Noname manuscript No.
(will be inserted by the editor)

Object Tracking via Dirichlet Process-based Appearance Models

Wayne State University

Raed Almomani · Ming Dong · Dongxiao Zhu

Received: date / Accepted: date

Abstract Object tracking is the process of locating objects of interest in video frames. Challenges still exist in handling appearance changes in object tracking for robotic vision. In this paper, we propose a novel Dirichlet Process-based Appearance Model (DPAM) for tracking. By explicitly introducing a new model variable into the traditional Dirichlet Process, we model the negative and positive target instances as the combination of multiple appearance models. Within each model, target instances are dynamically clustered based on their visual similarity. DPAM provides an infinite nonparametric mixture of distributions that can grow automatically with the complexity of the appearance data. In addition, prior off-line training or specifying the number of mixture components (clusters or parameters) is not required. We build a tracking system in which DPAM is applied to cluster negative and positive target samples and detect the new target location. Our experimental results on real-world videos show that our system achieves superior performance when compared with several state-of-the-art trackers.

Keywords Computer Vision · Object Tracking · Dirichlet Process · Appearance Model

Raed Almomani
Department of Computer Science
Wayne State University
Tel.: +1(313) 473-7370
E-mail: ec8951@wayne.edu

Ming Dong
Department of Computer Science
Wayne State University
Tel.: +1(313) 577-0725
E-mail: mdong@cs.wayne.edu

Dongxiao Zhu
Department of Computer Science
Wayne State University
Tel.: +1(313) 577-247
E-mail: dzhu@wayne.edu

1 Introduction

Object tracking is the process of locating objects of interest in video frames. Tracking systems are increasingly used in various applications such as surveillance, security and robotic vision. Tracking in robotic vision (i.e., with moving cameras) is considered more difficult than tracking with static camera videos as segmenting the foreground objects by background subtraction methods is not applicable. Although numerous approaches have been proposed for tracking a specific type of objects (e.g., humans [1], faces[2], rigid objects [3], mice [4]), robust tracking of a generic object is still a challenging problem in robotic vision research. Specifically, one of the major challenges comes from handling appearance variations caused by changes in scale, pose, illumination and occlusion during tracking [5].

Current tracking methods can be grouped in two main categories: discriminative and generative approaches [6–15]. Discriminative approaches deal with object tracking as a binary classification problem by finding the best location that separates the target from the background. The classifier can be built using off-line training. For example, Avidan [16] trained Support Vector Machine off-line and Lepetit et al. [17] trained randomized trees. The main problem with these methods is that a comprehensive training dataset that covers all appearance variations and different backgrounds is required beforehand. Other approaches applied adaptive classifiers where tracking results are used for classifier adaptation. To this end, Lim et al. [18] employed incremental subspace learning; Avidan [19] applied adaptive ensembles classifiers by constructed a feature vector for main frame pixels. Grabner and Bischof [20] used online boosting and Kalal et al. [21] applied bootstrapping binary classifiers. Babenko et al. [22] used online multiple instance learning and Williams et al. [23] sparse Bayesian learning. However, adaptive discriminative methods suffer from drifting caused by the accumulation of updating errors.

Generative approaches search in a video frame for the most similar location based on a target appearance model [24–27]. The previously observed target instances are used to learn the appearance model before adopting it to the current frame. Many generative methods employ static appearance models (e.g., randomized trees [3]). The training sets of static appearance models are collected manually or from the first frame only [1,3,28–31]. Generally, they are unable to cope with the sudden appearance changes, especially when prior knowledge about the target is limited. Subsequently, adaptive appearance models are proposed where a model is constantly updated during tracking [32–34]. Similar to the adaptive discriminative methods, adaptive generative approaches suffer from drifting.

In order to characterize appearance variations and handle drifting and occlusion problems, an appearance model should have the following desired properties. First, the capacity of the model should be adaptive to the appearance complexity. Second, the model should be built based on both initial target instances and online tracking results as complete off-line training is only applicable to very limited scenarios. Finally, for robust tracking, the performance of the model should not heavily rely on parameter tuning for each video.

In this paper, we propose a novel multiple appearance model based on Dirichlet Process (DP) to address the aforementioned challenges. Our method differs from

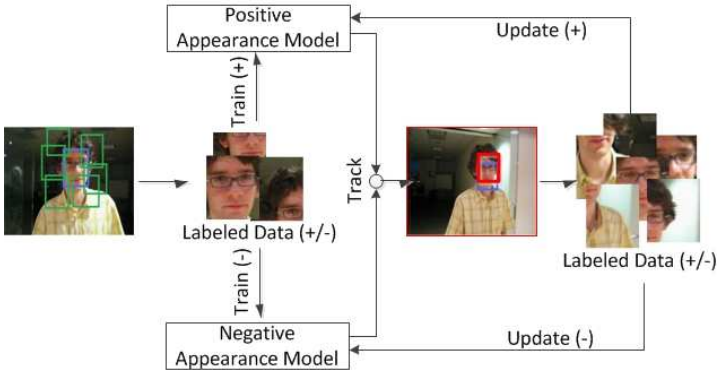

Fig. 1 Our system distributes target instances to positive and negative samples. Each group instances are clustered dynamically based on visual similarity.

the traditional DP by explicitly introducing a new model variable v, which categorizes the negative and positive target instances into different models. Within each model, target instances are dynamically clustered based on their visual similarity (see Fig. 1 for an illustrative example). DPAM provides an infinite nonparametric mixture of distributions that can grow automatically with the complexity of the appearance data. In addition, prior off-line training or specifying the number of mixture components (clusters or parameters) is not required. We build a tracking system in which DPAM is applied to cluster negative and positive target samples and detect the new target location. In our tracker, the target object can be arbitrarily chosen with no prior knowledge except its initial location in the first frame. Our experimental results on real-world videos show that our system can provide stable, robust tracking in complex scenes (e.g., with occlusions, illumination and pose variations) and achieve superior performance when compared with several state-of-the-art tracking systems.

The rest of this paper is organized as follows. We start with reviewing relevant works in Section 2. Section 3 describes DPAM, the model structure and the Bayesian decision in detail. Section 4 gives our tracking system. Section 5 presents the experiment results. Finally, Section 6 concludes.

2 Related works

Appearance modeling has been widely used in object tracking. In this section, we review related work in two categories: single appearance models and multiple appearance models.

In single appearance models, previously observed target instances are used to train the model, then the model is adapted to the current frame. Collins and Liu [25] utilized target instances to learn the discriminative color features that distinguish the target from the background. Aeschliman et al. [24] proposed a probabilistic frame-

work for joint segmentation and tracking. In [16] and [19], the target is represented by a binary classifier that is learned by Support Vector Machine and AdaBoost, respectively. Later on, Kalal et al. [21] used randomized trees. Grabner and Bischof [20] proposed an online boosting method to update an appearance model. Babenko et al. [22] applied online multiple instance learning to build a discriminative tracker. Zhou et al. [35] used SIFT features and mean shift. Godec et al. [36] built a tracking system by integrating hough forests with voting-based detection, back-projection and rough segmentation. He et al. [37] employed locality sensitive histogram to update the appearance model. Sevilla and Miller [38] used distribution fields to represent targets and images in tracking. However, due to the limitation of building only one appearance model that covers all target appearance changes, these methods update the model from subsets of the previous target instances [16, 19, 25] or the most recent ones [22, 20]. Therefore, they are intolerant of sudden appearance changes.

Multiple appearance models overcome the limitation by establishing several models and allowing each one to represent a specific target situation [39]. Kwon and Lee [40] decomposed the target appearance and motion into several models and assigned a tracker for each one. Kim et al. [41] trained and updated multiple classifiers to capture the changes of the appearance. Liu et al. [42] used the sparse representation to extract samples from the training set with minimal reconstruction errors. Avidan [19] combined multiple weak classifiers into a strong one. Han et al. [43] applied Kalman and Particle filters to evaluate the collected features from color and gradient orientation histograms. However, the performance of such models generally depends on the availability of comprehensive training sets and fine tuning of the model parameters for each video.

In this paper, we propose Dirichlet Process Appearance Model (DPAM) for object tracking, which is different from the aforementioned methods in several ways. First, the number of mixture components (clusters or parameters) is automatically determined based on the complexity of the appearance data. Thus, DPAM can be used to model various amounts of appearance changes and is widely applicable in object tracking. Second, DPAM is an online learning model that can handle significant and abrupt appearance variations during tracking. Finally, DPAM is a nonparametric method. Its performance does not depend on hand tuning of system parameters.

3 Dirichlet Process-based Appearance Model

In this section, we introduce DPAM in details. We begin with an overview of Dirichlet Process (DP) and our contribution and modification to the traditional DP model in Section 3.1, and the DPAM model structure in Section 3.2. Finally, the Bayesian decision in Section 3.3.

3.1 Dirichlet Process

Our goal is to learn a target appearance model during real time object tracking. Since the target data is unknown in advance, the capacity of the model should adapt to the

appearance complexity. So, we need a multiple-appearance model. According to De Finetti's theorem, the probability distribution of infinite exchangeable observations $\{x_1, x_2, ..., x_n\}$ is a mixture of probability distributions of these observations. That is [44],

$$p(x_1, x_2, ..., x_n) = \int_{\Theta} p(\theta) \prod_{i=1}^{n} p(x_i|\theta) d\theta, \tag{1}$$

where Θ is an infinite-dimensional mixture space of probability measures and $d\theta$ defines a probability measure over distributions.

Dirichlet Process (DP) [45] is a Bayesian nonparametric probabilistic model comes under De Finetti's theorem where a Dirichlet random variable θ with k-dimensionality have the property: $\theta_i \geq 0, \sum_{i=1}^{k} \theta_i = 1$. DP describes the distribution of θ with the following probability density:

$$DP(\alpha, \theta) = \frac{\Gamma(\Sigma_{i=1}^{k} \alpha_i)}{\Pi_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} ... \theta_k^{\alpha_k - 1}, \tag{2}$$

where the parameter α is a k-vector with components $\alpha_i > 1$ and Γ is the Gamma function.

As the number of clusters generally grows with the number of target instances, which is unknown in advance, an infinite DP is required where $k \to \infty$. The equations for the infinite DP are:

$$x_n \sim p(x|\theta_m), \tag{3}$$

$$\theta_m \sim G, \tag{4}$$

$$G \sim DP(\alpha, G_0), \tag{5}$$

where G_0 is the base distribution and α is the concentration parameter.

The advantage of using the infinite DP for target instance clustering over traditional clustering methods lies on the number of repetitions required to infer the number of clusters. The infinite DP automatically infers the number of clusters with a single repetition, while the traditional clustering methods need multiple repetitions to compare different hypotheses on the number clusters before determining the best one. Moreover, during testing, DP has the flexibility of allowing previously unseen data to form a new cluster.

The distribution over data partitions induced by DP is known as a Chinese Restaurant Process (CRP) [46]. CRP can potentially model an infinite number of mixture clusters regarding the input data, where each cluster can have infinite target's instances. If the target's instances $\{x_1, x_2, ..., x_n\}$ has occupied the clusters $\{\theta_1, ..., \theta_m\}$, when a new target's instance x_{n+1} comes, the probability of joining or creating a new cluster is given as:

$$p(x_{n+1} \in k | x_{1,...,n}, \alpha) = \left\{ \begin{array}{l} \frac{\alpha}{n+\alpha} \text{ if } k = \theta_{m+1} \\ \frac{L_k}{n+\alpha} \text{ if } k \in \theta_1, ..., \theta_m \end{array} \right\}, \tag{6}$$

where n is the total number of target instances, L_k is the number of target instances in cluster k and α the concentration parameter.

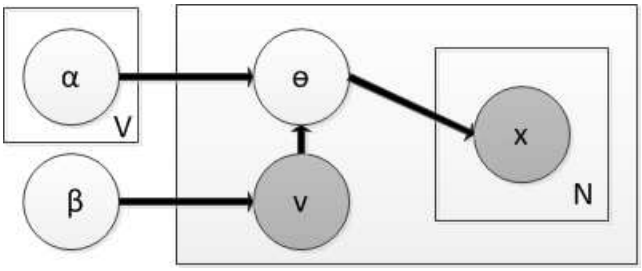

Fig. 2 Dirichlet Process-based Appearance Model (DPAM).

When used in tracking, CRP has the nice property where neither the number of clusters nor the number of target instances need to be known in advance. It can dynamically increase the number of clusters as data grows. In this paper, we propose a novel appearance model based on CRP to cluster the target instances and handle the appearance changes during tracking.

Generally, in order to detect the new location of the target, we need to model the appearance of both the target and its surrounding background. Thus, the proposed model, DPAM, includes two CRPs, one for positive samples and the other one for negative samples. Since DPAM could have more than one CRP model, Equation 6 is rewritten as follows:

$$p(x_{n+1} \in k | x^v_{1,...,n}, \alpha, v) = \left\{ \begin{array}{c} \frac{\alpha}{n^v+\alpha} \text{ if } k = \theta^v_{m+1} \\ \frac{L^v_k}{n^v+\alpha} \text{ if } k \in \theta_1, ..., \theta^v_m \end{array} \right\} \quad (7)$$

where v is a model variable that will be explained with details in the next section, n^v is the total number of target instances in model v, $\{\theta_1, ..., \theta^v_m\}$ are the clusters of model v, and L^v_k is the number of target instances in cluster k. When a new target instance comes, Equation 7 determines the order of the evaluation (joining an existing cluster or creating a new cluster). Specifically, it chooses the cluster with the highest number of images first. If the similarity is lower than the preset threshold, we move to the next highest cluster. The hyperparameter α is set to 1 to enforce our system to check all the existing clusters before creating a new one.

3.2 The Model Structure

Our appearance model is created based on CRP proposed by Aldous [46]. We differ from the traditional CRP by explicitly introducing a new model variable v to categorize data into different models. Here, we use v to indicate the positive (target) and negative (surrounding background) category, while in general, v can represent any desired categorization of the target. For example, we can build a model for each different object in multiple object tracking.

As shown in Fig.2, a feature vector x represents the target instance that is used as a base for clustering. A collection of N instances for the same tracked target is denoted by $X = \{x_1, x_2, ..., x_n\}$. Note that both x and v are shaded to indicate that they are observed variables.

In our model, the generative process of creating an object instance x is given in the following steps:

1. Choose the model variable label $v \sim p(v|\beta)$ for each instance, where $v = \{1, ..., V\}$, V is the total number of model variables and β is a dimensional vector of a multinomial distribution with length V.
2. Given the model variable label v, we draw a distribution by choosing $\theta^v \sim p(\theta|v, \alpha)$ for each instance, where θ is the parameter of a multinomial distribution for choosing the clusters; α is a $V \times Z$ matrix where V is the total number of model variables and Z is the total number of clusters under the model variables.
3. For each target instance:
   (a) choose cluster assignment $\theta_c \sim Mult(\theta^v)$
   (b) choose a target instance $x \sim p(x|\theta_c)$.

Given the parameters α and β, the generative equation can be known. The joint probability of an instance mixture θ, a set of N instances x and a model variable v is:

$$p(x, \theta, v|\alpha, \beta) = p(v|\beta)p(\theta|v, \alpha) \prod_{n=1}^{N} p(x_n|\theta) \tag{8}$$

$$p(v|\beta) = Mult(v|\beta) \tag{9}$$

$$p(\theta|v, \alpha) = \prod_{j=1}^{V} DP(\theta|\alpha_j)^{\delta(v,j)} \tag{10}$$

3.3 Bayesian Decision

In tracking, DPAM is employed to recognize a given target instance x. Specifically, the probability of the model variable v is computed as follow:

$$p(v|x, \alpha, \beta) \propto p(x|v, \alpha)p(v|\beta) \tag{11}$$

where $p(v|\alpha)$ is the probability of choosing a certain CRP, $p(x|v, \alpha)$ is the probability of choosing a certain cluster in that CRP, and α and β are parameters learned from the target's previously observed instance set. For convenience, the distribution of $p(v|\beta)$ is assumed to be a fixed uniform distribution: $p(v) = 1/V$, where V is the number of models. So, Equation 11 could be rewritten as,

$$p(v|x, \alpha, \beta) \propto p(x|v, \alpha). \tag{12}$$

The target recognition problem is solved by computing the maximal likelihood of (x) given the model variable v: $max_v p(x|v, \alpha)$. $p(x|v, \alpha)$ is obtained by,

$$p(x|v, \alpha) = max_D BD(x, y_d), \tag{13}$$

Algorithm 1 TRACKING SYSTEM

INPUT: The target location $\ell(x)^*_{t-1}$ in frame $t-1$.
OUTPUT: The target location $\ell(x)^*_t$ in frame t;

1. Extract the patches from the searching area $X = \{x|||\ell(x)^*_{t-1} - \ell(x)_t|| < \gamma\}$.
2. P-DPAM finds the top candidates that have highest probabilities $p(x|P-DPAM)$ (higher than a threshold ζ). If none of candidates is chosen, the full occlusion is considered. In full occlusion, patches are collected from the whole frame.
3. N-DPAM chooses a candidate from the top candidates that has the lowest probability $p(x|N-DPAM)$ to consider as new target location $\ell(x)^*_t$.
4. Extract the positive sample set: $X^P = \{x|||\ell(x)^*_t - \ell(x)_t|| < \eta\}$.
5. Extract the negative sample set: $X^N = \{x|\psi > ||\ell(x)^*_t - \ell(x)_t|| > \omega\}$.
6. Use X^P and X^N to update P-DPAM and N-DPAM, respectively.

where D is the number of clusters in v, y_d is the cluster centroid feature vector of cluster $d \in D$, and BD is the Bhattacharyya distance that is computed as:

$$BD(x,y) = \sqrt{1 - \frac{1}{\sqrt{\overline{xy}N^2}} \sum_I \sqrt{x(I).y(I)}}, \tag{14}$$

where N is the dimension of the feature vectors.

4 Tracking System

In this section, we introduce how DPAM is used to build the appearance model for positive and negative samples and track the target. The pipeline of our tracker is illustrated in Fig. 1, and the tracking procedure is summarized in Algorithm 1.

4.1 Target Tracking

The performance of the tracking system depends mainly on the effectiveness of the appearance model. In this paper, we build two models using DPAM, $P-DPAM$ for positive samples and $N-DPAM$ for the negative samples. In addition to the importance of choosing an effective appearance model, the method of choosing positive and negative samples when updating the appearance model is also important. In our system, we applied the most common technique for choosing positive and negative samples, in which the patch at the current tracker location is selected as the positive sample and the neighborhood samples around the current tracker location are considered as negative ones. Then, the positive sample is used to update $P-DPAM$ and the negative ones are used to update N-DPAM.

Before tracking starts, a user first chooses the target of interest $\ell(x)^*_t$. The system extracts the positive samples $x \in X^P$ within an integer radius η from the given target location $X^P = \{x|||\ell(x)^*_t - \ell(x)_t|| < \eta\}$. $\eta = 1$ gives only one positive sample while setting $\eta > 1$ provides multiple positive samples. For negative samples $x \in X^N$, the system extracts patches from an annular region surrounding the target location, defined by $X^N = \{x|\psi > ||\ell(x)^*_t - \ell(x)_t|| > \omega\}$, ψ and ω are parameters to

control the size of the region. The patches X^P and X^N are used to update P-DPAM and N-DPAM, respectively.

In tracking, our system finds the target location $\ell(x)_t^*$ in frame t by extracting and evaluating all patches $X = \{x|||\ell(x)_{t-1}^* - \ell(x)_t|| < \gamma\}$ that are within a search radius from the previous target location $\ell(x)_{t-1}^*$ in frame $t-1$. Based on Eq. 13, the appearance tracker first identify the candidate patches that have high probability belonging to the positive cluster $p(x|s = P - DPAM)$ (higher than a threshold ζ). Second, from the P-DPAM candidates patches, N-DPAM chooses a candidate that has lowest probability to belong to the negative appearance model $p(x|s = N - DPAM)$ to consider it as the new target location $\ell(x)_t^*$. If no patch is classified as positive patch in the first step, the target is considered fully occluded. In full occlusion, the entire frame will be used as the searching area. The detailed steps of tracking are given in Algorithm 1.

After the system detects the new tracker location, the system extracts the positive and negative samples. The positive samples are extracted according to $X^P = \{x|||\ell(x)_t^* - \ell(x)_t|| < \eta\}$ and the negative samples are extracted according to $X^N = \{x|\psi > ||\ell(x)_t^* - \ell(x)_t|| > \omega\}$. All the positive and negative samples are used to update $P - DPAM$ and $N - DPAM$, respectively. Since the number of clusters in $N - DPAM$ grows fast, we remove the cluster that is not updated for a certain number of frames.

4.2 Image Features

Image features, e.g., color and texture, that are sufficiently robust to changes are very important for appearance models. In general, using Gabor filter-based texture feature in tracking gives good results, but is computationally expensive. On the other hand, simple and fast-to-compute texture features (e.g., SIFT) do not provide accurate tracking results, especially in viewpoint change situations [47]. In the literature, global (from the entire bounding box of the object) and local (dividing the bounding box into sub-regions) Hue-Saturation-Value (HSV) color histograms are widely used as image descriptors in tracking systems due to their robustness and simplicity [48, 49]. Following these practices, in our system we define the target appearance as the composition of both global and local color histograms. As discussed in [50], human cognition about color is mainly based on hue (H), and then saturation (S), and finally value (V). Thus, we used 24 H channels, 12 S channels and 4 V channels. We chose a low bin number of V channel to reduce the influence of illumination changes [51–53]. So, a 40-bin [H=24, S=12, V=4] global HSV color feature is extracted from the entire bounding box of the target, In addition, the target is divided into 16 equal size sub-windows, and a 40-bin local color histogram is obtained from each. Thus, the final feature vector contains 17 40-bin HSV histograms.

5 Experiments

We evaluated our appearance model (DPAM) and tracking system on several challenging image sequences from PETS 2006 [54], AVSS 2007 [55], ViSOR [56] datasets

and publicly available video sequences[57]. These are challenging videos with multiple occlusions, pose variations, illumination and scaling changes.

Our tracking system is compared with several state-of-the-art trackers, i.e., Tracking-Learning-Detection (TLD) [21], Multiple Instance Learning (MIL) [22], Visual Tracking Decomposition (VTD) [40], Locality Sensitive Histogram (LSH) [37] and Distribution Field (DF) [38]. Specifically, TLD used positive and negative samples to train a online binary classifier; MIL used multiple instance learning to build a discriminative tracker; VTD applied both observation and motion models; LSH employed locality sensitive histogram to update the appearance model; and DF used distribution fields to represent targets and images in tracking.

In our comparison, either the binary or source codes for TLD, MIL, VTL, LSH and DF are obtained from their authors. The same initialization and default parameter settings are used in our evaluation. For our system, η (range of positive samples), ψ and ω (range of negative samples), γ (target search radius), and ζ (confidence threshold) are set to 1, 20, 5, 20, and 0.95, respectively, and used for all the video datasets evaluated in our experiments. These are the common parameters for any detection-based tracking algorithms. The values are typically selected based on the size of the target and the resolution of the video.

Our tracking system is implemented using OpenCV and C++ language on a machine that has a Quad (2.83GHz and 3.01GHz) processor and 4GB RAM. The performance of tracking is evaluated by using the mean center location errors between the tracking results and the ground truth. The center location error was computed for all the frames in which a method was able to return a target location. That is, full occlusion frames, when detected by an algorithm, will be excluded.

5.1 Evaluation of Clustering Results

In this section, we show the clustering performance of DPAM by comparing it with the Gaussian mixture model (GMM) and the mixture model of Kotz-type distributions (Kotz) using the Expectation Maximization (EM) algorithm. In both cases, EM starts from some initial estimate of model parameters and then proceeds to iteratively update them until finding the maximum likelihood. More specifically, GMM is a parametric probability density function represented as a weighted sum of Gaussian component densities. Kotz-type distribution has fatter tail regions compared to Gaussian distribution [58]. While Gaussian distribution is powerful in modeling rare tail events, which often represents data with low noise, Kotz-type distribution can be more amenable for modeling more frequent tail events, and thus may be more suitable for noisy data. The most general form of Kotz-type distribution is given by,

$$f(x,\mu,\Sigma) = c_p|\Sigma|^{-\frac{1}{2}}[(x-\mu)^T\Sigma^{-1}(x-\mu)]^{N-1}\exp\{-r[(x-\mu)^T\Sigma^{-1}(x-\mu)]^s\}, \tag{15}$$

where $c_p = \frac{s\Gamma(\frac{2}{p})}{\pi^{\frac{p}{2}}\Gamma(\frac{2N+p-2}{2s})} r^{\frac{2N+p-2}{2s}}$. In Kotz-type distributions, N, s and r are tuning parameters to modulate tail events, and p represents the dimension of the data. In practice, a number of special cases appeared in literature, such as in [59–62], in which N is routinely set to 1 for mathematical convenience. Since s and r are a pair of

Cluster #	2	3	4	5	6	7	8
GMM	1.2	1	1.38	1.22	1.25	1.16	1.29
Kotz (r=1)	1.23	1.02	1.47	0.97	1.15	1.16	0.81
Kotz (r=1/4)	1.13	1.02	1.38	1.3	1.15	1.16	1.08
DPAM	0.49	-	-	-	-	-	-

Cluster #	7	8	9	10	11	12	13
GMM	1.51	1.76	1.56	1.58	1.5	1.7	1.73
Kotz (r=1)	1.52	1.56	1.51	1.58	1.54	1.6	1.52
Kotz (r=1/4)	1.48	1.78	1.8	1.64	1.58	1.53	1.65
DPAM	-	-	-	0.38	-	-	-

Cluster #	4	5	6	7	8	9	10
GMM	1.5	1.49	1.5	1.6	1.63	1.69	1.5
Kotz (r=1)	1.49	1.53	1.55	1.52	1.52	1.52	1.49
Kotz (r=1/4)	1.49	1.49	1.56	1.44	1.58	1.54	1.55
DPAM	-	-	-	-	0.5	-	-

Fig. 3 Comparing clustering results between GMM, Kotz and DPAM.

covariates, we simply fix s at 1 and try a couple of different values of r around $\frac{1}{2}$, the Gaussian case, to examine the fat tail events and to demonstrate its impact on our results.

Sequence	TLD	VTD	MIL	LSH	DF	OUR
David	**12**	28	22	**12**	99	**11***
Dollar	65	75	22	**5**	80	**3***
Sylvester	57	13	**11**	17	31	**5***
Tiger 1	21	24	45	**13**	25	**8***
Surfer	**8***	25	12	15	99	**10**
Tiger 2	58	**13**	17	14	33	**12***
Twinning	30	11	**10***	14	37	**10***
Coke	17	**12**	34	31	34	**9***
Face Occluded 1	34	18	17	31	**11**	**10***

Table 1 The mean center location errors (pixels) between the tracking system results and their ground truth for the videos in Fig. 5. Bold and * indicate the best performance, and bold indicates the second best.

As a base for comparison, we used Davies-Bouldin Index (DBI):

$$DBI = \frac{1}{N}\sum_{i=1}^{N} max_{i\neq j}\left(\frac{M_i + M_j}{d(c_i, c_j)}\right), \tag{16}$$

where N is the number of clusters, c_k is the centroid of the k^{th} cluster, M_k is the average distance between all instances in k^{th} cluster and its centroid and $d(c_i, c_j)$ is the distance between the i^{th} and j^{th} cluster centroids. The clustering method that produces the smallest DBI value is considered the best.

Fig. 3 summarizes the comparison, in which three image sequences from PETS 2006, ViSOR and AVSS 2007 are used. As the number of clusters needs to be specified in GMM and Kotz, we run it with different number of clusters. Then, we run DPAM on the same image sequences where the number of clusters are automatically determined. Clearly, DPAM gives a higher performance in all three image sequences.

5.2 Tracking Results

In this section, we evaluated our tracker on matchmarking videos [57] and compared it with TLD, VTD, MIL, LSH and DF. The quantitative results are summarized in Table 1 and Fig. 4. Overall, our system provides the most accurate and robust tracking with average speed 20 fps on the 320*270 frame size.

Comparative tracking results of selected frames are presented in Fig. 5. Specifically, in Sylvester and David videos, the tracking results for the target under lighting, scale and pose changes are presented. Our tracker achieves the best performance compared with all other tracking systems. TLD and LSH provide the second best performance on David video, and MIL provides the second best performance on Sylvester video. Our system tracks the whole target in all video frames with high accuracy and robustness.

In Face Occluded 1 video, the main challenges are severe partial occlusion and appearance changes. DF achieved the second best performance on Face Occluded video because it is specifically designed to handle occlusion via distribution fields to represent targets and images in tracking. MIL achieved the third best performance on

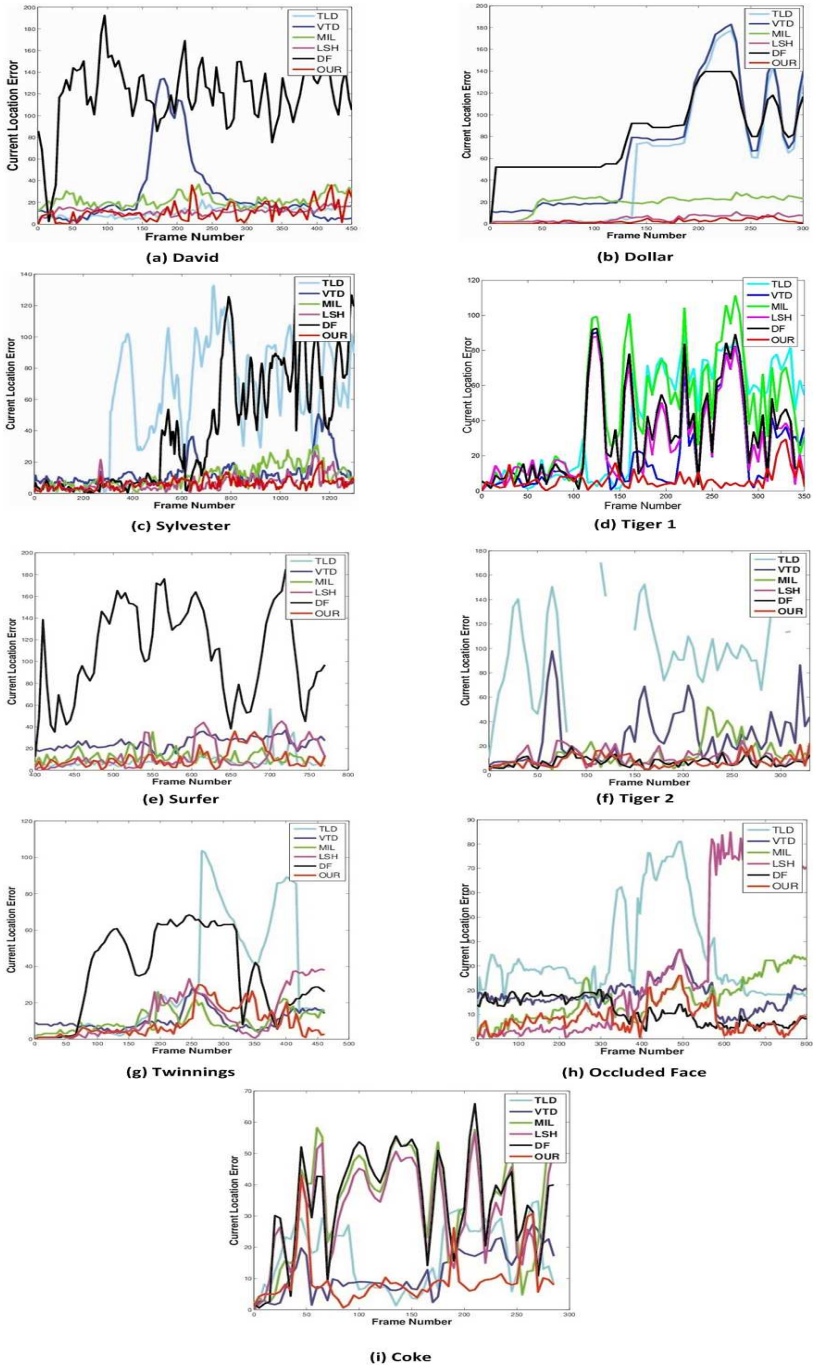

Fig. 4 The center location error plots.

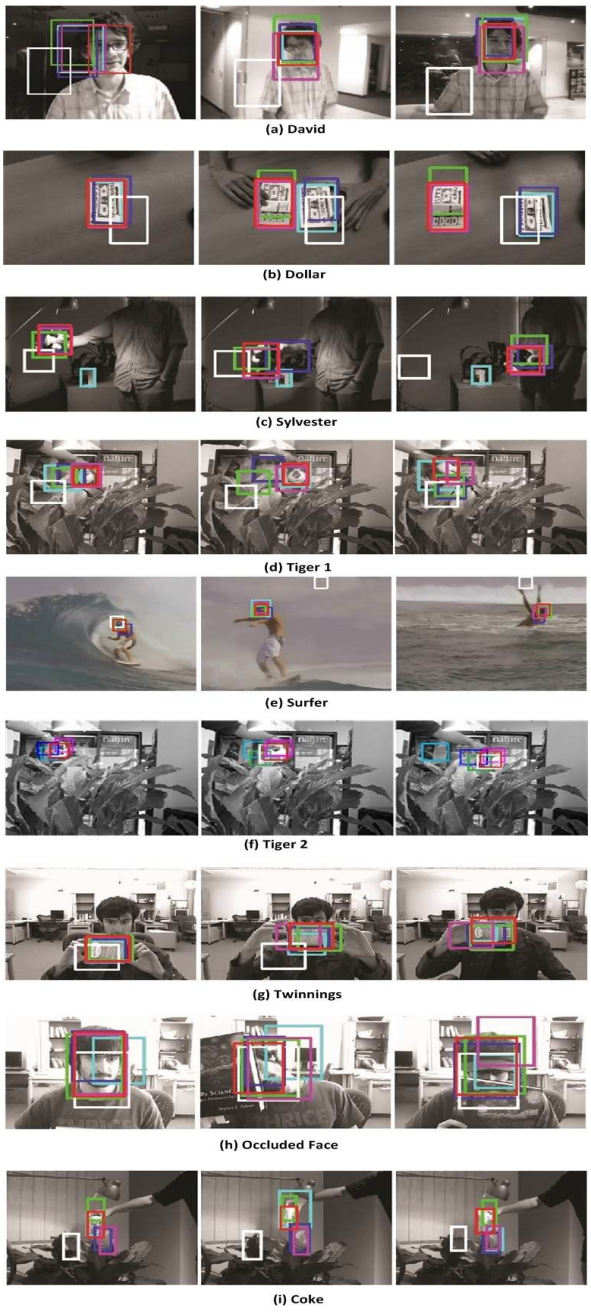

Fig. 5 Comparative tracking results of selected frames. The tracking results by TLD, VTD, MIL, LSH, DF, and ours, are represented by cyan, blue, green, magenta, white and red rectangles, respectively.

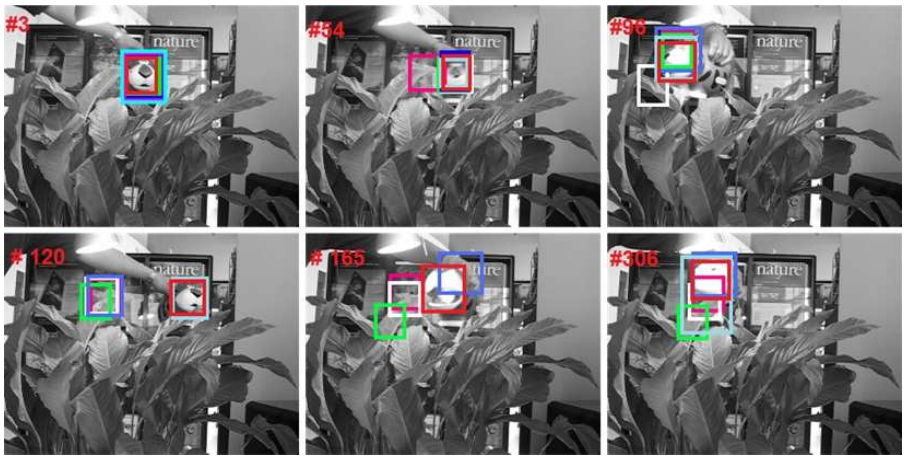

Fig. 6 Representative frames from different clusters of DPAM in Tiger 1 video. Different appearance changes are shown: view angle (54), scale (96), occlusion (120), appearance (165) and illumination (306) changes. The tracking results by TLD, VTD, MIL, LSH, DF, and DPAM, are represented by cyan, blue, green, magenta, white and red rectangles, respectively.

the video because it depends on patches during tracking. This highlights the advantages of using a dynamic appearance model. Obviously, our system tracked the target accurately in all situations and provides the most accurate and robust results.

In Tiger 1, Sylvester, David, Tiger 2 and Coke Can videos, the main challenges are appearance and pose changes, fast motion and frequent severe occlusions. In all videos, our system provides the best performance comparing with other systems because our tracker has the ability to create a new cluster for abrupt appearance or pose changes. In addition, our system keeps a target's previous appearance, which helps re-detect it after full or severe occlusion.

In Dollar video, two objects have exactly the same appearance, and thus presents a big challenge to track the right one. In Surfer video, the target is small and there is a pose and lighting changes. In Twinning video, the object appearance is changed totally. Again, our tracker achieved excellent tracking results on these three videos. The robustness of our system is clearly shown. TLD provides good performance on Surfer, but gets bad results when we have a big appearance change, i.e., in Twinning and Dollar videos. MIL provides similar performance as ours on Twinning video, and LSH provides the second best performance on Dollar Video.

5.3 A Case Study

In this section, we take a closer look at the tracking results of Tiger 1 video to clearly illustrate the advantage of DPAM. The video has 353 frames and many appearance changes due to heavy occlusion, scale change, 3D-rotation, and uneven illumination. Fig. 6 shows representative frames of the target appearance changes: view angle (frame 54), scale (frame 96), occlusion (frame 120), appearance (frame

165) and illumination (frame 306). These changes, in turn, produces many clusters in $P - DPAM$. Some representative frames in these clusters are shown in Fig. 6. Since determining the number of clusters in advance is generally not possible, DPAM provides a general model for tracking by growing dynamically with the complexity of the video.

MIL, VTD, LSH and DF all lost the target (the center location error is higher than 30) for the first time between frames 108 and 150 (frame 120 is used to show the tracking results), where the target changed its appearance (scale and rotation) during occlusion. TLD first lost the target on frame 165. The loss of tracking attributes to a couple of reasons. First, because of the fast appearance changes, KLT in TLD failed to track the target. Second, these methods do not keep the target's previous appearance. Their tracking mainly depends on the online updating of the corresponding classifier. In frame 165, background is added to the target box by TLD, leading to the misdetection in the subsequent frames. On the other hand, DPAM can handled sudden appearance changes by quickly creating a new cluster in $P - DPAM$. In addition, $P - DPAM$ explicitly keeps all the previous appearances of the target, which are used effectively for target detection. The percentages of correctly tracked frames are $46\%, 40\%, 70\%, 60\%, 55\%$ and 100% for LTD, MIL, VTD, LSH, DF and DPAM, respectively.

In our experiments, DPAM successfully tracks all the objects for the full length of each video sequence, which none of other trackers can achieve. Even when other methods track the target successfully, our method significantly improves the tracking accuracy, evidenced by the lowest average center location errors shown in Table 1.

6 CONCLUSION

In this paper, we propose a novel Dirichlet Process-based Appearance Model (DPAM) to handle target appearance changes during tracking. DPAM differs from the traditional Dirichlet Process by explicitly introducing a new model variable v, which categorizes the negative and positive target instances into different models and dynamically clusters them based on visual similarity. DPAM provides an infinite nonparametric mixture of distributions that can grow automatically with the complexity of the appearance data. In addition, prior off-line training or specifying the number of mixture components (clusters or parameters) is not required. Our tracking system with DPAM achieves superior performance when compared with several state-of-the-art trackers.

In the future, we plan to employ our model to more complicated tracking problems, e.g., multiple object tracking and deformable object tracking. Specifically, DPAM can be modified to have a model for the background and an individual model for each target to handle the target appearance changes. In this case, the number of DPAM models will grow automatically regarding the number of new targets, and the number of clusters in each model will grow regarding the target appearance changes.

References

1. M. Isard and J. MacCormick, "Bramble: A bayesian multiple-blob tracker," in *IEEE International Conference on Computer Vision*, pp. 34–41, 2001.
2. S. Birchfield, "Elliptical head tracking using intensity gradients and color histograms," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 232–237, 1998.
3. V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1465–1479, 2006.
4. K. Branson and S. Belongie, "Tracking multiple mouse contours (without too many samples)," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1039–1046, 2005.
5. G. Yu, Z. Hu, H. Lu, and W. Li, "Robust object tracking with occlusion handle," *Neural Computing and Applications*, pp. 1027–1034, 2011.
6. N. Wang, J. Wang, and D.-Y. Yeung, "Online robust non-negative dictionary learning for visual tracking," in *IEEE International Conference on Computer Vision*, pp. 657–664, 2013.
7. D. Chen, Z. Yuan, Y. Wu, G. Zhang, and N. Zheng, "Constructing adaptive complex cells for robust visual tracking," in *IEEE International Conference on Computer Vision*, pp. 1113–1120, 2013.
8. H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *European Conference on Computer Vision*, pp. 234–247, 2008.
9. X. Wang, G. Hua, and T. Han, "Discriminative tracking by metric learning," in *European Conference on Computer Vision*, pp. 200–214, 2010.
10. A. Li, F. Tang, Y. Guo, and H. Tao, "Discriminative nonorthogonal binary subspace tracking," in *European Conference on Computer Vision*, pp. 258–271, 2010.
11. R. Lefort, R. Fablet, and J. Boucher, "Weakly supervised classification of objects in images using soft random forests," in *European Conference on Computer Vision*, pp. 185–198, 2010.
12. R. Liu, J. Cheng, and H. Lu, "A robust boosting tracker with minimum error bound in a co-training framework," in *IEEE International Conference on Computer Vision*, pp. 1459–1466, 2009.
13. J. Santner, C. Leistner, A. Saffari, T. Pock, and H. Bischof, "Prost: Parallel robust online simple tracking," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 723–730, 2010.
14. H. Lu, Q. Zhou, D. Wang, and R. Xiang, "A co-training framework for visual tracking with multiple instance learning," in *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*, pp. 539–544, 2011.
15. T. Dinh and G. Medioni, "Co-training framework of generative and discriminative trackers with partial occlusion handling," in *IEEE Workshop on Applications of Computer Vision*, pp. 642–649, 2011.
16. S. Avidan, "Support vector tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1064–1072, 2004.
17. V. Lepetit, P. Lagger, and P. Fua, "Randomized trees for real-time keypoint recognition," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 775–781, 2005.
18. J. Lim, D. Ross, R. Lin, and M. Yang, "Incremental learning for visual tracking," in *Advances in neural information processing systems*, pp. 793–800, 2004.
19. S. Avidan, "Ensemble tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 261–271, 2007.
20. H. Grabner and H. Bischof, "On-line boosting and vision," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 260–267, 2006.
21. Z. Kalal, J. Matas, and K. Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 49–56, 2010.
22. B. Babenko, M. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 983–990, 2009.
23. O. Williams, A. Blake, and R. Cipolla, "Sparse bayesian learning for efficient visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1292–1304, 2005.
24. C. Aeschliman, J. Park, and A. Kak, "A probabilistic framework for joint segmentation and tracking," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1371–1378, 2010.
25. R. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1631–1643, 2005.
26. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 564–577, 2003.

27. X. Mei and H. Ling, "Robust visual tracking using l 1 minimization," in *IEEE International Conference on Computer Vision*, pp. 1436–1443, 2009.
28. G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1025–1039, 1998.
29. M. Black and A. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision*, pp. 63–84, 1998.
30. D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 142–149, 2000.
31. A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 798–805, 2006.
32. A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1296–1311, 2003.
33. L. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 810–815, 2004.
34. D. Ross, J. Lim, R. Lin, and M. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, pp. 125–141, 2008.
35. H. Zhou, Y. Yuan, and C. Shi, "Object tracking using sift features and mean shift," *Computer vision and image understanding*, pp. 345–352, 2009.
36. M. Godec, P. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," *Computer Vision and Image Understanding*, pp. 1245–1256, 2013.
37. S. He, Q. Yang, R. Lau, J. Wang, and M. Yang, "Visual tracking via locality sensitive histograms," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 2427–2434, 2013.
38. L. Lara and E. Learned-Miller, "Distribution fields for tracking," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1910–1917, 2012.
39. Q. Yu, T. B. Dinh, and G. Medioni, "Online tracking and reacquisition using co-trained generative and discriminative trackers," in *European Conference on Computer Vision*, pp. 678–691, 2008.
40. J. Kwon and K. M. Lee, "Visual tracking decomposition," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1269–1276, 2010.
41. T. Kim, T. Woodley, B. Stenger, and R. Cipolla, "Online multiple classifier boosting for object tracking," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1–6, 2010.
42. B. Liu, L. Yang, J. Huang, P. Meer, L. Gong, and C. Kulikowski, "Robust and fast collaborative tracking with two stage sparse optimization," in *European Conference on Computer Vision*, pp. 624–637, 2010.
43. Z. Han, Q. Ye, and J. Jiao, "Combined feature evaluation for adaptive visual object tracking," *Computer vision and image understanding*, pp. 69–80, 2011.
44. A. Cherian, V. Morellas, N. Papanikolopoulos, and S. Bedros, "Dirichlet process mixture models on symmetric positive definite matrices for appearance clustering in video surveillance applications," in *IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 3417–3424, 2011.
45. T. Ferguson, "A bayesian analysis of some nonparametric problems," *The annals of statistics*, pp. 209–230, 1973.
46. D. Aldous, "Exchangeability and related topics," *École d'Été de Probabilités de Saint-Flour*, pp. 1–198, 1985.
47. A. Ramisa, S. Vasudevan, D. Aldavert, R. Toledo, and R. L. de Mantaras, "Evaluation of the sift object recognition method in mobile robots," *International Conference of the Catalan Association for Artificial Intelligence*, pp. 56–73, 2009.
48. S.-I. Yu, Y. Yang, and A. Hauptmann, "Harry potter's marauder's map: Localizing and tracking multiple persons-of-interest by nonnegative discretization," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3714–3720, 2013.
49. D.-Y. Lee, J.-Y. Sim, and C.-S. Kim, "Visual tracking using pertinent patch selection and masking," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3486–3493, 2014.
50. Y. K. Jain and R. Yadav, "Content-based image retrieval approach using three features color, texture and shape," *International Journal of Computer Applications*, vol. 97, no. 17, 2014.
51. K. Nummiaro, E. Koller-Meier, and L. Van Gool, "Color features for tracking non-rigid objects," *ACTA Automatica Sinica*, vol. 29, no. 3, pp. 345–355, 2003.
52. J. Wang and Y. Yagi, "Integrating shape and color features for adaptive real-time object tracking," in *IEEE International Conference on Robotics and Biomimetics*, pp. 1–6, 2006.
53. B. Ogul and A. Temizel, "Person re-identification by combining features in a learning based framework," 2013.

54. "http://www.cvg.rdg.ac.uk/pets2006/data.html,"
55. "http://www.eecs.qmul.ac.uk/ãndrea/avss2007d.html,"
56. "http://www.openvisor.org,"
57. "http://vision.ucsd.edu/bbabenko/projectmiltrack.shtml,"
58. K. Plungpongpun and D. Naik, "Multivariate analysis of variance using a kotz type distribution," in *Proceeding of the World Congress on Engineering*, vol. 2, pp. 2–4, 2008.
59. K. Fang, S. Kotz, and K. Ng, "Symmetric multivariate and related distributions," Chapman&Hall, London, 1998.
60. E. Gómez, M. Gomez-Viilegas, and J. Marin, "A multivariate generalization of the power exponential family of distributions," *Communications in Statistics-Theory and Methods*, vol. 27, no. 3, pp. 589–600, 1998.
61. M. E. Johnson, *Multivariate statistical simulation: A guide to selecting and generating continuous multivariate distributions*. John Wiley & Sons, 2013.
62. D. N. Naik and K. Plungpongpun, "A kotz-type distribution for multivariate statistical inference," in *Advances in Distribution Theory, Order Statistics, and Inference*, pp. 111–124, 2006.