Visual Tracking in Background Subtracted Image Sequences via Multi-Bernoulli Filtering

Reza Hoseinnezhad, Ba-Ngu Vo and Ba-Tuong Vo

***Abstract*—This paper presents a novel method for simultaneous tracking of multiple non-stationary targets in video. Our method operates directly on the video data and does not require any detection. We propose a multi-target likelihood function for the background-subtracted grey-scale image data, which admits multi-target conjugate priors. This allows the multi-target posterior to be efficiently propagated forward using the multi-Bernoulli filter. Our method does not need any training pattern or target templates and makes no prior assumptions about object types or object appearance. Case studies from the CAVIAR dataset show that our method can automatically track multiple targets and quickly finds targets entering or leaving the scene.**

***Index Terms*—multi-object filtering, random finite set, multi-Bernoulli filter, finite set statistics, visual tracking.**

I. Introduction

We consider the problem of tracking multiple non-stationary targets, i.e. jointly estimate the number of targets and their individual states, directly from video data without the need for detection. Most multi-target tracking approaches for video data involve a detection operation to generate point measurements before multi-target tracking/filtering is applied [1]–[4]. Well-known multi-target tracking algorithms that do not require detection include Bramble [5], color-based visual tracking [6], [7], Bayesian existence process [8], multi-modal distributions [9]–[11]. In principle, detection incurs information loss which can significantly degrade tracking performance, especially in low signal to noise ratio. Tracking directly from the video data (without detection) is of fundamental interest as it avoids this type of information loss.

A recent approach to multi-target tracking that has attracted substantial interest is the random finite set (RFS) framework [12], [13]. Motivated by a fundamental consideration in estimation theory – estimation error – this approach represents the collection of target states, called the *multi-target state*, as a finite set (see [14], [15]). RFS multi-target filtering techniques such as Gaussian mixture and particle probability hypothesis density filters [16]–[18] have been applied to tracking from video data via detection in [2], [3], [19]. RFS-based techniques for tracking multiple targets directly from video data have also been investigated in and successfully demonstrated on tracking of sport players in [20], [21]. However, like many current detection-free methods, this technique requires prior information about the visual appearance of the desired targets, which are either prescribed or obtained from training data.

In this paper, we propose an RFS-based algorithm for tracking of multiple non-stationary objects from video data that requires neither detection nor prior information on target visual appearance. Our method is a combination of kernel density estimation and multi-Bernoulli filtering. Using kernel density estimation, our algorithm learns and updates the background which is then subtracted from the video frames yielding grey-scale foreground images. A tractable multi-target measurement model for the resulting grey-scale foreground image is proposed which allows the sequence of grey-scale foreground images to be processed directly by the multi-Bernoulli filter proposed in [14]. Preliminary results have been reported in [22]. The current correspondence presents a more complete study, with improved solution. Case studies from the CAVIAR data set show improved performance in terms of computational cost and accuracy, compared to recent tracking methods that use background subtraction.

Reza Hoseinnezhad is with the School of Aeropspace, Mechanical and Manufacturing Engineering, RMIT University, Victoria, Australia. email: rezah@rmit.edu.au.

Ba-Ngu Vo and Ba-Tuong Vo are with Department of Electrical and Computer Engineering, Curtin University of Technology, WA, Australia. email: {ba-ngu.vo,ba-tuong.vo}@curtin.edu.au.

B.-N. Vo and B.-T. Vo are supported by the Australian Research Council under the discovery grants FT0991854 and DE120102388.

Manuscript received mmmm dd, yyyy; revised mmmm dd, yyyy.

II. Visual Multi-Target Likelihood

Using kernel density estimation based background subtraction [23]–[25], each frame image is transformed to a grey-scale image in which each pixel value can be interpreted as the probability of the pixel belonging to the background. The resulting grey-scale image is then used as input to the multi-target filter.

A. Background Subtraction

It is assumed that 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:

$$r_i(k) = R_i(k)/\left(R_i(k)+G_i(k)+B_i(k)\right) \quad (1)$$

$$g_i(k) = G_i(k)/\left(R_i(k)+G_i(k)+B_i(k)\right) \quad (2)$$

$$I_i(k) = \left(R_i(k)+G_i(k)+B_i(k)\right)/256 \quad (3)$$

where the denominator value 256 corresponds to 8-bit colour quantization. It is observed in [24] that chromaticity colour is more robust to ambience light variations and shadows.

To compute the probability that the i-th pixel belongs to the 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 $5 \times 25 = 125$. These parameter values are used in our experiments.

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 $K_0\lfloor k/K_0 \rfloor$, $K_0(\lfloor k/K_0 \rfloor - 1)$, $\ldots$, $K_0(\lfloor k/K_0 \rfloor - N_0 + 1)$. The kernel density estimate of the likelihood of the event that the i-th pixel belongs to the background is then given by:

$$p_i(k) = \frac{1}{N_0}\sum_{\ell=0}^{N_0-1}\prod_{d=r,g,I}\mathcal{N}\left(d_i(k); d_i(k_{i\ell}), \sigma_{d,i}(k)^2\right) \quad (4)$$

where $d_i(k)$ means one of three possible chromaticity colour components at time k, i.e. $r_i(k)$, $g_i(k)$ or $I_i(k)$, the time k_ℓ is the time when the $(\ell-1)$-th image in the stack was recorded, i.e. $k_\ell = K_0(\lfloor k/K_0 \rfloor - \ell)$ and $\mathcal{N}(x; x_0, \sigma) \triangleq \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(x-x_0)^2}{2\sigma^2})$ and $\sigma_{r,i}(k)$, $\sigma_{g,i}(k)$, and $\sigma_{I,i}(k)$ are the bandwidth (scale) of Gaussian kernels for the rgI colours for the i-th pixel at time k.

When using kernel density estimation technique, the choice of suitable kernel bandwidth is a major issue. In theory, for an infinite number of samples, the estimate will approach the actual density regardless of the choice of bandwidth. In practice, only a finite number of samples are used. Thus, the choice of a suitable bandwidth is essential. A very small bandwidth results in a ragged density

estimate. On the other hand, a very wide bandwidth will result in an over-smoothed density estimate. In practice, variations in pixel intensity over time are different from one location to another in the image. Therefore, a different kernel bandwidth is used for each pixel in each colour channel.

At each time k, we estimate the kernel bandwidth $\sigma_{d,i}(k)$ for the colour channel $d = r, g, I$ and for a given pixel i, by computing the median absolute deviation (MAD) over the sample for consecutive intensity values of the pixel, which is given by:

$$\sigma_{d,i}(k) \quad = \quad \underset{\ell}{\text{median}} \ |d_i(k) - d_i(k_\ell)|. \tag{5}$$

The rationale behind the use of MAD estimate is that variations of each pixel's chromaticity colours over time are expected to have jumps because different objects (when an edge passes through the pixel) are projected onto the same pixel at different times. The median provides robustness and is not significantly affected by a small number of jumps.

Normalizing the $p_i(k)$ values to vary within [0,1], results in the following normalized y_i values:

$$y_i(k) = \frac{1}{N_0} \sum_{\ell=0}^{N_0-1} \exp\left[- \sum_{d=r,g,I} \frac{[d_i(k) - d_i(k_{i\ell})]^2}{2\sigma_{d,i}(k)^2} \right]. \tag{6}$$

B. Measurement Model

We use the notation $y = [y_1 \ \ldots \ y_m]$ for the background-subtracted grey-scale image and assume that each y_i, $i = 1, ..., m$, is normalized to lie in the interval $[0, 1]$ (hence the pixel value y_i can be interpreted as the probability that pixel i in the colour image belongs to the background). When necessary, we indicate the dependence of these values on the time index k by $y(k)$, $y_i(k)$.

We consider the following measurement model for the grey-scale background-subtracted image. Each target x illuminates a region $T(x)$ on the grey-scale image with intensity t distributed according to a probability density g_F that is strictly decreasing on $[0, 1]$. An appropriate choice of such function is $g_F(t) = \zeta_F \exp(-t/\delta_F)$ where δ_F is a control parameter to tune the sensitivity to large average pixel values, and ζ_F is a normalizing constant – see Fig. 1(a).

Let $\overline{y}(x)$ denote the average (or a weighted average) of all pixel values in $T(x)$, i.e.

$$\overline{y}(x) = \frac{1}{|T(x)|} \sum_{i \in T(x)} y_i \tag{7}$$

where $|T(x)|$ is the number of pixels within $T(x)$. Then the likelihood that target x illuminates the region $T(x)$ is $g_F(\overline{y}(x))$. Assuming that given a set X of targets the pixel values are statistically independent, then the likelihood that the set X of targets illuminates the region $\bigcup_{x \in X} T(x)$ in the background-subtracted grey-scale image is given by $\prod_{x \in X} g_F(\overline{y}(x))$. [1]

The rest of the pixels in the grey-scale image y that do not belong to $\bigcup_{x \in X} T(x)$, belong to the background. We assume that y is formed by overlaying the foreground pixels on an all background images with average intensity t distributed according to a probability density g_B which is an increasing function on [0,1] (the probability density of the background intensity should be large only for values that are close to 1). We choose the exponential function $g_B(t) = \zeta_B \exp(t/\delta_B)$ 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 normalizing constant – see Fig. 1(b).

[1]We will model the set of targets X as a random finite set, and apply a Bayesian filtering scheme to recursively estimate the distribution of X. The details are presented in Section III. In the update step of the filter, we will need a measurement model formulated as a likelihood function. It is important to note that in the likelihood function, the set of targets X is given.

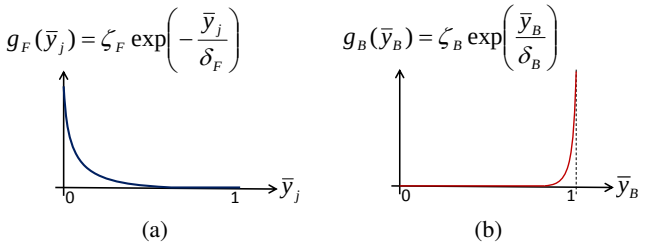

Fig. 1. (a) Foreground likelihood model (b) Background likelihood model

The control parameters δ_F and δ_B are fixed and manually chosen as small positive values within 0 and 1. In our experiments, we chose $\delta_F = 0.1$ and $\delta_B = 0.02$. We also tried alternative values close to 0.1 and 0.02, and no significant changes in the tracking results were observed. It is important to note that our choice of δ_B must be several times smaller than δ_F. The reason is that in presence of several targets, we expect a relatively large ratio of the background subtracted image to be white (compared with the area covered by each target). Thus, the average intensity of a background area is expected to be far closer to 1 than the average intensity of a foreground (single target) area to zero.

Given a set of target states X, let $\overline{y}_B(X)$ denote the average (or weighted average) of pixel intensities of the image constructed by filling up all the target regions of the image y with background pixel value of 1, i.e.

$$\overline{y}_B(X) = \frac{1}{m} \left(\sum_{i=1}^{m} y_i + \sum_{x \in X} \sum_{i \in T(x)} (1 - y_i) \right). \tag{8}$$

Then the likelihood that X generates the background region is $g_B(\overline{y}_B(X))$. Replacing $\overline{y}_B(X)$ from equation (8), we have

$$\begin{aligned}
g_B(\overline{y}_B(X)) &= \zeta_B \exp\left(\frac{\sum_{i=1}^{m} y_i + \sum_{x \in X} \sum_{i \in T(x)} (1-y_i)}{m\delta_B} \right) \\
&= \zeta_B \exp\left(\frac{\sum_{i=1}^{m} y_i}{m\delta_B} \right) \exp\left(\frac{\sum_{x \in X} \sum_{i \in T(x)} (1-y_i)}{m\delta_B} \right) \\
&= \zeta_B \exp\left(\frac{\sum_{i=1}^{m} y_i}{m\delta_B} \right) \prod_{x \in X} \exp\left(\frac{|T(x)| - \sum_{i \in T(x)} y_i}{m\delta_B} \right) \\
&= \zeta_B \exp\left(\frac{\sum_{i=1}^{m} y_i}{m\delta_B} \right) \prod_{x \in X} \exp\left(\frac{|T(x)|(1-\overline{y}(x))}{m\delta_B} \right).
\end{aligned}$$

Note that background average intensity $\overline{y}_B(X)$ lies in the interval $[0, 1]$ and is expected to be close to 1. Indeed, if there are any targets existing in the image but not included in the hypothesized state X, the low values of the pixels belonging to that target region will decrease $\overline{y}_B(X)$. If the average target size is small relative to the whole image, this decreasing effect is small. It is important to note that scattered noise (e.g. salt and pepper noise) in the background-subtracted image may cause reduction in $\overline{y}_B(X)$, 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).

Finally, the likelihood of the background-subtracted image y given multi-target state X is given by:

$$g(y|X) = g_B(\overline{y}_B(X)) \prod_{x \in X} g_F(\overline{y}(x)). \tag{9}$$

Substituting for the foreground and background likelihood functions yield the following form

$$g(y|X) = \underbrace{\zeta_B \exp\left(\frac{\sum_{i=1}^{m} y_i}{m\delta_B}\right)}_{f(y)} \times \prod_{x\in X} \underbrace{\left[\exp\left(\frac{|T(x)|(1-\overline{y}(x))}{m\delta_B}\right) g_F(\overline{y}(x))\right]}_{g_y(x)}. \quad (10)$$

III. Multi-Bernoulli Filter

Having constructed a measurement model for the background-subtracted image, we now proceed to describe the algorithm for estimating multiple targets from the background-subtracted data.

A. Bayes update

Based on a fundamental consideration in estimation theory – estimation error – it was shown that a finite set is a suitable representation of the multi-target state [14]. Hence, in the Bayesian estimation paradigm, the multi-target state is treated as a realization of a random finite set (RFS). In this work we use Mahler's Finite Set Statistics (FISST) notion of integration and density for dealing RFS [12], [26].

Given a prior multi-target probability density π, the posterior probability density $\pi(\cdot|y)$ of the multi-target state is given via Bayes rule:

$$\pi(X|y) = \frac{g(y|X)\pi(X)}{\int g(y|X)\pi(X)\delta X} \quad (11)$$

where $g(y|X)$ is likelihood of observation y given the multi-target state X, and the integral over the space of finite sets is defined as follows [12], [26]:

$$\int f(X)\delta X \triangleq \sum_{i=0}^{\infty} \frac{1}{i!} \int f(\{x_1,\ldots,x_i\})dx_1\ldots dx_i. \quad (12)$$

The Bayes update (11) is computationally intractable in general. Fortunately, it was shown in [14] that a likelihood function of the form (10) admits multi-target conjugate priors such as multi-Bernoulli, which can be efficiently computed.

A *multi-Bernoulli* RFS X is a union of a fixed number of independent RFSs $X^{(i)}$ that have probability $1-r^{(i)}$ of being empty, and probability $r^{(i)} \in (0,1)$ of being a singleton whose (only) element is distributed according to a probability density $p^{(i)}$ [26]

$$X = \bigcup_{i=1}^{M} X^{(i)}. \quad (13)$$

A multi-Bernoulli RFS is completely described by the multi-Bernoulli parameters $\{(r^{(i)},p^{(i)})\}_{i=1}^{M}$ and we use the notation $\pi = \{(r^{(i)},p^{(i)})\}_{i=1}^{M}$ to denote its probability density. The parameter $r^{(i)}$ is interpreted as the existence probability of the ith target while $p^{(i)}$ is the probability density of the target state conditional on its existence.

B. Multi-Bernoulli filter

Let π_k denote the multi-target posterior at time k (for convenience we drop the dependence on the observation history $[y(1),...,y(k)]$. Then, the *multi-target Bayes recursion* propagates π_k in time [12], [26] according to the following prediction and update steps:

$$\pi_{k|k-1}(X) = \int f_{k|k-1}(X|X')\pi_{k-1}(X')\delta X' \quad (14)$$

$$\pi_k(X) = \frac{g(y(k)|X)\pi_{k|k-1}(X)}{\int g(y(k)|X)\pi_{k|k-1}(X)\delta X}, \quad (15)$$

where $f_{k|k-1}(\cdot|\cdot)$ is the *multi-target transition density*, from time $k-1$ to k, which encapsulates the underlying models of motions, births and deaths.

In this paper we use the following standard multi-target transition model. Given a multi-target state X' at time $k-1$, each x' in X' either continues to exist at time k with probability $p_{S,k}(x')$ and moves to a new state x with probability density[2] $f_{k|k-1}(x|x')$, or dies with probability $1-p_{S,k}(x')$. Denote by $S_{k|k-1}(x')$ the RFS generated at time k by a state x' at time $k-1$, then the multi-target state X at time k is given by the union

$$X = \bigcup_{x'\in X'} S_{k|k-1}(x') \cup \Gamma_k, \quad (16)$$

where Γ_k denotes the multi-Bernoulli RFS of spontaneous births.

The multi-target Bayes recursion is computationally intractable in general. However, if π_{k-1} is a multi-Bernoulli, then $\pi_{k|k-1}$ and π_k are also multi-Bernoulli's. More concisely,

If $\pi_{k-1} = \{(r_{k-1}^{(i)}, p_{k-1}^{(i)})\}_{i=1}^{M_{k-1}}$, *then*

$$\pi_{k|k-1} = \{(r_{P,k|k-1}^{(i)}, p_{P,k|k-1}^{(i)})\}_{i=1}^{M_{k-1}} \cup \{(r_{\Gamma,k}^{(i)}, p_{\Gamma,k}^{(i)})\}_{i=1}^{M_{\Gamma,k}}, \quad (17)$$

where

$$r_{P,k|k-1}^{(i)} = r_{k-1}^{(i)} \int p_{k-1}^{(i)}(x')p_{S,k}(x')dx', \quad (18)$$

$$p_{P,k|k-1}^{(i)}(x) = \frac{\int f_{k|k-1}(x|x')p_{k-1}^{(i)}(x')p_{S,k}(x')dx'}{\int p_{k-1}^{(i)}(x')p_{S,k}(x')dx'}, \quad (19)$$

$f_{k|k-1}(\cdot|\zeta)$ = *single target transition density at time k, given previous state ζ,*

$p_{S,k}(\zeta)$ = *probability of target existence at time k, given previous state ζ,*

$\{(r_{\Gamma,k}^{(i)}, p_{\Gamma,k}^{(i)})\}_{i=1}^{M_{\Gamma,k}}$ = *parameters of the multi-Bernoulli RFS of births at time k.*

If $\pi_{k|k-1} = \{(r_{k|k-1}^{(i)}, p_{k|k-1}^{(i)}\}_{i=1}^{M_{k|k-1}}$, *then*

$$\pi_k = \{(r_k^{(i)}, p_k^{(i)})\}_{i=1}^{M_{k|k-1}} \quad (20)$$

where

$$r_k^{(i)} = \frac{r_{k|k-1}^{(i)} \int p_{k|k-1}^{(i)}(x)g_y(x)dx}{1 - r_{k|k-1}^{(i)} + r_{k|k-1}^{(i)} \int p_{k|k-1}^{(i)}(x), g_y(x)dx} \quad (21)$$

$$p_k^{(i)} = \frac{p_{k|k-1}^{(i)} g_y}{\int p_{k|k-1}^{(i)}(x), g_y(x)dx}. \quad (22)$$

These equations are the prediction and update step of the multi-Bernoulli filter proposed in [14].

C. Particle Implementation

We use the particle implementation of the multi-Bernoulli filter given in [14]. 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) $\{w_{k-1}^{(i,j)}, x_{k-1}^{(i,j)}\}_{j=1}^{L_{k-1}^{(i)}}$. In the prediction step, the birth particles are generated according to the birth model parameters. Using the transition density $f_{k|k-1}(\cdot|x_{k-1})$ as the

[2]The same notation is used for multi-object and single-object densities. There is no danger of confusion since for single-object the arguments are vectors whereas for multi-object the arguments are finite sets.

proposal, 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 by

$$\begin{aligned} x_{P,k|k-1}^{(i,j)} &\sim f_{k|k-1}(\cdot|x_{k-1}^{(i,j)}) \\ r_{P,k|k-1}^{(i)} &= r_{k-1}^{(i)} \sum_{j=1}^{L_{k-1}^{(i)}} w_{k-1}^{(i,j)} p_{S,k}(x_{k-1}^{(i,j)}) \\ w_{P,k|k-1}^{(i,j)} &= w_{k-1}^{(i,j)}. \end{aligned}$$

The particle approximation of the birth multi-Bernoulli parameters are sampled directly from $p_{\Gamma,k}^{(i)}$. In the update step, the predicted multi-Bernoulli parameters are updated as follows

$$\begin{aligned} r_k^{(i)} &= r_{k|k-1}^{(i)} \varrho_k^{(i)} / \left(1 - r_{k|k-1}^{(i)} + r_{k|k-1}^{(i)} \varrho_k^{(i)}\right) \\ w_k^{(i,j)} &= w_{k|k-1}^{(i,j)} \ g_{y_k}(x_{k|k-1}^{(i,j)}) / \varrho_k^{(i)} \end{aligned}$$

where $\varrho_k^{(i)} = \sum_{j=1}^{L_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} \ g_{y_k}(x_{k|k-1}^{(i,j)})$ [14].

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 $L_{\max}$ (similar to [20]–[22], [27]). 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. In our experiments, the overlap between two rectangular target regions was defined as the ratio of the intersection are to the area of the smaller target region. We merged every two targets with an overlap of more than 80%. 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.

IV. Simulation Results

We demonstrate our method for tracking moving people in three video sequences from the CAVIAR data set[3] which is a benchmark for visual tracking simulations. In our experiments, targets are modelled by rectangular blobs with constant survival probability p_S and the target state is a 4-D vector comprising the x and y location and width and height. The target dynamic is modelled by the random walk model $x(k+1) = x(k) + e(k)$ where $e(k)$ is Gaussian with zero mean and covariance $\Sigma = \mathrm{diag}(\sigma_x^2, \sigma_y^2, \sigma_h^2, \sigma_w^2)$. Thus, $f_{k|k-1}(x|x_{k-1}) = \mathcal{N}(x; x_{k-1}, \Sigma)$. This model is assumed to be the same for all targets. People targets can exhibit different movements and a fixed model needs to be sufficiently general to be compatible with all possible manoeuvres. This is the rationale behind our choice of the random walk model. In a more sophisticated solution, we could consider different models for different classes of people, and append a "class" variable to the target state. However, this would increase the computational cost, and we are not sure if the enhanced accuracy would be worthwhile.

For the birth model, 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, $M_\Gamma = 4$, $r_\Gamma^{(1)} = \cdots = r_\Gamma^{(4)} = 0.02$. The birth model and their parameters remain unchanged throughout the simulation. It is important to note that this birth model is well-suited for the case studies presented in this paper. In other applications, where additional practical information on the birth process (*e.g.* the entrance gate locations) are available, other birth models with different M_Γ and non-uniform densities need to be properly formulated.

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

The number of particles for each Bernoulli target is constrained between $L_{\min} = 100$ and $L_{\max} = 1000$, and the particles are resampled in every iteration (processing each frame of the video).

In principle, the multi-Bernoulli filtering scheme proposed here does not need any foreground-specific information relevant to targets. However, it is important to note that target size constraints are implicitly used when morphological operations (closing and opening) are applied on the background-subtracted image before it is input as measurement to the multi-Bernoulli filter. Indeed, such operations not only remove salt and pepper noise from the grey-scale image, but also implicitly impose constraints on the size and geometric shape of the target areas in the measurement model. We have used a 5-pixel value for the size of the structuring elements in our closing and opening operations.

The tracking videos are available to download as supplemental material or from our home page. [4]

The first video shows two persons each entering and leaving the lobby of a lab in INRIA. 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.

The snapshots of the third video in Figure 2 demonstrates that our method can accurately track multiple targets in the video. The tracking results in the frames shown in Fig. 2 also illustrate the ability of the proposed technique to detect the arrival of new targets, track them, and detect their departure from the scene.

We compare the performance of our method with two recent and popular visual tracking methods that can work on background-subtracted data. The first method [24] is based on nonparametric modeling of background objects using kernel density estimates (KDE). The second is a deterministic modeling technique for background and foreground objects using sample consensus (SACON) [28]. The Matlab code used in our case studies is provided by the authors of [28]. Note that if prior information about visual appearance of targets (foreground objects) are available, then both KDE and SACON can still be used to detect and track the foreground objects. However, to benchmark the tracking performance without prior information on the targets visual appearance, we have implemented the methods to process only background-subtracted images.

Tracking performance is quantified by the false alarm rate (FAR) and false negative rate (FNR) which represent the ratio of wrong detections and missed targets, respectively. To normalize these ratios, the number of false alarms or target misses are divided by the total number of ground truth targets over all frames. The ground truth number of targets and their ground truth states were also downloaded from the CAVIAR dataset. A false alarm or missed detection is determined based on comparing the estimation results with the ground truth. An estimated target state is a false alarm if it does not overlap any ground-truth target by at least 80%. A ground-truth target is considered being missed by the filter if it does not overlap any estimates by at least 80%. This is consistent with the criterion we applied to merge updated Bernoulli components, and the relatively large threshold of 80% demonstrates that the FAR and FNR quantities are highly sensitive to estimation accuracy. More precisely, a low

[4]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.

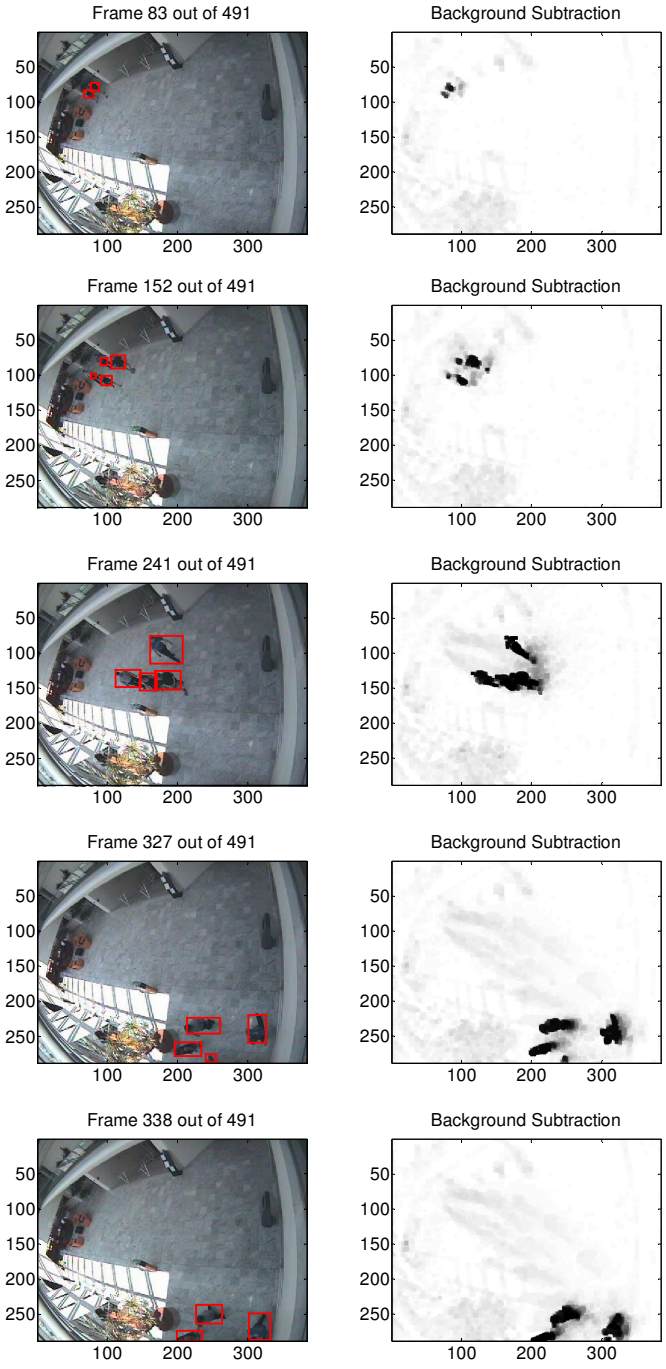

Fig. 2. Tracking of up to four people in a video sequence from CAVIAR dataset. The selected frames show that the method is capable of detecting and tracking multiple moving objects as they enter the scene, interact and leave the scene.

FAR and FNR can only be achieved if the filtering results are highly accurate.

Table I shows the FAR and FNR along with the number of processed frames per second (PFPS) – total number of image frames in the video divided by the total run time – for each of the videos. Observe that in the first two cases (video 1 and 2), where there is no more than one target at any time, our method is faster than the other tracking methods and better in terms of FAR and FNR. However, in the third case, where there are up to four targets, our method runs slightly slower than SACON and KDE methods but its FAR and FNR are significantly smaller.

The slower speed than SACON and KDE is due to the increase in the number of particles required to accommodate four targets. However, the superior accuracy, makes the proposed method worthwhile. Moreover, the multi-Bernoulli filter is highly parallelizable, and scalable.

Note that the processed frame rates (PFPS values varying from 1.78 to 5.15 frames per second as reported in Table I) correspond to coding our method in Matlab. If coded in C, the computation speed would substantially increase and real-time performance with frame rates of 15 and higher would be achieved. The actual number of frames that can be processed in every second also depends on other factors. For instance, a higher frame rate would be achieved via down-sampling the images before processing, decreasing the number of particles ($L_{\min}$ and $L_{\max}$ parameters) and decreasing the maximum number of Bernoulli targets (M). However, the mentioned variations could result in reduction of estimation accuracy. To strike the right balance between computation and accuracy, we need to tune the above mentioned parameters via trial and error.

V. 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 tractable multi-target measurement model has been proposed for the grey scale foreground image, which enabled an efficient multi-target filtering technique known as the multi-Bernoulli filter to be applied. The method has been evaluated in three tracking scenarios from the CAVIAR data sets, showing that multiple persons can be tracked accurately. Comparative results show that with a comparable computational cost, our method outperforms competitive and similar methods in terms of accuracy, especially for a relatively large number of targets.

References

[1] S.-W. Joo and R. Chellappa, "A multiple-hypothesis approach for multiobject visual tracking," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2849 – 2854, 2007.

[2] E. Maggio, M. Taj, and A. Cavallaro, "Efficient multitarget visual tracking using random finite sets," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 8, pp. 1016–1027, 2008.

[3] R. Hoseinnezhad, B.-N. Vo, and D. Suter, "Fast single-view people tracking," in *Proceedings of Cognitive Systems with Interactive Sensors Conference (COGIS'09)*, Paris, France, November 2009.

[4] K. Smith, D. Gatica-Perez, and J.-M. Odobez, "Using particles to track varying numbers of interacting people," in *CVPR'05*, vol. I, San Diego, CA, USA, 2005, pp. 962 – 969.

[5] M. Isard and J. MacCormick, "BraMBLe: a Bayesian multiple-blob tracker," in *ICCV'01*, vol. 2, Vancouver, British Columbia, Canada, 2001, pp. 34 – 41.

[6] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," Copenhagen, Denmark, 2002//, pp. 661 – 75.

[7] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "Object tracking with an adaptive color-based particle filter," Zurich, Switzerland, 2002//, pp. 353 – 60.

TABLE I
FALSE ALARM RATE, FALSE NEGATIVE RATE AND RUN TIMES OF THE TRACKING METHOD COMPARED WITH OUR METHOD.

	Video 1			Video 2			Video 3		
Method	FAR	FNR	PFPS	FAR	FNR	PFPS	FAR	FNR	PFPS
KDE [24]	0.001	0.015	4.88	0.057	0.326	3.73	0.046	0.396	2.68
SACON [28]	0.000	0.011	4.67	0.041	0.245	3.61	0.031	0.381	2.03
Our Method	0.000	0.005	5.15	0.017	0.105	3.92	0.004	0.055	1.78

FAR = False Alarm Rate, FNR = False Negative Rate, PFPS = Processed Frames Per Second.

[8] J. Vermaak, S. Maskell, M. Briers, and P. Perez, "Bayesian visual tracking with existence process," Atlanta, GA, USA, 2006//, pp. 721 – 4.

[9] K. Okuma, A. Taleghani, N. De Freitas, J. Little, and D. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *ECCV'04*, vol. 3021, 2004, pp. 28–39.

[10] J. Vermaak, A. Doucet, and P. Perez, "Maintaining multi-modality through mixture tracking," vol. 2, Nice, France, 2003, pp. 1110 – 1116.

[11] J. Czyz, B. Ristic, and B. Macq, "A particle filter for joint detection and tracking of color objects," *Image and Vision Computing*, vol. 25, no. 8, pp. 1271–1281, August 2007.

[12] R. Mahler, "Multi-target bayes filtering via first-order multi-target moments," *IEEE Trans. Aerospace & Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[13] ——, "PHD filters of higher order in target number," *IEEE Trans. Aero. Elec. Syst.*, vol. 43, no. 3, pp. 1523–1543, 2007.

[14] B.-N. Vo, B.-T. Vo, N.-T. Pham, and D. Suter, "Joint detection and estimation of multiple objects from image observations," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5129–5141, 2010.

[15] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation in multi-object filtering," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

[16] B.-N. Vo, S. Singh, and A. Doucet, "Sequential Monte Carlo methods for multi-target filtering with random finite sets," *IEEE Tran. AES*, vol. 41, no. 4, pp. 1224–1245, 2005.

[17] B.-N. Vo and W.-K. Ma, "The Gaussian mixture probability hypothesis density filter," *IEEE Trans. Signal Proc.*, vol. 54, no. 11, pp. 4091 – 104, 2006.

[18] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic implementations of the cardinalized probability hypothesis density filter," *IEEE Trans. Signal Proc.*, vol. 55, no. 7, pp. 3553–3567, 2007.

[19] N.-T. Pham, W. Huang, and S. Ong, "Probability hypothesis density approach for multi-camera multi-object tracking," in *Proc. ACCV'07*, vol. I, Tokyo, Japan, November 2007, pp. 875–884.

[20] R. Hoseinnezhad, B.-N. Vo, D. Suter, and B.-T. Vo, "Multi-object filtering from image sequence without detection," in *ICASSP*, Dallas, TX, March 2010, pp. 1154–1157.

[21] R. Hoseinnezhad, B.-N. Vo, B.-T. Vo, and D. Suter, "Visual tracking of numerous targets via multi-Bernoulli filtering of image data," *Pattern Recognition*, vol. 45, no. 10, pp. 3625–3635, 2012.

[22] R. Hoseinnezhad, B.-N. Vo, and T. N. Vu, "Visual tracking of multiple targets by multi-Bernoulli filtering of background subtracted image data," in *International conference on Swarm Intelligence (ICSI'2011), Lecture Notes in Computer Science (LNCS)*, ser. 2, Y. Tan, Ed., vol. 6729, Chongqing, China, June 2011, pp. 509–518.

[23] A. Tyagi, M. Keck, J. W. Davis, and G. Potamianos, "Kernel-based 3D tracking," in *CVPR'07*, Minneapolis, Minnesota, USA, 2007.

[24] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1151 – 1162, 2002.

[25] B. Han, D. Comaniciu, Y. Zhu, and L. S. Davis, "Sequential kernel density approximation and its application to real-time visual tracking," *PAMI*, vol. 30, no. 7, pp. 1186–1197, 2008.

[26] R. Mahler, *Statistical multisource-multitarget information fusion*. Norwood, MA, USA: Artech House, 2007.

[27] R. Hoseinnezhad, B.-N. Vo, B.-T. Vo, and D. Suter, "Bayesian integration of audio and visual information for multi-target tracking using a CB-MeMBer filter," in *ICASSP'2011*. Prague, Czech Republic: IEEE, May 2011, pp. 2300–2303.

[28] H. Wang and D. Suter, "A consensus-based method for tracking: Modelling background scenario and foreground appearance," *Pattern Recognition*, vol. 40, no. 3, pp. 1091 – 105, 2007.