Adaptive Object Tracking by Learning Hybrid Template Online

Xiaobai Liu, Liang Lin, Shuicheng Yan, *Senior Member, IEEE,* Hai Jin, *Senior Member, IEEE,* and Wenbin Jiang

***Abstract*—This paper presents an adaptive tracking algorithm by learning hybrid object templates online in video. The templates consist of multiple types of features, each of which describes one specific appearance structure, such as flatness, texture, or edge/corner. Our proposed solution consists of three aspects. First, in order to make the features of different types comparable with each other, a unified statistical measure is defined to select the most informative features to construct the hybrid template. Second, we propose a simple yet powerful generative model for representing objects. This model is characterized by its simplicity since it could be efficiently learnt from the currently observed frames. Last, we present an iterative procedure to learn the object template from the currently observed frames, and to locate every feature of the object template within the observed frames. The former step is referred to as *feature pursuit*, and the latter step is referred to as *feature alignment*, both of which are performed over a batch of observations. We fuse the results of feature alignment to locate objects within frames. The proposed solution to object tracking is in essence robust against various challenges, including background clutters, low-resolution, scale changes, and severe occlusions. Extensive experiments are conducted over several publicly available databases and the results with comparisons show that our tracking algorithm clearly outperforms the state-of-the-art methods.**

***Index Terms*—Adaptive tracking, hybrid template, matching pursuit.**

I. Introduction

THIS PAPER presents an adaptive tracking algorithm which learns hybrid templates for objects in video on the fly. Herein, the input video is captured by a fixed camera. The template consists of multiple types of features, including sketchs/edges, texture regions, and flatness regions. Sketch/edge regions usually consist of various links, ridges, or their compositions, such as corners and junctions. Texture regions are likely to contain a large number of objects that are either too small or too distant relative to the camera. In contrast, flatness regions are always filled with homogeneous color or intensity. For the ease of descriptions, we call these feature types as *sketch*, *texture*, and *flatness*, respectively. In the past literature [27], the sketch features are well known as the good features for tracking moving objects. In this paper, nevertheless, we argue that both texture and flatness features also contain discriminative information and are thus expected to have substantial contribution in object tracking, especially for the complex scenes with various challenges, such as background clutters, large illumination changes, and severe occlusions. Based on this argument, we propose to jointly track various types of features extracted from the foreground regions and fuse the results to locate the objects of interest within frames.

A. Related Works

There exist wide varieties of research on feature selection or feature combination in the visual tracking community [27], [28], and it is well known that employing multiple diverse features may lead to improved tracking performance [27]. For example, Stern and Efros [22] proposed to select color spaces adaptively by a weighted probability measure. Collins *et al.* [3] define a variance ratio of the foreground regions against the local background regions. Grabner *et al.* [12], [13] apply the online boosting techniques for realtime tracking to adaptively choose the best features according to the fitness with the strong classifier. Yang *et al.* [29] adopt the so-called visual attention map to discover the most salient patches to track. Although encouraging performance has been achieved, most of these efforts utilize one single feature type, e.g., color or sketch/corner, which may fail to work while the specific feature type has less discriminative power. For example, the object of interest that has similar sketch structure as the local background regions may challenge the corner-based tracking algorithms [27].

Therefore, it is better to employ multiple diverse types of features, rather than using one single type, and adaptively choose the most informative features while tracking the moving objects. There are also previous efforts [1], [14], [17],

Manuscript received July 12, 2009; revised March 27, 2010 and August 31, 2010; accepted November 16, 2010. Date of publication March 17, 2011; date of current version November 2, 2011. This work was performed when X. Liu was visiting the Lotus Hill Institute for Computer Vision and Information Science, Hubei, China. This work was supported by the CSIDM Project CSIDM-200803. This work was also supported in part by the National High Technology Research and Development Program of China, under Grant 2006AA01A115, the China National Natural Science Foundation, under Grants 60903173 and 60970156, the Fundamental Research Funds for the Central Universities, under Grant 2010620003162041, and the SYSU-Sugon High Performance Computing Typical Application Projects, under Grant 62000-1132001. This paper was recommended by Associate Editor D. Schonfeld.

X. Liu, H. Jin, and W. Jiang are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China.

L. Lin is with the Software School, Sun Yat-Sen University, Guangzhou 510275, China. (Corresponding author,e-mail: linlng@mail.sysu.edu.cn).

S. Yan is with the Department of Electrical and Computer Engineering, National University of Singapore, Kent Ridge 119077, Singapore.

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/TCSVT.2011.2129410

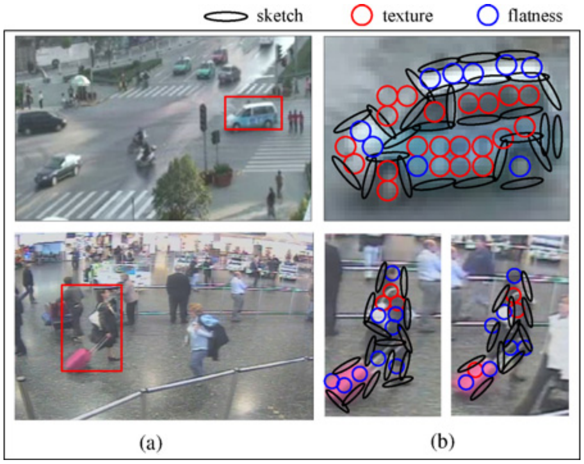


Fig. 1. Hybrid object template consists of multiple types of features, including sketch, texture, and flatness. (a) Objects within images. (b) Hybrid templates learned from the foreground regions. Each feature characterizes one localized image patch.

[28] which focused on how to combine multiple visual cues to achieve robust object tracking. Hua *et al.* [10] proposed to study the problem of inconsistency with respect to the measurement for fusing visual cues. One recently attempt by Birchfield *et al.* [2] aimed to explore the image gradient information to enhance the performance of object tracking. These approaches are likely to first predict a single target model at every tracking step and then fuse the learnt model to estimate the target state for the newly observed frames. However, fusing various cues is not trivial due to the uncertainties and high-dimensional target distribution [1]. Therefore, in this paper, our goal is not to approximate a fused distribution. Instead, we argue that different types of features should have equally important roles in tracking and propose to represent objects of interest using the hybrid templates. Jointly and adaptively tracking diverse types of features, especially the texture features and the flatness features, distinguishes our approach from the traditional corner-based algorithms [2], [18], [21] which usually ignore the homogeneous features. As Fig. 1 illustrates, the flatness region in the woman's white bag (in the bottom row of the figure) is notably distinctive against the local surroundings and thus should also be considered as a "good" feature to track.

B. Method Overview

We represent objects using the so-called *hybrid templates*, which consist of three different types of features, namely, sketch, texture, and flatness. For one given frame, we partition it into a set of patches of equal size and describe each patch as one certain feature. Features belonging to different types usually bear different appearance distributions and thus are likely to be complementary with each other while describing the same object. In addition, different features may have different discriminative power along with the changes of the local surrounding background, and thus they should have different confidences in the final decision of object tracking. Once a hybrid template is constructed, each feature within the template can be tracked by matching it into the incoming frames, and jointly tracking all the features is able to exploit the inter-patch geometry structure information, which usually leads to an enhanced tracking performance. As the discriminative power of features change along with the object movements, the hybrid template should be adaptively updated by either adjusting the feature confidences, or substituting the old features with the newly discovered ones from the currently observed frames.

Based on the philosophy of hybrid feature tracking, we propose an adaptive tracking algorithm. Fig. 2 summarizes the diagram of our algorithm. The basic idea is to track objects by learning the hybrid templates and updating the learnt templates adaptively. More specially, given a set of input frames each containing the object of interest, our goals are twofold. One is to learn the hybrid template by extracting the most discriminative features from the foreground regions. The other one is to locate each feature of the hybrid template in the observed frames. We call the above two procedures as *feature pursuit* and *feature alignment*, respectively. These two steps are mutually supportive. On the one hand, accurately locating the object of interest within each observed frame is able to improve the quality of template learning. On the other hand, a high quality object template is very likely to improve the accuracy of feature alignment. Therefore, in this paper, we propose to alternately perform these two procedures for object tracking.

Moreover, we propose a simple yet efficient generative model to guide the procedure of feature pursuit. The model is formulated as maximizing the likelihood ratio of the feature distributions over the foreground regions against the feature distributions over the local background regions. Taking the likelihood ratio as a general metric, different types of features are made comparable with each other and are able to compete to explain the same foreground region. This generative model is characterized by its simplicity as it could be efficiently learnt from the currently observed frames.

The remainder of this paper is organized as follows. We first introduce the object representation of hybrid template in Section II and then develop an adaptive tracking algorithm in Section III. Extensive experiments with comparisons are reported in Section IV. Last, we conclude this paper and discuss the future works in Section V.

II. Hybrid Template Representation

A. Background and Motivation

Generally, objects or scenes can appear at a wide range of distances or scales in the view of camera, and the same structure at different scales may produce images with different statistical properties. Wu *et al.* [26] proposed an information scaling theory, which showed that the entropy rate of the image data and the perceptual uncertainty usually change along with the viewing distance, as well as the camera resolution. Based on this theory, they proposed the primal sketch model [9] which integrates the sparse coding formulation and the Markov random field theory, and further extended the model to propose a novel object representation, i.e., the active basis model [25]. In the above methods, the local image patterns are categorized

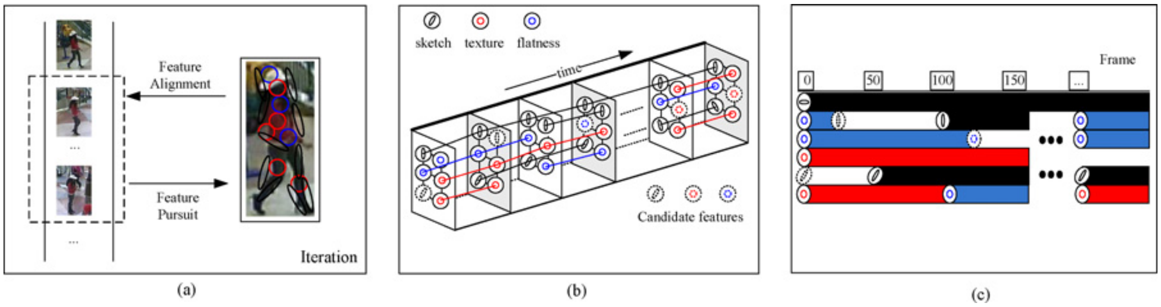


Fig. 2. Diagram of the proposed algorithm. (a) Tracking by learning. Given a set of input frames, two iterative procedures, namely, feature pursuit and feature alignment, are alternately performed to learn the object hybrid templates and match the features of the learnt template into each frame. (b) Hybrid templates are updated by substituting the old less discriminative features with the new more discriminative ones from the current observations. To allow a gradual update of the hybrid template, a set of candidate features are also kept. (c) Features of the hybrid template usually span different numbers of frames.

into the sketch-able regions and the non-sketch-able regions by a sketch-ability criterion. Based on the same methodology of categorization, we further divide the non-sketch-able regions into two types, namely, the texture regions and the flatness regions, and use all the three types of features to build the hybrid templates while representing objects. Different types of features usually capture different information of appearance and are expected to be complementary with each other while describing the same object.

In this paper, we first define a probabilistic metric to choose the discriminative or confident features, and then use the selected features to construct the hybrid template. Each feature within the template is matched into the current frame and all the matching results are fused to estimate the object location as well as object scale. It is predicted that, jointly tracking different features of the template can exploit the inter-feature geometry structure, which may lead to an enhanced performance for visual tracking.

B. Object Model

Let $\{I^1, \cdots, I^T\}$ denote the consecutive T frames containing the object of interest and $\mathbf{X}^t$ denote the object bounding box to predict at the tth frame. We assume that each frame contains one single object and shall discuss the case of multiple objects tracking in Section III. Given $\mathbf{X}^t$ and I^t, we can crop the images for the foreground regions and denote the cropping process as $J^t = I^t[\mathbf{X}^t]$. Thus, our goals are twofold. One is to calculate the object locations $\{\mathbf{X}^t\}_{t=1}^T$ and the other one is to further learn a hybrid template $\mathbf{B}$ from the cropped images $\{J^t\}_{t=1}^T$, which consists of n ($n = 10 \sim 100$) features as follows:

$$\mathbf{B} = (B_1, \cdots, B_i, \cdots, B_n) \quad (1)$$

$$B_i = (l_i, \vec{x}_i, s_i) \quad (2)$$

where B_i denotes the ith feature of the template $\mathbf{B}$, l_i denotes the feature type, vector $\vec{x}_i$ denotes the feature location, and s_i denotes the scale factor. In this paper, the feature scale is fixed to be a constant, which is chosen such that every feature is large enough to be tracked.

The hybrid template consists of a set of features, each characterizing one local image patch. These features can be learnt from the observed frames and are allowed to slightly perturb their locations before they are linearly combined to generate the observed frames. We illustrate the basic idea in Fig. 1, where the right column shows the learnt hybrid templates. The sketch features are illustrated by the ellipsoids at certain positions and with certain orientations, while the texture features and flatness features are shown by circles in red and blue colors, respectively. Formally, letting $B_i^t = (l_i, \vec{x}_i^t, s_i)$ denote the perturbed version of B_i at the frame t, we have $\vec{x}_i^t = \vec{x}_i \pm \vec{\delta}_i^t$, where the vector $\vec{\delta}_{t,i}$ represents the shift in location at the horizontal and vertical directions. These activities are used to account for the local shape deformations. Thus, the deformed template $\mathbf{B}^t$ contains additional information about the translation of each feature, denoted as follows:

$$\mathbf{B}^t = (B_1^t, \cdots, B_i^t, \cdots, B_n^t). \quad (3)$$

We restrict the displacement in location $\vec{\delta}_i^t$ to be smaller than a small pre-specified value a.

C. Probabilistic Formulation of Modeling Objects

Suppose $\{J^t\}_{t=1}^T$ are cropped from the frame sequence, we can formulate the goal of hybrid template learning as maximizing a posterior probability, denoted as $p(J^t, \mathbf{B}^t|\mathbf{B}, \Theta)$, where Θ denotes the parameters set. Furthermore, we have

$$p(J^t, \mathbf{B}^t|\mathbf{B}; \Theta) \quad \propto \quad p(\mathbf{B}^t|\mathbf{B})p(J^t|\mathbf{B}, \mathbf{B}^t) \quad (4)$$

where $p(J^t|\mathbf{B}, \mathbf{B}^t)$ indicates the likelihood term and $p(\mathbf{B}^t|\mathbf{B})$ indicates the prior term.

Since the deformation from $\mathbf{B}$ to $\mathbf{B}^t$ is performed for each feature independently, we can further factorize the prior model $p(\mathbf{B}^t|\mathbf{B})$ as follows:

$$p(\mathbf{B}^t|\mathbf{B}) \propto \prod_i^n p(B_i^t|B_i) \quad (5)$$

where $p(B_i^t|B_i)$ is the probability in term of the deformation between feature B_i and B_i^t. In this paper, the related energy function of $p(B_i^t|B_i)$ is calculated as the shift in translation, namely, $\vec{\delta}_i^t$.

Let J_i^t indicate the image patch matched by the feature B_i^t. To simplify the model, we assume that: 1) the features of the hybrid template $\mathbf{B}^t$ are approximately orthogonal to each other, i.e., one image patch J_i^t can only be explained by one single feature B_i^t, and 2) the deformation from the the given $\mathbf{B}$ to B_i^t, $i = 1, \ldots, n$ are independent under $p(J^t|\mathbf{B}, \mathbf{B}^t)$.

Let $q(J^t)$ denote the reference distribution pooled over the local background regions surrounding the objects of interest, following the active basis model in [25], we can define a log-probability ratio model as follows:

$$\frac{p(J^t|\mathbf{B}, \mathbf{B}^t)}{q(J^t)} = \prod_{i}^{n} \frac{p(J_i^t|B_i^t)}{q(J_i^t)}. \tag{6}$$

The hybrid templates, including the feature types, location, and scales of the feature elements, can be learnt from the observed images by maximizing (6).

In this paper, one feature descriptor may consist of one or multiple filters. Let K_i denote the number of filters, and $F_j(J_i^t)$ denote the response of the jth filter projected on the image patch J_i^t. Being similar to the active basis model proposed in [25], the probability ratio of different feature types can share the same form, as follows:

$$\frac{p(J_i^t|B_i^t)}{q(J_i^t)} = \frac{1}{Z(\lambda_i)} \exp\{-\lambda_i \sum_{j=1}^{K_i} h[F_j(J_i^t)]\} \tag{7}$$

where $h(\cdot)$ is a nonnegative function of filter responses, λ_i is the weight of the ith feature, and $Z(\lambda_i) = \sum_i \exp(-\lambda_i \sum_{t,j} h[F_j(J_i^t)])$ is the normalization factor. Note that: 1) the function $h(\cdot)$ can be designed to perform various transformations, such as whitening and Sigmoid, up to the types of features to model, and 2) different features of the same object template may have different feature weights. While it is usually difficult to accurately calculate the feature weight λ_i and the normalization factor $Z(\lambda_i)$, we use a simple method to directly estimate them as follows.

Parameters Estimation: For a feature B_k, let $r_k = \sum_{t=1}^{T} r_k^t$ and $r_k^t = \sum_j h[F_j(J_i^t)]$, we need to find λ_k such that

$$\bar{r} = E_p(\lambda_k) = r_k \tag{8}$$

which matches the theoretical mean value $\bar{r} = E_p(\lambda_k)$ and the observed mean value r_k. We calculate λ_k by searching a lookup table. Let $\{\hat{\lambda}_1, \hat{\lambda}_2, \cdots, \hat{\lambda}_M\}$ denote M possible values of λ_k, which are sorted in ascending order. For each $\hat{\lambda}_m$, $m \in [1, M]$, we can estimate $Z(\hat{\lambda}_m)$ and $E_p(\hat{\lambda}_m)$ by

$$Z(\hat{\lambda}_m) = \sum_i \exp\{-\hat{\lambda}_m r_i\} \tag{9}$$

$$E_p(\hat{\lambda}_m) = \sum_i r_i \exp\{-\hat{\lambda}_m r_i\} \frac{1}{Z(\hat{\lambda}_m)}. \tag{10}$$

Thus, the lookup table consists of two vectors, $\{Z(\hat{\lambda}_1), \cdots, Z(\hat{\lambda}_M)\}$ and $\{E_p(\hat{\lambda}_1), \cdots, E_p(\hat{\lambda}_M)\}$, both of which have the ascending order. Therefore, for a given r_k, we can first determine the appropriate position m according to the inequations $E_p(\hat{\lambda}_m) \leq r_k \leq E_p(\hat{\lambda}_{m+1})$, and further estimate the parameter λ_k by performing linear interpolation between λ_m and λ_{m+1}, namely

$$\lambda_k = \hat{\lambda}_m + (\hat{\lambda}_{m+1} - \hat{\lambda}_m) \frac{r_k - E_p(\hat{\lambda}_m)}{E_p(\hat{\lambda}_{m+1}) - E_p(\hat{\lambda}_m)}. \tag{11}$$

We can also compute $Z(\lambda_k)$ in a similar way. Note that this estimation approach is reasonable because λ_k is 1-D and a moderate sample size M would be able to provide a robust estimation for $Z(\lambda_k)$. In this paper, we set $M = 40$ empirically.

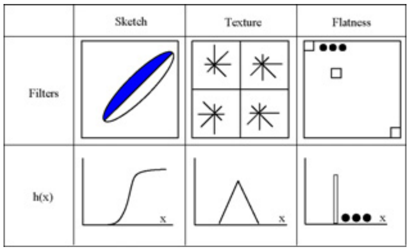


Fig. 3. Illustration of the filters for different types of features (see text for more details).

D. Feature Set

In this section, we discuss how to generalize the unified model 7 for each of the three types of features, namely, sketch, texture, and flatness. The main issues include the designs of filters as well as the filter transformation function $h(\cdot)$.

1) *Sketch:* We describe the sketch features using Gabor wavelets. Let $G_i = (\theta_i, \vec{x}_i, s_i)$ denote the wavelet for the feature B_i, where θ_i denotes the orientation of the wavelet, $\vec{x}_i$ denotes the location, and s_i denotes the scale factor. We set the location and scale of the wavelet G_i the same as that of the feature B_i. Let $G_{\cos,\vec{x},\theta}$ and $G_{\sin,\vec{x},\theta}$ be the Gabor cosine and Gabor sine filters at location $\vec{x}$ and orientation θ. Let $< J^t, G_i >$ denote the coefficient of the Gabor wavelet G_i projected on the image J^t, or the filter response, we have $< J^t, G_i >= \left\|\left\langle J_i^t, G_{\cos,\vec{x}_i,\theta_i}\right\rangle\right\|^2 + \left\|\left\langle J_i^t, G_{\sin,\vec{x}_i,\theta_i}\right\rangle\right\|^2$. Thus, the orientation of G_i can be determined to maximize the filter response as follows:

$$\theta_i = \arg\max_{\theta} \sum_t \max_{\vec{\zeta}_i^t} \left\|\left\langle J_i^t, G_{\cos,\vec{x}_i+\vec{\zeta}_i^t,\theta}\right\rangle\right\|^2 + \left\|\left\langle J_i^t, G_{\sin,\vec{x}_i+\vec{\zeta}_i^t,\theta}\right\rangle\right\|^2 \tag{12}$$

where $\theta \in [-\pi, \pi]$, and the vector $\vec{\zeta}_i^t$ contains the allowed local displacements at the horizontal and vertical directions. This local maximization operation is used for deforming the Gabor basis to fit the observed images.

The transformation function $h(\cdot)$ for sketch features is chosen as a crude approximation to the whitening transformation function as follows:

$$h^{\text{sk}}[< J^t, G_i >] = \min(< J^t, G_i >, \zeta) \tag{13}$$

where ζ is a threshold, and set to be $\zeta = 16$ in this paper.

Thus, the probability model 7 can be generalized for the sketch features as follows:

$$\frac{p(J_i^t|B_i^t)}{q(J_i^t)} = \frac{1}{Z(\lambda_i^{\text{sk}})} \exp\{-\lambda_i^{\text{sk}} h^{\text{sk}}[< J^t, G_i >]\}. \tag{14}$$

2) *Texture:* We use the locally normalized histogram of oriented gradient (HOG) [6] to describe texture features. As illustrated in Fig. 3, we uniformly divide each patch into 2 × 2 cells of equal size, and extract a 9-bin HOG for each cell. We further concatenate these four HOG histograms to form a 36-bin histogram, each bin of which is considered as one independent filter, denoted as $F_j^{\text{txt}}(J_i^t)$, and thus the total number of filters for texture features is $K = 36$. Since the probability distributions over texture regions usually bear the

bandpass form [26], as illustrated in Fig. 3, we can define the corresponding filter transformation as follows:

$$h^{\text{txt}}[F_j^{\text{txt}}(J_i^t)] = \left| F_j^{\text{txt}}(J_i^t) - \frac{1}{K}\sum_{l=1}^{K} F_l^{\text{txt}}(J_i^t) \right|^2 . \tag{15}$$

Thus, the unified model 7 can be generalized for texture features as follows:

$$\frac{p(J_i^t|B_i^t)}{q(J_i^t)} = \frac{1}{Z(\lambda_i^{\text{txt}})} \exp\{-\lambda_i^{\text{txt}} \sum_j h^{\text{txt}}[F_j^{\text{txt}}(J_i^t)]\}. \tag{16}$$

3) *Flatness:* We describe the flatness features using the intensity values at different positions. The range of intensity value is evenly quantized into multiple bins, e.g., 16. Thus, the total number of filters for the flatness features is the product of the patch size and the number of bins. This leads to a large number of filter parameters to estimate, which may be computationally infeasible. Since the intensities at different positions within the same flatness patch are usually concentrated to one specific value, it is reasonable to assume that filters of the same bin at different positions have the same response. This assumption reduces the number of filters for flatness features to be the same as the number of bins. Formally, let $J_i^t(\vec{x})$ denote the intensity value at the position $\vec{x}$, $|J_i^t|$ denote the number of pixels within the patch J_i^t, $F_j^{\text{fl}}[J_i^t(\vec{x})]$ be the response of the jth filter at the position $\vec{x}$, and $F_j^{\text{fl}}(J_i^t)$ denote the corresponding position-free filter response. Actually, $F_j^{\text{fl}}[J_i^t]$ is calculated as the fraction of pixels with that particular intensity bin and we have

$$F_j^{\text{fl}}[J_i^t] = \frac{1}{|J_i^t|} \sum_{\vec{x}\in J_i^t} F_j^{\text{fl}}[J_i^t(\vec{x})] \tag{17}$$

$$F_j^{\text{fl}}[J_i^t(\vec{x})] = \begin{cases} 1, & J_i^t(\vec{x}) \in jthbin \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

The transformation function for the flatness features is simplified as $h^{\text{fl}}[F_j^{\text{fl}}(J_i^t)] = F_j^{\text{fl}}(J_i^t)$, since the intensity distributions over flatness patches are usually with the form of an indicator function, as illustrated in Fig. 3. Accordingly, we have the model for flatness features as follows:

$$\frac{p(J_i^t|B_i^t)}{q(J_i^t)} = \frac{1}{Z(\lambda_i)} \exp\{-\sum_j \lambda_i h^{\text{fl}}[F_j^{\text{fl}}(J_i^t)]\}. \tag{19}$$

In summary, our proposed hybrid template consists of three types of features, namely, sketch, texture, and flatness, each of which is described as one or multiple filters. For a given image patch, we first associate it with different feature types and calculate the likelihood ratios defined in 14, 16, and 19, respectively. Next, the feature type can be heuristically determined in favor of the type that achieves the maximum ratio. In this way, the ratio is used as a general metric, with which the features of different types are made comparable to each other and thus can compete to explain the same patch of the observed frame.

III. Object Tracking Via Learning Hybrid Template

As mentioned above, our goals are twofolds: 1) to estimate the optimal object location $\mathbf{X}^t$ within each observed frame I^t, and 2) to build and maintain the hybrid object template from the foreground images $\{J^t = I^t[\mathbf{X}^t]\}$. We formulate these two targets as maximizing a posterior as follows:

$$\max_{\{\mathbf{X}^t\},\{\mathbf{B}^t\}} p\left(\{\mathbf{X}^t\}, \{\mathbf{B}^t\}, \{I^t\}|\mathbf{B}; \Theta\right) \tag{20}$$

which can be further rewritten as follows:

$$\max_{\{\mathbf{X}^t\},\{\mathbf{B}^t\}} p(\{\mathbf{B}^t\}, \{I^t\}|\mathbf{B}, \{\mathbf{X}^t\}; \Theta) p(\{\mathbf{X}^t\}; \Delta) \tag{21}$$

or

$$\begin{aligned} & \max_{\{\mathbf{X}^t\},\{\mathbf{B}^t\}} p(\{\mathbf{X}^t\}, \{I^t\}|\mathbf{B}, \{\mathbf{B}^t\}; \Theta) p(\{\mathbf{B}^t\}|\mathbf{B}) \\ \Rightarrow & \max_{\{\mathbf{X}^t\},\{\mathbf{B}^t\}} p(\{J^t\}|\mathbf{B}, \{\mathbf{B}^t\}; \Theta) p(\{\mathbf{B}^t\}|\mathbf{B}) \end{aligned} \tag{22}$$

where $p(\{\mathbf{X}^t\}; \Delta)$ is the prior term and Δ indicates the motion parameter which shall be further defined in the following sections.

On the one hand, 21 indicates that if we have an accurate object position $\mathbf{X}^t$ in the frame I^t, we can crop the image of foreground region, denoted as $J^t = I^t[\mathbf{X}^t]$, and learn the hybrid template from these roughly aligned images $\{J^t\}_{t=1}^T$. On the other hand, 22 indicates that if we have a good object template $\mathbf{B}$, the tracking task may degenerate to matching the given object template with the observed frames. More specially, we can match every feature within $\mathbf{B}$ with the local patches in the frame I^t to estimate the feature positions within I^t, and further fuse all the estimated positions of individual features to determine the object bounding box $\mathbf{X}^t$. Therefore, we could introduce an iterative procedure to alternately optimize the object location and the hybrid template.

A. Feature Pursuit on Roughly Aligned Images

Suppose the object bounding boxes $\{\mathbf{X}^t\}_{t=1}^T$ are estimated and the corresponding images of foreground regions, i.e., $\{J^t\}_{t=1}^T$, are cropped, we propose a feature pursuit procedure to learn the templates $\{\mathbf{B}^t\}_{t=1}^T$. We formulate this task by maximizing the probability ratio, namely, $\prod_t \frac{p(J^t, \mathbf{B}^t|\mathbf{B})}{q(J^t)}$, as introduced in Section II-C, with the given distribution $q(J^t)$ over the local background regions. After expanding the above ratio and applying the logarithm, we have

$$\begin{aligned} & \max_{\mathbf{B}^t} \prod_t \frac{p(J^t, \mathbf{B}^t|\mathbf{B})}{q(J^t)} \\ \Rightarrow & \max_{\mathbf{B}^t} \prod_t \frac{p(\mathbf{B}^t|\mathbf{B}) p(J^t|\mathbf{B}, \mathbf{B}^t)}{q(J^t)} \\ \Rightarrow & \max_{\mathbf{B}^t} \sum_{i,t} \log p(B_i^t|B_i) + \log \frac{p(J_i^t|B_i^t)}{q(J_i^t)} \\ \Rightarrow & \max_{\mathbf{B}^t} \sum_i G_D(B_i) + G_F(B_i) \\ \Rightarrow & \max_{\mathbf{B}^t} \sum_i G(B_i) \end{aligned} \tag{23}$$

Algorithm 1 Feature pursuit on roughly aligned images

```
1: Input: Images $J = \{J^1, \cdots, J^T\}$ that contains the same object only.
   Output: Hybrid template $\mathbf{B} = \{B_1, \cdots, B_n\}$ and parameter $\lambda = \{\lambda_\mathbf{1}, \ldots, \lambda_\mathbf{n}\}$.
2: Collect candidate feature set: We partition each image into a set of patches of equal size, and collect the patches that have the same position but locate within different images to form one candidate feature, denoted as $B_j$. Let $\Omega = \{B_j\}$ denote the candidate feature set.
3: Parameter estimation: For each candidate feature $B_j \in \Omega$, estimate its feature weight $\lambda_j$ using 11.
4: Feature measurement: For each $B_j \in \Omega$, calculate its feature gain $G(B_j) = G_D(B_j)+G_F(B_j)$ using 23. Let $i = 1$ and $\mathbf{B} = \emptyset$.
5: Feature selection: Let $B_i$ denote the candidate feature that has the largest gain among all the features in $\Omega$. $\mathbf{B} = \mathbf{B} \bigcup B_i$ and $i = i + 1$.
6: Local inhibition:
    6.1 let $\nabla B_i = \{B_j = \{\ell_j, \vec{x}_j, s_j\}; \forall j, |\vec{x}_i - \vec{x}_j| \leq \epsilon\}$ for a small $\epsilon > 0$;
    6.2 $\Omega \leftarrow \Omega \setminus (\nabla B_i \cup B_i)$.
7: Let $i \leftarrow i + 1$, if $i \leq n$, go to 5; otherwise, exit.
```

where $G(B_i)$ indicates the gain of feature B_i, i.e., how well the feature B_i fits with the observed frames. We further divide $G(B_i)$ into two independent parts, namely, the deformation part $G_D(B_i) = \frac{1}{T}\sum_t \log p(B_i^t|B_i)$ and the feature fitting part $G_F(B_i) = \sum_i \frac{1}{T}\sum_t \log \frac{p(J_i^t|B_i^t)}{q(J_i^t)}$.

We apply the matching pursuit algorithm [20] to learn the object template $\{\mathbf{B}\}$ from the observed images $\{J^t\}_{t=1}^T$, by maximizing the total gain defined in 23. Fig. 4 illustrates the feature pursuit procedure. When a feature is selected, the feature should be shared by all the observed images, in the sense that a perturbed version of this feature is generated to fit with the specific image. Therefore, the perturbed versions of the selected feature locate within different images often characterize the same pattern. For example, in Fig. 4, when the feature with yellow color is selected, it is attracted to the nearby region of "white hat" in each observed image.

We summarize the pursuit procedure in Algorithm 1. Note that we assume one image patch can only be explained by one feature, which leads to the local inhibition operation after selecting the current feature, as described in Step 6. Herein, we set ϵ as the half of the lattice width such that there is no overlapping between the selected features. However, it is possible to allow overlapping to some extent in implementation.

B. Feature Alignment Via Heuristic Search

Suppose the object template $\mathbf{B}$ is learnt, the goal of feature alignment is to match the features of $\mathbf{B}$ into the observed frames $[I^1, \ldots, I^t, \ldots, I^T]$. In order to exploit the inter-feature geometry structure information, we represent the features of $\mathbf{B}$ using an adjacent graph, denoted as $G^S = (V^S, E^S)$, where V^S indicates a set of graph vertices each representing one feature from the template $\mathbf{B}$, and E^S indicates a set of graph edges each linking two spatially adjacent features. For each vertex of G^S, there are at most four edges linking to other four nearest vertices. In addition, we evenly partition every observed frame I^t into a set of local patches of equal size, and further collect the patches that roughly have the same position but locate within different frames to construct one candidate feature. Let Ω denote the candidate feature set. Taking these candidates features as graph vertices, we build another adjacent graph, denoted as $G^T = (V^T, E^T)$. Note that, every vertex in V^T indicates a set of image patches. In this context, we can formulate the problem of feature alignment between $\mathbf{B}$ and the candidate feature set Ω as seeking the optimal correspondence between the source graph G^S and the target graph G^T. Formally, we denote the desired mapping relationship as $C : V^S \to V^T$.

We use the branch-and-bound method to heuristically seek the optimal correspondence between the source graph G^S and the target graph G^T. Each vertex $v \in V^S$ is initially matched to all vertices in the graph G^T, and we denote this initial mapping as $\mathcal{M}(v) = V^T$. The branch-and-bound method starts with a graph vertex in G^S and branch to other vertices while pruning the bad mappings. Each graph vertex in G^S is originally matched to a set of vertices in G^T, depicted as the plots in the lines.

Let $\mathcal{U}$ denote the seed set that stores the currently explored vertices of G^S and $\mathcal{C}$ be a set of mapping, each indicating one possible correspondence of $\mathcal{U}$. The feature alignment procedure contains three main repetitive steps: 1) *select* one vertex from V^S and add it into the seed set $\mathcal{U}$; 2) *branch* the seed set $\mathcal{U}$ to one of its adjacent vertices u', and meanwhile, branch every mapping of $\mathcal{U}$ to the matches of u', i.e., $\mathcal{C} = \mathcal{C} \otimes \mathcal{M}(u')$, where $\otimes$ denotes the Cartesian product of two sets; and 3) *prune* the bad mappings that achieve poor confidences in term of a bound cost function, which returns the shape similarity between two position sets. The above steps are conducted iteratively until the seed set $\mathcal{U}$ contains all the vertices of V^S.

We use the squared Procrustes distance [16] to define the bound function $BCost(\cdot)$. Denote Y and Y' as two matched position sets, and their corresponding complex forms as $\mathcal{J}(Y)$ and $\mathcal{J}(Y')$. We have

$$BCost(Y, Y') = 1 - \frac{|\mathcal{J}(Y)^* \cdot \mathcal{J}(Y')|^2}{\mathcal{J}(Y)^* \cdot \mathcal{J}(Y) \cdot \mathcal{J}(Y')^* \cdot \mathcal{J}(Y')} \tag{24}$$

where $\mathcal{J}(Y)^*$ and $\mathcal{J}(Y')^*$ are the conjugations of $\mathcal{J}(Y)$ and $\mathcal{J}(Y')$ [16].

The procedure of feature alignment via heuristic search is summarized in Algorithm 2. Note the following.

1) The inputs of Algorithm 2 include the object template $\mathbf{B}$, which is previously learnt from the frame sequence, and the currently observed frames $\{I^t\}_{t=1}^T$. The outputs are the optimal correspondence between the feature set V^S and the candidate feature set V^T extracted from the observed frames.
2) While constructing the target graph G^T, each vertex indicates a set of image patches which roughly have the same positions but locate within different frames.
3) In Step 3, we initially match each vertex $v \in V^S$ to all the vertices V^T in the graph G^T. In implementation,

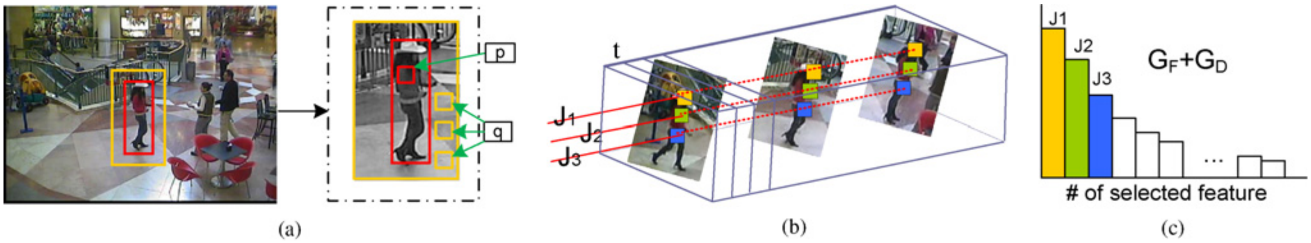


Fig. 4. Feature pursuit procedure. (a) Input frame. The red box indicates the foreground region and the yellow box indicates the local background region surrounding the object. p denotes the feature distribution pooled over the foreground region and q denotes the feature distribution pooled over the local surrounding background region. (b) Shared feature pursuit. While one patch of one image is selected, it must be shared by other images. (c) Illustrations of information gain $G_F + G_D$ for different patches. The patches are selected sequentially according to the information gain.

Algorithm 2 Feature alignment via heuristic search

```
1: **Input**: Template $\mathbf{B}$ and the observed frames $\{I^t\}, t = 1, \dots, T$.
   **Output**: The optimal correspondence $C^*$; object bounding box $\mathbf{X}^t$ within $I^t$.
2: Build the source graph $G^S = (V^S, E^S)$ for the object template $\mathbf{B}$.
   Collect candidate features from the observed frames and build the target graph $G^T = (V^T, E^T)$.
3: For each $v \in V^S$, set $\mathcal{M}(v) = V^T$.
4: Select $v_0 \in V^S$, set $\mathcal{U} = \{v_0\}$, $\mathcal{C} = \{\mathcal{M}(v_0)\}$, $\mathcal{V} = V^S \setminus \{v_0\}$.
5: While $\mathcal{V} \neq \emptyset$:
    5.1 select one vertex $u \in \mathcal{U}, u' \in \nabla u, u' \in \mathcal{V}$, where $\nabla u$ indicates the set of vertices being adjacent to the vertex $u$;
    5.2 branch the seed set, $\mathcal{U} = \mathcal{U} \cup \{u'\}, \mathcal{V} = \mathcal{V} \setminus \{u'\}$, $\mathcal{C} = \mathcal{C} \otimes \mathcal{M}(u')$, $\otimes$ indicates the Cartesian product of two sets;
    5.3 for each $C \in \mathcal{C}$,
        if $BCost(C, \mathcal{U}) > \eta$ (set to be $\eta = 0.5$), $\mathcal{C} = \mathcal{C} \setminus \{C\}$.
6: Set $C^* = \arg\min_C BCost(C, \mathcal{U})$, for $\forall C \in \mathcal{C}$.
7: Estimate the object location and scale, namely, $\mathbf{X}^t$, according to $C^* : V^S \rightarrow V^T$.
```

we can compute the appearance similarities between v and the vertices in V^T, and remove the bad matches to improve the computational efficiency.

4) We denote $\mathcal{C}$ as a set of mappings, i.e., each element of $\mathcal{C}$ represents one possible correspondence between the seed set $\mathcal{U}$ and the candidate feature set. In Step 5.2, while branching the seed set to one adjacent vertex u', we set $\mathcal{C} = \mathcal{C} \otimes \mathcal{M}(u')$ to branch every mapping in $\mathcal{C}$ to the matches of u'.

5) The branch-and-bound method may generate multiple mappings, and in Step 6, we set C^* as the best mapping, which bears the minimum cost in terms of $BCost()$.

C. Simultaneous Feature Alignment and Feature Pursuit (SFAFP)

We alternately perform Algorithm 1, i.e., the feature pursuit procedure, and Algorithm 2, i.e., the feature alignment procedure, to learn the object template from the frame sequence $\{I^t\}$, and meanwhile, track each feature of the template for the observed frames.

Algorithm 3 Procedure for SFAFP

```
1: **Input:** Observed frame sequence $I = \{I^1, \cdots, I^T\}$, initial hybrid template $\mathbf{B} = \{B_1, \dots, B_n\}$, and parameters $\lambda = \{\lambda_1, \dots, \lambda_n\}$.
2: Do (iteration body):
    1) *Feature alignment.*
        1.1 Call Algorithm 2 with the inputs of $\{I^t\}$ and $\mathbf{B}$, to obtain the optimal mapping $C^* : V^S \rightarrow V^T$ and the object bounding boxes $\{\mathbf{X}^t\}, t \in [1, T]$;
        1.2 For each $t \in [1, T]$, crop the image of foreground region, denoted as $J^t = I^t[\mathbf{X}^t]$.
    2) Set $\mathcal{N} = \emptyset$; for each $B_j \in V^T, \forall B_i \in V^S, C^*(B_i) \neq B_j$, set $\mathcal{N} = \mathcal{N} \bigcup B_j$.
    3) *Feature pursuit.*
       Call Algorithm 1 with the inputs of $\{J^t\}_{t=1}^T$ to obtain a new object template $\hat{\mathbf{B}} = \{\hat{B}_1, \dots, \hat{B}_n\}$ and the feature parameters $\hat{\lambda} = \{\hat{\lambda}_1, \dots, \hat{\lambda}_n\}$.
    4) *Template update.*
        4.1 *Parameters update:* For each $B_i = (l_i, \vec{x}_i, \theta_i) \in \mathbf{B}$, $i \in [1, n]$
            $$\begin{aligned} \lambda_i &\leftarrow (1-\beta)\lambda_i + \beta\hat{\lambda}_i \\ \vec{x}_i &\leftarrow (1-\beta)\vec{x}_i + \beta\hat{\vec{x}}_i \end{aligned}$$
            where $\beta$ is the constant forgotten factor.
        4.2 *Features update:* First, select $m > 1$ features from the newly occurred feature set $\mathcal{N}$ which achieve the most feature gains and add them into the template $\mathbf{B}$. Second, re-rank all features of $\mathbf{B}$ according to feature parameters $\lambda_i$. Last, the top $n$ features are labeled as formal features, the bottom feature is removed, and the remaining features are labeled as candidate features.
3: **Output:** Object location $\{\mathbf{X}^t\}_{t=1}^T$ and refined hybrid template $\mathbf{B}$.
```

We summarize the tracking procedure in Algorithm 3. Note the following.

1) The inputs of this procedure are a set of observed frames and the hybrid template $\mathbf{B}$ learnt from the previous frames. The choice of the observed window size, namely, T, is essentially a tradeoff between performance and efficiency. We set $T = 15$ empirically in this paper. We

Algorithm 4 Procedure for Real-time Adaptive Object Tracking

```
1: Input: Real-time Video Sequence
2: Initialization
     2.1 For each observed frame,
         Update the background modeling module [11] using the current observations;
     2.2 Set templates set $\mathcal{R} = \emptyset$;
3: For each currently observed frame $I^t$,
     3.1 Extract foreground regions using the background model [11];
     3.2 For each object, $(\mathbf{B}, \lambda) \in \mathcal{R}$, (tracking)
       3.2.1 Call Algorithm 3 with the inputs of frames $\{I^{t-T}, \cdots, I^t\}$ and $(\mathbf{B}, \lambda)$;
       3.2.2 Output: the object bounding box $\mathbf{X}^t$ in frame $I^t$;
     3.3 For each foreground bounding box $\mathbf{X}$ in frame $I^t$, which is not matched with any interest objects,
       3.3.1 Crop image $J = I^t[\mathbf{X}]$;
       3.3.2 Call Algorithm 1 with the input of a single image $J$, to obtain the initial template $(\mathbf{B}^{new}, \lambda^{new})$;
       3.3.3 $\mathcal{R} = \mathcal{R} \cup \{(\mathbf{B}^{new}, \lambda^{new})\}$;
```

shall extend Algorithm 3 to tackle the real-time frame sequence in the later section.

2) The algorithm contains three main steps as follows:
   a) feature alignment;
   b) feature pursuit;
   c) template update.

   The first two steps are used to alternately optimize the object locations $\{\mathbf{X}^t\}$ and the object template $\mathbf{B}$, while the third step is used to gradually update the learnt template.

3) All the features in the hybrid template are labeled as "formal" or "candidate" according to their weights, namely, λ_i. Only the formal features are used while conducting feature alignment.

D. Motion Prior

We introduce the multiframe motion prior, formulated as $p(\{\mathbf{X}^t\}; \Delta)$ in 21 and 22, into our tracking algorithm. The model is first learnt from the current tracking results and then applied to the the newly occurred frames to predict the object locations and scales within frames. Formally, we denote the motion parameter as $\Delta = (\mathbf{v}, \eta, \sigma)$, which consists of three components, i.e., the velocity $\mathbf{v}$, the initial position η, and the noise level σ. Each component has four sub-components, including its horizontal coordinate, vertical coordinate, width, and height. Formally, we have $\mathbf{v} = (v_x, v_y, v_w, v_h)'$, $\eta = (\eta_x, \eta_y, \eta_w, \eta_h)'$, and $\sigma = (\sigma_x, \sigma_y, \sigma_w, \sigma_h)'$. Thus, we can estimate the object bounding box $\hat{\mathbf{X}}_t$ at frame t by $\hat{\mathbf{X}}^t = \eta + \mathbf{v}t$. Herein, we assume $\hat{\mathbf{X}}_t$ follow with a Gaussian distribution, i.e., $\hat{\mathbf{X}}_t \sim \mathcal{N}(\mathbf{X}_t, \sigma)$, which can be learnt from the previous tracking results. This multiframe motion prior can be used to improve the robustness against the common tracking challenges, such as background clutters, full/partially occlusions, or objects intersections.

E. Adaptive Object Tracking for Real Video Sequence

Taking all above components, we now extend the proposed SFAFP procedure in Algorithm 3 to track multiple objects in real-time video sequences. As illustrated in Fig. 2, the observed frames are accumulated from the real video sequence and taken as inputs to call Algorithm 3. We move the observation window of size T with a fixed step, e.g., two frames, and process each window step by step. Herein, we summarize the entire real-time tracking procedure in Algorithm 4 and remark several discussions as follows.

1) The real-time tracking algorithm begins with an initialization stage, where the background model [11] is built for detecting the foreground regions within each frame.
2) While processing the tth frame, the current learnt templates $\mathcal{R}$ as well as the previous T frames are used as inputs to call Algorithm 3. This leads to a sliding-window-based tracking procedure, which is also widely employed in the previous papers [27].
3) In Step 3, we first track each object in $\mathcal{R}$ into the current frame I^t, and then detect the newly occurred objects from the remaining foreground regions using the approach proposed by Hu *et al.* [11]. Next, for each new object, the Steps 4.1.1–4.1.3 are conducted to initialize the corresponding object template.

F. Discussions About Model Drift

Online model-based tracking algorithms are usually exposed to the risk of model drift that roots in their ill-posed nature [27]. Although it is still an open problem in tracking community, we enforce three novel characteristics of our approach.

1) Our method adaptively updates the learnt object templates by using both the jump dynamic, such as replacing the older and lower confident features with the newer and higher confident ones, and the diffusion dynamic, such as updating the feature parameters. Thus, the overall update procedure is well driven by the bottom data information, as justified in [32].
2) The model pursuit procedure is conducted on a batch of deferred observations, which is more robust compared to the traditional sequential inference [27], especially when there exist various scene noises, lighting changes or occlusions in video scenes.
3) In the step of feature alignment, we only use the formal features, namely, the features that are more confident, for localizing the objects. Although the above components cannot completely solve the problem of model drift, we can obtain encouraging experiment results over challenging video scenarios as demonstrated in the later section.

IV. Experiments

In this section, we apply our approach to visual tracking and carry out the experiments with comparisons to the state-of-the-art algorithms.

A. Experiment Settings

1) *Parameters:* We start by introducing the parameter setting for our proposed approach. The size of the observed

TABLE I

DETAILS OF DATABASES USED IN THIS PAPER

	TRECVID08 [23]	LHI [30]	PETS [8]	I-80
No. of clips	10	8	8	8
No. of frames	13 450	8644	6455	7920
No. of objects	436	241	112	104

window is set as $T = 15$ frames. The size of image patch is fixed to be 12×12 pixels. The allowed displacement in location is set to be $a = 5$ pixels. We use three types of features to describe image patches.

1) For sketch features, the size of the Gabor wavelets is fixed to be 12×8 pixels. The orientation θ takes 15 equally spaced angles within the range of $[0, 2\pi]$.
2) For texture features, we first partition each patch into four cells, as illustrated in Fig. 3. Then, we extract a 9-bin normalized gradient histogram from each cell and concatenate them to form a 36-bin histogram.
3) For flatness features, we use a 16-bin local histogram of image intensity. Note that, for texture and flatness features, each bin of the histogram is considered as one filter in the unified probability model 7. We fix the number of features in the hybrid template to be $n = 60$.

We perform Algorithm 4 every two frames and update the hybrid template through replacing one candidate feature with a newly selected feature, and gradually update model parameters with the forgotten factor $\beta = 0.2$ (see Algorithm 3). The maximum number of iterations is set to be 10 empirically.

2) *Platform:* We implement Algorithm 4 using the C/C++ language and integrate it with a surveillance system, INTMON [11], which has been applied in industry. The system is able to process 15–20 f/s on an Intel Xeon X5450 Computer with 3.0 GHz central processing unit and memory size of 4 GB.

3) *Dataset:* The video clips we used are selected from four public datasets: TRECVID08 [23], Lotus Hill Institute (LHI) [30], PETS [8], and I-80.[1] These datasets include challenging scenes with severe occlusions, scale changes, or complex background structure. We manually annotate the object bounding boxes within each frame as the ground truths of object trajectories. Table I depicts the details of each database.

4) *Benchmark:* The benchmark baselines include four state-of-the-art algorithms as follows.

a) The method proposed by Birchfield *et al.* [2] which combines the ideas of Lucas-Kanade and Horn-Schunck to jointly trace sparse interest points and edges, named *JLK*.
b) The spatial selection algorithm for attentional visual tracking (*AVT*) proposed by Yang *et al.* in [29].
c) The online feature space selection algorithm proposed by Collins *et al.* [3] (*Collins*), which uses the two-class log-likelihood variance ratio to measure feature salience.
d) The particle filtering (*PF*) tracking method [4]. We keep the parameter settings of the above baselines the same as in their original papers.

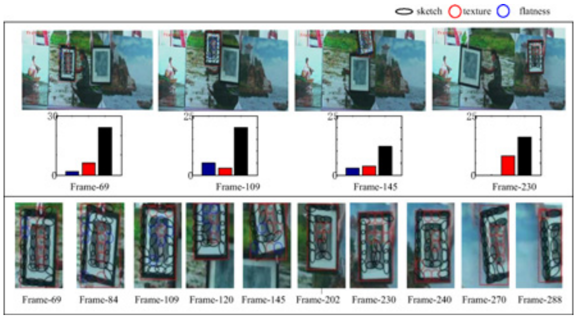


Fig. 5. Adaptive tracking by feature pursuit. Each type of feature is plotted with different color.

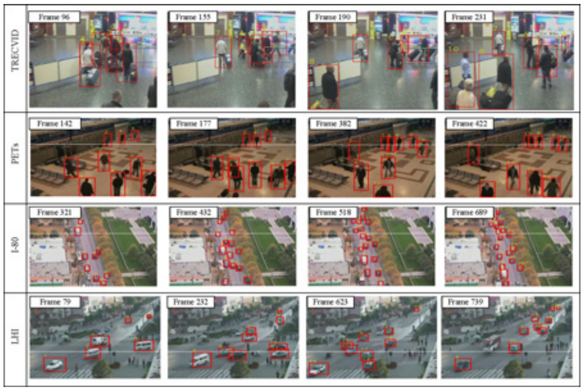


Fig. 6. Sample results on TRECVID08, PETs, I-80, and LHI test videos. Each row shows the images of one scene overlaid with the object bounding boxes.

5) *Metric:* The evaluation metrics we use are listed as follows.

a) *Recall* (frame-based), defined as number of correctly matched objects/total number of ground-truth objects.
b) *Precision* (frame-based), number of correctly matched objects/total number of output objects.
c) *Fa/Frm*, number of false alarms per frame (the smaller is the better), calculated by averaging over all the test videos.
d) *Mostly tracked (%)*, MT, percentage of ground truth trajectories which are covered by tracker output for more than 80% in length.
e) *Mostly lost (%)*, ML, percentage of ground-truth trajectories which are covered by tracker output for less than 20% in length (the smaller the letter).

A program is written to compute above metrics automatically. The key point is the matching between ground-truth and the tracking results, which is non-trivial itself. We implemented this part by the Hungarian algorithm based on the VACE evaluation software [15].

B. Results with Analysis

Fig. 5 depicts a set of hybrid templates learnt from one test video from the LHI database [30]. The top row shows four frames overlaid with tracking results and the numbers of three different types of features contained in the current template. The bottom row shows the learnt hybrid templates

[1] Available at http://ngsim.fhwa.dot.gov.

TABLE II
RESULTS ON TRECVID08 DATABASE [23]

	Recall (%)	Precision (%)	Fa/Frm	MT (%)	ML (%)
PF [4]	68.3	63.50	3.352	65.6	15.4
Collins [3]	79.4	75.1	1.723	71.4	12.2
JLK [2]	84.7	85.9	0.224	81.5	3.9
AVT [29]	82.4	85.1	0.245	82.6	5.6%
Ours	**88.3**	**87.7**	**0.145**	**83.6**	**3.6**

TABLE III
RESULTS ON LHI DATABASE [30]

	Recall (%)	Precision (%)	Fa/Frm	MT (%)	ML (%)
PF [4]	75.4	73.2	0.558	76.9	15.8
Collins [3]	80.5	81.6	0.347	84.7	8.2
JLK [2]	86.3	84.7	0.257	87.5	6.3
AVT [29]	87.2	83.4	0.369	88.6	3.6
Ours	**89.3**	**85.7**	**0.235**	**90.5**	**2.8**

TABLE IV
RESULTS ON PETS DATABASE [8]

	Recall (%)	Precision (%)	Fa/Frm	MT (%)	ML (%)
PF [4]	65.6	68.0	1.251	72.1	19.2
Collins [3]	78.1	80.6	0.859	76.8	8.7
JLK [2]	85.2	**86.4**	0.378	79.2	4.9
AVT [29]	86.6	84.1	0.457	81.3	4.5
Ours	**87.1**	85.9	**0.360**	**82.7**	**4.3**

while tackling ten different frames. From the results, we can draw the following conclusions.

1) The hybrid templates are adaptively updated according to the local background regions.
2) The feature parameters, such as orientation and locations, are also adaptively changed to maximize the information gains defined in 23.
3) Most of the foreground regions of interest have been explained by more than one types of features, which coincides with our motivation, namely, different types of features play equally important roles in visual tracking.

Fig. 6 shows several video frame sequences overlaid with the tracking results. Most of the videos are very challenging due to the crowded objects, scale changes, severe occlusions, and low resolution. For example, in the first scenario from the TRECVID test videos, the pedestrians with the IDs of #0, #3, #4, #5, #6, #7 step out the airport with severe inter-object occlusions and interactions. Also, there are about 10 pedestrians in the second scenario from the PETs database and about 15 cars in the third scenario from the I-80 database. Our method can work very well on above videos against various challenges. In addition, it is interesting to observe that, while there are severe occlusions in the videos, our method can still work correctly, because it combined the multiframe motion prior to predict the object locations.

Tables II–V show the comparisons of different metrics among our approach and other four algorithms, including PF [4], Collins [3], JLK [2], and AVT [29], on the TRECVID08 [23], LHI [30], PETs, and I-80 datasets. From the results, we can have the following observations.

TABLE V
RESULTS ON I-80 DATABASE

	Recall (%)	Precision (%)	Fa/Frm	MT (%)	ML (%)
PF [4]	67.4	65.1	3.48	69.8	12.5
Collins [3]	80.8	81.4	0.926	72.3	7.8
JLK [2]	89.1	85.6	0.318	85.2	5.1
AVT [29]	88.1	86.8	0.227	85.7	4.9
Ours	**90.2**	**87.5**	**0.139**	**86.3**	**2.3**

1) Among all the algorithms, ours achieves the best recall rates and precision rates on all the four databases. For JLK [2], which uses the similar idea of jointly tracking multiple features, although its performance is already good, our approach outperforms it with the margin of 3.6 percentage in term of the precision rate and the margin of 1.8 percentage in term of the recall rate on the TRECVID database.
2) Our approach achieves the false alarm number per frame (Fa/Frm) of 0.145, 0.235, 0.360, and 0.139, on the TRECVID08, LHI, PETs, and I-80 databases, respectively. In contrast, the corresponding best results of other four baselines are 0.224, 0.257, 0.378, and 0.227, which is much lower than the proposed solution. In addition, the MT (and ML) of our approach are 83.6% (3.6%), 90.5% (2.8%), 82.7% (4.3%), and 86.3% (2.3%) on the TRECVID08, LHI, PETs, and I-80 databases, respectively. These results also clearly outperform the corresponding best results achieved by other four baselines, namely, 82.6% (3.9%), 88.6% (3.6%), 81.3% (4.5%), and 85.7% (4.9%). The comparisons on the above three metrics show that our approach is much more applicable for practical applications, e.g., video surveillance.

V. CONCLUSION

This paper proposed to jointly track different types of features, including sketch, texture, and flatness, by representing the objects of interest with the hybrid templates. A simple yet effective generative model was developed to learn the hybrid template from the batch of observations, and meanwhile to estimate the object location and scale robustly. Extensive experiments with comparisons showed that our algorithm clearly outperforms several popular tracking algorithms and works very well over various challenging scenarios against various challenges, such as background clutters, scale changes, or frequent object intersection.

The proposed model in 7 naturally decomposes a color image into three components, i.e., flatness regions, texture, and sketch/shape. The proposed feature pursuit algorithm have to compare among these three dictionaries. Each time, we choose a sketch, a texture feature or a flatness feature so that it tells the maximum statistic difference between the foreground and background images, i.e., achieves the maximum likelihood ratio as defined in 7. Actually, our work can be considered as a practical extension of the perceptual transition theory proposed by Wang and Zhu [24] and the active basis theory proposed by Wu *et al.* [25].

We plan to further investigate this work from two aspects. First, the hybrid template can be extended by introducing a hierarchical structure of image patches. The patches at multilevel scales may be overlapped within each other and thus expected to capture rich information with images. Second, in order to improve the robustness against large-scale changes, we plan to introduce additional feature generation stage when tracking the hybrid templates. This idea is motivated by the following observation: when an object moves toward (or away from) the camera, one older patches shall be split into several new patches (or several older patches shall be merged into one single patch).

Acknowledgment

The authors would like to thank Prof. S.-C. Zhu for extensive discussions. The data used in this paper were provided by the Lotus Hill Annotation Project [30].

References

[1] V. Badrinarayanan, P. Perez, F. L. Clerc, and L. Oisel, "Probabilistic color and adaptive multi-feature tracking with dynamically switched priority between cues," in *Proc. IEEE Conf. Comput. Vision*, vol. 1. Oct. 2007, pp. 1–8.

[2] S. T. Birchfield and S. J. Pundlik, "Joint tracking of features and edges," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2008, pp. 1–6.

[3] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1631–1643, Oct. 2005.

[4] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, 2000.

[5] P. Dollar, Z. Tu, H. Tao, and S. Belongie, "Feature mining for image classification," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2007, pp. 1–8.

[6] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. Eur. Conf. Comput. Vision*, 2006, pp. 428–441.

[7] J. Friedman, "Exploratory projection pursuit," *J. Am. Statistic. Assoc.*, vol. 82, no. 397, pp. 249–266, Mar. 1987.

[8] R. B. Fisher, "PETS04 surveillance ground truth data set," in *Proc. IEEE Int. Work. Performance Eval. Track. Surveillance*, May 2004, pp. 1–5.

[9] C. Guo, S.-C. Zhu, and Y. Wu, "Primal sketch: Integrating structure and texture," in *Proc. Special Issue Generative Model Based Vision, Comput. Vision Image Understand.*, vol. 106. 2007, pp. 5–19.

[10] G. Hua and Y. Wu, "Measurement integration under inconsistency for robust tracking," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 1. Jun. 2006, pp. 650–657.

[11] W. Hu, H. Gong, and S.-C. Zhu, "An integrated background model for video surveillance based on primal sketch and 3-D scene geometry," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2008, pp. 1–8.

[12] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting," in *Proc. British Mach. Vision Conf.*, vol. 1. 2006, pp. 47–56.

[13] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 1. Jun. 2006, pp. 260–267.

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

[15] R. Kasturi, D. Goldgof, V. Manohar, M. Boonstra, and V. Korzhova, "Performance evaluation protocol for face, person and vehicle detection and tracking in video analysis and content extraction," in *Proc. Workshop Classification Events, Activities Relationships*, Apr. 2006.

[16] M. A. Koschat and D. F. Swayne, "A weighted procrustes criterion," *Psychometrika*, vol. 56, no. 2, pp. 229–239, 1991.

[17] I. Leichter, M. Lindenbaum, and E. Rivlin, "A generalised framework for combining visual trackers: The black boxes approach," *Int. J. Comput. Vision*, vol. 67, no. 3, pp. 343–363, 2006.

[18] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.

[19] L. Lin, K. Zeng, X. Liu, and S.-C. Zhu, "Layered graph matching by composite cluster sampling with collaborative and competitive interactions," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2009, pp. 1351–1358.

[20] S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[21] J. Shi and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 1. Jun. 1994, pp. 593–600.

[22] H. Stern and B. Efros, "Adaptive color space switching for face tracking in multi-colored lighting environments," in *Proc. IEEE Conf. Automat. Face Gesture Recog.*, vol. 1. May 2002, pp. 249–254.

[23] A. Smeaton, P. Over, and W. Kraaij, "Evaluation compaigns and TRECVid," in *Proc. ACM Int. Workshop Multimedia Inform. Retrieval*, 2006, pp. 321–330.

[24] Y. Wang and S.-C. Zhu, "Perceptual scale-space and its applications," *Int. J. Comput. Vision*, vol. 2, no. 4, pp. 259–362, Jan. 2006.

[25] Y. Wu, Z. Si, and S.-C. Zhu, "Deformable template as active basis," in *Proc. Int. Conf. Comput. Vision*, vol. 1. 2007, pp. 1–8.

[26] Y. Wu, S.-C. Zhu, and C.-E. Guo, "From information scaling of natural images to regimes of statistical models," *Quarter. Appl. Math.*, vol. 66, no. 1, pp. 81–122, 2008.

[27] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Survey*, vol. 38, no. 4, p. 13, Dec. 2006.

[28] M. Yang and Y. Wu, "Tracking non-stationary appearances and dynamic feature selection," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 2. Jun. 2005, pp. 1059–1066.

[29] M. Yang, J. Yuan, and Y. Wu, "Spatial selection for attentional visual tracking," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, vol. 1. Jun. 2007, pp. 1–8.

[30] B. Yao, X. Yang, and S.-C. Zhu, "Introduction to a large scale general purpose groundtruth dataset: Methodology, annotation tool, and benchmarks," in *Proc. IEEE Conf. Energy Minimiz. Methods Comput. Vision Patt. Recog.*, Aug. 2007, pp. 169–183.

[31] J. Yuan, J. Luo, and Y. Wu, "Mining compositional features for boosting," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, Jun. 2008, pp. 1–8.

[32] S.-C. Zhu, "Stochastic jump-diffusion process for computing medial axes in Markov random fields," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 21, no. 11, pp. 1158–1169, Nov. 1999.

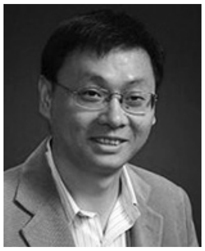
Xiaobai Liu has been pursuing the Ph.D. degree from the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China, since September 2006.

Since December 2008, he has been a Research Associate with Prof. S. Yan, Learning and Vision Group, National University of Singapore, Singapore. From September 2007 to December 2008, he was a Research Associate with the Lotus Hill Research Institute, Ezhou, China, under the supervision of Prof. S.-C. Zhu. He has published more than ten articles over a series of research topics. His current research interests include computer vision, machine learning, and large-scale image retrieval.

Liang Lin received the B.S. and Ph.D. degrees from the Beijing Institute of Technology (BIT), Beijing, China, in 1999 and 2008, respectively. From 2006 to 2007, he was a joint Ph.D. student with the Department of Statistics, University of California, Los Angeles (UCLA).

He was a Post-Doctoral Research Fellow with the Center for Image and Vision Science, UCLA. From 2007 to 2009, he was a Senior Research Scientist with the Lotus Hill Research Institute, Ezhou, China. He is currently an Associate Professor with the Software School of Sun Yat-Sen University, Guangzhou, China. He has published more than 30 academic papers. His current research interests include but are not limited to computer vision, pattern recognition, computer graphics, and virtual reality.

Dr. Lin has received a number of honors, including several scholarships while pursuing the Ph.D. degree, the Beijing Excellent Students Award in 2007, the Excellent Ph.D. Thesis of BIT in 2008, and the Best Paper Runners-Up Award in NPAR 2010.



Shuicheng Yan (M'06–SM'09) received the Ph.D. degree from the School of Mathematical Sciences, Peking University, Beijing, China, in 2004. He spent three years as a Post-Doctoral Fellow at the Chinese University of Hong Kong, Shatin, Hong Kong, and then at the University of Illinois at Urbana-Champaign, Urbana.

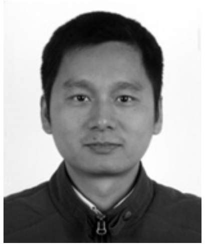
He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His current research interests include computer vision (biometrics, surveillance, and Internet vision), multimedia (video event analysis, image annotation, and media search), machine learning (feature extraction, sparsity/non-negativity analysis, and large-scale machine learning), and medical image analysis. He has authored or co-authored over 140 technical papers over a wide range of research topics.

Dr. Yan has served on the Editorial Board of the *International Journal of Computer Mathematics*. He has served as a Guest Editor of the special issue for *Pattern Recognition Letters*. Currently, he is the Guest Editor of the special issue for *Computer Vision and Image Understanding*. He has served as a Co-Chair of the IEEE International Workshop on Video-Oriented Object and Event Classification, in 2009, held jointly with ICCV, in 2009. He was the Special Session Chair of the Pacific-Rim Symposium on Image and Video Technology, in 2010. He is currently an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Hai Jin (M'98–SM'06) received the Ph.D. degree in computer engineering from the Huazhong University of Science and Technology (HUST), Hubei, China, in 1994.

He is a Professor of computer science and engineering with HUST. He is currently the Dean of the School of Computer Science and Technology at HUST. From 1998 to 2000, he was with the University of Hong Kong, Shatin, Hong Kong. From 1999 to 2000, he was a Visiting Scholar with the University of Southern California, Los Angeles. He is the Chief Scientist of ChinaGrid, the largest grid computing project in China. He has co-authored 15 books and published over 400 research papers. His current research interests include computer architecture, virtualization technology, cluster computing and grid computing, peer-to-peer computing, network storage, and network security.

Dr. Jin was awarded the German Academic Exchange Service Fellowship to visit the Technical University of Chemnitz, Chemnitz, Germany, in 1996. He was awarded the Excellent Youth Award from the National Science Foundation of China in 2001. He is a member of the ACM. He is the member of the Grid Forum Steering Group. He is the Chair of the Steering Committee of the International Conference on Grid and Pervasive Computing, Asia-Pacific Services Computing Conference. He is a member of the Steering Committee of the IEEE/ACM International Symposium on Cluster Computing and the Grid, the IFIP International Conference on Network and Parallel Computing, the International Conference on Grid and Cooperative Computing, the International Conference on Autonomic and Trusted Computing, and the International Conference on Ubiquitous Intelligence and Computing.



Wenbin Jiang received the Ph.D. degree in information and communication engineering from the Huazhong University of Science and Technology (HUST), Hubei, China, in 2004.

He is currently an Associate Professor with the School of Computer Science and Technology, HUST. He was with Aizu University, Fukushima, Japan, for research visiting in 2005. He has published about 40 research papers. His current research interests include multimedia, ubiquitous computing, data management, and so on.

Dr. Jiang has been the PC Chair, the Publicity Chair, and a PC Member of more than 50 international conferences.