Fixed-lag parameter learning for visual tracking

S. Hernández
Laboratorio Geoespacial
Universidad Católica del Maule
shernandez@ucm.cl

Abstract

In this paper, we propose an approach to learn parameters for robust visual object tracking. Rather than using pre-trained models with off-line features, Bayesian parameter inference using Markov Chain Monte Carlo and particle filtering is performed. Particle Marginal Metropolis-Hastings (PMMH) has been proposed in the literature for batch parameter inference in linear Gaussian multi-target tracking systems. Instead, a fixed-lag parameter inference algorithm for tracking a single target from discrete visual features is analyzed. Robustness of the method is measured in terms of re-initialization rates and frames-per-second, and we conclude that the algorithm is well suited for visual tracking used in a wide variety of scenarios.

1. Introduction

Visual tracking is an important task in applications requiring video analysis, such as surveillance, robotics, human-computer interaction and medical diagnostic among others. Essentially, the problem consists of estimating the state (e.g. position and velocity) of a moving target from a sequence of images. Broadly speaking, visual tracking can be divided into three main components : object detection, prediction of the target state and updating the estimation using current measurements. These subtasks have a natural interpretation as model-based inference problem such as the state-space representation [10].

Traditional tracking systems based on the state-space representation use a fixed model (consisting of a dynamic and observation model) with known parameters, however this approach is prone to fail when the object suffers from appearance changes [15]. Sources of variation may differ from shape deformation, posture changes and changing scene conditions. Therefore, updating strategies based on on-line learning have been proposed in order to cope with variations of the original model [11].

Furthermore, long-term tracking systems are required to track arbitrary objects from an initial bounding box without relying on prior information about the object [2]. Recent approaches for this task combine tracking, learning and detection in order to develop a continuous adaption framework. However, these methods couple supervised learning with tracking and therefore cannot be used to sequentially update the tracking system [6].

1.1. Related Work

Pose, shape and dynamic deformations are considered as intrinsic variations while occlusions, camera motion and illumination changes are regarded as extrinsic variation. On-line and off-line learning is a key feature for handling these variations in an adaptive fashion, however these methods may also introduce extra variation and lead to drifting [15].

A generative approach for adapting to appearance and velocity changes within the standard state-space model was presented in [16]. Object appearance was modeled as Gaussian mixture from image intensities and model updating is performed with the Expectation-Maximization algorithm.

Parameter learning for state space models has been receiving increased attention in the signal processing community. An offline Markov Chain Monte Carlo (MCMC) method for tracking and learning multiple targets with non-linear and non-Gaussian likelihoods was proposed in [1, 14]. The MCMC algorithm uses continuous (dynamic and appearance model) and discrete variables (data association model), however sampling is carried out in a reduced model by analytically integrating the continuous variables.

In the context of visual tracking, discrete features are important because they lead to robust appearance representations [4]. MCMC methods were developed for adapting both continuous and discrete variables without an explicit detection mechanism [12]. The method uses frame differencing and therefore is only restricted to static camera scenarios.

In this paper, we use a standard color observation model and a constant velocity model and develop an MCMC methodology for adapting all model parameters. Similar to the aforementioned MCMC approaches, our method is able to process several images in batch so parameter learning is

performed off-line. However, an on-line parameter estimation method is devised using a few MCMC steps using a small batch of measurements (fixed-lag) within the recursive state estimation.

2. State Space Model

Let $\mathbf{x}_k \in \mathbb{R}^{n_x}$ be the state of the target at time k and $\mathbf{y}_k \in \mathbb{R}^{n_y}$ the observed measurements. In this case, we consider a generative model in the form of:

$$\begin{aligned} \theta &\sim p(\theta_x, \theta_y) \\ \mathbf{x}_0 &\sim p(\mathbf{x}_0) \\ \mathbf{x}_k &\sim p(\mathbf{x}_k|\mathbf{x}_{k-1}, \theta_x) \\ \mathbf{y}_k &\sim p(\mathbf{y}_k|\mathbf{x}_k, \theta_y) \end{aligned}$$

Given a set of batch measurements $\mathbf{y}_{1:T}$, we would like to estimate the target states $x_{1:T}$ given the parameter θ. Using Bayes rule, we can calculate the full posterior distribution:

$$p(\mathbf{x}_{0:T}, \theta|\mathbf{y}_{1:T}) \propto p(\mathbf{y}_{1:T}|\mathbf{x}_{0:T}, \theta)p(\mathbf{x}_{0:T}|\theta)p(\theta)$$

Traditionally, in visual tracking problems it is assumed that the parameter vector θ is known and we only need to estimate recursively the states $\mathbf{x}_{0:T}$. This approach, makes it difficult to cope with appearance changes that are commonly solved using heuristic model update strategies. Instead, we are interested in updating the model using the marginal distribution:

$$p(\theta|\mathbf{y}_{1:T}) = \int p(\mathbf{x}_{0:T}, \theta|\mathbf{y}_{1:T}) d\mathbf{x}_{0:T} \tag{1}$$

2.1. Particle MCMC

Due to its latent structure, direct parameter estimation using the marginal likelihood (Equation 1) is not feasible. Instead, we can estimate θ using the marginal posterior distribution $p(\theta|\mathbf{y}_{1:T}) \propto p(\mathbf{y}_{1:T}|\theta)p(\theta)$, leading to the following energy function:

$$\psi(\theta) = -\log p(\mathbf{y}_{1:T}|\theta) - \log p(\theta) \tag{2}$$

The above energy function could be used for parameter learning using *maximum a posteriori* (MAP) estimation or Bayesian inference using MCMC, however these approaches require analytic marginalization of the states $\mathbf{x}_{0:T}$. Sequential Monte Carlo (SMC) techniques such as particle filtering and smoothing are standard tools for recursively approximating integrals and therefore can be used to perform inference in general state space models. However, performing inference in both the states and the parameters of the model is a complex problem that requires to calculate intractable integrals. Nevertheless, it has been shown that standard SMC methods can be also used to build a proposal distributions for MCMC models, such that:

$$\hat{p}(\theta|\mathbf{y}_{1:T}) = \sum_j^n \frac{1}{N} p(\mathbf{x}_{0:T}^j, \theta|\mathbf{y}_{1:T}) \tag{3}$$

Algorithm 1 can be used for batch parameter estimation:

Algorithm 1 Batch parameter estimation

Require: $\{\mathbf{x}_0, \theta^0, \mathbf{y}_{1:T}, M, N\}$
$\{w_T^j, \mathbf{x}_{0:T}^j; j = 1, \ldots, n\} \sim p_{\theta^0}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$
$\hat{p}_{\theta^0}(\mathbf{y}_{1:T}) = \prod_{k=1}^T \hat{p}(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \theta^0)$
for $i = 1 : M$ **do**
  $\theta^* \sim q(\theta^*|\theta^{i-1})$
  $\{w_T^j, \mathbf{x}_{0:T}^j; j = 1, \ldots, n\}^* \sim p_{\theta^*}(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$
  $\hat{p}_{\theta^*}(\mathbf{y}_{1:T}) = \prod_{k=1}^T \hat{p}(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \theta^*)$
  $\alpha = \min\left\{1, \frac{p_{\theta^*}(\mathbf{y}_{1:T})p(\theta^*)q(\theta^{i-1}|\theta^*)}{p_{\theta^{i-1}}(\mathbf{y}_{1:T})p(\theta^{i-1})q(\theta^*|\theta^{i-1})}\right\}$
  $u \sim \mathcal{U}(0, 1)$
  if $u \leq \alpha$ **then**
    $\theta^i \leftarrow \theta*$
  else
    $\theta^i \leftarrow \theta^{i-1}$
  end if
end for

3. Fixed-lag parameter estimation

Now we are interested in on-line parameter estimation. Since at any time $k < T$ we don't have access to the complete data, the estimation problem becomes more cumbersome than batch MCMC. If we now consider a fixed-lag $L \leq k$, a sequential estimation procedure could target $p(\theta|\mathbf{y}_{1:L})$ and use MCMC steps within SMC [3]. Algorithm 2 can be used for on-line parameter estimation:

It is worth noting that for large fixed-lag windows, the memory requirements may increase exponentially over time.

3.1. Numerical Example

In order to illustrate the convergence results of the fixed-lag parameter estimation technique, we use a 1-dimensional non-linear state space model specified by:

$$\begin{aligned} x_{k+1} &= 0.5x_k + 25x_k/(1 + x_k^2) + 8\cos(1.2 * k) + v_k \\ y_k &= 0.05x_k^2 + e_k \end{aligned}$$

Algorithm 2 Fixed-lag parameter estimation

Require: $\{\mathbf{x}_0, \theta^0, \mathbf{y}_{1:T}, M, N, L\}$
$\{w_0^j = 1/n, \mathbf{x}_0^j; j = 1, \ldots, n\} \sim p_{\theta^0}(x_0)$
for $k = 1 : T$ **do**
  $\mathbf{x}_k \sim p(\mathbf{x}_k, \mathbf{x}_{k-1}, \theta^{k-1})$
  $\{w_k^j, \mathbf{x}_{0:k}^j; j = 1, \ldots, n\}^k \sim p_{\theta^k}(\mathbf{x}_{1:k}|\mathbf{y}_{1:k})$
  $\hat{p}_{\theta^k}(\mathbf{y}_{1:L}) = \prod_{k=1}^{L} \hat{p}(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \theta^k)$
  for $i = 1 : M$ **do**
    $\theta^* \sim q(\theta^*|\theta^{k-1})$
    $\{w_k^j, \mathbf{x}_{0:k}^j; j = 1, \ldots, n\}^* \sim p_{\theta^*}(\mathbf{x}_{1:k}|\mathbf{y}_{1:k})$
    $\hat{p}_{\theta^*}(\mathbf{y}_{1:L}) = \prod_{k=1}^{L} \hat{p}(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \theta^*)$
    $\alpha = \min\left\{1, \frac{p_{\theta^*}(\mathbf{y}_{1:L})p(\theta^*)q(\theta^k|\theta^*)}{p_{\theta^k}(\mathbf{y}_{1:L})p(\theta^k)q(\theta^*|\theta^k)}\right\}$
    $u \sim \mathcal{U}(0, 1)$
    if $u \leq \alpha$ **then**
      $\theta^k \leftarrow \theta*$
    end if
  end for
end for

where $v_k \sim \mathcal{N}(0, q)$ and $e_k \sim \mathcal{N}(0, r)$ are Gaussian noise sources with unknown parameters $\theta = \{q = 0.1, r = 1\}$ with independent Inverse gamma priors $\mathcal{IG}(0.1, 0.1)$. Batch and fixed-lag estimation techniques are compared with different number of particles N and MCMC iterations M. The results over 100 different runs are summarized in Table 1.

	N	M	$\hat{r}$	$\hat{q}$	RMSE	Time[sec]
Online ($L = k$)	500	100	0.18	1.17	1.16	142.85
Online ($L = 3$)	500	100	0.37	1.59	2.21	14.26
Batch	500	3000	0.12	1.10	0.53	85.17

Table 1. Numerical results for batch and online (fixed-lag) parameter estimation. A fixed-lag $L = k$ uses all data up to time k and therefore increases computation time. Instead, online parameter estimation using only a small lag $L = 3$ makes use of a non-sliding window to calculate the acceptance probability of the estimates.

4. On-line parameter estimation for visual object tracking

Visual tracking relies on object appearance representations that can be regarded as generative or discriminative approaches. Generative appearance models are based on probabilistic reasoning and require training data in order to learn the conditional densities to be used in the state estimation problem.

Generative appearance models used in visual tracking have traditionally been based on color [13] and texture [9] templates which are difficult to update. Instead, we use a generative model that is suitable for on-line parameter learning [4].

4.1. Dynamic model

The object model used in our approach uses an area approximation and a constant velocity model of a rectangle that bounds the target of interest. At time k, the object is represented as $\mathbf{x}_k = \{x_k, z_k, \dot{x}_k, \dot{z}_k, w_k, h_k\}$, where (x, z) represents the left-most pixel of the bounding box, $(\dot{x}_k, \dot{z}_k)$ the constant velocity on each one of the axis and (w_k, h_k) the width and height of the bounding box. The dynamic model can be written as:

$$x_k = x_{k-1} + \dot{x}_{k-1} + \mathcal{N}(0, \sigma_x^2)$$
$$z_k = z_{k-1} + \dot{z}_{k-1} + \mathcal{N}(0, \sigma_z^2)$$

The velocities $(\dot{x}_k, \dot{z}_k)$ are independently sampled from a Gaussian distribution $\mathcal{N}(0, \sigma_v^2)$ while the width and height of the bounding box are kept constant $(w_k, h_k) = (w_0, h_0)$. Consequently, the parameters of the dynamic model can be written as $\theta_x = (\sigma_x, \sigma_z, \sigma_v, w_0, h_0)$.

This simple model cannot be used when the target is under velocity, aspect-ratio or size changes. Nevertheless, introducing some extra parameters the model can be modified in order to cope with those variations.

4.2. Observation model

Since gradient orientation histograms are expensive to compute, we only rely on color histograms for our observation model. In the proposed approach, an initial bounding box is used to generate a multinomial distribution that represents the H and S channels of the color histogram in the HSV color space.

The multinomial distribution specifies the probability of observing a vector with discrete elements $\mathbf{y} = (y_1, \ldots, y_{N_b})$, containing counts on each one of the N_b bins of the color histogram. The likelihood of a color histogram given certain parameter vector $\boldsymbol{\theta}_y = (\theta_1, \ldots, \theta_{N_b})$ can therefore be written as:

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{n!}{\prod_i y_i!} \prod_i \theta_i^{y_i} \tag{4}$$

where $n = \sum_i y_i$ correspond to the total number of pixels in the ROI.

This model has been extensively used within the natural language processing community for text analytics. For the image classification task, it has been recently reported that better discriminative power can be achieved when the histogram counts are normalized [7].

4.3. Model updating

SMC techniques work well when all parameters θ of the model are known, but this approach is largely sub-optimal

when the problem consists of jointly estimating the states of several targets (multi-target) and the model parameters [8, 5].

In this paper, we concentrate on the single target case with unknown parameters $\boldsymbol{\theta}$. Algorithm 2 is used to sequentially sample both the target states and model parameters, according to:

$$q(\theta_x^*|\theta_x^{i-1}) = \mathcal{N}(\theta_x^{i-1}, \mathbf{I}_x)$$
$$q(\theta_y^*|\theta_y^{i-1}) = \Big(\frac{\mathcal{G}(\theta_y^{i-1}[1], \beta)}{\sum_j \mathcal{G}(\theta_y^{i-1}[j], \beta)}, \ldots, \frac{\mathcal{G}(\theta_y^{i-1}[N_b], \beta)}{\sum_j \mathcal{G}(\theta_y^{i-1}[j], \beta)}\Big)$$

where $\mathcal{G}(\theta_y^{i-1}[j], \beta)$ is a sample from a Gamma distribution whose shape parameter is taken from the observed color histogram. It's important to notice that since the support of the proposal distribution $q(\theta_x^*|\cdot)$ is larger than the target distribution, some samples must be discarded.

5. Experimental Results

In order to demonstrate the advantages of the proposed model updating approach over other discriminative approaches, we evaluate the tracking results on the Visual Object Challenge 2015 (VOT) dataset. This dataset contains several sequences containing moving camera, target pose, velocity and scale changes, varying illumination conditions and occlusions among other difficulties [1].

Table 2 summarizes the static parameters used for initialization:

parameter	value
σ_x^2	0.1
σ_y^2	0.1
σ_v^2	0.1
N_b	9
α	$(1/N_b, \ldots, 1/N_b)$

Table 2. Model parameters

In order to quantitatively analyze the performance the proposed approach, we use two commonly used evaluation metrics. A pixel-wise comparison between the annotated ground truth and the estimated object state is performed on each frame and the average precision $\bar{P}$ and recall $\bar{R}$ are calculated as follows:

$$\bar{P} = \frac{1}{|F|}\sum_i^{|F|} \frac{|TP|_i}{|TP|_i + |FP|_i}$$
$$\bar{R} = \frac{1}{|F|}\sum_i^{|F|} \frac{|TP|_i}{|TP|_i + |FP|_i}$$

where $|F|$ is the total number of frames in the sequence, and $|TP|_i$ and $|FP|_i$ correspond to $i-th$ frame true positive and false positives.

Evaluation is performed according to the VOT challenge protocol and the PMMH learning method is tested against a baseline tracker (particle filter without parameter learning). Table 3 shows the parameters used for the tracking system.

parameter	value
N	100
M	3
L	3
β	1

Table 3. Static tracker parameters

Both trackers are initialized using the first frame of each sequence and are allowed to re-initialize when the overlap between the ground-truth and the estimate is lower than a threshold $t = 0.1$. Table 4 shows the average performance measure for 15 runs of each method on the VOT2015 dataset.

method	accuracy	recall
pmmh	0.57	0.57
tracker	0.59	0.58

Table 4. Average accuracy and recall for the PMMH and the baseline tracker.

The system was tested on a server equipped with a Intel(R) Xeon(R) CPU E5-2620 v3 2.40GHz. Both trackers are implemented in C++, without resorting to any parallel or thread-level optimizations. In all evaluations, all images and ground-truth data are pre-loaded into memory and the running time of the algorithm is measured in terms of number of frames per second (FPS) (see Figure 1).

In order to evaluate the posterior distribution of the state and observation parameters, PMMH requires complete rejuvenation of the particle paths up to a fixed-lag $L = 3$. In terms of processing speed, the baseline method clearly outperforms the proposed approach. However, if we measure robustness in terms of the number of re-initializations, we can see the advantage of using PMMH (see Figure 2).

Figures 4 and 6 show the number of re-initializations required to complete the sequences : *bolt1*,*bolt2*,*fish1* and *fish2*.

[1] http://www.votchallenge.net/vot2015/

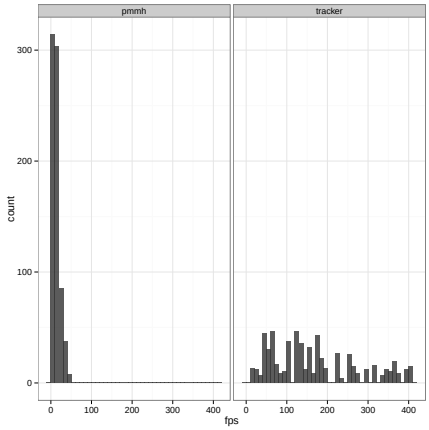

Figure 1. Frames per second (fps) for PMMH and baseline tracker. Compared to the baseline tracker (right), PMMH tracking (left) shows increased complexity.

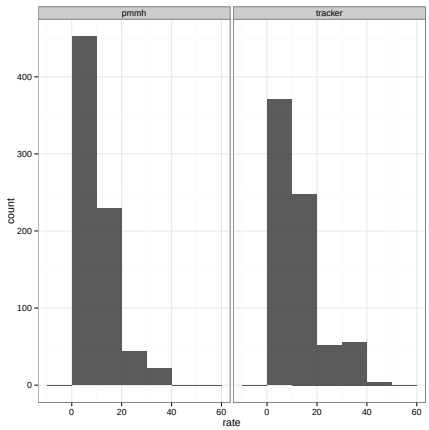

Figure 2. Robustness of each method is measured in terms of the number of re-initializations required to complete the sequence (re-initialization rate). Although PMMH adds complexity to the baseline tracker, target tracking with parameter learning requires less re-initializations.

Figure 3. Frame # 100 of the bolt1 sequence. Ground truth is shown in green and estimate is shown in red.

Table 5 shows the overall results on the VOT2015 dataset averaged over 15 different runs.

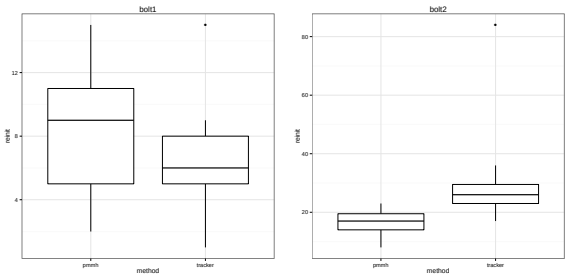

Figure 4. Number of re-initializations required to complete *bolt1* and *bolt2* sequences

Figure 5. Frame # 65 of the fish1 sequence. PMMH (left image) is able to better adapt to shape and appearance changes.

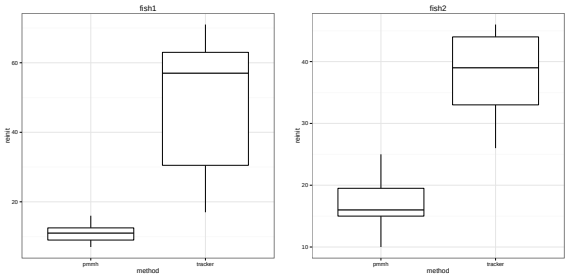

Figure 6. Number of re-initializations required to complete *fish1* and *fish2* sequences

6. Acknowledgments

This work was supported by CONICYT/FONDECYT grant, project **Robust Multi-Target Tracking using Discrete Visual Features**, code 11140598.

7. Conclusions

A Bayesian approach for parameter inference in single target tracking systems was presented. The method achieves improved performance in several tracking scenarios when compared with a baseline particle filter with manually tuned parameters. Algorithm complexity and poor MCMC convergence can be seen as the major drawbacks of the proposed approach. Future work will take into account these issues and will compare the results with other state-of-the-art tracking systems.

	pmmh		tracker	
sequence	fps	rate %	fps	rate %
bag	4.89	5.48	59.89	11.02
ball1	17.40	18.03	105.00	16.43
ball2	31.43	19.51	41.00	21.95
basketball	8.25	5.78	128.89	8.79
birds1	15.92	30.05	248.60	34.08
birds2	9.54	3.30	140.14	2.56
blanket	17.53	1.13	225.00	1.90
bolt1	17.68	2.42	291.67	1.89
bolt2	17.49	5.64	293.00	10.13
book	12.41	14.70	175.00	14.02
butterfly	8.11	5.83	134.22	5.65
car1	7.49	12.17	110.12	20.58
car2	31.77	4.46	393.00	6.81
crossing	7.06	7.79	111.35	7.94
dinosaur	7.44	4.97	115.46	6.02
fernando	3.47	11.53	51.17	16.12
fish1	16.92	2.99	268.40	13.01
fish2	12.94	5.48	161.89	12.04
fish3	14.97	1.95	236.43	7.49
fish4	15.28	4.83	215.97	7.80
girl	5.42	1.82	89.29	0.53
glove	13.47	14.83	120.00	17.29
godfather	19.79	4.21	329.40	4.15
gymnastics2	3.28	5.19	44.50	10.21
gymnastics3	1.49	7.01	21.58	7.29
gymnastics4	4.02	3.48	54.55	3.38
hand	16.58	9.51	267.00	12.16
handball1	35.42	13.79	377.00	18.45
handball2	20.40	14.93	402.00	26.80
helicopter	2.99	4.17	41.15	3.61
leaves	14.91	20.74	63.00	21.59
marching	4.10	6.83	60.30	10.15
matrix	13.97	12.80	100.00	12.50
motocross1	4.56	13.68	68.33	14.33
motocross2	1.26	2.95	18.64	0.91
nature	3.16	4.77	43.94	6.55
pedestrian1	22.00	11.43	140.00	13.78
pedestrian2	12.96	8.74	194.09	3.74
rabbit	28.79	32.11	158.00	34.90
racing	5.37	6.24	71.07	6.79
road	14.17	10.84	241.80	12.54
shaking	9.99	8.80	129.78	11.82
sheep	18.60	3.24	251.00	4.30
soccer1	9.62	7.79	139.38	8.21
soldier	5.28	7.39	96.60	12.75
sphere	5.79	15.39	82.63	19.70
tiger	8.77	8.77	127.75	9.21
traffic	16.32	9.42	191.00	5.05
tunnel	22.02	17.16	312.00	10.21
wiper	19.03	15.01	316.64	16.88

Table 5. Average performance (fps) and robustness (re-initialization rate) of PMMH and baseline tracker on the VOT2015 dataset.

References

[1] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

[2] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619–1632, Aug 2011.

[3] W. R. Gilks and C. Berzuini. Following a moving target : Monte carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146, 2001.

[4] S. Hernandez and M. Hernandez. Likelihood function for multi-target color tracking using discrete finite mixtures. In *Advances in Artificial Intelligence–IBERAMIA 2014*, pages 182–193. Springer, 2014.

[5] L. Jiang, S. Singh, and S. Yildirim. Bayesian tracking and parameter learning for non-linear multiple target tracking models. *Signal Processing, IEEE Transactions on*, PP(99):1–1, 2015.

[6] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.

[7] T. Kobayashi. Dirichlet-based histogram feature transform for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3278–3285, June 2014.

[8] J. Kokkala and S. Sarkka. Combining particle MCMC with rao-blackwellized monte carlo data association for parameter estimation in multiple target tracking. *Digital Signal Processing*, 47:84 – 95, 2015.

[9] W.-L. Lu, K. Okuma, and J. J. Little. Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image and Vision Computing*, 27(1):189–205, 2009.

[10] E. Maggio and A. Cavallaro. *Video tracking: theory and practice*. John Wiley & Sons, 2011.

[11] I. Matthews, T. Ishikawa, and S. Baker. The template update problem. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):810–815, 2004.

[12] W. Neiswanger, F. Wood, and E. Xing. The dependent Dirichlet process mixture of objects for detection-free tracking and object modeling. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 660–668, 2014.

[13] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Computer visionECCV 2002*, pages 661–675. Springer, 2002.

[14] T. Vu, B.-N. Vo, and R. Evans. A particle marginal metropolis-hastings multi-target tracker. *Signal Processing, IEEE Transactions on*, 62(15):3953–3964, Aug 2014.

[15] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, Nov. 2011.

[16] S. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. *Image Processing, IEEE Transactions on*, 13(11):1491–1506, Nov 2004.