A BAYESIAN HIERARCHICAL APPEARANCE MODEL FOR ROBUST OBJECT TRACKING

Raed Almomani, Ming Dong and Dongxiao Zhu

Wayne State University
Computer Science Department
Detroit, MI 48202

ABSTRACT

In tracking, one of the major challenges comes from handling appearance variations caused by changes in scale, pose, illumination and occlusion. In this paper, we propose a novel Bayesian Hierarchical Appearance Model (BHAM) for robust object tracking. Our idea is to model the appearance of a target as a combination of multiple appearance models, each covering the target appearance changes under a given view angle. Specifically, target instances are modeled by Dirichlet Process and dynamically clustered based on their visual similarity. Thus, BHAM provides an infinite nonparametric mixture of distributions that can grow automatically with the complexity of the appearance data. We built an object tracking system by integrating BHAM with background subtraction and the KLT tracker. Our experimental results on real-world videos show that our system has superior performance when compared with several state-of-the-art trackers.

Index Terms— Computer Vision, Object Tracking, Appearance Model.

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. Although many numerous approaches provide promising results for tracking a specific object, providing a general tracking system is still a challenging problem. In tracking, handling appearance variations caused by changes in scale, pose, illumination and occlusion stands as one of the major challenges [1].

Current tracking methods can be grouped in two main categories: discriminative and generative approaches [2]. 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 [3] trained Support Vector Machines; Lepetit et al. [4] trained randomized trees; and Williams et al. [5] used sparse Bayesian learning. Later on, more sophisticated classifiers are employed. Kalal et al. [6] applied bootstrapping binary classifiers, and Babenko et al. [7] used online multiple instance learning. However, adaptive discriminative methods suffer from drifting caused by the accumulation of updating errors.

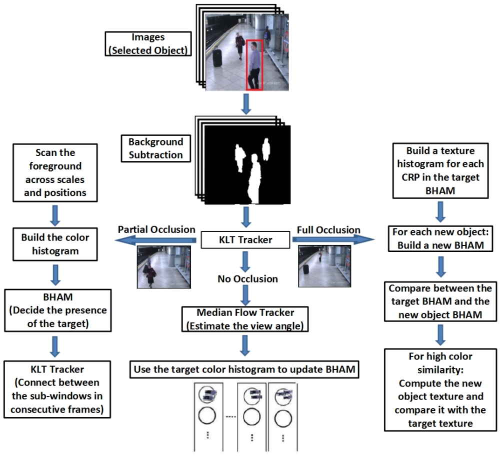


Fig. 1. The pipeline of our tracking system.

Generative approaches search in a video frame for the most similar location based on a target appearance model. The previously observed target instances are used to learn the appearance model before adopting it to the current frame. Many generative methods learn a static appearance model before adopting it to the current frame. The training sets of static appearance models are collected manually from the first frame only [8]. 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 [9]. Similar to the adaptive discriminative methods, adaptive generative approaches suffer from drifting as well.

In this paper, we propose a novel Bayesian Hierarchical Appearance Model (BHAM) to address these challenges. Our main idea is to model the appearance of an object as the combination of multiple appearance models, each covering the target appearance changes for a given view angle. Within each model, target instances are modeled by Dirichlet Process (DP) and dynamically clustered based on their visual similarity. BHAM differs 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, BHAM can be used to model various amounts of appearance changes and is widely applicable in object tracking. Second, BHAM is an online learning model that can handle significant and abrupt appearance variations during tracking. Finally, BHAM is a nonparametric method. Its performance does not depend on hand tuning of system parameters.

2. BAYESIAN HIERARCHICAL APPEARANCE MODEL

2.1. Chinese restaurant 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 grow with the data complexity. DP is a Bayesian nonparametric probabilistic model 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:

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

where the parameter α is a k-vector with components $\alpha_i > 1$ and Γ is the Gamma function [10].

The distribution over data partitions induced by DP is known as a Chinese Restaurant Process (CRP) [11]. CRP can potentially model an infinite number of mixture clusters regarding the input data, where each cluster can have infinite target's instances.

2.2. BHAM

Generally, tracking and re-identifying systems create a target appearance model by averaging feature vectors from all target instances. The accuracy of these systems are badly affected when the target instances are captured from different view angles. For example, for a person with a blue t-shirt and a red backpack, the feature vectors (e.g., color histogram) from front-facing camera instances are totaly different from back-facing camera instances. Thus, the proposed model, BHAM, includes eight CRPs, each representing the target's appearance in one of eight different view angles. To cover 360 degrees, forty five degree is adapted as the difference between two consecutive view angles and the upper-left corner of frames is considered as the zero degree. The target directions (view angles) can typically be determined according to a static point and other motion information [12].

Since the object observed from the same view angle usually shares similar texture, we introduce the hierarchical model, in which we pool all the target images under the same view and model them using one CRP. Subsequently, we can build one representative texture feature for each CRP, which achieves a good balance on the computation time and discriminative capability. In addition, the hierarchy also helps disentangle viewpoints from other factors such as illumination so that our model is more robust and accurate.

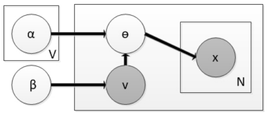


Fig. 2. Bayesian Hierarchical Appearance Model (BHAM).

As BHAM has eight CRPs, each of which could be written as follows:

$$p(x_{n+1} \in k | x^v_{1,...,n}, \alpha, v) = \left\{ \begin{array}{c} \frac{\alpha}{n^v + \alpha} \text{ if } k = \theta^v_{m+1} \\ \frac{L^v_k}{n^v + \alpha} \text{ if } k \in \theta_1, ..., \theta^v_m \end{array} \right\} \tag{2}$$

where n^v is the total number of target instances in model v, L^v_k is the number of target instances in model v in cluster k and $\{\theta_1, ..., \theta^v_m\}$ are the clusters of model v. When a new target instance comes, Equation 2 determines the order of the evaluation (joining an existing cluster or creating a new cluster).

In our method, we employ the median flow tracker [13] to compute the motion directions (2D view angles). To handle the appearance changes under the same view angle (e.g., illumination variations), a CRP is employed. The CRP is built based on the accumulated target's instances over time. Instances are clustered into different groups regarding the appearance similarity and Equation 2. The averaging of the feature vectors in each group represents the group center.

For a new target instance, features are extracted as a vector (A). The median flow tracker is used to determine the view angle model, and Equation 2 is used to select a cluster within the model with the highest probability. Then, the similarity (Bhattacharyya distance) between the new instance and the cluster center (B) is computed,

$$BD(A, B) = \sqrt{1 - \frac{1}{\sqrt{\bar{A}\bar{B}N^2}} \sum_I \sqrt{A(I).B(I)}}, \tag{3}$$

where N is the dimension of the feature vectors. If the similarity is beyond a threshold, the cluster is updated to incorporate the new instance (e.g., the number of instances in the cluster and the cluster center). The model will be updated as well (e.g., the total number of instances).

2.3. Model structure

Our appearance model is created based on CRP proposed by Aldous [11]. We differ from the standard CRP by explicitly introducing a new variable (view angle) for classification.

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 x is shaded to indicate that it is an observed variable.

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

1. Choose the view angle label $v \sim p(v|\beta)$ for each instance, where $v = \{1, ..., V\}$, V is the total number of view angles and β is a dimensional vector of a multinomial distribution with length V.

2. Given the view angle label v, we draw samples from 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 view angles and Z is the total number of clusters under the view angles.

3. For each target instance:

   (a) choose cluster assignment $\theta_c \sim \mathrm{Mult}(\theta^v)$,

   (b) choose a target instance $x \sim p(x|\theta_c)$.

Given the observed 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 view angle v is:

$$p(x, \theta, v|\alpha, \beta) = p(v|\beta)p(\theta|v, \alpha) \prod_{n=1}^{N} p(x_n|\theta), \qquad (4)$$

$$p(v|\beta) = Mult(v|\beta), \qquad (5)$$

$$p(\theta|v, \alpha) = \prod_{j=1}^{V} DP(\theta|\alpha_j)^{\delta(v,j)}. \qquad (6)$$

In tracking, BHAM is employed to recognize the target in partial and full occlusions. In these cases, decisions are made based on either an instance of the target (partial occlusion) or a collection of instances of a newly tracked target (full occlusion) by maximum a posteriori probability (MAP) estimate.

3. TRACKING WITH ONLINE BHAM

BHAM is a general object tracking method that can be used to track many kinds of object. As an example, in this section we introduce in details a BHAM-based pedestrian tracking system. The overview of the tracker is shown in Fig. 1.

3.1. Image features

Image features that are sufficiently robust to changes, such as self-occlusion and illumination, are very important for an appearance model. Here, we define the target appearance as composition of two kinds of features: a global color feature: Hue Saturation Value (HSV) histogram, and a local texture-based feature computed by Schmid and Gabor filters. These features are extracted from torso and legs of a pedestrian.

3.2. Target tracking

In our system, BHAM is applied to detect the target during partial occlusion and recognize the target after full occlusion. After a user selects a target, background subtraction is applied to segment the moving objects from the background (i.e., a mixture of Gaussians) as the first step. Then, KLT features connect between blobs in consecutive frames. KLT features detect partial occlusion when a blob in the current frame is matched to more than one blob in the previous frame. All previous blobs that cannot be matched in the current frame are considered as fully occluded.

During tracking, the median flow tracker is applied to estimate the view angle of the target. BHAM clusters the same view angle instances in one group as an intermediate step and each resulting group is divided into different subgroups based on the appearance similarity as the final step. If the view angle of the target is estimated incorrectly and the target instance is misclassified accordingly, the noisy target instance will most likely be grouped to a new cluster in the corresponding view model as it is an outlier in that model. We will remove these noisy clusters with a low number of samples from our model so that they will not adversely affect the appearance model and the tracking results.

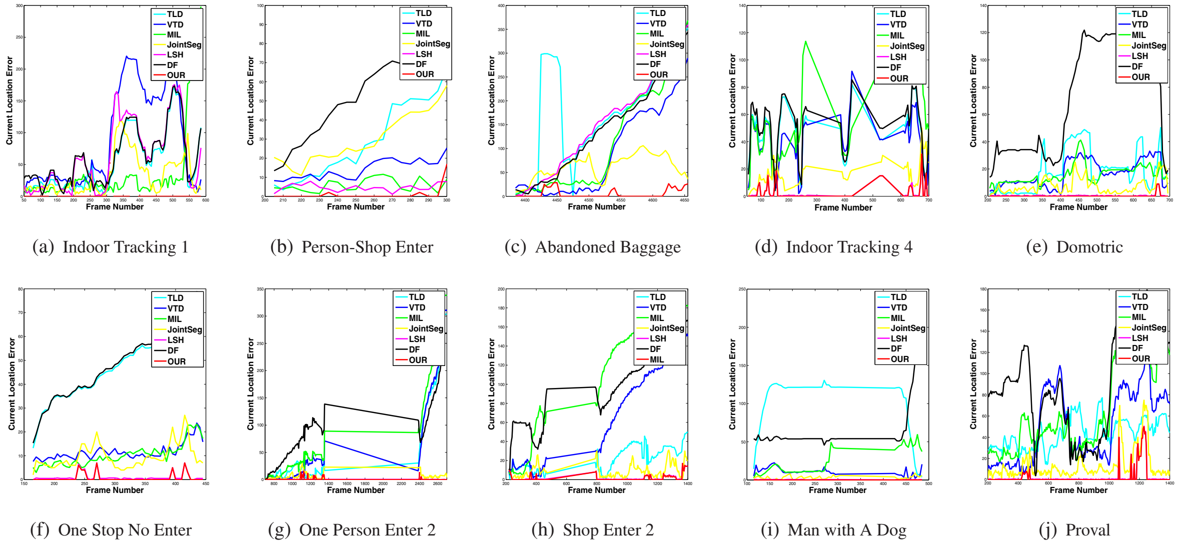
Based on the hierarchical model, both global (color histograms) and local features (texture histograms) are extracted. Specifically, the average HSV histogram is obtained for each cluster based on all the instances in the cluster as it can be computed very quickly. On the other hand, due to the high computation complexity, the average texture histogram is built only at the higher level (view angle level) based on the representative samples in the corresponding CRP (one from each cluster). In addition, it is only computed in full occlusion cases.

During partial occlusion, the foreground in the input image is scanned across positions and scales by applying the fast scanning window strategy [14]. At each sub-window an 80-bin color histogram is built and sent to BHAM to decide about the presence of the object. For sub-windows with the probability higher than a threshold, a KLT tracker is used to connect between the sub-windows in the previous frame and in the current one.

In full occlusion, our system builds a BHAM for each new object (KLT features are used to track all the objects to distinguish between old and new ones) after tracks it for a certain number of frames (e.g., 10). This gives us a more accurate appearance model than single instance-based methods. First, the color similarity between the new object and the target is computed based on the Bhattacharyya distance. Only for a new object with high color similarity, we further computes

Table 1. The average center location errors (pixels) between the tracking results and the corresponding ground truth for the videos in Fig. 4. Red indicates the best performance and blue indicates the second best.

Sequence	JSeg	TLD	VTD	MIL	LSH	DF	OUR
Indoor Tracking 1	60	186	99	123	161	150	9
Person-Shop Enter	54	62	71	70	60	62	5
Abandoned Baggage	54	62	71	70	62	61	5
Indoor Tracking 4	16	46	39	62	3	50	2
Domotric	8	19	16	15	2	72	1
One Stop No Enter	10	48	12	10	2	49	1
One Person Enter 2	7	58	59	77	2	83	1
Shop Enter 2	9	22	72	115	5	100	2
Man with A Dog	3	102	11	24	1	67	1
Proval	12	43	49	64	3	87	2



(a) Indoor Tracking 1

(b) Person-Shop Enter

(c) Abandoned Baggage

(d) Indoor Tracking 4

(e) Domotric

(f) One Stop No Enter

(g) One Person Enter 2

(h) Shop Enter 2

(i) Man with A Dog

(j) Proval

Fig. 3. The center location errors for videos from the AVSS, CAVIAR and ViSOR datasets

its texture and compares it with the target. In this way, color features are used to quickly rule out the dissimilar objects so that the computing of the expensive local texture features can be avoided. Finally, the object with a high similarity on both color and texture is recognized as the target for continuous tracking.

4. EXPERIMENTS

We evaluated our appearance model (BHAM) and tracking system on several challenging image sequences from AVSS 2007 [1], CAVIAR [2] and ViSOR [3] datasets. These are challenging videos with multiple interaction targets, occlusions, pose variations, illumination and scaling changes. We compared our tracking system with several state-of-the-art trackers, i.e., Tracking-Learning-Detection (TLD), Joint Segmentation (JointSeg), Multiple Instance Learning (MIL), Visual Tracking Decomposition (VTD), Locality Sensitive Histogram (LSH) and Distribution Field (DF) [15],[16] and [17]. In our comparison, either the binary or source codes for TLD, JointSeg, MIL, VTL, LSH and DF are obtained from their authors. The same initialization and default parameter settings are used in our evaluation. BHAM is implemented using OpenCV and C++ language on a machine that has a Quad (2.83GHz and 3.01GHz) processor and 4GB RAM. The average speed for BHAM is 23 fps with 320*270 frame size.

We manually labeled the ground truth center of each object every 5 frames for all the video sequences that are used in our experiments. The performance of tracking is evalu-

[1] http://www.eecs.qmul.ac.uk/ãndrea/avss2007d.html
[2] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1
[3] http://www.openvisor.org

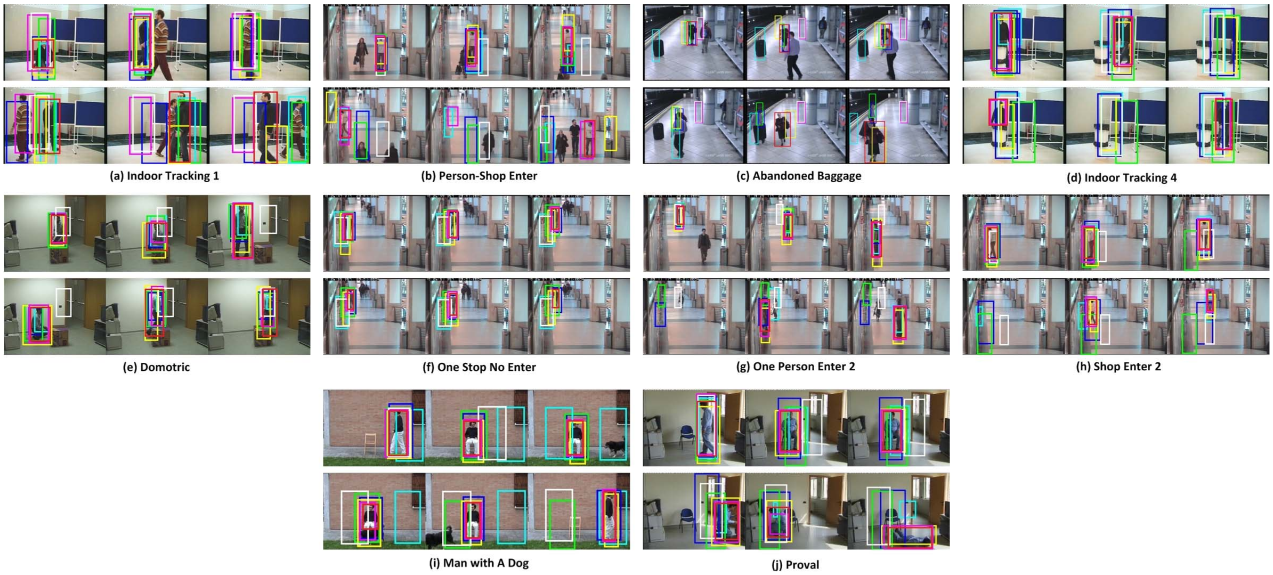

(a) Indoor Tracking 1
(b) Person-Shop Enter
(c) Abandoned Baggage
(d) Indoor Tracking 4
(e) Domotric
(f) One Stop No Enter
(g) One Person Enter 2
(h) Shop Enter 2
(i) Man with A Dog
(j) Proval

Fig. 4. Comparative tracking results of selected frames. The tracked target is highlighted by different colors: TLD (cyan), VTD (blue), MIL (green), JointSeg (yellow), LSH (magenta), DF (white) and our system (red).

ated only in the labeled frames by using the mean center location errors between the tracking results and the ground truth. The error is reported for the frames in which a method was able to track the target and is summarized in Table 1 and Fig. 3. Overall, our system provides the most accurate and robust tracking.

Comparative tracking results of selected frames from AVSS, CAVIAR, ViSOR datasets are presented in Fig. 4. Specifically, in the video of Indoor Tracking 1 from IEEE ViSOR 2007, the tracking results for the target under severe partial and full occlusion, scale and pose changes are presented. TLD, VTD, MIL, LSH, DF and JointSeg give false detections and track the wrong target or a part of the target in many scenarios while BHAM tracks the whole target in all the video frames. In addition, TLD, VTD, MIL, LSH and DF frequently fail to recognize the target after occlusion while JointSeg and BHAM recognize the target with high accuracy and robustness. Obviously, BHAM tracks the target accurately in all different situations and gives the most accurate and robust results.

Person-Shop Enter and Abandoned Baggage videos are from the CAVIAR and IEEE AVSS 2007 datasets. The main challenges are severe partial and full occlusion and appearance changes. VTD, MIL and JointSeg fail to detect and track the target during and after partial occlusion. TLD shows a bad target detection during the partial occlusion and a very well recognition after the full occlusion. BHAM tracks the target successfully during the partial occlusion and gives accurate target recognition after the occlusion. TLD, VTD, MIL and LSH give false detection during the target full occlusion while JointSeg and BHAM identify the full occlusion and stop the tracking. DF fails totally to track the target. Finally, all tracking system except our method failed to re-identify the target after the full occlusion. BHAM gives the highest accuracy before, during and after the occlusion.

Indoor Tracking 4 and Domotric videos are from the ViSOR dataset. TLD, VTD, MIL and DF fail to stop tracking the target during full occlusion, while LSH and BHAM detect the full occlusion correctly. Some trackers can detect the target during partial occlusion, however, the detection is not very accurate as parts from the background or other objects are included. One Stop No Enter, One Person Enter 2, and Shop Enter 2 videos are from the CAVIAR dataset. Those three videos are challenging due to scale and appearance changes during the target tracking. DF, MIL and VTD lost the target, while TLD and JointSeg tracked the target with a high detection error. LSH and BHAM tracked the target nicely all the time. In addition, they stop tracking the target during the full occlusion.

Finally, Proval video is also from the CAVIAR dataset. The target changes his pose many times in both the video. TLD, DF and MIL do not recognize the target correctly in the video. Frequently, they track the background or part of the target instead of the actual target. LSH and BHAM track the target with high accuracy. The robustness of our system is clearly shown. In our experiments, BHAM successfully tracks all 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 center location error shown in our experiments.

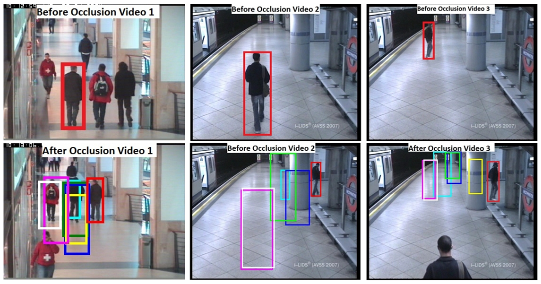


Fig. 5. Recognizing targets after full occlusion. The systems are TLD (cyan), VTD (blue), MIL (green), JointSeg (yellow), LSH (magenta), DF (white) and our system (red).

4.1. Tracking during full occlusion

Fig. 5 shows the comparison between TLD, VTD, MIL, JointSeg, LSH, DF and BHAM on full occlusion scenarios. The first video has multiple objects presented and some of them have a similar appearance to the target. The second video has only one object (the target) and the target changes his direction during full occlusion. The third video has only one object (the target) and the target does not change his direction during the full occlusion. In all cases, the objects are recognized and tracked nicely in BHAM while other methods fail.

In our experiments, BHAM successfully tracks all 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 center location error shown in our experiments.

5. CONCLUSION

In this paper, we propose a novel Bayesian Hierarchical Appearance Model (BHAM) to handle target appearance changes during tracking. Our tracking system with BHAM shows superior performance when compared with several state-of-the-art trackers. In the future, we plan to employ our model to solve more complicated tracking problems, e.g., multiple object tracking and deformable objects tracking.

6. REFERENCES

[1] Alper Yilmaz, Omar Javed, and Mubarak Shah, "Object tracking: A survey," *ACM Computing Surveys*, pp. 1–45, 2006.

[2] Thang Ba Dinh and Gérard Medioni, "Co-training framework of generative and discriminative trackers with partial occlusion handling," in *WACV*, 2011, pp. 642–649.

[3] Shai Avidan, "Support vector tracking," *TPAMI*, pp. 1064–1072, 2004.

[4] Vincent Lepetit, Pascal Lagger, and Pascal Fua, "Randomized trees for real-time keypoint recognition," in *CVPR*, 2005, pp. 775–781.

[5] Oliver Williams, Andrew Blake, and Roberto Cipolla, "Sparse bayesian learning for efficient visual tracking," *TPAMI*, pp. 1292–1304, 2005.

[6] Zdenek Kalal, Jiri Matas, and Krystian Mikolajczyk, "Pn learning: Bootstrapping binary classifiers by structural constraints," in *CVPR*, 2010, pp. 49–56.

[7] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie, "Visual tracking with online multiple instance learning," in *CVPR*, 2009, pp. 983–990.

[8] Vincent Lepetit and Pascal Fua, "Keypoint recognition using randomized trees," *TPAMI*, pp. 1465–1479, 2006.

[9] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang, "Incremental learning for robust visual tracking," *IJCV*, pp. 125–141, 2008.

[10] Anoop Cherian, Vassilios Morellas, Nikolaos Papanikolopoulos, and Saad J Bedros, "Dirichlet process mixture models on symmetric positive definite matrices for appearance clustering in video surveillance applications," in *CVPR*, 2011, pp. 3417–3424.

[11] David Aldous, "Exchangeability and related topics," *École d'Été de Probabilités de Saint-Flour*, pp. 1–198, 1985.

[12] Imran Saleemi, Lance Hartung, and Mubarak Shah, "Scene understanding by statistical modeling of motion patterns," in *CVPR*, 2010, pp. 2069–2076.

[13] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas, "Forward-backward error: Automatic detection of tracking failures," in *ICPR*, 2010, pp. 2756–2759.

[14] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, 2001, pp. 511–518.

[15] Chad Aeschliman, Johnny Park, and Avinash C Kak, "A probabilistic framework for joint segmentation and tracking," in *CVPR*, 2010, pp. 1371–1378.

[16] Shengfeng He, Qingxiong Yang, Rynson WH Lau, Jiang Wang, and Ming-Hsuan Yang, "Visual tracking via locality sensitive histograms," in *CVPR*, 2013, pp. 2427–2434.

[17] Laura Sevilla Lara and Erik G Learned-Miller, "Distribution fields for tracking," in *CVPR*, 2012, pp. 1910–1917.