tracking and recognition using probabilistic appearance manifolds. Comput. Vis. Image Underst. **99**(3), 303–331 (2005)
32. Levy, A., Lindenbaum, M.: Sequential Karhunen-Loeve basis extraction and its application to images. IEEE Trans. Image Process. **9**(8), 1371–1374 (2000)
33. Li, R., Yang, M.-H., Sclaroff, S., Tian, T.-P.: Monocular tracking of 3D human motion with a coordinated mixture of factor analyzers. In: Proceedings of European Conference on Computer Vision, pp. 137–150 (2006)
34. Li, R., Tian, T.-P., Sclaroff, S., Yang, M.-H.: 3D human motion tracking with a coordinated mixture of factor analyzers. Int. J. Comput. Vis. **87**(1–2), 170–190 (2010)
35. Lim, J., Ross, D., Lin, R.-S., Yang, M.-H.: Incremental learning for visual tracking. In: Advances in Neural Information Processing Systems, pp. 793–800. MIT Press, Cambridge (2005)

36. Lin, R.-S., Liu, C.-B., Yang, M.-H., Ahuja, N., Levinson, S.: Learning nonlinear manifolds from time series. In: Proceedings of European Conference on Computer Vision, pp. 239–250 (2004)
37. Lin, R.-S., Ross, D., Lim, J., Yang, M.-H.: Adaptive discriminative generative model and its applications. In: Advances in Neural Information Processing Systems, pp. 801–808. MIT Press, Cambridge (2005)
38. Mohan, A., Papageorgiou, C., Poggio, T.: Example-based object detection in images by components. IEEE Trans. Pattern Anal. Mach. Intell. **23**(4), 349–361 (2001)
39. Moselund, T., Granum, E.: A survey of computer vision-based human motion capture. Comput. Vis. Image Underst. **81**(3), 231–268 (2001)
40. Murase, H., Nayar, S.: Visual learning and recognition of 3d objects from appearance. Int. J. Comput. Vis. **14**(1), 5–24 (1995)
41. Nejhum, S.M.S., Ho, J., Yang, M.-H.: Online articulate object tracking with appearance and shape. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008
42. Oza, N.C.: Online Ensemble Learning. Ph.D. Thesis, University of California, Berkeley (2001)
43. Pentland, A., Moghaddam, B., Starner, T., Oligide, O., Turk, M.: View-based and modular eigenspaces for face recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 84–91 (1994)
44. Porikli, F.: Integral histogram: A fast way to extract histograms in Cartesian spaces. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 829–836 (2005)
45. Ramanan, D., Forsyth, D.: Finding and tracking people from the bottom up. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 467–474 (2003)
46. Ronfard, R., Schmid, C., Triggs, B.: Learning to parse pictures of people. In: Proceedings of the Seventh European Conference on Computer Vision, pp. 700–714 (2002)
47. Ross, D., Lim, J., Yang, M.-H.: Adaptive probabilistic visual tracking with incremental subspace update. In: Proceedings of European Conference on Computer Vision, pp. 470–482 (2004)
48. Ross, D., Lim, J., Lin, R.-S., Yang, M.-H.: Incremental learning for robust visual tracking. Int. J. Comput. Vis. **77**(1–3), 125–141 (2008)
49. Roweis, S., Saul, L., Hinton, G.E.: Global coordination of local linear models. In: Advances in Neural Information Processing Systems, pp. 889–896 (2001)
50. Rowley, H., Baluja, S., Kanade, T.: Neural network-based face detection. IEEE Trans. Pattern Anal. Mach. Intell. **20**(1), 23–38 (1998)
51. Schneiderman, H., Kanade, T.: Object detection using the statistics of parts. Int. J. Comput. Vis. **56**(3), 151–177 (2004)
52. Sidenbladh, H., Black, M.: Learning image statistics for Bayesian tracking. In: Proceedings of IEEE International Conference on Computer Vision, pp. 709–716 (2001)
53. Sigal, L., Bhatia, S., Roth, S., Black, M., Isard, M.: Tracking loose-limbed people. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 421–428 (2004)
54. Sigal, L., Black, M.: HumanEva: synchronized video and motion capture dataset for evaluation of articulated human motion. Technical Report CS-06-08, Brown University (2006)
55. Sminchisescu, C., Triggs, B.: Covariance scaled sampling for monocular 3D body tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 447–454 (2001)
56. Sung, K.-K., Poggio, T.: Example-based learning for view-based human face detection. IEEE Trans. Pattern Anal. Mach. Intell. **20**(1), 39–51 (1998)
57. Teh, Y.W., Roweis, S.: Automatic alignment of local representations. In: Advances in Neural Information Processing Systems, pp. 841–848 (2002)
58. Tian, T.-P., Li, R., Sclaroff, S.: Tracking human body pose on a learned smooth space. Technical Report 2005-029, Boston University (2005)
59. Toyama, K., Blake, A.: Probabilistic tracking with exemplars in a metric space. Int. J. Comput. Vis. **48**(1), 9–19 (2002)

60. Urtasun, R., Fleet, D., Hertzmann, A., Fua, P.: Priors for people tracking from small training sets. In: Proceedings of IEEE International Conference on Computer Vision, pp. 403–410 (2005)
61. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 511–518 (2001)
62. Viola, P., Platt, J.C., Zhang, C.: Multiple instance boosting for object detection. In: Advances in Neural Information Processing Systems, pp. 1417–1426. MIT Press, Cambridge (2005)
63. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Comput. Surv. **38**(4), 1–45 (2006)