Visual Tracking of Multiple Targets by Multi-Bernoulli Filtering of Background Subtracted Image Data

Reza Hoseinnezhad[1], Ba-Ngu Vo[2], and Truong Nguyen Vu[3]

[1] RMIT University, Victoria, Australia, rezah@rmit.edu.au
[2] The University of Western Australia, WA, Australia, ba-ngu.vo@uwa.edu.au
[3] Vietnam Academy of Science and Technology, Ho Chi Minh City, Vietnam.

Abstract. Most visual multi-target tracking techniques in the literature employ a detection routine to map the image data to point measurements that are usually further processed by a filter. In this paper, we present a visual tracking technique based on a multi-target filtering algorithm that operates directly on the image observations and does not require any detection nor training patterns. Instead, we use the recent history of image data for non-parametric background subtraction and apply an efficient multi-target filtering technique, known as the multi-Bernoulli filter, on the resulting grey scale image data. In our experiments, we applied our method to track multiple people in three video sequences from the CAVIAR dataset. The results show that our method can automatically track multiple interacting targets and quickly finds targets entering or leaving the scene.

Keywords: visual tracking, Bayesian estimation, multi-target tracking, random finite sets, multi-Bernoulli

1 Introduction

Single-view visual tracking techniques invariably consist of *detection* followed by *filtering*. A detection module generates point measurements from the images in the video sequence which are then utilised as inputs by a filtering module, which estimates the number of targets and their states (properties such as location and size). Detection is an integral part of single-view visual tracking techniques. There is a large body of literature on models and techniques for detecting targets based on various background and foreground models. One of the most popular approaches is the detection of targets based on matching colour histograms of rectangular blobs [1, 2]. Other recent methods include a game-theoretic approach [3], using human shape models [4, 5], multi-modal representations [6], sample-based detection [7], range segmentation [8] and a multi-step detection scheme including median filtering, thresholding, binary morphology and connected components analysis [9].

Detection compresses the information on the image into a finite set of points measurements, and is efficient in terms of memory as well as computational requirements. However, this approach may not be adequate when the information loss incurred in the detection process becomes significant. Another problem with using detection is the selection of a suitable measurement model for the filtering algorithm. Modelling the detection process in a computationally tractable manner is a difficult problem. In practice, the selection of the measurement model is done on an ad-hoc basis and requires the manual tuning of model parameters.

Using random finite set (RFS) theory, a tractable framework for tracking multiple targets from video data without detection was recently introduced in [10]. This work led to a novel method for tracking multiple targets in video and has been successfully demonstrated on sport players tracking [11]. However, this method requires prior information about the visual appearance of the targets to be tracked, and is most useful in cases where visual target model is available either a priori or from training data. In many applications, such as people surveillance, there is no prior information about the visual appearance of the targets and a new algorithm is needed.

This paper presents a novel algorithm that tracks multiple moving targets directly from the image without any training data. Our proposed algorithm gradually learns and updates a probabilistic background model based on kernel density estimation. This resulting background model is then substracted to generate a grey scale foreground image from which the multi-target posterior distribution can be computed analytically using the multi-Bernoulli update of [10]. A sequential Monte Carlo implementation of the multi-Bernoulli filter is detailed and demonstrated through case studies involving people tracking in video sequences.

2 Background

In the context of jointly estimating the number of states and their values, the collection of states, referred to as the *multi-target state*, is naturally represented as a *finite set*. The rationale behind this representation traces back to a fundamental consideration in estimation theory–estimation error, see for example [10]. Since the state and measurement are treated as realisations of random variables in the Bayesian estimation paradigm, the finite-set-valued (multi-target) state X is modelled as a *random* finite set (RFS). Mahler's Finite Set Statistics (FISST) provides powerfull yet practical mathematical tools for dealing with RFSs [12], [13], based on a notion of integration and density that is consistent with the well-established point process theory [14]. FISST has attracted substantial interest from academia as well as the commercial sector with the developments of the Probability Hypothesis Density (PHD) and Cardinalized PHD filters [12],[14], [15], [16], [17].

Let us denote the frame image observation by $y = [y_1 \ \dots \ y_m]$. Then, using the FISST notion of integration and density, we can compute the posterior probability density $\pi(\cdot|y)$ of the multi-target state from the prior density via Bayes rule:

$$\pi(X|y) = \frac{\mathbf{g}(y|X)\pi(X)}{\int \mathbf{g}(y|X)\pi(X)\delta X} \tag{1}$$

where $\mathbf{g}(y|X)$ is probability density (likelihood) of observation y given the multi-target state X and the integral over the space of finite sets is defined as follows:

$$\int f(X)\delta X \triangleq \sum_{i=0}^{\infty} \frac{1}{i!} \int f(\{x_1, \ldots, x_i\}) dx_1 \ldots dx_i. \tag{2}$$

In this paper, the finite set of targets, X, is modelled by a *multi-Bernoulli* RFS which is defined as the union of M independent RFSs $X^{(i)}$ where M is the maximum number of targets.

$$X = \bigcup\nolimits_{i=1}^{M} X^{(i)}.$$

In this representation, each $X^{(i)}$ is either empty or a singleton with probabilities $1 - r^{(i)}$ and $r^{(i)}$, respectively. In the case where $X^{(i)}$ is a singleton, its only element is distributed according to a probability density $p^{(i)}(\cdot)$. Thus, a complete representation of the multi-target state is given by $\{(r^{(i)}, p^{(i)})\}_{i=1}^{M}$.

The Bayes update (1) is computationally intractable in general. Fortunately, it was shown in [10] that if the likelihood function has the following separable form:

$$\mathbf{g}(y|X) = f(y) \prod_{x \in X} g(x, y) \tag{3}$$

and the multi-target RFS has a multi-Bernoulli prior distribution $\{(r^{(i)}, p^{(i)})\}_{i=1}^{M}$, then the posterior distribution of X, given by Bayes rule (1), is also multi-Bernoulli with the parameters $\{(r^{(i)}_{\text{updated}}, p^{(i)}_{\text{updated}})\}_{i=1}^{M}$ where:

$$r^{(i)}_{\text{updated}} = \frac{r^{(i)}\langle p^{(i)}(\cdot), g(\cdot, y)\rangle}{1 - r^{(i)} + r^{(i)}\langle p^{(i)}(\cdot), g(\cdot, y)\rangle} \tag{4}$$

$$p^{(i)}_{\text{updated}}(\cdot) = \frac{p^{(i)}(\cdot) g(\cdot, y)}{\langle p^{(i)}(\cdot), g(\cdot, y)\rangle} \tag{5}$$

and $\langle f_1, f_2\rangle$ denotes the standard inner product $\int f_1(x) f_2(x) dx$.

In the next section, we show that the likelihood function for the image after background subtraction satisfies the above separable form. Using the update results (4) and (5), we detail a recursive filtering scheme that takes the background subtracted images as input to directly track multiple targets.

3 Visual Likelihood

Using background subtraction, each frame image is transformed into a grey scale image in which each pixel value is the probability density of the pixel belonging to the background. The background subtraction method used in this work is based on kernel density estimation which has been quite popular in visual tracking [18–20]. The resulting grey scale image is then used as input to the multi-target filter. For simplicity of notation, we will use the y and y_i symbols for the background subtracted grey scale image and its pixel values (which are indeed the probability density values of the actual pixel in the colour image to belong to background). We also assume that the y_i values are normalised to the interval $[0, 1]$.

3.1 Background Subtraction

It is assumed that the pixel i in the k-th colour image frame of the video has an RGB colour denoted by $[R_i(k)\ G_i(k)\ B_i(k)]^\top$. We first convert the RGB colour to chromaticity (rgI) colours by:

$$r_i(k) = R_i(k) / \left(R_i(k) + G_i(k) + B_i(k)\right) \tag{6}$$

$$g_i(k) = G_i(k) / \left(R_i(k) + G_i(k) + B_i(k)\right) \tag{7}$$

$$I_i(k) = \left(R_i(k) + G_i(k) + B_i(k)\right) / 256 \tag{8}$$

where the denominator 256 applies to 8-bit colour quantisation. It is observed in [19] that chromaticity colour is more robust to ambience light variations and shadows. Note that the above colour components all vary within the interval $[0, 1]$.

To compute the kernel density estimate of the i-th pixel to belong to background, we keep a stack of N_0 image frames (each in the form of a 3-D array including all rgI colours of the pixels) and update the contents of the stack regularly after every K_0 frames. The interpretation of the parameter K_0 can be explained via an example: if the frame rate of the video is 25 and we are looking for moving targets that are not stationary for more than 5 seconds, we can choose K_0 in the range of $5 \times 25 = 125$.

The stack of images will initially contain all pixel values recorded at the sampling times 0, K_0, $2K_0$, …, $(N_0 - 1)K_0$. This stack will be then updated (first at the time $k = N_0K_0$ then in each K_0 frames) by removing the first image from the bottom of the stack and appending the most recently recorded image (e.g. at the sampling time N_0K_0). More precisely, at time k (for $k \geq N_0K_0$), the stack will contain the rgI values of all pixels at the times $K_0\lfloor k/K_0 \rfloor$, $K_0(\lfloor k/K_0 \rfloor - 1)$, … , $K_0(\lfloor k/K_0 \rfloor - N_0 + 1)$. The kernel density estimate of the likelihood of the event that the i-th pixel belongs to the background is then given by:

$$p_i(k) = \frac{1}{N_0} \sum_{\ell=0}^{N_0-1} \prod_{d=r,g,I} \mathcal{N}\left(d_i(k); d_i(K_0(\lfloor k/K_0 \rfloor - \ell)), \sigma_d^2\right) \tag{9}$$

where $\mathcal{N}(x; x_0, \sigma) \triangleq \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-x_0)^2}{2\sigma^2})$ and σ_r, σ_g and σ_I are the bandwidth of Gaussian kernels for the rgI colours and are user-defined parameters chosen between 0 and 1. To normalise the $p_i(k)$ values to vary within [0,1], the density normalisation factors $\frac{1}{\sqrt{(2\pi)^3\sigma_r\sigma_g\sigma_I}}$ are removed which results in the following normalised y_i values:

$$y_i(k) = \frac{1}{N_0}\sum_{\ell=0}^{N_0-1}\prod_{d=r,g,I}\exp\left[-\frac{[d_i(k)-d_i(K_0(\lfloor k/K_0\rfloor-\ell))]^2}{2\sigma_d^2}\right]. \tag{10}$$

3.2 Likelihood Model

Having obtained a grey scale image with pixel values y_i, we need to compute its likelihood for a given set of multi-target state $X = \{x_j | j = 1, \ldots, n\}$. Each target region is denoted by $T(x_j)$ within which the average (or a weighted average) of all pixel values can be computed:

$$\overline{y}_j = \sum_{i\in T(x_j)} y_i \Big/ m_j \tag{11}$$

where $m_j = |T(x_j)|$ is the number of pixels within the region $T(x_j)$ defined by the state x_j. The likelihood of the region $T(x_j)$ to include a target is expressed as a function of $\overline{y}_j$ denoted by $g_{F}(\overline{y}_j)$. This function should be a strictly decreasing function in [0,1]. An appropriate choice of such function is $g_{F}(\overline{y}_j) = \zeta_{F}\exp(-\overline{y}_j/\delta_{F})$ where δ_{F} is a control parameter to tune the sensitivity to large average pixel values, and ζ_{F} is a normalising constant – see Fig. 1(a). Based on independence assumptions, the likelihood of all elements of the state set X to include target regions in the background-subtracted image is given by $\prod_{j=1}^{n} g_{F}(\overline{y}_j)$.

The rest of the pixels in the image that do not belong to any of the regions $T(x_j)$ $(j = 1, \ldots, n)$, are highly likely to belong to background. This is an important condition as otherwise, there might be more than n targets in the scene and this violates the premise that there are n targets. Let us denote the rest of the image by:

$$y_{-X} \triangleq \Upsilon(y) - \bigcup_{j=1}^{n}\{y_i | i \in T(x_j)\} \tag{12}$$

where $\Upsilon(y)$ is the result of mapping the matrix y to a set containing all the pixel values. We also construct a new image by filling up all the target regions with background pixels (all y_i values equal to 1), and denote the set of its pixel values by $\Taurus(y;X)$ which can also be expressed as below:

$$\Taurus(y;X) = \left[\bigcup_{j=1}^{n}\{\underbrace{1,\cdots,1}_{m_j\ \text{times}}\}\right]\bigcup y_{-X}. \tag{13}$$

The average (or weighted average) of pixels belonging to $\Taurus(y;X)$ is given by:

$$\overline{y}_{B} = \frac{1}{m}\left(\sum_{i=1}^{m} y_i + \sum_{j=1}^{n}\sum_{i\in T(x_j)}(1-y_i)\right). \tag{14}$$

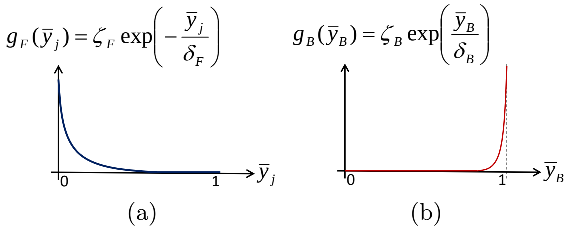

Fig. 1. (a) Foreground likelihood model (b) Background likelihood model

This average is within [0,1] and expected to be very close to 1. Indeed, if there are any targets existing in the image but not included in the hypothesised state X, the low values of the pixels belonging to that target region will decrease $\overline{y}_B$. If the average target size is small relative to the whole image, this decreasing effect can be small. Therefore, the likelihood of $\overline{y}_B$ to represent background region should be large only for $\overline{y}_B$ values that are very close to 1. It is important to note that scattered noise (e.g. salt and pepper noise) in the background-subtracted image may cause reduction in $\overline{y}_B$, similar to the effect of small size targets. To prevent this, we remove such tiny noise and other areas of image (containing small-values pixels) by morphologically closing the image (erosion followed by dilation of the image using a small structural element).

We denote the likelihood of $\overline{y}_B$ to represent an all background region by $g_B(\overline{y}_B)$ which is expected to be an increasing function of $\overline{y}_B$ in [0,1]. We choose the exponential function $g_B(\overline{y}_B) = \zeta_B \exp(\overline{y}_B/\delta_B)$ where δ_B is a control parameter to tune the sensitivity to deviations of the average pixel value from 1, and ζ_B is a normalising constant – see Fig. 1(b). As we will see later, the exponential form is a necessity here to provide a separable form for the total likelihood. Replacing $\overline{y}_B$ from equation (14), we derive:

$$g_B(\overline{y}_B) = \zeta_B \ \exp\left(\frac{\sum_{i=1}^{m} y_i + \sum_{j=1}^{n} \sum_{i\in T(x_j)} (1-y_i)}{m\ \delta_B}\right) \tag{15}$$

$$= \zeta_B \ \exp\left(\frac{\sum_{i=1}^{m} y_i}{m\ \delta_B}\right) \exp\left(\frac{\sum_{j=1}^{n} \sum_{i\in T(x_j)} (1-y_i)}{m\ \delta_B}\right) \tag{16}$$

$$= \zeta_B \ \exp\left(\frac{\sum_{i=1}^{m} y_i}{m\ \delta_B}\right) \prod_{j=1}^{n} \exp\left(\frac{m_j - \sum_{i\in T(x_j)} y_i}{m\ \delta_B}\right) \tag{17}$$

$$= \zeta_B \ \exp\left(\frac{\sum_{i=1}^{m} y_i}{m\ \delta_B}\right) \prod_{j=1}^{n} \exp\left(\frac{m_j(1-\overline{y}_j)}{m\ \delta_B}\right). \tag{18}$$

Finally, the total likelihood of the image y for the given set of states X is given by:

$$g(y|X) = g_B(\overline{y}_B) \prod_{j=1}^{n} g_F(\overline{y}_j). \tag{19}$$

By substituting the foreground and background likelihood functions, we derive the following separable form:

$$g(y|X) = \underbrace{\zeta_B \ \exp\left(\frac{\sum_{i=1}^{m} y_i}{m \ \delta_B}\right)}_{f(y)} \ \prod_{j=1}^{n} \underbrace{\left[\exp\left(\frac{m_j(1-\overline{y}_j)}{m \ \delta_B}\right) g_F(\overline{y}_j)\right]}_{g_y(x_j)} . \qquad (20)$$

4 Monte Carlo Implementation

Our implementation is based on the method presented in [10], adapted to the likelihood function defined in (20) for multi-target visual tracking. Suppose that at time $k-1$, the posterior density $\{r_{k-1}^{(i)}, p_{k-1}^{(i)}\}_{i=1}^{M_{k-1}}$ is given and each $p_{k-1}^{(i)}$ is represented by a set of weighted samples (particles) $\{w_{k-1}^{(i,j)}, x_{k-1}^{(i,j)}\}_{j=1}^{L_{k-1}^{(i)}}$. More precisely,

$$p_{k-1}^{(i)}(x) = \sum_{j=1}^{L_{k-1}^{(i)}} w_{(k-1)}^{(i,j)} \delta_{x_{k-1}^{(i,j)}}(x). \qquad (21)$$

We assume a constant survival probability P_S, and consider a predefined model for birth particles denoted by known parameters $\{r_{\Gamma}^{(i)}, p_{\Gamma,k}^{(i)}\}_{i=1}^{M_\Gamma}$ where the density $p_{\Gamma,k}^{(i)}$ is represented by the particles $\{w_{\Gamma,k}^{(i,j)}, x_{\Gamma,k}^{(i,j)}\}_{j=1}^{L_\Gamma}$. In our experiments, we assume that with a constant probability of 0.02, one target appears in each of the four quarters of the image planes, with the location of the target being uniformly distributed within the quarter. Thus, $M_\Gamma = 4$, $r_\Gamma^{(1)} = \cdots = r_\Gamma^{(4)} = 0.02$ and the birth particles are sampled with uniform distribution and weights.

Similar to many other particle filtering schemes, in each iteration, the particles are predicted then updated. In the prediction step, the birth particles are generated according to the birth model parameters. The multi-Bernoulli parameters from the previous iteration, $\{r_{k-1}^{(i)}, w_{k-1}^{(i,j)}, x_{k-1}^{(i,j)}\}$, are propagated forward:

$$x_{k|k-1}^{(i,j)} \sim f_{k|k-1}(\cdot|x_{k-1}^{(i,j)}) \ ; \ r_{k|k-1}^{(i)} = P_S \ r_{k-1}^{(i)} \ ; \ w_{k|k-1}^{(i,j)} = w_{k-1}^{(i,j)}. \qquad (22)$$

The proposal density equals the state transition density $f_{k|k-1}(\cdot|x_{k-1})$. In our experiments, the targets are modelled by rectangular blobs and the target state is a 4-tuple vector comprising the x and y location and width and height. The target dynamic is modelled by $x(k+1) = x(k) + e(k)$ where $e(k)$ is a 4-dimensional Gaussian variable with zero mean and variance $\Sigma = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_h^2, \sigma_w^2)$. Thus, $f_{k|k-1}(x|x_{k-1}) = \mathcal{N}(x, \Sigma)$.

In the update step, the predicted multi-Bernoulli parameters are updated using the likelihood function (20) and update formulas (4) and (5) which translate to:

$$r_k^{(i)} = r_{k|k-1}^{(i)} \varrho_k^{(i)} / \left(1 - r_{k|k-1}^{(i)} + r_{k|k-1}^{(i)} \varrho_k^{(i)}\right) \qquad (23)$$

$$w_k^{(i,j)} = w_{k|k-1}^{(i,j)} \ \ g_{y_k}(x_{k|k-1}^{(i,j)}) / \varrho_k^{(i)} \qquad (24)$$

where $\varrho_k^{(i)} = \sum_{j=1}^{L_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} \;\; g_{y_k}(x_{k|k-1}^{(i,j)})$ [10].

Similar to the MeMBer filter [21], the updated particles are resampled with the number of particles reallocated in proportion to the probability of existence as well as restricted between a minimum $L_{\min}$ and maximum $L_{\max}$. To reduce the growing number of multi-Bernoulli parameters, those with probabilities of existence less then a small threshold (set at 0.01) are removed. In addition, the targets with substantial overlap are merged. Finally, the number of targets and their states are estimated via finding the multi-Bernoulli parameters with existence probabilities larger than a threshold (set at 0.5 in our experiments). Each target state estimate is then given by the weighted average of the particles of the corresponding density.

5 Tracking Experiments

We demonstrate our method for tracking moving people in three video sequences from the CAVIAR dataset[4] which is a benchmark for visual tracking experiments. The tracking results are available to download and view from our home page.[5] The first video shows two persons each entering the lobby of a lab in INRIA and leaving the environment. The second video shows people walking in a shopping centre and occasionally visiting a shop that is in the front view of the camera. The third video shows four people entering the same place as in the first video, walking together and leaving the lobby. Except for a small number of frames, the four people are relatively accurately detected and tracked at all times. In this video, we also show the background subtracted (grey scale) images to give an indication of how our tracking method uses the results of background subtraction.

Figure 2 shows snapshots of the third video. It demonstrates that in general, our method can accurately track multiple targets in the video. The tracking results in the frames shown in Fig. 2 also present the ability of our tracking technique in detecting the arrival of new targets into the scene and tracking them while moving and interacting with other targets, and detecting their departure from the scene.

6 Conclusions

A novel algorithm for tracking multiple targets directly from image observations has been presented. Using kernel density estimation, the proposed algorithm gradually learns and updates a probabilistic background model which is then used to generate a grey scale foreground image. A separable likelihood function has been derived for the grey scale foreground image, which enabled

[4] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

[5] Video 1: www.dlsweb.rmit.edu.au/eng1/Mechatronics/Case01.mpg
Video 2: www.dlsweb.rmit.edu.au/eng1/Mechatronics/Case02.mpg
Video 3: www.dlsweb.rmit.edu.au/eng1/Mechatronics/Case03.mpg.

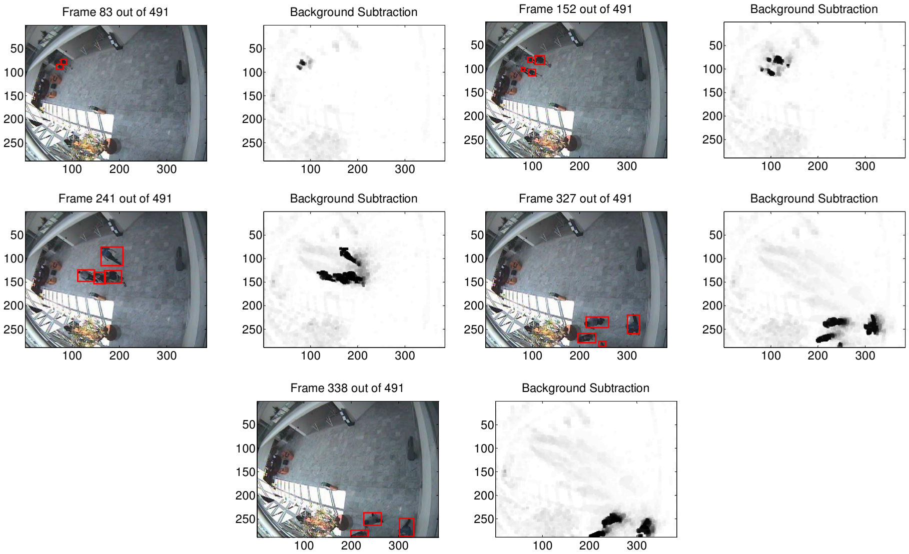

Fig. 2. Tracking of up to four people in a video sequence from CAVIAR dataset. The selected frames show that the method is capable of detecting and tracking multiple moving objects as they enter the scene, interact and leave the scene.

an efficient multi-target filtering technique called multi-Bernoulli filtering to be applied. The method has been evaluated in three tracking scenarios from the CAVIAR datasets, showing that multiple persons can be tracked accurately.

Acknowledgement

This work was supported by ARC Discovery Project grant DP0880553. Authors thank Dr Branko Ristic from DSTO, Australia for his contribution.

References

1. Okuma, K., Taleghani, A., De Freitas, N., Little, J., Lowe, D.: A boosted particle filter: Multitarget detection and tracking. In: ECCV'04. Volume 3021. (2004) 28–39
2. Kristan, M., Per, J., Pere, M., Kovacic, S.: Closed-world tracking of multiple interacting targets for indoor-sports applications. Computer Vision and Image Understanding **113**(5) (2009) 598 – 611
3. Yang, M., Yu, T., Wu, Y.: Game-theoretic multiple target tracking. In: ICCV'07, Rio de Janeiro, Brazil (2007) http://dx.doi.org/10.1109/ICCV.2007.4408942.
4. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. IJCV **75**(2) (2007) 247 – 266
5. Zhao, T., Nevatia, R., Wu, B.: Segmentation and tracking of multiple humans in crowded environments. PAMI **30**(7) (2008) 1198 – 211

6. Apewokin, S., Valentine, B., Bales, R., Wills, L., Wills, S.: Tracking multiple pedestrians in real-time using kinematics. In: CVPR'08 Workshops, Anchorage, AK, United states (2008) http://dx.doi.org/10.1109/CVPRW.2008.4563149.
7. Zhu, L., Zhou, J., Song, J.: Tracking multiple objects through occlusion with online sampling and position estimation. Pattern Recognition **41**(8) (2008) 2447 – 2460
8. Parvizi, E., Wu, Q.J.: Multiple object tracking based on adaptive depth segmentation. In: Canadian Conference on Computer and Robot Vision – CRV 2008, Windsor, ON, Canada (2008) 273 – 277
9. Abbott, R., Williams, L.: Multiple target tracking with lazy background subtraction and connected components analysis. Machine Vision and Applications **20**(2) (2009) 93 – 101
10. Vo, B.N., Vo, B.T., Pham, N.T., Suter, D.: Bayesian multi-object estimation from image observations. In: Fusion'09, Seatle, Washington (2009) 890–898
11. Hoseinnezhad, R., Vo, B.N., Suter, D., Vo, B.T.: Multi-object filtering from image sequence without detection. In: ICASSP, Dallas, TX (2010) 1154–1157
12. Mahler, R.: Multi-target bayes filtering via first-order multi-target moments. IEEE Trans. Aerospace & Electronic Systems **39**(4) (2003) 1152–1178
13. Mahler, R.: Statistical multisource-multitarget information fusion. Artech House (2007)
14. Vo, B.N., Singh, S., Doucet, A.: Sequential Monte Carlo methods for multi-target filtering with random finite sets. IEEE Tran. AES **41**(4) (2005) 1224–1245
15. Vo, B.N., Ma, W.K.: The Gaussian mixture probability hypothesis density filter. IEEE Trans. Signal Proc. **54**(11) (2006) 4091 – 104
16. Mahler, R.: Phd filters of higher order in target number. IEEE Trans. Aerospace & Electronic Systems **43**(4) (2007) 1523–1543
17. Vo, B.T., Vo, B.N., Cantoni, A.: Analytic implementations of the Cardinalized Probability Hypothesis Density filter. IEEE Trans. Signal Processing **55**(7) (2007) 3553–3567
18. Tyagi, A., Keck, M., Davis, J.W., Potamianos, G.: Kernel-based 3D tracking. In: CVPR'07, Minneapolis, Minnesota, USA (2007)
19. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of the IEEE **90**(7) (2002) 1151 – 1162
20. Han, B., Comaniciu, D., Zhu, Y., Davis, L.S.: Sequential kernel density approximation and its application to real-time visual tracking. PAMI **30**(7) (2008) 1186–1197
21. Vo, B.T., Vo, B.N., Cantoni, A.: The cardinality balanced multi-target multi-Bernoulli filter and its implementations. IEEE Transactions on Signal Processing **57**(2) (2009) 409–423