Visual Tracking of Numerous Targets via Multi-Bernoulli Filtering of Image Data

Reza Hoseinnezhad[a,*], Ba-Ngu Vo[b], Ba-Tuong Vo[b], David Suter[c]

[a] *RMIT University, Victoria, Australia*
[b] *The University of Western Australia, WA, Australia*
[c] *The University of Adelaide, SA, Australia*

Abstract

This paper presents a novel Bayesian method to track multiple targets in an image sequence without explicit detection. Our method is formulated based on finite set representation of the multi-target state and the recently developed multi-Bernoulli filter. Experimental results on sport player and cell tracking studies show that our method can automatically track numerous targets, and it outperforms the state-of-the-art in terms of false positive (false alarm) and false negative (missing) rates as detection error measures, and in terms of label switching rate and lost tracks ratio as tracking error measures.

Keywords: Random finite sets, Multi–target tracking, Visual tracking, Track–before–detect.

1. Introduction

Reliable visual tracking of multiple objects is an indispensable task in many emerging computer vision applications such as automatic video surveillance and robotics. In a multiple object setting, not only the states of the objects vary with time, but the number of objects also changes due to objects appearing and disappearing. At each sampling time, the states of these objects are partially observed as an image. The aim of multiple-object tracking is to jointly estimate the time-varying number of objects and their states from a stream of noisy images. The unknown and stochastically time-varying number of objects, the context-rich nature of visual measurements, and the sheer size of the measurements themselves, pose significant challenges in theory and practice.

Visual tracking techniques in the literature can be divided into three categories (see Figure 1). The first category, applies detection to each individual image – See Figure 1(a). Many detection techniques include a search routine to find the best regions occupied by targets. Data association and labeling are usually applied after detection and may involve specific sampling or search routines to resolve the occlusion (two targets merged into one detected region) and splitting (more than two separate detections belonging to the same target) events [1]. Popular detection approaches include the detection of targets based on matching color histograms of rectangular blobs [2], and background/foreground modeling via kernel density estimation [3]. Other recent methods include a game-theoretic approach [4], a deterministic method using sample consensus [5], human shape models [6], multi-modal representations [7], sample-based detection [8] and lazy background subtraction [9].

*Corresponding Author
Email addresses: rezah@rmit.edu.au (Reza Hoseinnezhad), Ba-Ngu.Vo@uwa.edu.au (Ba-Ngu Vo), Ba-Tuong.Vo@uwa.edu.au (Ba-Tuong Vo), David.Suter@adelaide.edu.au (David Suter)

Preprint: Pattern Recognition, Vol. 45, Issue 10, pp. 3625–3636, October 2012

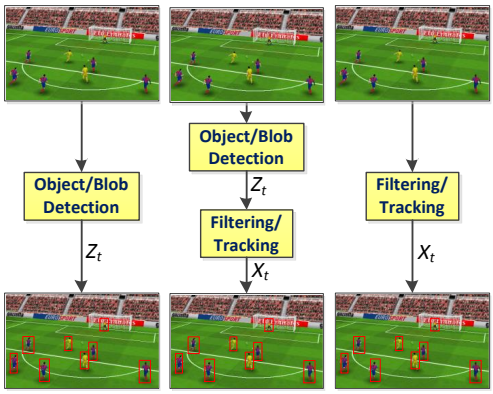


Figure 1: Three different approaches to visual tracking: (a) tracking based on detection only (b) Tracking by filtering the detection results to make the best use of the temporal information (c) Tracking by a filter directly operating on the whole image data.

In applications with relatively large number of targets (e.g. sports player tracking), detection methods are more likely to miss targets as well as declaring false targets, and visual tracking methods in the second category are preferred. Here, an additional filtering operation is required to keep track of all targets including those missed in the detection process –see Figure 1(b). Indeed, filtering makes it possible to exploit the available *temporal information* embedded in the data (such as the information about the dynamics of targets, the possible entry and exit areas, and the interactions between targets), and thus, yielding fewer missed targets and lower rates of false alarm compared to detection-only methods.

Typically, in the filtering module, motion correspondence or data association is first determined followed by the application of standard filtering techniques such as Kalman or particle filtering. The literature of visual tracking methods has a significant overlap with the target tracking literature as the filtering module essentially functions as a multi-target tracker. Indeed, the introduction of multi-target tracking techniques such as JPDA [10] and MHT [11] into visual tracking by Cox and Hingorani [12] and Chang et al. [13] had a big impact on the development of visual tracking techniques to date [14].

The third category–the focus of this paper–is of fundamental interest as it by-passes the detection module and exploits the spatio-temporal information directly from the image sequence – See Figure 1(c). This paper presents a novel multi-target visual tracking method which directly processes the sequence of images, without the need for explicit detection (extracting any point measurements). In a track without detection scheme, the rationale is to use all information in the image sequence, which should in principle produce better results (than detection-only or filtering detected outputs), especially in applications with low signal to noise ratio. The argument is that detection modules are designed to compress the information and therefore, while conceptually simpler and computationally cheaper, are far from completely loss-less.

Our proposed visual tracking technique is based on a Bayesian multi-target filtering solution developed from the random finite set (RFS) framework, known as the multi-Bernoulli filter [15]. In this framework the multi-target state is modeled as a RFS whose posterior distribution is propagated forward in time via the multi-target Bayes filter. The multi-Bernoulli filter is a tractable solution to the multi-target Bayes filter which propagates the multi-Bernoulli parameters of the multi-target posterior distribution. The salient features of the multi-Bernoulli filter are that it operates in the single-target state space, is highly paralellizable, and amenable to multiple sensor fusion. Our method

performs at its best in applications with numerous targets of similar visual patterns (e.g. sport players with specific short and shirt colors). Experimental results on several case studies show that our proposed method can automatically track numerous targets coming in and out of the scene.

The outline of this paper is as follows. In section 2, the current trend of visual track–before–detect is briefly reviewed and the existing challenges are discussed. Also, Bayesian multi-object estimation in a random finite set framework is introduced to allow the definition and computation of multi-object miss–distances which is a fundamental necessity in estimation theory. Section 3 presents basic definitions and the principles of finite set statistics (FISST). Section 4 presents the proposed visual multi-target tracking method and its implementation. The experimental results and a comparison of the performance of our method with other state-of-the-art techniques are given in section 5. A conclusion and possible extensions are presented in section 6.

2. Background

Tracking without detection can be regarded as a track-before-detect approach. Track-before-detect techniques are often required in tracking from radar imagery applications with low signal to noise ratio [16]. A number of tracking without detection methods have also appeared in the computer vision literature. Perhaps the most well-known method is Bramble [17]. Other tracking methods that can be categorized as track-before-detect include the works of Perez et al. [18] and Nummiaro et al. [19] based on color-based probabilistic tracking, Vermaak et al. [20] based on incorporating an existence process into a Bayesian filtering scheme for multi-target tracking, Okuma et al. [21], Vermaak et al. [22], and Czyz et al. [23] based on using multi-modality of distributions to track multiple targets in video.

2.1. Multi-target Estimation

The distinguishing feature of our approach is the representation of the multi-target (or multi-object) state as a finite set. Indeed, from an estimation viewpoint, the multi-target state should be represented by a finite set. Since the goal of estimation is to determine an estimate that is "close" to the true value, estimation is rather futile without a mathematically consistent notion of estimation error. The fundamental question is how to define a suitable metric between the true and estimated values. While the traditional practice of simply stacking individual states into a single vector does not admit a satisfactory metric for multi-target estimation error [15], a finite set representation of the multi-target state admits a mathematically consistent notion of estimation error since distance between sets is a well understood concept [24].

In the Bayesian estimation paradigm, the state and measurement are treated as realizations of random variables. Since the (multi-target) state is a finite set, the concept of a random finite set (RFS) is required to cast the multi-object estimation problem in the Bayesian framework. Mahler's Finite Set Statistics (FISST) provides powerful yet practical mathematical tools for dealing with RFSs [25]. The centerpiece of the RFS approach is the so-called Bayes multi-target filter, a generalization of the standard Bayes filter to propagate the posterior distribution of the multi-target state forward in time.

Direct application of the Bayes multi-target filter requires computation in large-dimensional spaces similar to methods such as Bramble [17], the work of Perez et al. [18] and Czyz et al. [23], which use sequential Monte Carlo (SMC) to approximate high dimensional integrals. While SMC is a versatile tool, it still suffers from the curse of dimensionality and sampling from high dimensional spaces is a challenging problem in computational statistics [26]. This is also the case with multi-target visual tracking methods that use reversible jump Markov chain Monte Carlo (RJMCMC) [27–30] to address state spaces with varying dimensionality. Moreover, RJMCMC is a batch processing approach and requires an unknown "burn in" period, thereby not suitable to be used in real-time. In the presence of numerous targets, SMC and RJMCMC become computationally prohibitive due to the very large number of samples required to approximate posterior distribution.

To circumvent the curse of dimensionality, a number of alternative approaches which exploit multi-modality on the single-target state space have been proposed. Examples of such methods include the multi-modal tracking methods of Vermaak et al. [22], Okuma et al. [21], Mahler's Probability Hypothesis Density (PHD) filter [25] and Cardinalized PHD filter [31]. These techniques do not suffer from high dimensionality due to numerous targets. Indeed, the sport player tracking example presented in [21] and the results of applying the method of [21] to sport player and cell tracking reported in this paper show that the multi-modality-based approach can be tractable in cases with numerous targets. The PHD and CPHD filters have been proposed as moment and cardinality approximation to the multi-target Bayes filter, while the relationship between the multi-modal tracking approaches of [21, 22] with Bayesian multi-target filtering have not been studied.

This paper presents a multi-target tracking method based on multi-Bernoulli filtering from image observations. In [32], multi-target filter based on multi-Bernoulli RFS was proposed for tracking multiple targets from point measurements. Apart from the use of multi-Bernoulli approximation for the distribution of the multi-target RFS state, this paper is very different to the current work where image data are directly processed without detection. In [15], random finite set conjugate priors including multi-Bernoulli are presented along with a generic multi-Bernoulli filtering scheme for a particular class of multi-target measurement models for radar images. Our paper proposes a novel multi-object measurement model for image observations in video sequences that admits multi-Bernoulli conjugate prior and hence, the generic multi-Bernoulli filter of [15] can be applied. We also incorporate labels to the filter so that the tracjectories of targets can be estimated.

2.2. Track Management

An essential component of any multi-target tracking system is *track management*: keeping an *identity* or *label* for each target.Track management is often performed by solving the *data association* problem (finding out which target corresponds to which measurement e.g. MHT [14]). This approach suffers from the combinatorial explosion in the number of hypothesis and results in a computational bottleneck in presence of numerous objects.

Similar to other methods formulated in the RFS framework [33, 34], the multi-Bernoulli filter does not require data association[1]. The multi-Bernoulli filter provides estimates for the number and states of targets without needing to know which track each target belongs to. Further processing is required to determine the target trajectories. Note that the RFS formulation can accommodate target trajectories by augmenting an identity variable to the state of each object. However, in addition to substantial savings in computational cost, managing the target labels separately after estimation of the number and states of targets in each frame gives us more flexibility in using the past history of state estimates efficiently in tracking.

In this paper, we introduce a graph theoretic label management technique which is similar to [35] but specially tailored to our intended applications of sport player and cell tracking with numerous targets. Our proposed technique uses a long history of past estimates to find out which label fits best to which currently estimated object. This makes the whole multi-target tracking method to exhibit satisfactory performance in handling occlusions which commonly occur in presence of numerous moving targets. It is also important to mention that although our label management technique is designed for images recorded by a static camera, the flexibility of separate label management allows the integration of alternative methods (specially tailored to the applications involving moving or zooming cameras) to the multi-object filter.

[1] Although the methods of [33, 34] are also formulated in the RFS framework, none of them directly process the image data as filter inputs.

3. Definitions and Notations

Mahler's Finite Set Statistics (FISST) provides practical mathematical tools for dealing with RFSs [25, 36], based on a notion of integration and density that is consistent with point process theory [37]. New efficient algorithms such as the Probability Hypothesis Density (PHD) [25] , Cardinalized PHD [31] and Multi-target Multi-Bernoulli filters [25] have attracted substantial interest from academia, defense, and commercial sectors. Some of these techniques have already been applied to various problems in visual tracking where point measurements (the results of target detections from image observations) are fed into RFS-based multi-target filters to return an estimate for the multi-target state. Our method is a novel track–before–detect method based on modelling the multi-object prior and posterior via multi-Bernoulli distribution.

In this section, we review the elements of FISST pertinent to this work. An RFS X on $\mathcal{X} \subseteq \mathbb{R}^d$ can be completely specified by a discrete distribution that characterizes the cardinality (number of points), and a family of joint distributions that characterize the distribution of the points conditional on the cardinality (symmetry is required to allow for all possible permutations).

In order to define the density of a RFS, we first need to define the notion of integration for finite sets. Consider a function $f : \mathcal{F}(\mathcal{X}) \to \mathbb{R}$ where $\mathcal{F}(\mathcal{X})$ denotes the set containing all finite subsets of $\mathcal{X}$. The integral of this function over a closed subset S of $\mathcal{X}$ is given as follows [25]:

$$\begin{aligned}\int_S f(X)\delta X &\triangleq \sum_{n=0}^{\infty} \frac{1}{n!} \int_{\underbrace{S \times \cdots \times S}_{n \text{ times}}} f(\{x_1, \ldots, x_n\})\, dx_1 \ldots dx_n \\ &= f(\emptyset) + \int_S f(\{x\})\, dx + \frac{1}{2}\int_{S\times S} f(\{x_1, x_2\})\, dx_1 dx_2 + \cdots \end{aligned} \tag{1}$$

The FISST density of a RFS is a function $\pi : \mathcal{F}(\mathcal{X}) \to \mathbb{R}^+ \cup \{0\}$ that satisfies the following equation:

$$\forall S \subseteq \mathcal{X}, \qquad P(X \subseteq S) = \int_S \pi(X)\, \delta X. \tag{2}$$

Set integrals and set derivatives in the FISST framework have been proved to be consistent with the measure theoretic notions of integration/density in probability theory [37].

In this paper, an image frame in a video sequence is denoted by $\mathbf{y} = [y_1 \ \cdots \ y_m]^\top$ where y_i is the i-th pixel value and m is the total number of pixels in each frame. Suppose there are n targets in the scene with states $x_1, \ldots, x_n$, at a particular instant. As discussed previously, we represent the multi-target state by a finite set $X = \{x_1, \ldots, x_n\}$. For a given multi-target state X, the conditional density of the image observation (likelihood) is denoted by $g(\mathbf{y}|X)$ and is assumed to be known.

In this work, we employ a special type of RFS called *multi-Bernoulli* RFS, which is defined as the union of M independent Bernoulli RFSs $X^{(i)}$. Each Bernoulli RFS is either empty or singleton with probabilities $1-r^{(i)}$ and $r^{(i)}$, respectively. The parameter $r^{(i)}$ is called the *probability of existence* of the i-th Bernoulli *component*. In the case $X^{(i)}$ is a singleton, its (only) element is distributed according to a probability density $p^{(i)}(\cdot)$. It has been shown that any multi-Bernoulli RFS is completely characterized by its parameters $\{(r^{(i)}, p^{(i)}(\cdot))\}_{i=1}^{M}$ [25, 32].

Models based on the finite set representation (including the multi-Bernoulli model) admit mathematically consistent error metrics for multi-object estimation [15, 24]. Compared with other RFS models such as Poisson and IID cluster, the multi-Bernoulli RFS has a significant advantage in SMC implementation. Multi-object estimation for Poisson and IID cluster models is cumbersome and error prone when particles are used to approximate the intensity function or PHD. Indeed, the PHD/CPHD conjugate prior for image data do not involve any normalization, and in SMC implementation, result in high variance of the particle weights. The multi-Bernoulli model does not suffer from this problem.

4. Visual Tracking Using the Multi-Bernoulli Filter

A multi-target Bayes filter is formulated in a RFS framework as follows. At time k, the posterior density of multi-target state is denoted by $\pi_k(\cdot|\mathbf{y}_{1:k})$. The multi-target Bayes recursion propagates the multi-target posterior density $\pi_k(\cdot|\mathbf{y}_{1:k})$, in time according to the following prediction and update stages:

$$\text{Prediction:} \quad \pi_{k|k-1}(X_k|\mathbf{y}_{1:k-1}) = \int f_{k|k-1}(X_k|X)\pi_{k-1}(X|\mathbf{y}_{1:k-1})\delta X \tag{3}$$

$$\text{Update:} \quad \pi_k(X_k|\mathbf{y}_{1:k}) = \frac{g(\mathbf{y}_k|X_k)\pi_{k|k-1}(X_k|\mathbf{y}_{1:k-1})}{\int g(\mathbf{y}_k|X)\pi_{k|k-1}(X|\mathbf{y}_{1:k-1})\delta X} \tag{4}$$

where $f_{k|k-1}(\cdot|\cdot)$ is the multi-target transition density from $k-1$ to k, and $g(\cdot|\cdot)$ is the measurement likelihood function. In sections 4.1-4.3, we will detail the multi-target transition density and measurement likelihood function. In section 4.4, we present the multi-Bernoulli filter to approximate the Bayes multi-target recursion along with a particle implementation. A graph-based method to effectively label the targets and estimate their trajectories using the outputs of multi-Bernoulli filter will be presented in section 4.5.

4.1. Multi-Target Dynamic Model

Formulation of the multi-target transition density in RFS framework enables us to apply various motion-related constraints in a mathematically consistent manner. Indeed, in addition to the underlying model of motion, physical constraints for the states of entering and exiting targets such as entrance/exit gateways (in people tracking) and road areas (in vehicle tracking) are captured in the multi-target transition density.

For a given multi-target state at time $k-1$, each target x_{k-1} is assumed to either continue to exist at time k with survival probability $p_{S,k}(x_{k-1})$ or die with probability $1-p_{S,k}(x_{k-1})$. If it survives, it moves to a new state x_k with probability density $f_{k|k-1}(x_k|x_{k-1})$ which is a known stochastic model for the single target motion.[2] Thus, given a state x_{k-1} at time $k-1$, its behavior at time k is modeled by a (single) Bernoulli RFS denoted by $S_{k|k-1}(x_{k-1})$, with parameters $r = p_{S,k}(x_{k-1})$ and $p(\cdot) = f_{k|k-1}(\cdot|x_{k-1})$. The multi-target state X_k at time k is given by the union of the Bernoulli sets and the new targets possibly appeared (born) at time k:

$$X_k = \left[\bigcup_{x_{k-1}\in X_{k-1}} S_{k|k-1}(x_{k-1})\right] \cup \Gamma_k \tag{5}$$

where Γ_k denotes the multi-Bernoulli RFS of new targets. Assuming that the RFSs constituting the above union are mutually independent, X_k is a multi-Bernoulli RFS conditional on X_{k-1}. The independence assumption neglects possible interactions between targets. Modeling target interaction will need substantially larger number of particles to be included in the implementation, and in case of numerous targets, combinatorial explosion can make the filter numerically intractable. The independence assumption, quite common in target tracking, aims to simplify calculation for real-time implementation [25, 38].

[2]Despite we use the same notation $f_{k|k-1}(\cdot|\cdot)$ both for single-target and for multi-target transition densities, there is no danger of confusion because for single-target, the arguments are vectors whereas for multi-target, the arguments are finite sets.

4.2. Observation Model and Likelihood Function

The performance of all visual tracking methods that include stochastic filtering, significantly depends on how the background and target regions are modeled and matched with regions within the image observations. Various background/foreground modeling methods have been introduced in the computer vision literature. There is a trade-off between computation time and accuracy of observation modeling. Sophisticated models, such as the articulated human foreground model presented in [6], include high-dimensional states which need substantial computation for effective sampling of particles within the object filtering scheme. Without loss of generality, we assume that each target is represented as a rectangular blob and each single-target state vector comprises the location and size of the blob in pixels, although blobs/templates of arbitrary shapes are possible. To allow the users to specify exactly what to track, our model employs a database of training blobs provided by the user.

Consider the image $\mathbf{y}$ in one frame from a video sequence. Given a multi-target state $X = \{x_1, \ldots, x_n\}$, we derive a measurement likelihood function $g(\mathbf{y}|X)$. From the image $\mathbf{y}$, we can compute the HSV color histogram of the image blob corresponding to each target location x_i. HSV color histograms are advantageous over RGB color histograms because the HSV coding decouples the intensity (i.e., value) from color (i.e., Hue and Saturation), and it is therefore relatively insensitive to illumination conditions.

We denote the histogram values for the i-th target x_i by the vector $v_i = \mathscr{H}_f(\mathbf{y}_{x_i})$. It is important to note that each histogram vector only depends on the contents of the image region covered by the target, which is denoted by $\mathbf{y}_{x_i}$. To ensure that histogram values represent probability distributions, each vector is normalized to sum to 1.

For the set of pixels that do not belong to any target (hypothetically background pixels), the HSV histogram can also be computed, and the values recorded are denoted by the vector $v_b = \mathscr{H}_b(\mathbf{y})$. The background color histogram can be reasonably assumed to have time-invariant statistics. Note that this is not equivalent to assuming a static background. Indeed, the background may change, but the proportion of the colors contributing to it are assumed to vary slightly with time. In applications where each target covers a relatively small portion of the image, movement of the targets will cause negligible changes in the color histogram of the unoccupied parts of the image (background), and we can reasonably assume that each component of the color histogram values vary only slightly (they are almost constant). Thus, the background histogram vector is only a function of the image $\mathbf{y}$.

We assume that the histograms of individual targets and the histogram of the background are mutually independent from each other, noting that as long as the targets do not largely occlude each other, they do not substantially affect each other's color histograms. Thus, the likelihood function can be formulated as follows:

$$g(\mathbf{y}|X) = g_b(\mathscr{H}_b(\mathbf{y})) \prod_{i=1}^{n} g_f(\mathscr{H}_f(\mathbf{y}_{x_i})) \tag{6}$$

where $g_b(\mathscr{H}_b(\mathbf{y}))$ is the likelihood of background histogram to be given by $\mathscr{H}_b(\mathbf{y})$, and $g_f(\mathscr{H}_f(\mathbf{y}_{x_i}))$ is the likelihood that a target is present in the image $\mathbf{y}$ with state x_i. The likelihood can also be expressed in terms of the observation and state values:

$$g(\mathbf{y}|X) = \mathbb{g}_b(\mathbf{y}) \prod_{i=1}^{n} \mathbb{g}_f(\mathbf{y}; x_i) \tag{7}$$

where $\mathbb{g}_b(\mathbf{y}) \triangleq g_b(\mathscr{H}_b(\mathbf{y}))$ and $\mathbb{g}_f(\mathbf{y}; x_i) \triangleq g_f(\mathscr{H}_f(\mathbf{y}_{x_i}))$. This likelihood function is said to be *separable* since it can be written as a product of function $\mathbb{g}_f(\mathbf{y}; x_i)$ values each depending only on one of target states, and the normalising factor, $\mathbb{g}_b(\mathbf{y})$, is independent from target states (only a function of $\mathbf{y}$). As it will be discussed later in section 4.4, it is this separable form of the likelihood function that enables the filter to update the multi-target state directly from image observation –see equations (14) and (15).

Figure 2: Samples of the training blobs for tracking the players of the red shirt team in PETS database.

4.3. Computing the measurement likelihood

In tracking, we (the user) need to specify what types of objects/targets to be tracked. For applications such as radar, the target profile/characteristics are known and the measurement likelihood function is given a priori. In this work, we consider tracking applications where an ensemble of training data is available in the form of blobs each containing a single target (from which a color histogram can be computed). Figure 2 show 30 samples of the 850 training blobs used to track the members of the red team in a football game (more details of the case study will be presented in section 5). The likelihood terms $g_i(\cdot)$ can be then computed using kernel density estimation over the training data using (8) explained below. The training database contains n_{train} vectors $\{v_j^*\}_{j=1}^{n_{\text{train}}}$ and each vector corresponds to the HSV color histogram of a training blob. For a given histogram v_i, the likelihood term g_i is then given by the following kernel density estimate:

$$g_f(v_i) = \frac{\xi}{n_{\text{train}}\, h^N} \sum_{j=1}^{n_{\text{train}}} \kappa\left(\frac{d(v_i, v_j^*)}{h}\right) \tag{8}$$

where $\kappa(\cdot)$ is the kernel function (Gaussians used in our experiments), h is the kernel bandwidth, N is the total number of bins in each histogram and $d(v_i, v_j^*)$ is the Bhattacharyya distance [18, 19, 21]:

$$d(v_i, v_j^*) = \left(1 - \sum_{r=1}^{N} \sqrt{v_j^*(r) v_i(r)}\right)^{\frac{1}{2}} \tag{9}$$

and ξ is the normalizing factor to ensure that the kernel density integrates to 1.

Sometimes the spatial layout of the blob color can also be exploited and embedded within the likelihood function. For example, the upper and lower halves of sport player images usually have different colors. In such cases, we follow [18, 19, 21] to divide each blob belonging to a single target into an upper and a lower region. For the case studies presented in this paper which involve sports player tracking, the upper and lower halves of the rectangular blobs are processed separately. This division takes place for both the training blobs and the blobs in the tracking region. We compute and save the HSV color histograms of the upper and lower sub-blobs of all the training blobs. For the i-th target, the histogram of the upper and lower sub-blobs, denoted by v_i^{upper} and v_i^{lower}, are also computed. Then the likelihood of each sub-blob are calculated using (8), and the blob likelihood is calculated from the upper and lower sub-blob likelihoods:

$$g_f(v_i) \quad = \quad g_f^{\text{upper}}(v_i^{\text{upper}})\, g_f^{\text{lower}}(v_i^{\text{lower}}). \tag{10}$$

It is important to note that it is not necessary to formulate and compute the $g_b(\cdot)$ function in our likelihood equation because as we will explain in section 4.4, only the g_f terms appear in the update formulas of the multi-Bernoulli filter.

4.4. Particle Multi-Bernoulli Filter

The multi-Bernoulli filter for the image data is derived from the Bayes filter (3)-(4) and hence, consists of a prediction step and an update step. The prediction and update steps propagate the multi-Bernoulli parameters of the multi-target posterior distribution forward in time.

The prediction step is formulated based on the dynamic model (5). Mahler [25] has shown that given the multi-Bernoulli parameters of X_{k-1} denoted by $\pi_{k-1} = \left\{\left(r_{k-1}^{(i)}, p_{k-1}^{(i)}(\cdot)\right)\right\}_{i=1}^{M_{k-1}}$, the predicted state is also a multi-Bernoulli RFS given below:

$$\pi_{k|k-1} = \left\{\left(r_{k|k-1}^{(i)}, p_{k|k-1}^{(i)}(\cdot)\right)\right\}_{i=1}^{M_{k-1}} \cup \left\{\left(r_{\Gamma,k}^{(i)}, p_{\Gamma,k}^{(i)}(\cdot)\right)\right\}_{i=1}^{M_{\Gamma,k}} \tag{11}$$

where $\left\{\left(r_{\Gamma,k}^{(i)}, p_{\Gamma,k}^{(i)}(\cdot)\right)\right\}_{i=1}^{M_{\Gamma,k}}$ are the multi-Bernoulli parameters of the RFS of spontaneously born targets, and the parameters $\left\{\left(r_{k|k-1}^{(i)}, p_{k|k-1}^{(i)}(\cdot)\right)\right\}_{i=1}^{M_{k-1}}$ are given by:

$$r_{k|k-1}^{(i)} = r_{k-1}^{(i)} \langle p_{k-1}^{(i)}(\cdot), p_{S,k}(\cdot)\rangle \tag{12}$$

$$p_{k|k-1}^{(i)}(x) = \frac{\langle f_{k|k-1}(x|\cdot), p_{k-1}^{(i)}(\cdot) p_{S,k}(\cdot)\rangle}{\langle p_{k-1}^{(i)}(\cdot), p_{S,k}(\cdot)\rangle} \tag{13}$$

where $\langle f_1(\cdot), f_2(\cdot)\rangle$ denotes the standard inner product $\int f_1(x) f_2(x) dx$. The birth parameters $\left\{\left(r_{\Gamma,k}^{(i)}, p_{\Gamma,k}^{(i)}(\cdot)\right)\right\}_{i=1}^{M_{\Gamma,k}}$ are user-defined and in section 5, we will explain how they are determined.

The predicted multi-Bernoulli parameters are then used as prior in the update step of the filtering process. Given a multi-Bernoulli prior $\{(r_{k|k-1}^{(i)}, p_{k|k-1}^{(i)}(\cdot))\}_{i=1}^{M}$ for the multi-target RFS, and a separable likelihood function (7), it follows from Corollary 3 in [15] that the posterior distribution of X, given by Bayes rule (4), is also multi-Bernoulli with the parameters:

$$r_k^{(i)} = \frac{r_{k|k-1}^{(i)} \langle p_{k|k-1}^{(i)}(\cdot), \mathrm{g}_f(\cdot, \mathbf{y})\rangle}{1 - r_{k|k-1}^{(i)} + r_{k|k-1}^{(i)} \langle p_{k|k-1}^{(i)}(\cdot), \mathrm{g}_f(\cdot, \mathbf{y})\rangle} \tag{14}$$

$$p_k^{(i)}(\cdot) = \frac{p_{k|k-1}^{(i)}(\cdot) \mathrm{g}_f(\cdot, \mathbf{y})}{\langle p_{k|k-1}^{(i)}(\cdot), \mathrm{g}_f(\cdot, \mathbf{y})\rangle}. \tag{15}$$

This result is one of the few known RFS conjugate priors – see [15] for other examples. Using the update results (14) and (15), we devise a complete object filtering scheme that takes the raw image sequence as input to directly track multiple targets.

An important characteristic of multi-Bernoulli update is the straightforward implementation of *multi-sensor* update. Assume that at time k, there exist $\mathfrak{M}$ images $\{\mathbf{y}_k^1, \ldots, \mathbf{y}_k^{\mathfrak{M}}\}$. For example, in a multiple-view tracking application, the images can be the multiple camera images transformed to a common image coordinate system using affine, homography or fundamental matrix transformations. Utilization of all image data then simply takes place in the update step of our filter where the calculations (14)-(15) are sequentially repeated for $\mathfrak{M}$ times in each of which $\mathrm{g}_f(\cdot, \mathbf{y})$ is replaced with $\mathrm{g}_f^s(\cdot, \mathbf{y}^s)$ for $s = 1, \ldots, \mathfrak{M}$. Note that each likelihood function $\mathrm{g}_f^s(\cdot, \mathbf{y}^s)$ incorporates the transformation to the common image coordinates. The multi-sensor update result is independent of the order of the individual updates [15].

The multi-Bernoulli framework also allows straightforward fusion with detection-based measurements. Suppose there are two measurements: one image measurement and one set of point measurements of the multi-target state. For example, the point measurements can be a set of blobs returned by a detection algorithm. In this case, data fusion can be performed by updating with the image measurement using equations (14)-(15) followed by the multi-Bernoulli update in [31] with the point measurements. Here the order of the update matters. The reason is that the multi-Bernoulli

update in [31] is not exact and it is sensible to perform the most accurate update (with the image measurement) first.

Since the Bayes recursion is generally intractable, a sequential Monte Carlo (SMC) implementation of our multi-Bernoulli filter is presented in this section. 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,

$$p_{k-1}^{(i)}(x) = \sum_{j=1}^{L_{k-1}^{(i)}} w_{(k-1)}^{(i,j)} \delta_{x_{k-1}^{(i,j)}}(x). \tag{16}$$

Two proposal densities $q_k^{(i)}(\cdot|x_{k-1}, \mathbf{y}_k)$ and $b_k^{(i)}(\cdot|\mathbf{y}_k)$ are chosen for the particles corresponding to existing targets and new born targets, respectively. Given these proposal densities, the prediction equations lead to the following equations for the parameters of the predicted particles [31]:

$$r_{P,k|k-1}^{(i)} = r_{k-1}^{(i)} \sum_{j=1}^{L_{k-1}^{(i)}} w_{k-1}^{(i,j)}\, p_{S,k}(x_{k-1}^{(i,j)}) \tag{17}$$

$$p_{P,k|k-1}^{(i)}(x) = \sum_{j=1}^{L_{k-1}^{(i)}} \tilde{w}_{P,k|k-1}^{(i,j)}\, \delta_{x_{P,k|k-1}^{(i,j)}}(x) \tag{18}$$

$$r_{\Gamma,k}^{(i)} = \text{parameters known from a presumed birth model} \tag{19}$$

$$p_{\Gamma,k}^{(i)}(x) = \sum_{j=1}^{L_{\Gamma,k}^{(i)}} \tilde{w}_{\Gamma,k}^{(i,j)}\, \delta_{x_{\Gamma,k}^{(i,j)}}(x) \tag{20}$$

where

$$\begin{aligned}
x_{P,k|k-1}^{(i,j)} &\sim q_k^{(i)}(\cdot|x_{k-1}^{(i,j)}, \mathbf{y}_k), && j = 1, \ldots, L_{k-1}^{(i)} \\
w_{P,k|k-1}^{(i,j)} &= \left[w_{k-1}^{(i,j)} f_{k|k-1}(x_{P,k|k-1}^{(i,j)}|x_{k-1}^{(i,j)})\; p_{S,k}(x_{k-1}^{(i,j)})\right] \Big/ q_k^{(i)}(x_{P,k|k-1}^{(i,j)}|x_{k-1}^{(i,j)}, \mathbf{y}_k) && j = 1, \ldots, L_{k-1}^{(i)} \\
\tilde{w}_{P,k|k-1}^{(i,j)} &= w_{P,k|k-1}^{(i,j)} \Big/ \textstyle\sum_{\ell=1}^{L_{k-1}^{(i)}} w_{P,k|k-1}^{(i,\ell)} && j = 1, \ldots, L_{k-1}^{(i)} \\
x_{\Gamma,k}^{(i,j)} &\sim b_k^{(i)}(\cdot|\mathbf{y}_k) && j = 1, \ldots, L_{\Gamma,k}^{(i)} \\
w_{\Gamma,k}^{(i,j)} &= p_{\Gamma,k}(x_{\Gamma,k}^{(i,j)}) \Big/ b_k^{(i)}(x_{\Gamma,k}^{(i,j)}|\mathbf{y}_k) && j = 1, \ldots, L_{\Gamma,k}^{(i)} \\
\tilde{w}_{\Gamma,k}^{(i,j)} &= w_{\Gamma,k}^{(i,j)} \Big/ \textstyle\sum_{\ell=1}^{L_{\Gamma,k}^{(i)}} w_{\Gamma,k}^{(i,\ell)}. && j = 1, \ldots, L_{\Gamma,k}^{(i)}
\end{aligned}$$

Let us denote the predicted multi-Bernoulli distribution by $\left\{\left(r_{k|k-1}^{(i)}, p_{k|k-1}^{(i)}(\cdot)\right)\right\}_{i=1}^{M_{k|k-1}}$ where $M_{k|k-1} = M_{k-1} + M_{\Gamma,k}$ and each predicted Bernoulli component density $p_{k|k-1}^{(i)}(\cdot)$ comprises $L_{k|k-1}^{(i)}$ particles, i.e.

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

Indeed, for $i = 1, \ldots, M_{k-1}$,

$$r_{k|k-1}^{(i)} = r_{P,k|k-1}^{(i)}\ ,\ L_{k|k-1}^{(i)} = L_{k-1}^{(i)}\ ,\ w_{k|k-1}^{(i,j)} = \tilde{w}_{P,k|k-1}^{(i,j)}\ ,\ x_{k|k-1}^{(i,j)} = x_{P,k|k-1}^{(i,j)}$$

and for $i = M_{k-1} + 1, \ldots, M_{k-1} + M_{\Gamma,k}$,

$$r_{k|k-1}^{(i)} = r_{\Gamma,k}^{(i-M_{k-1})}\ ,\ L_{k|k-1}^{(i)} = L_{\Gamma,k}^{(i-M_{k-1})}\ ,\ w_{k|k-1}^{(i,j)} = \tilde{w}_{\Gamma,k}^{(i-M_{k-1},j)}\ ,\ x_{k|k-1}^{(i,j)} = x_{\Gamma,k}^{(i-M_{k-1},j)}.$$

It is important to notice the highly parallelizable nature of the computations. Indeed, the predicted states of surviving and new Bernoulli targets can be computed in separate $M_{k-1} + M_{\Gamma,k}$ parallel processors.

In the update step, the predicted multi-Bernoulli parameters are updated using the likelihood function (8) and update formulas (14) and (15) which translate to:

$$r_k^{(i)} = r_{k|k-1}^{(i)} \varrho_k^{(i)} \Big/ \left(1 - r_{k|k-1}^{(i)} + r_{k|k-1}^{(i)} \varrho_k^{(i)}\right) \tag{22}$$

$$w_k^{(i,j)} = w_{k|k-1}^{(i,j)} \ \mathrm{g}_f(\mathbf{y_k}; x_{k|k-1}^{(i,j)}) / \varrho_k^{(i)} \tag{23}$$

where $\varrho_k^{(i)} = \sum_{j=1}^{L_{k|k-1}^{(i)}} w_{k|k-1}^{(i,j)} \ \mathrm{g}_f(\mathbf{y_k}; x_{k|k-1}^{(i,j)})$.

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 than a small threshold (set at 0.01) are removed. It is important to note that similar to the prediction step, the state update calculations are highly paralellizable. Indeed, all computations can be performed using separate M_k parallel processors.

Having the updated parameters of the multi-Bernoulli RFS representing the multi-target state, the number of targets and their states can be estimated via finding the existence probabilities that are larger than a threshold (set at 0.5 in our experiments). The EAP estimate for each target state can then be computed by the weighted average of the particles of the corresponding density.

4.5. Label Management

We first tried the common approach to label management with multi-object filters developed based on multi-Bernoulli approximations, where each Bernoulli target at time k simply inherits the label of its predecessor (in prediction step) if it belongs to one of the first predicted M_{k-1} Bernoulli targets, otherwise, it belongs to a newly born target and is assigned a new label, and the labels remain unchanged during the update step of the filter [25, 31]. However, this method works best when the targets are well separated. Indeed, when two players occlude each other and they are merged in one frame and separate in another frame, one of them would be detected by the filter as a newly born target, and would be assigned a new label (not its pre-merging label). The is the limitation of filtering which looks back one step at a time.

An effective remedy to this problem is to look at the results over several time steps. To implement this, we need a label management scheme with a *memory* to keep the labels of missed targets for later use in case they reappear. The recorded labels will only be removed from memory if they are missed for several consecutive frames. We have devised such a memory in a fundamental way using the framework suggested by Shafique and Shah [35] for matching in monocular image sequences. They construct a digraph containing all the targets detected in several consecutive frames (different layers of the digraph), and solve the label management problem via a non-iterative greedy search on the graph to find an optimum path cover which in tracking terms is an ensemble of tracks. In the method of [35], the digraph contains three types of edges: old edges which are the edges already existing in the tracks found at previous frame, extension edges which are the edges extending the previous tracks to one of the currently detected targets, and the correction edges which correct a previous mismatch (one or more old edges) by using the newly obtained information. The matching is performed by finding a set of vertex disjoint paths (tracks) that optimizes the total cost.[3] A candidate solution may contain all three types of edges, a correction edge would always replace some existing or old

[3]The term "gain" was used in [35], and it is maximized in their method. In our technique, nearest neighbor graph search is employed and therefore, we use the term "cost" which is the distance to be minimized.

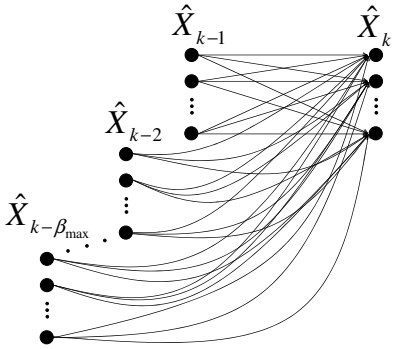


Figure 3: Schematic diagram of the bipartite graph matching used for label management.

edge which are called "false hypotheses". To replace the false hypotheses with correction edges, a recursive scheme is introduced in [35] as well as a non-recursive heuristic that is computationally cheaper.

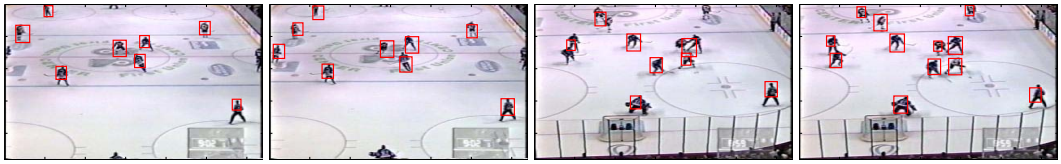
We have followed Shafique and Shah [35] to solve the label management problem in a graph theoretical framework in which the previously missed targets are taken into account besides the most recent targets, for their association with currently detected targets. However, to make the whole tracking feasible in real-time, we do not consider correction links. We suggest one should employ a three-stage constrained nearest neighbor graph search to find candidate tracks in a bipartite graph shown in Figure 3. The search is constrained in the sense that an edge is chosen only if its cost is less than a certain limit (this accounts for a limit for target displacements from one frame to the next). In stage 1, the subgraph including current estimates $\hat{X}_k$ and most recent estimates $\hat{X}_{k-1}$ is searched for associations. In stage 2, the constrained search is carried out for the subgraph including the previously missed targets $(\hat{X}_{k-2}, \ldots, \hat{X}_{k-\beta_{\max}})$ and the current estimates not associated yet. Finally, the remained current estimates are checked for their eligibility to be considered as newly born targets based on their distance from the image borders.

5. Simulation Results

We have evaluated the performance of our method in four challenging case studies, three of which involve tracking of multiple sport players. In the third case study, we detect and track multiple cells splitting and dying in a micro-well, using microscopic image sequences. Videos of all case studies are provided as supplemental material.

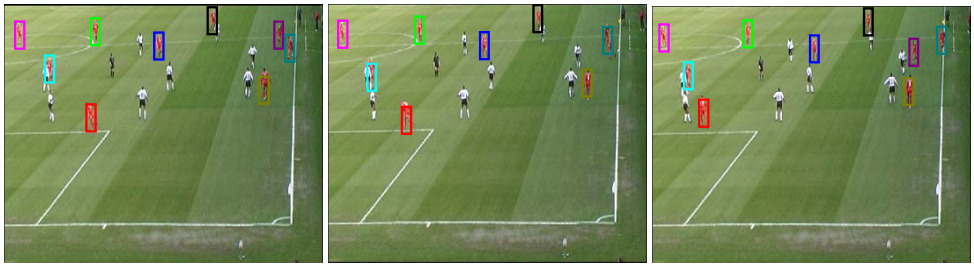
In all cases, we assume a constant survival probability P_S, and consider a predefined model for birth particles denoted by known parameters $\{r_\Gamma^{(i)}, p_{\Gamma,k}^{(i)}\}_{i=1}^{M_\Gamma}$ where the density $p_{\Gamma,k}^{(i)}$ is represented by the particles $\{w_{\Gamma,k}^{(i,j)}, x_{\Gamma,k}^{(i,j)}\}_{j=1}^{L_\Gamma}$. 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.

For the proposal density we use the state transition density $f_{k|k-1}(\cdot|x_{k-1})$. In the first three experiments, the targets are modeled by rectangular blobs and the target state is a 4-tuple vector comprising the x and y location and width and height. In the cell tracking experiment, we have used circular blobs, each state denoted by $[x\ y\ r]^\top$ where x and y are the center location and r is the blob radius. For the target dynamics, we have assumed the least informative model (random walk) to show the tracking power of our method. In a random walk model, each single target state follows the equation



(a) Frame 17 (b) Frame 22 (c) Frame 96 (d) Frame 101

Figure 4: Four snapshots of the results of hockey player tracking: (a) A new player is entering the scene from left side (b) The new player is detected and tracked (c) Three couples of very close players are merged into single targets (d) The merged targets are separately tracked as soon as the players get separated.



(a) Frame 46 (b) Frame 53 (c) Frame 79

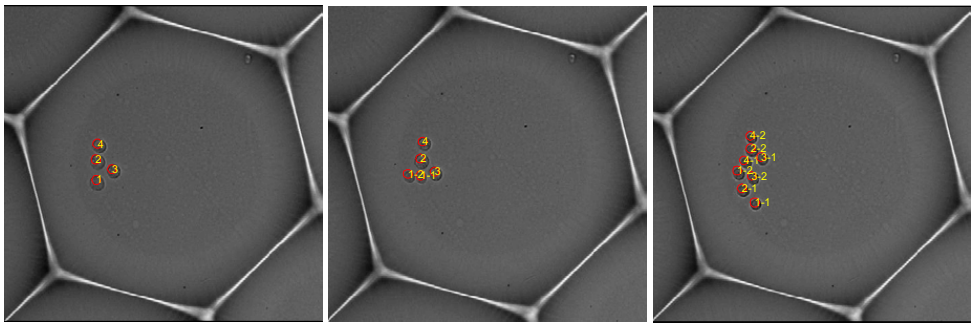
Figure 5: The tracking results for the red team players in an outdoor soccer game: (a) nine players are detected and tracked (b) Two players are merged – look at the right side of the image (c) The players are separated and tracked with their previous labels (border colors).

$x(k+1) = x(k) + e(k)$ where $e(k)$ is a 4-dimensional Gaussian variable with zero mean and variance $\psi = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_h^2, \sigma_w^2)$ in the case of a rectangular blob model. Similarly, for a circular blob model, $e(k)$ is a 3-dimensional Gaussian variable with zero mean and variance $\psi = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_r^2)$. In both cases, the single target transition density function is $f_{k|k-1}(x|x_{k-1}) = \mathcal{N}(x; x_{k-1}, \psi)$.

Other parameters of the filter are initialized as follows. The number of Bernoulli components is set at an over-estimate of the maximum number of targets that can possibly exist in the scene. In the sport player tracking scenarios, we have $M_0 = 15$, and in the cell tracking, $M_0 = 20$. The existence probabilities of all Bernoulli components are set at a small value ($r_0^{(i)} = 0.01, \;\; i = 1, \ldots, M_0$ in our case studies). Depending on the actual number of targets in the scene, some of these values gradually increase as the measurements are incorporated in the update step of estimation process. For each Bernoulli component, the particles are initially chosen with equal weights with their states uniformly distributed in their admissible ranges. For example, in player tracking scenarios, the location of the particle blobs are initialized uniformly distributed in the image plane while the width and height of the blobs uniformly distributed in the practical range of the size of a blob that can contain a target (e.g. 10-20 pixels $\times$ 15-30 pixels).

The first experiment includes 101 frames of a hockey game (benchmarked in [21]). Snapshots of the tracking results are presented in Figure 4. We recorded the HSV histograms of 1500 training rectangular blobs, each manually selected to contain a player.

In the second experiment, we tracked the red team players in 2500 frames of an outdoor soccer image sequence downloaded from the PETS database website (`http://www.cvg.cs.rdg.ac.uk/VSPETS/vspets-db.html`). Two videos (one for "training" and one for "testing") were used. We recorded the HSV histograms of only 850 blobs selected from the training image frames, and used them to compute the kernel density estimates in the likelihood function. In this experiment, we have also labeled the players (see the color blobs). Figure 5 shows snapshots of the tracking results for the PETS outdoor football dataset.



(a) Fr‘ame 170 (b) Frame 177 (c) Frame 421

Figure 6: Snapshots of cell tracking results: (a) There are four cells in the micro-well initially, labeled with numbers 1–4. (b) The cell number 1 is split and the new two cells are tracked with new labels 1-1 and 1-2. (c) After a while, all four cells are split and their labels assigned accordingly. The new cells continue to split and at the end of the experiment, there are 9 alive and 4 dead cells.

The third case study involves detection and tracking of up to 13 cells contained in a micro-well from which two microscopic images are taken in every 8 minutes. The first image is a photographic image (called the “bright field (BF)” image). The second image is obtained via contaminating the cells with green fluorescent protein (GFP) and exposing them to blue light which causes them to exhibit bright green fluorescence. In these images, called “GFP images”, the living cells appear in green color against the dark background, and dead cells do not appear (though they do in the BF images). Therefore, to track only the living cells, the GFP images are also needed.

In this experiment, we demonstrate the ability of the multi-Bernoulli visual filter to fuse data from different sources, namely bright field images and detections from GFP images where most target regions commonly have a relatively good contrast to the background. Applying denoising and level thresholding followed by a few morphological operations on each image, we can obtain a set of point measurements where each measurement is an estimate of the location and size of a cell. Cells can be missed in this detection process, and false alarms are also possible (though they are rare). However, combining these information with the BF image data (color histogram information) will compensate for the misses and false alarms in most of the times and is expected to drastically improve the performance of our multi-target tracking method.

In order to effectively fuse the information encapsulated in the two image sequences, we update the multi-target state twice. First, we use the point measurements extracted from the GFP image within the cardinality-balanced MeMBer filter [31]. Then, we update the multi-Bernoulli parameters again, using equations (22)-(23) in which the bright field image information and training blobs are used.

The sequence includes 1339 frames. There are initially four cells in the micro-well, and they split each into two cells every now and then, and the new cells are split later and so on. There are up to 13 cells to be tracked in the experiment, four of which die during the last 200 frames. This is a challenging case, as the cells are usually very close to each other and there is a high chance of them being merged in the tracking results. In addition, there is sometimes a low contrast between the cells and their surroundings. Furthermore, both the shape and contents of the cells vary with time, although the color histogram of their contents do not change substantially.

Snapshots of the cell tracking results are shown in Figure 6. Each cell is labeled with a sequence of numbers using the label management method presented in section 4.5. The labeling of new detected cells, however, is a bit different as new cells are the result of a cell being split not a new cell entering the scene. Therefore, each new cell has a parent and inherits the label of its parent as part of its own label.

Table 1: Comparative results for detection performance of the examined methods.

Method	Hockey		PETS Football		Cell Tracking		Indoor Football	
	FAR	FNR	FAR	FNR	FAR	FNR	FAR	FNR
KDE[3]	8%	11%	3%	9%	48%	62%	4%	5%
SACON[5]	11%	16%	6%	13%	56%	75%	5%	3%
BPF[21]	4%	9%	7%	8%	13%	21%	8%	11%
Our Method	0.5%	2%	0.25%	4%	0.3%	12%	2.1%	8%

Table 2: Comparative results for tracking performance of the examined methods.

Method	PETS Football		Cell Tracking	
	LSR	LTR	LSR	LTR
KDE[3]	54%	13%	75%	62%
SACON[5]	41%	21%	86%	53%
BPF[21]	28%	5%	33%	10%
Our Method	20%	2%	11%	1.5%

Figures 4–6 show that our method is capable of tracking numerous targets at the same time. In addition, they demonstrate the ability of the method to inherently detect and track the arrival of new targets. When there is substantial occlusion, the merging step in our tracking method may result in a single blob for the occluding targets. However, as Figures 4–6 show, as soon as the moving targets separate from each other, the tracker corrects its results. We have quantitatively compared the performance of our method with three visual tracking methods: visual tracking via kernel density estimation (KDE) [3], sample consensus (SACON) [5] and boosted particle filter [21].

In KDE, kernel density estimates are used for nonparametric modelling of background and foreground objects. KDE is straightforward and we developed the code for our experiments. SACON is deterministic and models the background and foreground objects using sample consensus. We used the Matlab code provided by the authors of [5]. BPF is based on training a detection module using a colour histogram dataset and ADABoost, then a mixture particle filter. We used the Matlab code provided in the website of first author of [21], `http://people.cs.ubc.ca/~okumak/research.html`

The detection performance is quantified in terms of two measures, false negative rate (FNR) and false alarm rate (FAR). The two rates are defined as the total number of targets that are missed and the total number of non-existing targets that are detected, both normalized over the accumulative total number of true targets over all frames. These measures have been computed for the all case studies and the results are presented in Table 1.

To evaluate the tracking performance of our method compared to the others, two other measures are computed, label switching rate (LSR) and lost tracks ratio (LTR). The label switching rate is the number of label switching events normalized over total number of ground truth tracks crossing events. The lost tracks ratio is the number of tracks lost for more than 50% of their lifetime normalized over total number of ground truth tracks. The track switching events happen when two targets get very close to each other (and they are sometimes merged into one target in detection results) and after they are separated, their labels are switched.

With KDE and SACON, the outputs contain only target states and we assign labels to targets by applying our label management method directly to the outputs of these methods. The comparison results for tracking performance of the examined methods are presented in Table 2.

The comparison results presented in Tables 1 and 2 demonstrate that our method significantly outperforms the other examined method in detection as well as tracking. In the cell tracking case study, the large errors of the compared methods are not tolerable. This is because of the low signal to

(a) Frame 180 (b) Frame 193 (c) Frame 200 (d) Frame 205

Figure 7: Four snapshots of football player tracking: (a) A new player enters the scene (b) The player is detected and tracked – see the left side of the image (c) Two very close players are merged into a single target (d) They are separately tracked when the players get apart.

noise ratio in the BF images and the large detection errors of the two methods in such cases, which also leads to a lot of missed tracks and switched labels in the tracking. Our method handles the low signal to noise ratio very well as it directly processes the whole image and uses all the information embedded and not just the point measurements extracted from it (which are usually highly erroneous in low SNR).

5.1. Limitations

Our method works best with relatively small-size targets. This is probably due to the fact that with large targets, the background histogram likelihood $g_b(v_b)$ in equation (6) will substantially vary with targets states. As a result, the term $\mathrm{g}_b(\mathbf{y})$ in equation (7) will need to be corrected to $\mathrm{g}_b(\mathbf{y}; X)$ which will no longer have a separable form as required by the Bayes filter employed by our method. To examine the extent of this limitation, we employed our method to jointly detect and track a number of indoor football players over 700 frames of a video sequence. Unlike our other experiments, in this sequence, the players normally occupy a relatively large portion of the image. Indeed, each player roughly covers 2.3% of the image area and in presence of 5 targets in average, a total of 11.5% of the image data would belong to target areas. This is while in the cases involving the tracking of hockey players, outdoor soccer players and cells, the targets roughly cover only 1.5%, 1.2%, and 1.3% of the image area in average.

We have compared the false alarm and false negative rates of our method with the other techniques and the results are listed in Table 1. In this experiment, KDE and SACON seem to result in smaller false negative rate which is (as previously mentioned) probably due to the inefficiency caused by inaccurate assumptions about the background histogram. Nonetheless, our technique is robust enough to exhibit a satisfactory performance even with large targets and new players entering the scene are effectively detected and tracked and those leaving the scene are quickly removed from the tracking results.

The number of training blobs was 2200 and we deliberately did not select any training blob containing one specific player (the one with the light pink stripy shirt). The main purpose of this exclusion was to examine the selectiveness of tracking (e.g. tracking the players and not the referee). As it appears in the tracking results (snapshots shown in Figure 7), that player is not picked by the tracker at any time.

6. Conclusion and Discussion

We have presented a novel multi-target tracking method capable of jointly detecting and tracking directly from image observations without the need for any separate target detection and extraction of point measurements. We derived a separable multi-target measurement likelihood function, which

enables us to exploit the efficiency of the multi-Bernoulli filter to estimate the number and states of targets. A graph-theoretic label management technique is then applied to multi-Bernoulli filter output to obtain target trajectories. The four case studies have demonstrated that our method is capable of detection and tracking of numerous interacting targets, and can tolerate a low level of signal to noise ratio. Comparison results also show that in general, our method outperforms the other examined techniques in terms of both false alarm and false negative rates (detection error) and label switching rates and lost track ratios (tracking error).

Our method is most useful when a relatively large training dataset for the targets of interest are available. In applications where large training datasets are not available we need to develop new separable likelihood functions that does not rely on training data. Instead, the target feature can be employed within the routine (including its likelihood function) as they become available. For example, background pixel likelihoods obtained by kernel density estimation would be good candidates for such features. Another solution is to pick previously detected objects (with large probability of existence) and use them as training blobs to compute the likelihood function. This approach might be sensitive to errors and further developments are needed to achieve an acceptable level of robustness.

The proposed algorithm can be significantly accelerated via parallel implementation. The multi-Bernoulli filter is highly parallelizable since there is very little overlap between different Bernoulli components in the propagation of the parameters $r^{(i)}$ and $p^{(i)}(\cdot)$. We believe that the multi-Bernoulli algorithm can be implemented in M parallel processors, where M is the maximum number of Bernoulli targets. The parallelization can be implemented in more depth, where the computation of the particle weights for each Bernoulli target are also implemented in parallel based on the work of Medeiros et al. [39], presenting a parallel implementation of color-based particle filters (based on color histograms). An implementation of color-based particle filtering, suitable for SIMD processors, has been presented in [39]. A faster implementation of the multi-Bernoulli filter presented in this paper can also be designed based on efficient particle sampling in which the particle distribution is hierarchically guided from coarse to fine templates.

Acknowledgment

This work was supported by Australian Research Council through the ARC Discovery Project grant DP0880553. Authors would like to thank Dr John Markham and Dr Cameron Wellard at Walter and Eliza Hall Institute of Medical Research (WEHI) for provision of cell images, and Dr Hanzi Wang at the University of Adelaide for provision of the Matlab source code for SACON tracking method.

References

[1] Q. Yu, G. Medioni, Multiple-target tracking by spatiotemporal Monte Carlo Markov chain data association, PAMI 31 (12) (2009) 2196–2210.

[2] M. Kristan, J. Per, M. Pere, S. Kovacic, Closed-world tracking of multiple interacting targets for indoor-sports applications, CVIU 113 (5) (2009) 598 – 611.

[3] A. Elgammal, R. Duraiswami, D. Harwood, L. S. Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, Proc. of the IEEE 90 (7) (2002) 1151 – 1162.

[4] M. Yang, T. Yu, Y. Wu, Game-theoretic multiple target tracking, in: ICCV'07, Rio de Janeiro, Brazil, 2007.

[5] H. Wang, D. Suter, A consensus-based method for tracking: Modelling background scenario and foreground appearance, Pattern Recognition 40 (3) (2007) 1091 – 105.

[6] T. Zhao, R. Nevatia, B. Wu, Segmentation and tracking of multiple humans in crowded environments, PAMI 30 (7) (2008) 1198 – 211.

[7] S. Apewokin, B. Valentine, R. Bales, L. Wills, S. Wills, Tracking multiple pedestrians in real-time using kinematics, in: CVPR'08 Workshops, Anchorage, AK, United states, 2008.

[8] L. Zhu, J. Zhou, J. Song, Tracking multiple objects through occlusion with online sampling and position estimation, Pattern Recognition 41 (8) (2008) 2447 – 2460.

[9] R. Abbott, L. Williams, Multiple target tracking with lazy background subtraction and connected components analysis, Machine Vision and Applications 20 (2) (2009) 93 – 101.
[10] Y. Bar-Shalom, T. Fortmann, M. Scheffe, Joint probabilistic data association for multiple targets in clutter, Proc. Conf. Information Sciences and Systems.
[11] D. Reid, An algorithm for tracking multiple targets, IEEE Transactions on Automatic Control 24 (6) (1979) 843–854.
[12] I. Cox, S. Hingorani, An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking, PAMI 18 (2) (1996) 138–150.
[13] K.-C. Chang, S. Mori, C.-Y. Chong, Evaluating a multiple-hypothesis multitarget tracking algorithm, IEEE Tran. AES 20 (2) (1994) 578–590.
[14] S.-W. Joo, R. Chellappa, A multiple-hypothesis approach for multiobject visual tracking, IEEE TIP 16 (11) (2007) 2849 – 2854.
[15] B.-N. Vo, B.-T. Vo, N.-T. Pham, D. Suter, Joint detection and estimation of multiple objects from image observations, IEEE TSP 58 (10) (2010) 5129–5141.
[16] S. Buzzi, M. Lops, L. Venturino, M. Ferri, Track-before-detect procedures in a multi-target environment, IEEE Transactions on Aerospace and Electronic Systems 44 (3) (2008/07/) 1135 – 50.
[17] M. Isard, J. MacCormick, BraMBLe: a Bayesian multiple-blob tracker, in: ICCV'01, Vol. 2, Vancouver, British Columbia, Canada, 2001, pp. 34 – 41.
[18] P. Perez, C. Hue, J. Vermaak, M. Gangnet, Color-based probabilistic tracking, in: ECCV, Copenhagen, Denmark, 2002, pp. 661 – 75.
[19] K. Nummiaro, E. Koller-Meier, L. Van Gool, Object tracking with an adaptive color-based particle filter, in: DAGM, Zurich, Switzerland, 2002, pp. 353 – 60.
[20] J. Vermaak, S. Maskell, M. Briers, P. Perez, Bayesian visual tracking with existence process, in: ICIP, Atlanta, GA, USA, 2006, pp. 721 – 4.
[21] K. Okuma, A. Taleghani, N. De Freitas, J. Little, D. Lowe, A boosted particle filter: Multitarget detection and tracking, in: ECCV'04, Vol. 3021, 2004, pp. 28–39.
[22] J. Vermaak, A. Doucet, P. Perez, Maintaining multi-modality through mixture tracking, Vol. 2, Nice, France, 2003, pp. 1110 – 1116.
[23] J. Czyz, B. Ristic, B. Macq, A particle filter for joint detection and tracking of color objects, Image and Vision Computing 25 (8) (2007) 1271–1281.
[24] D. Schuhmacher, B.-T. Vo, B.-N. Vo, A consistent metric for performance evaluation in multi-object filtering, IEEE TSP 56 (8) (2008) 3447–3457.
[25] R. Mahler, Statistical multisource-multitarget information fusion, Artech House, Norwood, MA, USA, 2007.
[26] P. Moral, Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications, Springer series in statistics: Probability and its applications, Springer, 2004.
[27] K. Smith, D. Gatica-Perez, J. Odobez, Using particles to track varying numbers of objects, in: CVPR, San Diego, USA, 2005.
[28] K. Smith, S. Ba, D. Gatica-Perez, J. Odobez, Tracking the multi-person wandering visual focus of attention, in: Intl. Conference on Multimodal Interfaces (ICMI), Banff, Canada, 2006.
[29] Z. Khan, T. Balch, F. Dellaert, MCMC-based particle filtering for tracking a variable number of interacting targets, PAMI 27 (11) (2005) 1805–1918.
[30] T. Zhao, R. Nevatia, Tracking multiple humans in crowded environment, in: CVPR, Vol. II, Washington D.C., USA, 2004, pp. 406 – 413.
[31] B.-T. Vo, B.-N. Vo, A. Cantoni, Analytic implementations of the Cardinalized Probability Hypothesis Density filter, IEEE TSP 55 (7) (2007) 3553–3567.
[32] B.-T. Vo, B.-N. Vo, A. Cantoni, The cardinality balanced multi-target multi-Bernoulli filter and its implementations, IEEE TSP 57 (2) (2009) 409–423.
[33] N. T. Pham, H. Weimin, S. Ong, Tracking multiple objects using probability hypothesis density filter and color measurements, in: IEEE Int. Conf. Multimedia and Expo, ICME'07, Beijing, China, 2007, pp. 1511 – 1514.
[34] Y.-D. Wang, J.-K. Wu, W. Huang, A. A. Kassim, Gaussian mixture probability hypothesis density for visual people tracking, in: Int. Conf. Information Fusion, Quebec City, Canada, 2007.
[35] K. Shafique, M. Shah, A noniterative greedy algorithm for multiframe point correspondence, PAMI 27 (1) (2005) 51–65.
[36] R. P. Mahler, Multitarget Bayes filtering via first-order multitarget moments, IEEE Trans. Aerospace and Elec. Sys. 39 (4) (2003) 1152 – 1178.
[37] B.-N. Vo, S. Singh, A. Doucet, Sequential Monte Carlo methods for multi-target filtering with random finite sets, IEEE Tran. AES 41 (4) (2005) 1224–1245.
[38] S. S. Blackman, R. Popoli, Design and Analysis of Modern Tracking Systems, Artech House, 1999.
[39] H. Medeiros, J. Park, A. Kak, A parallel color-based particle filter for object tracking, in: CVPR, Anchorage, Alaska, USA, 2008, pp. 1–8.