What Are We Tracking: A Unified Approach of Tracking and Recognition

Jialue Fan, Xiaohui Shen, *Student Member, IEEE*, and Ying Wu, *Senior Member, IEEE*

***Abstract*— Tracking is essentially a matching problem. While traditional tracking methods mostly focus on low-level image correspondences between frames, we argue that high-level semantic correspondences are indispensable to make tracking more reliable. Based on that, a unified approach of low-level object tracking and high-level recognition is proposed for single object tracking, in which the target category is actively recognized during tracking. High-level offline models corresponding to the recognized category are then adaptively selected and combined with low-level online tracking models so as to achieve better tracking performance. Extensive experimental results show that our approach outperforms state-of-the-art online models in many challenging tracking scenarios such as drastic view change, scale change, background clutter, and morphable objects.**

***Index Terms*— Object recognition, video analysis, visual tracking.**

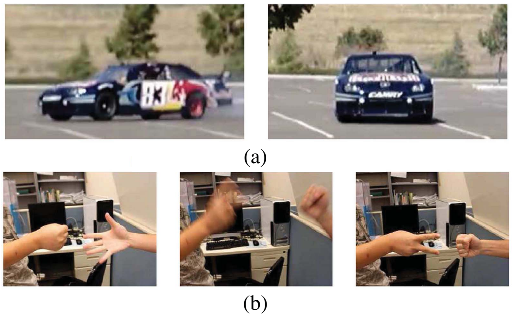


Fig. 1. Traditional methods may fail in these complicated scenarios, while our approach handles them well. (a) Rapid appearance change. (b) Object morphing.

I. Introduction

Tracking is closely related to constructing correspondences between frames. Traditional tracking approaches focus on finding *low-level* correspondences based on image evidence. Online models for low-level correspondences are generally employed to adapt to the changing appearances of the target [1]–[3]. However, one notable shortcoming of these online models is that they are constructed and updated based on the previous appearance of the target without much semantic understanding. Therefore, they are limited in predicting unprecedented states of the target due to significant view changes and occlusion, and easily drift in the case when the appearance of the target changes too fast. Figure 1(a) shows an example where the visual appearance of the target changes dramatically in a very short time period, making low-level image correspondences unstable. Without other information, it is very likely to cause tracking failure, no matter what online model is used. However, if we can recognize this target as a car at a higher level, the tracking task becomes to find the same *car* in the subsequent images instead of finding the object with the same low-level appearance. Therefore, the discriminative information provided by the car category, i.e., the *high-level* correspondences, can be utilized to help successfully track the target. In other words, to make tracking consistently effective in various challenging scenarios, it is necessary to combine both low-level and high-level correspondences.

Some offline-trained high-level detectors with semantic meanings have already been introduced into the tracking-by-detection scheme for some specific tracking tasks, especially for human tracking [4]–[6] and vehicle tracking [7], which largely improves the tracking performance. However, these models assume the semantic meanings of targets are already known before tracking, and accordingly cannot be applied to many general applications. Consider a video surveillance scenario with a complex scene, the categories of the moving objects cannot be predicted. Nevertheless, every moving object should be correctly tracked for subsequent analysis, no matter whether it is a human, a car or even an animal. In other cases, the category of the target might change because of object morphing and camouflage (e.g., in Fig. 1(b) the states of the hand are switching between “rock”, “paper”, “scissor”), in which those pre-determined detectors are likely to fail.

After all, tracking is not the final goal of video analysis but an intermediate task for some succeeding high-level processing like event detection and scene understanding. Essentially, an ideal tracking system should *actively understand* the target, and adaptively incorporate high-level semantic correspondences and low-level image correspondences. Towards this end, this paper proposes a unified approach for object tracking and recognition. In our approach, once an object is discovered and tracked, the tracking results are continuously fed forward to the upper-level video-based recognition scheme, in which

Manuscript received February 27, 2012; revised August 15, 2012; accepted August 28, 2012. Date of publication September 13, 2012; date of current version January 10, 2013. This work was supported in part by the National Science Foundation under Grant IIS-0347877 and Grant IIS-0916607, and the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant ARO W911NF-08-1-0504. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zhou Wang.

J. Fan was with Northwestern University, Evanston, IL 60208 USA (e-mail: jialue.fan@u.northwestern.edu).

X. Shen and Y. Wu are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208 USA (e-mail: xsh835@eecs.northwestern.edu; yingwu@eecs.northwestern.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2012.2218827

dynamic programming is adopted to recognize the category of the object in the current frame. Based on the feedback from the recognition results, different off-line models dedicated to specific categories are adaptively selected, and the location of the tracked object in the next frame is determined by integrated optimization of these selected detectors and the tracking evidence.

Compared with online tracking models and previous tracking-by-detection schemes, our framework has the following advantages.

1) Unlike previous tracking-by-detection methods in which the offline detectors are fixed for one category, our framework can actively recognize the target and adaptively utilize the high-level semantic information to improve the robustness of tracking. Besides, our combination of object tracking and recognition is not based on the discrete, sparse output of the detectors, but achieved by an integrated optimization scheme, which accordingly makes our tracking method more flexible to difficult scenarios.
2) Our approach is not only able to handle many difficult tracking scenarios such as background clutter, view changes, and severe occlusion, but also works well in some extreme situations (e.g., tracking a morphable object). Moreover, the output of our approach is further used for video understanding and event detection.

II. Related Work

Tracking has been an active research topic for decades [8]–[21], and a review of all the tracking methods is beyond the scope of this paper. Here we only mention some examples that are mostly related to our work.

Traditional online models for tracking include appearance-based templates [22] (e.g., color regions [23], and stable structures [24]), and online classifiers trained by boosting [3]. However, the efficacy of these online models heavily relies on the past tracking performance and tends to drift when the appearance of the objects keeps changing and tracking errors are accumulated. Therefore, much effort has been devoted to model updating to enhance their discriminative power and prevent the models from drifting [1], [2], [22], [25]–[29]. Nevertheless, these learning methods are still based on the immediate samples of the targets in a limited time period. If the object appearance abruptly changes to some states that have not been seen before, these models are very likely to fail.

More recently, some pre-trained offline models and databases have been incorporated to the online tracking models for some specific tracking tasks [5], [7], [30]–[32]. In [5], a human body is represented as an assembly of body parts. The responses of these part detectors are then combined as the observations for tracking. To address the occlusion problem in people tracking, Andriluka *et al.* [30] extract people-tracklets from consecutive frames and thus build models of the individual people, which is an extension of [31]. When the targets are pedestrians and vehicles, [7] formulates object detection and space-time trajectory estimation in a coupled optimization problem. [32] assigns a class-specific codebook to each tracked target. However, these methods assume the categories of the targets are known before tracking, which is quite a strong assumption. When the objects of interests are unknown, such prior knowledge would not be available.

There are also some work aiming to perform simultaneous tracking and recognition [33]–[37]. Reference [33] embeds the motion model and appearance model in a particle filter for face tracking, and constructs the intra- and extra-personal spaces for recognition. Image variations are further modeled via the union of several submanifolds in [34]. SURFTrac [36] tracks the objects by interest point matching and updating, and then continuously extracts feature descriptors for recognition. In [37], Rotation-Invariant Fast Features (RIFF) are used for unified tracking and recognition. In [35], an illumination aware MCMC particle filter is exploited for long-term outdoor multi-object simultaneous tracking and classification. However, these methods still treat tracking and recognition as independent steps and use conventional tracking approaches without the help of the higher-level recognition feedback. Different from this scheme, our method focuses on the information fusion of recognition and tracking, in which the recognition results are fed back to select different models and combine them in a unified optimization framework; and the tracking results are meanwhile fed forward to the recognition system, which hereby forms a closed-loop adaptation. Moreover, the recognition modules in these methods are different from the standard approaches in object recognition literature, because these methods recognize specific object instances (e.g., people identification [33], [34]) rather than object categories. On the contrary, we focus on semantic object recognition which yields object categories as the output.

III. Formulation

A. Overview Description

The framework of the proposed method is shown in Fig. 2. The object of interest is initialized by a user-specified bounding box, but its category is not provided. This target may or may not have a semantic meaning. Therefore, in the first few frames when the tracker does not know the target category, tracking the target only relies on the online target model, which is the same as traditional tracking. Meanwhile, video-based object recognition is applied on the tracked objects. When the target is recognized properly, the offline target model will be automatically incorporated to provide more information about the target.

At time t, we denote the target state by $\mathbf{x}_t$, and the target category by c_t.[1] Denote the image sequence by $\mathcal{I}_t = \{\mathbf{I}_1, \ldots, \mathbf{I}_t\}$, where $\mathbf{I}_t$ is the input image at time t. So the target measurement at time t is $\mathbf{z}_t = \mathbf{I}_t(\mathbf{x}_t)$. In a visual tracking framework, the online target model is generally based on low-level features. We denote the online target model by

[1] The motivation of defining the time varying c_t is that we wish to consider a general case where c_t may vary during time. For example, c_t can be used to describe the target status for morphable objects. For the conventional object recognition scenario, the target category is fixed, but we can still use c_t because the sequence $\{c_1, c_2, \ldots\}$ converges to the true target category c as the evidence is accumulated.

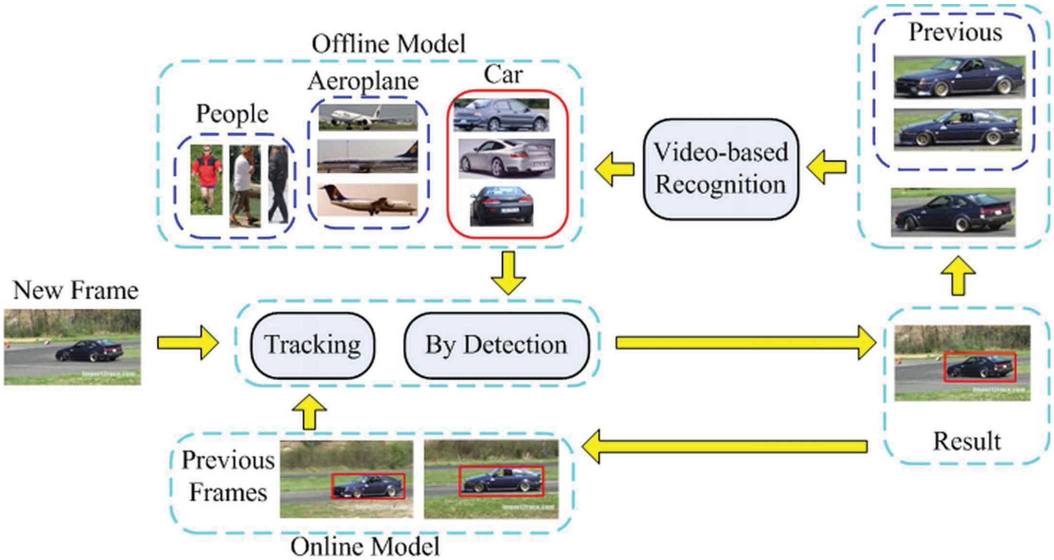


Fig. 2. Framework of the proposed method. In the first few frames, the object category is unknown, so our tracking procedure only relies on the online target model, which is the same as traditional tracking. Meanwhile, video-based object recognition is applied on the tracked objects. When the target is recognized properly, the offline target model will be automatically incorporated to provide more information about the target.

$M_t^L = \Psi(\mathbf{z}_1, \ldots, \mathbf{z}_t)$, where Ψ is a mapping (e.g., extracting feature descriptors from the target).

For the object category, we consider there are N different object classes (denoted by $C^1, \ldots, C^N$). For the regions not belonging to any known class or even with no semantic meanings, we denote by C^0 the complementary set, namely the "others" class. Hence, $c_t \in \{C^0, C^1, \ldots, C^N\}$. Each object class C^i is associated with a specific offline model ($M_{C^i}^H$), which is an abstraction of a specific object class.[2]

At time t, our objective is to estimate $\mathbf{x}_t$ and c_t, based on the input image sequence $\mathcal{I}_t$ as well as the offline model. Generally, it is very difficult to estimate $\mathbf{x}_t$ and c_t simultaneously. Therefore we employ a two-step EM-like method here: at time t, we first estimate $\mathbf{x}_t$ (i.e., "tracking"), and then estimate c_t based on the new tracking result $\mathbf{z}_t = \mathbf{I}_t(\mathbf{x}_t)$ (i.e., "recognition"). In the next subsections, we will present these two steps in details.

B. Tracking Procedure

Different from traditional tracking, the estimation of $\mathbf{x}_t$ in our approach is based on the online target model M_{t-1}^L, the offline model $M_{c_{t-1}}^H$ selected by the previous recognition result c_{t-1}, and the current input image $\mathbf{I}_t$. In a Bayesian perspective, we have

$$\begin{aligned}\mathbf{x}_t^* &= \arg\max_{\mathbf{x}_t \in \mathbf{\Omega}} p(\mathbf{x}_t | M_{t-1}^L, M_{c_{t-1}}^H, \mathbf{I}_t)\\ &= \arg\max_{\mathbf{x}_t \in \mathbf{\Omega}} p(M_{t-1}^L, M_{c_{t-1}}^H | \mathbf{x}_t, \mathbf{I}_t) p(\mathbf{x}_t | \mathbf{I}_t)\\ &= \arg\max_{\mathbf{x}_t \in \mathbf{\Omega}} p(M_{t-1}^L | \mathbf{x}_t, \mathbf{I}_t) p(M_{c_{t-1}}^H | \mathbf{x}_t, \mathbf{I}_t) p(\mathbf{x}_t | \mathbf{I}_t)\\ &= \arg\max_{\mathbf{x}_t \in \mathbf{\Omega}} p(M_{t-1}^L | \mathbf{x}_t, \mathbf{I}_t) p(M_{c_{t-1}}^H | \mathbf{x}_t, \mathbf{I}_t). \qquad (1)\end{aligned}$$

In the third equation of Eq. 1, we assume M_{t-1}^L and $M_{c_{t-1}}^H$ are conditionally independent given image $\mathbf{I}_t$ and position $\mathbf{x}_t$, because M_{t-1}^L can be viewed as the target online appearance variation, and $M_{c_{t-1}}^H$ is related to the intrinsic appearance of the target. This argues that the two models are independent given the image observations. Also, as will be shown in Eq. 7, c_t does not depend on the online tracking model, once the image measurement $\mathbf{I}_t(\mathbf{x}_t)$ is given. The last equation means that we consider $p(\mathbf{x}_t|\mathbf{I}_t) = \frac{1}{|\Omega|}$ is a uniform distribution in the search space of the target.[3]

When the recognition result is not available ($c_{t-1} = C^0$), the problem is simplified as $\mathbf{x}_t^* = \arg\max_{\mathbf{x}_t \in \mathbf{\Omega}} p(\mathbf{x}_t | M_{t-1}^L, \mathbf{I}_t)$, where only the online object model is considered. For every frame, the online target model is updated as $M_t^L = \Psi(\mathbf{z}_1, \ldots, \mathbf{z}_t)$. In this paper, the state $\mathbf{x}_t = \{x, y, w, h\}$ consists of the target central position (x, y), its width w, and its height h.

Maximizing the likelihood term in Eq. 1 can be formulated as an energy minimization problem by defining the energy term $E = -\ln p$. Therefore

$$\mathbf{x}_t^* = \arg\min_{\mathbf{x}_t \in \mathbf{\Omega}} E(\mathbf{x}_t) = \arg\min_{\mathbf{x}_t \in \mathbf{\Omega}} E_t(\mathbf{x}_t) + E_d(\mathbf{x}_t) \qquad (2)$$

where $E_t(\mathbf{x}_t) = -\ln p(M_{t-1}^L | \mathbf{x}_t, \mathbf{I}_t)$ is the energy term related to tracking, and $E_d(\mathbf{x}_t) = -\ln p(M_{c_{t-1}}^H | \mathbf{x}_t, \mathbf{I}_t)$ is the energy term related to detection. The term "detection" is consistent with the widely used term "tracking-by-detection" in state-of-the-art literature. Please note that we absorb the normalization factors into the energy terms without confusion. Both energy terms are further decomposed.

1) Tracking Term: The widely used correspondences in object tracking are point correspondences and region correspondences. The point correspondences reflect the local

[2] As the online target model is related to "low-level" features, the offline model is related to object recognition which is a "high-level" task, we use the superscript "L" and "H" for online and offline models respectively.

[3] Like [1]–[3], [10], [12], we do not model the dynamics of the target in our framework, since the dynamic models in current literature may not be appropriate for scale change, abrupt motion, etc.

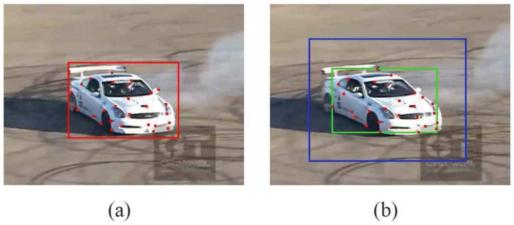


Fig. 3. Determine the search range. (a) Previous frame. The red bounding box shows the target location at the previous frame. (b) Current frame. The red points are corresponding matching points between two frames. The green box is the inner boundary, while the blue one is the outer boundary.

information, while the region correspondences reflect the global information. As contour correspondences may not always be reliable in cluttered situation, we do not employ them here. Therefore, the tracking term $E_t(\mathbf{x}_t)$ can be written as the weighted sum of the energy terms of these two types, i.e., $E_t(\mathbf{x}_t) = \omega_s E_s(\mathbf{x}_t) + \omega_h E_h(\mathbf{x}_t)$.

For point correspondences, we choose Harris-corner points with SIFT descriptor as our salient points, and denote the set of the salient points on the target at time t by s_i^t. For each salient point s_i^{t-1} on the target, we record its relative position w.r.t the target center as $\mathbf{l}_i^{t-1}$, normalized by the target size. And for each s_i^{t-1}, we find its correspondence b_i^t at time t by SIFT matching, with the matching error ω_i^t (Fig. 3 gives one example of corresponding matching points, shown by red points). Given the candidate region $\mathbf{x}_t$, the relative position of b_i^t is uniquely determined, denoted by $\mathbf{g}_i^t$. We assume that the relative position of the same salient point w.r.t the target cannot change rapidly, so:

$$E_s(\mathbf{x}_t) = \sum_i \|\mathbf{l}_i^{t-1} - \mathbf{g}_i^t\|^2 e^{-\omega_i^t/\sigma}. \tag{3}$$

Ideally, if the target movement is only translation or scaling, $E_s = 0$. So E_s is related to the deformation of the target.

For region correspondences, we employ the color histogram matching. We obtain the target histogram h^{t-1} at time $t-1$ in the HSV color space. Then:

$$E_h(\mathbf{x}_t) = \|h(\mathbf{x}_t)^t - h^{t-1}\|^2 \tag{4}$$

where $\|\cdot\|$ is the L_2 norm (outlier saturated). The target histogram $h(\mathbf{x}_t)^t$ can be quickly computed with an integral image.

2) Detection Term: $E_d(\mathbf{x}_t)$ measures the difference between the target candidate $\mathbf{x}_t$ and the specific offline model $M^H_{c_{t-1}}$ of class c_{t-1}. This is quite related to image object recognition. We define a cost $U(d, c) = -\ln p(d|c)$ where d is the measurement of the target instance and $c \in \{C^0, C^1, \ldots, C^N\}$ is a specific class. The object detection for a specific class c is indeed $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\Omega} U(d(\mathbf{x}), c)$, while the recognition procedure is formulated as finding the best c^* such that $c^* = \arg \min_{c\in\{C^0,C^1,\ldots,C^N\}} U(d, c)$. Therefore, we use the same cost function $U(d, c)$ for object detection and recognition.

In object detection, an exemplar-based energy term $E_d(\mathbf{x}_t) = U(\mathbf{z}_t, c_{t-1})$ (recall $\mathbf{z}_t = \mathbf{I}_t(\mathbf{x}_t)$) is designed for each specific class c_{t-1}, which can be decomposed as the weighted sum $E_d(\mathbf{x}_t) = \omega_p E_p(\mathbf{x}_t) + \omega_e E_e(\mathbf{x}_t)$, where $E_p(\mathbf{x}_t)$ is related to the pyramid matching of salient points [38], and $E_e(\mathbf{x}_t)$ is related to the pyramid matching of the histograms of edge directions:

$$E_p(\mathbf{x}_t) = \frac{1}{r}\sum_{j=1}^{r} \|\mathbf{f}_p(\mathbf{z}_t) - NN_{j,c_{t-1}}\mathbf{f}_p(\mathbf{z}_t)\|^2 \tag{5}$$

$$E_e(\mathbf{x}_t) = \frac{1}{r}\sum_{j=1}^{r} \|\mathbf{f}_e(\mathbf{z}_t) - NN_{j,c_{t-1}}\mathbf{f}_e(\mathbf{z}_t)\|^2 \tag{6}$$

where $\mathbf{f}_p(\cdot)$ and $\mathbf{f}_e(\cdot)$ are the features extracted for spatial pyramid matching of SIFT descriptors and edge histograms, respectively (more details are given in Sec. III-C). $NN_{j,c_{t-1}}\mathbf{f}_p(\mathbf{z}_t)$ is the jth nearest neighbor of $\mathbf{z}_t$ in the feature space $\mathbf{f}_p$ from the training samples of class c_{t-1}. Eq. 6 is defined in the same way. Please note that the edge histograms for all hypothesis can be quickly computed with the integral image. Both terms are commonly used in object detection methods [39]. When $c_{t-1} = C^0$, the offline model is not activated, and thus $E_d(\mathbf{x}_t) = 0$.

3) Optimization Method: As the optimization problem $\mathbf{x}_t^* = \arg\min_{\mathbf{x}_t\in\Omega} E(\mathbf{x}_t)$ does not have an analytic solution, we obtain $\mathbf{x}_t^*$ via exhaustive search in $\mathbf{\Omega}$.[4] To reduce the search range, we perform a coarse-to-fine search in the space $\mathbf{\Omega}$. We construct a subset $\mathbf{\Omega}' \subset \mathbf{\Omega}$ with spacing m pixels, and define $\mathbf{x}_t^{**} = \arg\min_{\mathbf{x}_t\in\Omega'} E_t(\mathbf{x}_t) + E_d(\mathbf{x}_t)$, which is a suboptimal solution. We then start from $\mathbf{x}_t^{**}$ and perform the local search every $m/2$ pixels, and the local optimum is treated as our tracking result. The parameter m depends on the target size, and we set $m = 10$ for the general case.

The search range $\mathbf{\Omega}$ is determined as follows. At time t, we have obtained the matching points b_i^t in the target. Therefore, for any candidate region $\mathbf{x}_t$, $b_i^t \in \mathbf{x}_t$. This gives the inner boundary of the candidate regions. For the outer boundary, we can estimate the rough global motion $(\Delta x_t, \Delta y_t)$ of the target by simply averaging the motion of each salient point. The outer boundary is then the rectangle with the center $(x_{t-1} + \Delta x_t, y_{t-1} + \Delta y_t)$ and the width/height $\omega_{t-1} + (\omega_{t-1} + h_{t-1})/4, h_{t-1} + (\omega_{t-1} + h_{t-1})/4$. The illustration is shown in Fig. 3. In Fig. 3, the candidate bounding box should contain the green box, and should be contained in the blue box.

C. Video-Based Object Recognition

Given the current target state $\mathbf{x}_t$, we compute the target category c_t based on the target measurement $\mathbf{z}_t = \mathbf{I}_t(\mathbf{x}_t)$. This can be formulated as an object recognition problem in a single image. However, recognition in a single image may not achieve good performance. As the decision is only made on one view of the target, recognition can be difficult due to complex situations such as partial occlusion. Therefore, we instead find the optimal sequence $\{c_1, \ldots, c_t\}$ given the measurement

[4] The inference algorithm is usually used in "particle-filter"-style object tracking research. And the gradient-descent method is usually used when the objective function has a nice property (like mean-shift). For other tracking research which does not reside in those two directions, the naive exhaustive search is often applied [2], [3], [10], [12].

$\mathbf{z}_1, \dots, \mathbf{z}_t$, which is indeed the *video-based object recognition*. Denote by $\underline{c}_t = \{c_1, \dots, c_t\}$, $\underline{\mathbf{z}}_t = \{\mathbf{z}_1, \dots, \mathbf{z}_t\}$, we have

$$\begin{aligned}\{c_1^*, \dots, c_t^*\} &= \arg\max\ \ p(\underline{c}_t|\underline{\mathbf{z}}_t)\\ &= \arg\max p(\underline{\mathbf{z}}_t|\underline{c}_t)p(\underline{c}_t)\\ &= \arg\max \prod_{i=1}^{t} p(\mathbf{z}_i|c_i)p(c_i|c_{i-1}). \qquad (7)\end{aligned}$$

The last equation assumes that $\mathbf{z}_i$ are independent of $\mathbf{z}_j$ and c_j $(i \neq j)$, and $\{c_i\}$ is a Markov chain.

Finding the optimal sequence $\{c_i\}$ in Eq. 7 is equivalent to the problem of finding the best hidden state sequence in an HMM. We can employ the Viterbi algorithm [40] (indeed a dynamic programming approach) for the inference. In practice, at time t, we do not have to estimate the sequence $\{c_1, \dots, c_t\}$ starting from the first frame because of the computational complexity. Instead, we construct a time window which only considers the recent T frames, i.e., the sequence $\{c_{t-T+1}, \dots, c_t\}$. The prior term is $p(c_{t-T+1}) = p(c_{t-T+1}|c_{t-T})$ where c_{t-T} is known.

As above, we model the probability $p(c_i|c_{i-1})$ and $p(\mathbf{z}_i|c_i)$ by using the energy terms: the transition cost $V(c_i, c_j) = -\ln p(c_i|c_j)$ which measures how likely the state c_j switches to c_i (it can be manually set as fixed values), and the cost $E_r(\mathbf{z}_i, c_i) = -\ln p(\mathbf{z}_i|c_i) \triangleq U(\mathbf{z}_i, c_i)$, which is related to object recognition in a single frame. We use the same $U(\cdot, \cdot)$ as described in Sec. III-B, This is the Naive-Bayes NN classifier, which overcomes the inferior performance of conventional NN-based image classifiers [41].

Now we give some more details in obtaining $\mathbf{f}_p(\cdot)$ and $\mathbf{f}_e(\cdot)$. In the salient point representation, we extract Harris-corner points with SIFT descriptors, and quantize them using a 300 entry codebook that was created by K-means clustering a random subset of 20,000 descriptors. We use a two-level pyramid to construct the feature $\mathbf{f}_p(\cdot)$.[5] To construct a histogram of edge directions, we use $[-1, 0, 1]$ gradient filter with no smoothing, and nine different directions are extracted. For color images, we compute separate gradients for each color channel, and take the one with the largest norm as the pixel's gradient vector [42]. The edge histogram $\mathbf{f}_e(\cdot)$ is represented using a uniformly weighted spatial pyramid with three levels [38]. KD-tree is used for the efficiency of NN search in order to reduce the computational complexity [41]. We choose $r = 15$ in our experiment.

The training images for object recognition are collected from PASCAL VOC Challenge 2007 data set [43]. We consider some often seen moving objects as object classes. Specifically, we consider six classes: aeroplane (A), boat (B), car (C), people (P), quadruped (Q), and others (O, i.e., C^0). The "quadruped" class includes horse/cat/dog, because the shape of these animals is very similar in many cases. For the "people" class, as the Pascal VOC 2007 data set includes various people postures like sitting, which is not good for recognition of moving persons, we use the training samples from INRIA dataset [31] instead. The class C^0 includes some static object classes: chair/sofa/table/monitor. We also include some natural scene images into this class. The natural scene images are from [44]. In order to avoid wrong recognition result, the object is recognized as class C^0 if $p(\mathbf{z}_i|c_i)$ is low for all $C^i, i = 1, \dots, N$. Then our tracking procedure is simplified as $\mathbf{x}_t^* = \arg\max_{\mathbf{x}_t \in \Omega} p(\mathbf{x}_t|M_{t-1}^L, \mathbf{I}_t)$.

1) Tracking Morphable Objects: The target category c can be extended to describe different status of the object. For example, if we want to track morphable objects, we regard $\{C^1, \dots, C^N\}$ as the different status of the object, and C^0 as the "others" status. Then the parameter c_t describes the object status at time t. By adjusting the transition cost $V(c_i, c_j)$, the proposed recognition scheme can be easily applied to this scenario.

IV. Experiments

In this section, we first go through the technical details. Then we compare the proposed video-based object recognition method with still-image object recognition, followed by the sensitivity analysis of the parameters. The tracking performance in various scenarios is then evaluated.

A. Technical Details

For technical details, the parameters w_s, w_h, w_p, and w_e are chosen to make the energy terms comparable. In practice, the weight of each term is adaptively adjusted by a confidence score, which measures the performance of one term at current frame. This ensures that the inaccuracy of one term will not impact our results, even in some extreme cases (e.g., $w_s = 0.15\sigma_s$ is the product of the normalizing constant 0.15 and the confidence score $\sigma_s \in [0, 1]$. Similarly, $w_h = 2\sigma_h$, $w_p = 0.3\sigma_p$, and $w_e = 2.5\sigma_e$).

σ_s is the (time varying) confidence score corresponding to the energy term w_s, which can be obtained by:

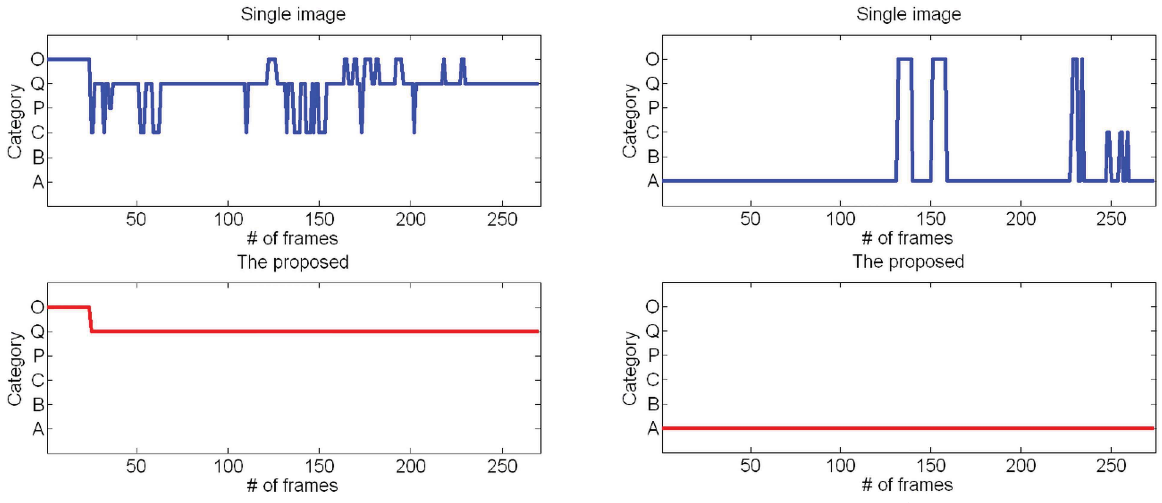
$$\sigma_s = \frac{\sum_{\mathbf{x}_t \in \Omega'} 1(E_s(\mathbf{x}_t) < T_s)}{|\Omega'|} \qquad (8)$$

where $\mathbf{x}_t$ is one candidate region at frame t, Ω' is the search space, $1(\cdot)$ is the indicator function, and T_s is the threshold of E_s. Eq. 8 means that the more candidate regions are larger than the threshold, the less confident the energy term is. Likewise we can define the confidence score for other energy terms.

The thresholds for the offline model (e.g., T_p and T_e) are determined based on the training data. Different categories have different thresholds. For a certain category $\hat{c}$, we compute $T_p(\hat{c})$ as follows (similarly for $T_e(\hat{c})$):

We choose one image from the training data, and collect some image regions close to the target location as the positive samples. For each image region $\mathbf{z}$, we obtain the average distance $d_p = E_p(\mathbf{z}, \hat{c})$ between this region and its nearest neighbors from remaining training samples. Intuitively, d_p is less than $T_p(\hat{c})$, since it is the positive data. For all the training images, and all the positive image regions we collected for each image, we can compute d_p similarly. Therefore, we obtain the distribution of d_p. Similarly, we collect negative samples which are far from the labeled image region in each

[5] We compute the NN-distance on the pyramid SIFT features rather than on local image descriptors in [41], because SIFT feature is more robust to noise, and the pyramid matching can encode some spatial information.



(a) (b)

Fig. 4. Comparison of video-based object recognition and image object recognition. (a) *Dog*. (b) *Plane*. We show these two because the recognition is relatively very challenging.

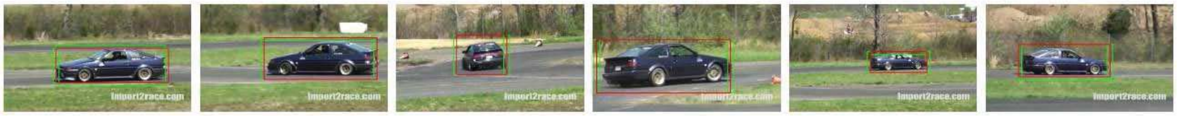


Fig. 5. Tracking a car (frames 16, 34, 75, 109, 145, 247). The red box is the tracking result (the optimum). The green bounding boxes are the target candidates with 2nd–5th best energy scores.

image, and obtain the distribution of d_n for negative samples. So $T_p(\hat{c})$ is essentially the Bayes optimal decision boundary, and it can be easily obtained numerically based on these two distributions. The thresholds for the online model (e.g., T_s and T_h) are determined empirically.

Although we can use same energy terms for all classes, our algorithm is flexible in that the energy term $E_d(\mathbf{x}_t)$ can have different choices for different object classes. We find that "people" is a special object class, because the SVM classifier for object detection usually works well in this class. However it is not always the case for other classes. Therefore, in case of human detection, we change the detection term to the energy term of a linear SVM classifier $E_d(\mathbf{x}_t) = -\ln \frac{\Phi(\mathbf{z}_t)+1}{2}$, where $\Phi(\mathbf{z}_t) \in [-1, 1]$ is the SVM score. We also choose the HOG feature for "people" class. The object detection is essentially a "one-against-others" classifier, where this task is simpler than the object recognition task. So we can simplify the detection term by discarding the energy term E_p from E_d, so as to reduce the computational complexity. In object recognition, E_p is still included. As the pyramid matching of the salient points only needs to compute once (at the tracked bounding box), the complexity is mild.

B. Recognition Evaluation

We show the recognition evaluation in the dog and plane sequences, in which our recognition results are compared with object recognition in a single image, i.e., the label of the object is solely determined by the recognition procedure at the current frame. To make the comparison fair, the same input bounding boxes which are obtained by the proposed tracking method are provided to these two methods. In Fig. 4, we find that the accuracy of our video-based recognition increases from 72% to 91% in the dog sequence, and increases from 90% to 100% in the plane sequence, which demonstrates that video-based recognition outperforms object recognition in a single image. Therefore, consistent tracking improves the recognition performance in our approach.

The proposed method is better than the simple majority voting on image sequences (with 82% accuracy for the dog sequence). The reason is that we are dealing with video-based recognition task, rather than an image-set-based one. Hence, the transition and continuity in the video frames are important clues, which are properly adopted in our method.

C. Sensitivity Analysis

The values of the normalizing constants (w_s, w_h, w_p, and w_e) are determined empirically. We tried different thresholds ($\pm 20\%$) in the experiments, and most results vary in a small range (Table II). The reason is that the object recognition on our selected object classes is very successful in literature, and our online SIFT/histogram matching has superior performance in accurately localizing the target.

The intermediate results are shown in Fig. 5. In each image, the red bounding box is the final result by our method, and the green bounding boxes are the target candidates with 2nd–5th best energy score. From Fig. 5, most regions are overlapped between these bounding boxes, which also means that our model is not sensitive to noises.

D. Tracking Evaluation

We test our approach on many difficult real videos containing various object classes. Most videos are downloaded from

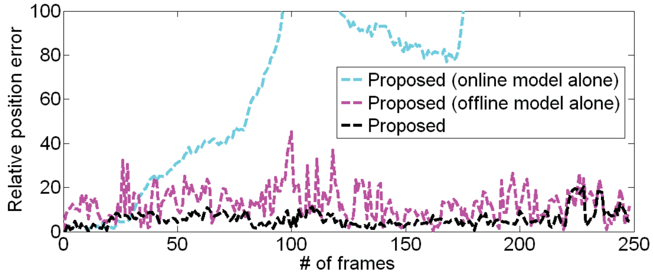


Fig. 6. Quantitative evaluation on individual components of the proposed method. *Car*.

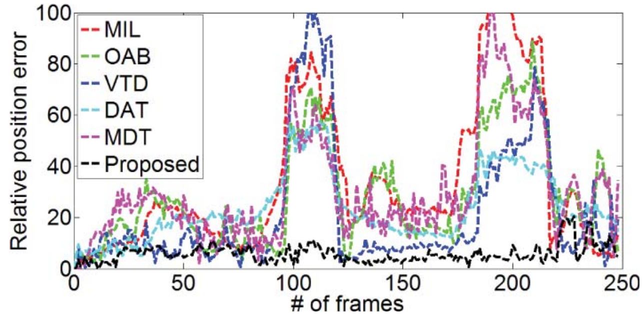


Fig. 7. Quantitative comparison with baseline methods. *Car*.

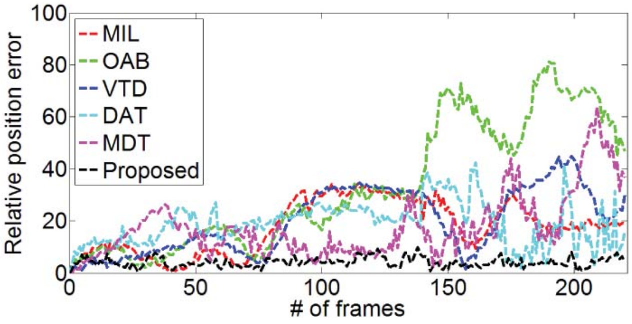


Fig. 8. Quantitative comparison with baseline methods. *Car2*.

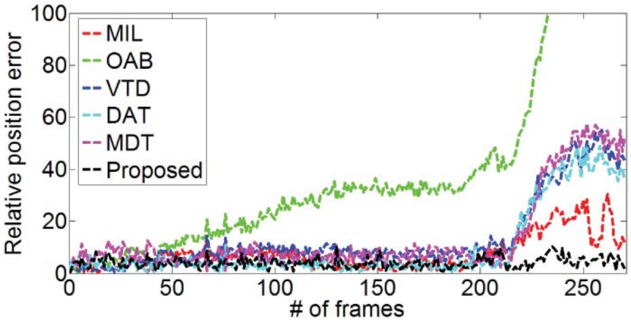


Fig. 9. Quantitative comparison with baseline methods. *Dog*.

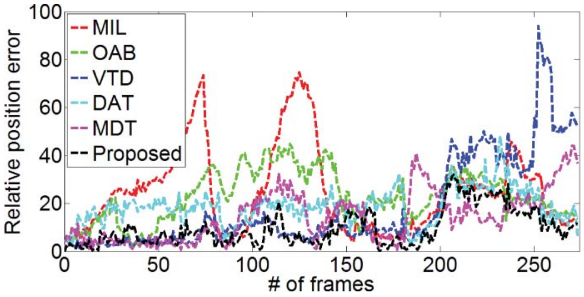


Fig. 10. Quantitative comparison with baseline methods. *Plane*.

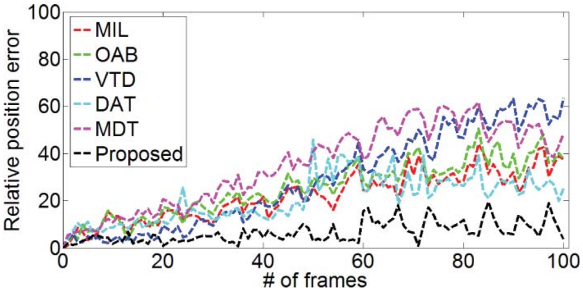


Fig. 11. Quantitative comparison with baseline methods. *People*.

Youtube. The algorithm is implemented in Matlab and runs $2 \sim 0.5$ frames per second on average depending on the object size. We compared the performance using online model alone, offline model alone, and both combined in our algorithm, as shown in Fig. 6. The combination of online and offline models performs better than either model alone. The online model does not handle large view changes, while the offline model does not always achieve the correct localization. However, when combined, these two models compensate for each other.

We also compared our method with five state-of-art online learning trackers, i.e., the multiple instance learning (MIL) tracker [2], the online AdaBoost (OAB) tracker [3], visual tracking decomposition (VTD) tracker [45], discriminative attentional tracker (DAT) [26], and metric differential tracker (MDT) [29]. Since the object class information is *unknown* in the beginning, which makes the offline learning based methods (like people tracking-by-detection [7]) infeasible. For [36], [37], the recognition part is merely object identification which simply matches the target to one image in the database, which is completely different from our method. Therefore, we did not include those methods for comparison. In addition to obtaining the target location, our method also recognizes the target at every frame.

The quantitative comparison in Fig. 7, 8, 9, 10, 11, and Table I (see the columns: MIL, OAB, VTD, DAT, MDT, Proposed) shows our method obtains a higher accuracy. In Fig. 12, a car is moving rapidly with dramatic view changes. The reference methods can hardly track the car because they only rely on low-level features extracted online. Without knowing what the target is, the trackers will drift when the car changes from a side view to the rear view. On the contrary, our method successfully recognizes it as a car during tracking, and is robust to view change since there are many cars of rear view in the training samples. This illustrates that the superiority of our method is actually due to the benefit from the recognition feedback we introduced. Note that we can deal with aspect change, as our parameter space is (x, y, w, h). In Fig. 13, the baseline online trackers can only track the local region of the white dog, as the low-level feature information is not enough to estimate the correct scale in this case. In contrast,

TABLE I

AVERAGE CENTER LOCATION ERRORS (IN PIXELS). SO: "SIMULTANEOUS OPTIMIZATION," AI: "AUTOMATIC INITIALIZATION"

Video Clip	MIL	OAB	VTD	DAT	MDT	Proposed	Proposed (SO)	Proposed (AI)
Car	33.8	28.6	21.4	23.7	30.2	5.8	6.2	15.9
Car2	17.6	33.0	20.0	18.9	17.6	4.2	4.7	13.4
Dog	7.6	46.1	13.5	10.0	14.0	4.0	5.3	8.1
Plane	25.0	22.9	17.9	18.6	14.5	9.6	10.5	13.9
People	22.0	25.4	28.5	20.8	35.1	6.6	6.4	11.7

TABLE II

AVERAGE CENTER LOCATION ERRORS (IN PIXELS) FOR PARAMETER SENSITIVITY ANALYSIS. FOR EACH COLUMN, WE CHANGE THE VALUE OF ONLY ONE PARAMETER, AND FIX THE OTHERS

Video Clip	Proposed	$w_s(-20\%)$	$w_s(+20\%)$	$w_h(-20\%)$	$w_h(+20\%)$	$w_p(-20\%)$	$w_p(+20\%)$	$w_e(-20\%)$	$w_e(+20\%)$
Car	5.8	7.1	6.6	6.2	5.5	6.4	7.2	7.0	5.6
Car2	4.2	5.6	4.7	4.0	5.3	4.5	5.5	5.7	6.2
Plane	9.6	10.4	9.2	10.1	11.4	8.8	12.9	10.1	12.5

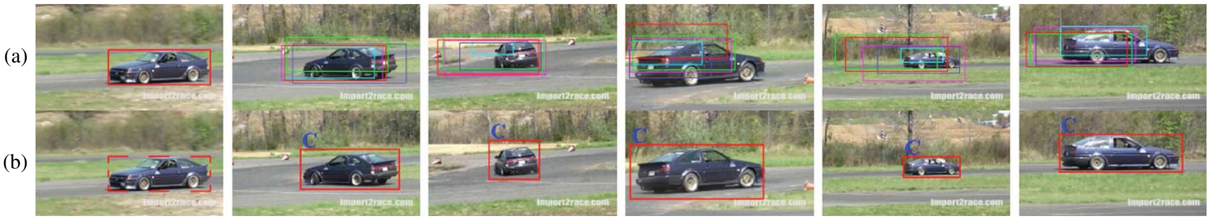


Fig. 12. Tracking a car (frames 0, 51, 75, 109, 145, 210). (a) Red/green/purple/cyan/magenta bounding boxes are generated from the MIL/OAB/VTD/DAT/MDT trackers, respectively. (b) On the top of the bounding box, we show the recognition result ("C" stands for *Car*). The dashed bounding box means that we do not know the target category at the beginning.

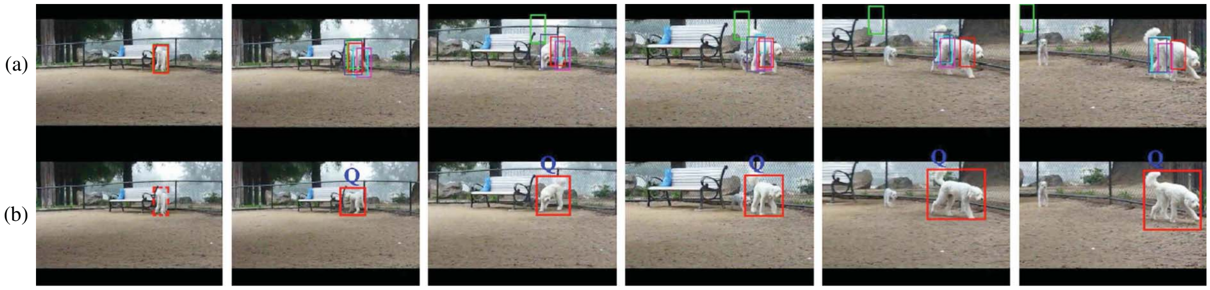


Fig. 13. Tracking a dog (frames 0, 45, 205, 223, 245, 262). (a) Red/green/purple/cyan/magenta bounding boxes are generated from the MIL/OAB/VTD/DAT/MDT trackers, respectively. (b) On the top of the bounding box, we show the recognition result ("Q" stands for *Quadruped*). The dashed bounding box means that we do not know the target category at the beginning.

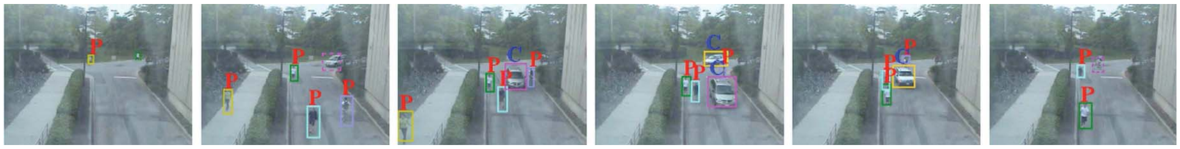


Fig. 14. Multiple target tracking and recognition (frame 71, 293, 336, 364, 403, 455). We assign an object id to one target (shown in different colors). The recognition result is given when the target is successfully recognized.

our method can well handle scale change by recognizing the target as a quadruped.

Figure 14 is a video surveillance scenario. After background subtraction, every new moving blob is tracked and recognized. The recognition result activates the offline model to help tracking. As the existing target would also generate blobs in background subtraction, we only detect new moving objects that are far from the existing target in order to avoid such redundancy. As observed in the experiment, our tracker can successfully track and recognize people and cars (except in the rightmost picture, a man is bending his body and riding a bicycle, where the contour is different from the moving person in the training set). This result can be further used for high-level tasks like abnormal video event detection, etc.

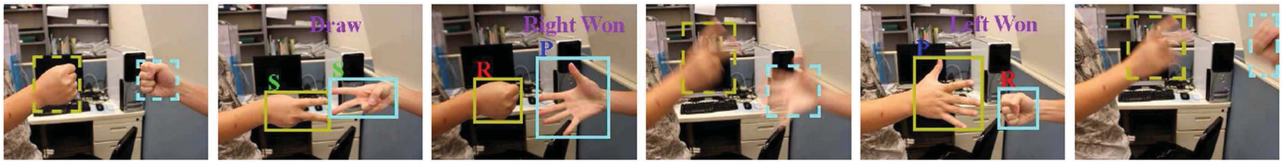


Fig. 15. "Rock-paper-scissors" game (frames 0, 73, 120, 160, 238, 252). The proposed method automatically tracks the hands, recognizes the hand gesture, and decides the winner of the game. "R," "P," and "S" stand for "rock," "paper," and "scissor," respectively.

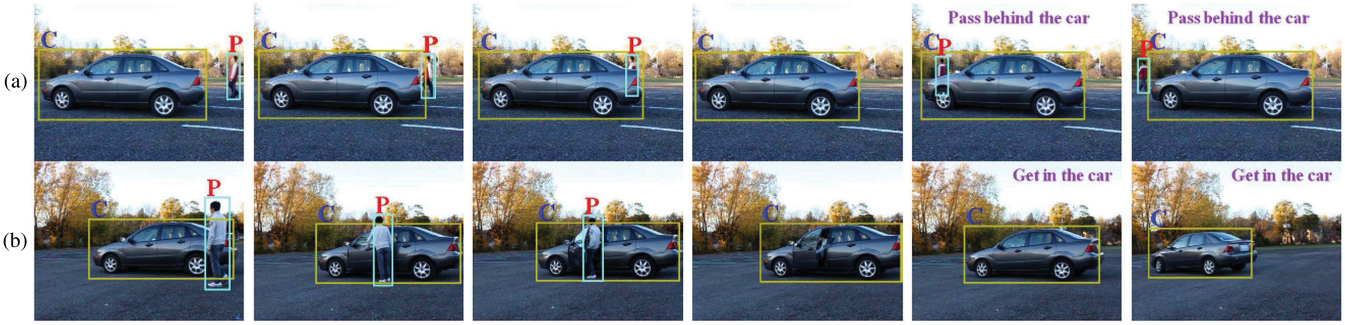


Fig. 16. Example of scene understanding. The system detects cars and pedestrians in the video. (a) (Frames 0, 25, 41, 130, 208, 254), the system understands the human "passes behind the car," while (b) (frames 0, 76, 112, 142, 305, 388), the system understands the human "gets in the car." "P" and "C" stand for *People* and *Car*, respectively.

Figure 15 is about tracking a morphable object. Two person play a "rock-paper-scissors" game, where they present their hand gestures ("rock", "paper", "scissor") at the same time, and judge who wins the game. Our objective is to automatically track the morphable hands and decide the winner. This is generally a very challenging problem, as the hand moves abruptly and quickly, and there is a strong motion blur when the hand moves. To handle this situation we train a simple skin color model (a multi-modal Gaussian model) to detect the human skin region for each frame. For recognition, we collect image exemplars of these three gestures as training samples. For each gesture, we have captured ten different views, and then rotate the images under different rotation angles to obtain more training images. We do not have training samples in the "others" class, but set a threshold on the cost $U(\mathbf{z}, c)$ for each gesture. The hand gesture whose cost is higher than all the thresholds in the existing classes is treated as the "others" class. As the hand gesture switches between different states very quickly, we lower the transition score between states. The excellent results are shown in Fig. 15.

Tracking by recognition has important applications in video understanding. Figure 16 is a simple application of understanding the activities in the video. In the first frame, we run the car detector and the pedestrian detector (the conventional sliding window approach based on our energy term) to detect cars and pedestrians. The detected objects are then tracked. When the human disappears in the video, we maintain the human model, and apply the human detector near the border of the car. If the human reappears (top row), the human detector and the maintained target model can relocate the human, so the system can understand the scenario as "the human passes behind the car". If the human does not appear and meanwhile the car starts moving, the system can understand it as "the human gets in the car".

E. About the EM-Like Solution

Recall that we employ the two-step EM-like method to estimate $\mathbf{x}_t$ and c_t. To evaluate its effect on tracking accuracy, we compare this scheme to the simultaneous optimization of $\mathbf{x}_t$ and c_t.

It is not obvious that how to set up the simultaneous optimization baseline. One possible formulation is:

$$\begin{aligned}&\{\mathbf{x}_t^*, c_t^*\}\\ &= \arg \min_{\mathbf{x}_t \in \Omega, c_t \in C^{0:N}} E_t(\mathbf{x}_t) + E_d(\mathbf{x}_t, c_t) - \lambda \ln p(\underline{c}_t | \underline{\mathbf{z}}_t)\\ &= \arg \min_{\mathbf{x}_t \in \Omega, c_t \in C^{0:N}} E_t(\mathbf{x}_t) + (1+\lambda) E_d(\mathbf{x}_t, c_t) + \lambda V(c_t, c_{t-1}).\end{aligned} \tag{9}$$

Note that c_t is a discrete variable, and its range $C^{0:N} \triangleq \{C^0, C^1, \ldots, C^N\}$ is not an *ordered* set.

The result of this formulation is shown in Table I (with $\lambda = 0.1$, see column Proposed (SO)). It illustrates the EM-like solution and the simultaneous optimization are comparable, and the latter is not guaranteed to be better. The reasons are: (1) the simultaneous solution achieves the minimum of RHS in Eq. 9, but it may not be the real target location and (2) conventionally, the reason the EM-like solution is inferior to the simultaneous optimized solution is that none of the variables is estimated correctly. It often occurs when these variables are continuous variables. However, our c_t is the discrete variable in a non-ordered set, and we have successfully obtained the correct object category in the EM-like solution. With these prerequisites, these two formulations are essentially the same, as c_t becomes a constant.

F. About Automatic Initialization

This work is about generic object tracking, where the object category is unknown at the beginning. While most generic

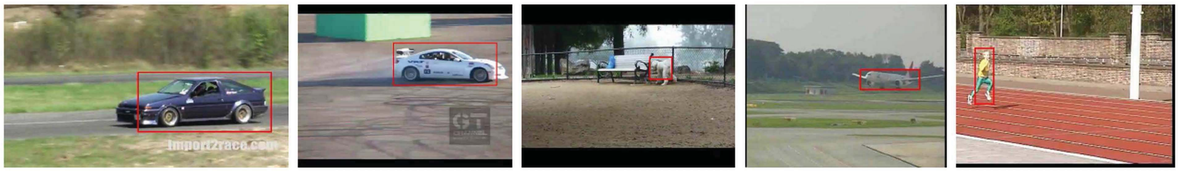


Fig. 17. Automatic initialization for testing videos, *Car* (0), *Car2* (0), *Dog* (45), *Plane* (0), and *People* (0).

object tracking methods use manual initialization, our method can automatically initialize the object by detection in the first frame. we tested the automatic initialization performance of our method. In the first frame, the target location is initialized via object detection (our five categories).[6] The results are shown in Fig. 17 and Table I (see column Proposed (AI)). We observe that the result is worse than the manual initialization, as the initialization error is induced through object detection. But the automatic initialization still performs better than baseline methods in most cases. This further verifies that the feedback of recognition module helps our object tracking.

G. About the Number of Object Categories

Generally speaking, when the number of object categories increases, the ambiguity of the tracking problem also increases. Here, we investigate the performance of the proposed algorithm versus different number of object categories.

In the first experiment, we obtain the recognition accuracy (the same way as Sec. IV.B) when the number of categories varies, where all the object categories are included in the extreme case. The result for dog sequence is shown in Fig. 18. Denote all the categories of PASCAL VOC 2007 by $C^{1:N} =$ {dog, plane, bicycle, bird, boat, bottle, bus, car, cat, cow, horse, motorbike, person, potted plant, sheep, train}, where $N = 16$. We also have the same C^0 as explained before (including four classes in PASCAL VOC 2007). Each time we choose C^0 and a subset of N_1 categories $C^{1:N_1}$, and evaluate the recognition accuracy. From Fig. 18, it is clear that the accuracy decreases as N_1 increases. We observe that there is an obvious drop in recognition accuracy when we add "cat" into the object category list, this is because "dog" and "cat" are sometimes difficult to distinguish using our classifier.

We further explore how the increased object categories affects the tracking performance. We consider the extreme case, i.e., all the object categories in PASCAL VOC, and perform tracking on this video sequence. In this experiment, the dog is wrongly recognized by "cat", and the tracking results are shown in Fig 19. We find that the tracker still keeps track on the target, though the result is worse than our previous result when $N = 5$. This is because the visual appearances of "cat" and "dog" look similar (they are both quadruped), and thus the detection result is more or less on the target.

We also perform similar experiments on the other four video sequences, and find that there is no additional ambiguity introduced. The reason is that the targets (car, plane, people) in those videos are relatively different from other categories, and thus rarely misclassified.

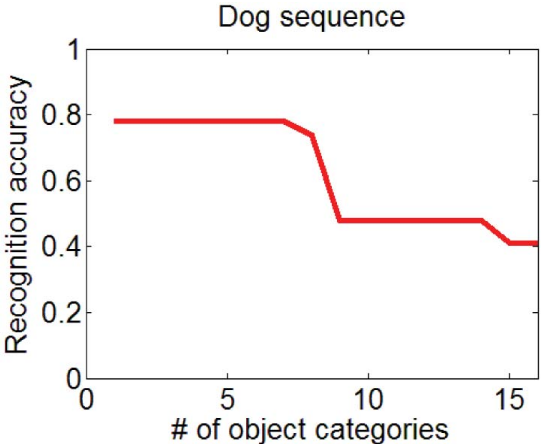


Fig. 18. Recognition accuracy with different number of object categories. *Dog*. The number N_1 of object categories means that our classifier contains the classes $C^{1:N_1}$ other than C^0.

H. Discussion

In this subsection, we discuss several issues about the proposed method.

1) Number of Object Categories: Currently, we consider five categories (people, car, aeroplane, boat, and quadruped) as they are the most frequently tracked objects in tracking applications. But our method does not restrict the number of object classes, and it can be generalized to more classes. Meanwhile, though our video-based recognition method can improve the recognition accuracies, if the recognition approaches incorporated in our framework have unsatisfactory performance, they will largely corrupt our tracking as well as the recognition results. Unfortunately, the state-of-the-art object recognition techniques (see PASCAL VOC Challenge results for example) are not good enough to recognize all object classes with high accuracy. With the development of object recognition research, if more object categories can be reliably recognized, they will be naturally included in our framework.

2) Closed-Loop Framework Design: In our closed-loop framework, wrong recognition probably leads to error propagation. Here, we discuss the robustness of the proposed method: (1) The object categories selected in our method generally yield good results in state-of-the-art object recognition literature. In addition, the inter-class distance is sufficiently large among these classes. So the single image recognition is usually reliable in our experiments. Furthermore, we employ the video-based object recognition to further improve the

[6] If the object is not detected in the first image, we perform object detection in the next frames until the target is initialized.

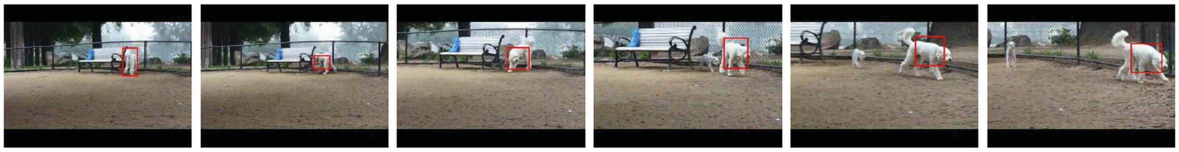


Fig. 19. Tracking a *Dog* (frames 0, 45, 205, 223, 245, 262) using 16 categories. The tracking ambiguity increases and the dog is wrongly classified as *Cat*.

recognition accuracy (Fig. 4). (2) The tracking result is determined by both online target model and offline model. So the online target model may adjust the target location in some cases. (3) In our work, we have designed the "others" class. The purpose of introducing this "others" class is that we want to combine online and offline model only if the object is recognized with a high confidence. If the classifier is not confident enough to recognize the object, we treat it as the "others" class. Then our tracker becomes the regular online tracking (offline model is disabled), just like what we do in the first few frames when the object category is unknown.

3) Dataset: Our current design is not appropriate for some tracking dataset such as MIL/VTD. The major reason is that: for the testing data, there is a gap between object tracking literature (MIL/VTD dataset) and object recognition literature (PASCAL VOC series). As we know, for all the machine learning techniques applied in computer vision problems, the training data should have some consistency with the testing data. Therefore, the data inconsistency of MIL/VTD dataset and PASCAL VOC series causes this problem here. We further explain it as follows:

We think that the tracking term E_t and the detection term E_d should be consistent to each other, since they are both about matching the target representation to the candidate model (online vs. offline model). In the state-of-the-art object recognition literature, the researchers all use "salient point" (bag-of-words) representation (or some variants). Inspired by those works, we use "salient point" representation in E_d and also E_t.

Unfortunately, this "salient point" representation in E_t is not appropriate for MIL/VTD dataset, due to data inconsistency. When we look at the testing data in PASCAL VOC series, the objects are all of large-size and in a relatively fine resolution. On the contrary, in the MIL/VTD dataset from object tracking literature, the objects are of small-size and not in a fine resolution. So the "salient point" representation has problems in those dataset. Therefore, we create our own dataset for testing, where the data is consistent to PASCAL VOC series. Our dataset is also very challenging in that the baseline methods have large tracking errors. Moreover, most videos in our dataset contain the cases where the view change of the target switches to the unprecedented status. This case cannot be essentially handled by current online tracking methods, as the appearance they learned is based on the previous status.

To make the object recognition techniques work for MIL/VTD dataset, we have two comments: (1) Solving the root of the problem depends on the progress of the object recognition algorithms. (2) One artificial solution may be that we adapt E_t to those dataset, which would break the consistency between E_t and E_d.

4) Limitation: This paper is only the start of an important conversation about introducing high-level semantic information to generic object tracking. Through the above discussions, we summarize the limitations in our method: (1) The ambiguity of the tracking problem increases, as the number of object categories increases. (2) The wrong recognition result probably leads to error propagation. (3) The current design may not be appropriate for some tracking dataset, due to data type inconsistency. These limitations may be resolved via the progress on robust detection, or the progress on large scale robust object recognition.

V. Conclusion

As a mid-level task, visual tracking plays an important role for high-level semantic understanding or video analysis. Meanwhile the high-level understanding (e.g., object recognition) should feed back some guidance for low-level tracking. Motivated by this, we propose a unified approach to object tracking and recognition. In our framework, once the objects are discovered and tracked, the tracking result is fed forward to the object recognition module. The recognition result is fed back to activate the off-line model to and help improve tracking. Extensive experiments demonstrate the efficiency of the proposed method.

References

[1] S. Avidan, "Ensemble tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 2, pp. 261–271, Feb. 2007.

[2] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 983–990.

[3] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. British Mach. Vis. Conf.*, 2006, pp. 1–10.

[4] Y. Li, H. Ai, T. Yamashita, S. Lao, and M. Kawade, "Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1728–1740, Oct. 2008.

[5] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors," *Int. J. Comput. Vis.*, vol. 75, no. 2, pp. 247–266, Nov. 2007.

[6] C. Huang, B. Wu, and R. Nevatia, "Robust object tracking by hierarchical association of detection responses," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 788–801.

[7] B. Leibe, K. Schindler, N. Cornelis, and L. V. Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 10, pp. 1683–1698, Oct. 2008.

[8] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, pp. pp. 1–13, 2006.

[9] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Proc. Eur. Conf. Comput. Vis.*, 1996, pp. 343–356.

[10] Z. Kalal, J. Matas, and K. Mikolajczyk, "P-N learning: Bootstrapping binary classifiers by structural constraints," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 49–56.

[11] A. Srivastava and E. Klassen, "Bayesian and geometric subspace tracking," *Adv. Appl. Probab.*, vol. 36, pp. 43–56, Dec. 2004.

[12] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 728–735.

[13] A. Tyagi and J. W. Davis, "A recursive filter for linear systems on Riemannian manifolds," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[14] J. Kwon, K. M. Lee, and F. C. Park, "Visual tracking via geometric particle filtering on the affine group with optimal importance functions," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 991–998.

[15] X. Mei and H. Ling, "Robust visual tracking using $\ell 1$ minimization," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 1436–1443.

[16] R. Li and R. Chellappa, "Aligning spatio-temporal signals on a special manifold," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 547–560.

[17] C. Bibby and I. Reid, "Real-time tracking of multiple occluding objects using level sets," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1307–1314.

[18] N. Alt, S. Hinterstoisser, and N. Navab, "Rapid selection of reliable templates for visual tracking," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1355–1362.

[19] J. Fan, "Toward robust visual tracking: Creating reliable observations from videos," Ph.D. thesis, Dept. Comput. Eng., Northwestern Univ., Evanston, IL, 2011.

[20] X. Mei, H. Ling, and Y. Wu, "Minimum error bounded efficient $\ell 1$ tracker with occlusion detection," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1257–1264.

[21] X. Li, A. Dick, H. Wang, C. Shen, and A. Van den Hengel, "Graph mode-based contextual kernels for robust SVM tracking," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 1156–1163.

[22] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2003.

[23] G. Hager, M. Dewan, and C. Stewart, "Multiple kernel tracking with SSD," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. 790–797.

[24] A. D. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1296–1311, Oct. 2003.

[25] J. Fan, X. Shen, and Y. Wu, "Scribble tracker: A matting-based approach for robust tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1633–1644, Aug. 2012.

[26] J. Fan, Y. Wu, and S. Dai, "Discriminative spatial attention for robust tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 480–493.

[27] R. Li, R. Chellappa, and S. K. Zhou, "Learning multi-modal densities on discriminative temporal interaction manifold for group activity recognition," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 2450–2457.

[28] M. Yang, Y. Wu, and G. Hua, "Context-aware visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 7, pp. 1195–1209, Jul. 2009.

[29] N. Jiang, W. Liu, and Y. Wu, "Adaptive and discriminative metric differential tracking," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1161–1168.

[30] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[31] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 1–8.

[32] J. Gall, N. Razavi, and L. V. Gool, "On-line adaption of class-specific codebooks for instance tracking," in *Proc. 21st British Mach. Vis. Conf.*, 2010, pp. 1–12.

[33] S. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1434–1456, Nov. 2004.

[34] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Visual tracking and recognition using probabilistic appearance manifolds," *Comput. Vis. Image Understand.*, vol. 99, no. 3, pp. 303–331, 2005.

[35] F. Bardet, T. Chateau, and D. Ramadasan, "Illumination aware MCMC particle filter for long-term outdoor multi-object simultaneous tracking and classification," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 1623–1630.

[36] D.-N. Ta, W.-C. Chen, N. Gelfand, and K. Pulli, "Surftrac: Efficient tracking and continuous object recognition using local feature descriptors," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 1–8.

[37] G. Takacs, V. Chandrasekhar, S. Tsai, D. Chen, R. Grzeszczuk, and B. Girod, "Unified real-time tracking and recognition with rotation-invariant fast features," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 934–941.

[38] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 2169–2178.

[39] O. Chum and A. Zisserman, "An exemplar model for learning object classes," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[40] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

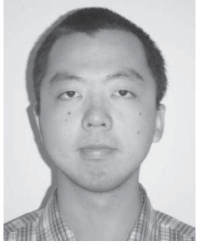
[41] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[42] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.

[43] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (2007). *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results* [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

[44] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 524–531.

[45] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1–8.

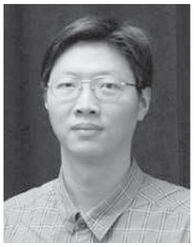


Jialue Fan received the B.E. and M.S. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2005 and 2007, respectively, and the Ph.D. degree in electrical and computer engineering from Northwestern University, Evanston, IL, in 2011.

He was a Summer Intern with NEC Labs America, Cupertino, CA, Siemens Corporate Research, Princeton, NJ, and Adobe Systems Inc., San Jose, CA, in 2008, 2009, and 2010, respectively. His current research interests include computer vision, and image and video processing.

Xiaohui Shen (S'11) received the B.S. and M.S. degrees in automation from Tsinghua University, Beijing, China, in 2005 and 2008 respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL.

He was a Summer Intern with Nokia Research Center, Santa Monica, CA, in 2010, and Adobe Systems Inc., San Jose, CA, in 2011 and 2012, respectively. His current research interests include image and video processing, and computer vision.



Ying Wu (SM'06) received the BS degree in automation from Huazhong University of Science and Technology, Wuhan, China, the MS degree in automation from Tsinghua University, Beijing, China, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Urbana, in 1994, 1997, and 2001, respectively.

He joined the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, as an Assistant Professor, in 2001, where he is currently an Associate Professor of electrical engineering and computer science. His current research interests include computer vision, image and video analysis, pattern recognition, machine learning, multimedia data mining, and human-computer interaction.

Dr. Wu was the recipient of the Robert T. Chien Award at UIUC in 2001 and the NSF CAREER Award in 2003. He is an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the *SPIE Journal of Electronic Imaging*, and the *IAPR Journal of Machine Vision and Applications.*