Using Particles to Track Varying Numbers of Interacting People

Kevin Smith, Daniel Gatica-Perez, and Jean-Marc Odobez *
IDIAP Research Institute, Martigny, Switzerland
{smith, gatica, odobez}@idiap.ch

Abstract

In this paper, we present a Bayesian framework for the fully automatic tracking of a variable number of interacting targets using a fixed camera. This framework uses a joint multi-object state-space formulation and a trans-dimensional Markov Chain Monte Carlo (MCMC) particle filter to recursively estimate the multi-object configuration and efficiently search the state-space. We also define a global observation model comprised of color and binary measurements capable of discriminating between different numbers of objects in the scene. We present results which show that our method is capable of tracking varying numbers of people through several challenging real-world tracking situations such as full/partial occlusion and entering/leaving the scene.

1. Introduction

Tracking a fixed number of independent, hand-initialized objects is a well studied problem. However, the automatic detection and tracking of a variable number of interacting objects is still difficult, implying three challenging tasks: (1) reliably estimating the number of objects in the scene, (2) keeping the algorithm computationally tractable when multiple objects appear simultaneously, and (3) modeling interactions between varying numbers of objects.

To address the first problem, we define an observation model which uses segmented binary information taken from background subtraction, along with foreground and background color information, to predict the number of objects in the scene. The works of [13, 5] highlight the need for a global observation model to track multi-object configurations of varying size. The work in [13] used binary observations derived from background subtraction, but in a substantially different way than we present here. In [5], a grid-based global appearance likelihood combining ideas from background modeling and Bayesian correlation is used.

To keep the algorithm computationally tractable, we use a Bayesian approach to define a robust and efficient particle filter (PF) capable of automatically tracking a variable number of interacting objects. In this framework, a trans-dimensional (reversible jump) MCMC algorithm is used to sample from the distribution of states given observations. The benefits of our formulation include: efficient sampling, a state vector of variable dimension (to handle varying numbers of objects), an explicit model for proximity-based interactions, and the ability to handle multi-modality in a multi-object observation model.

There is an abundance of literature devoted to the PF approach to multi-object tracking (MOT) [9, 13, 5, 10, 14]. Methods using a single-object state-space model are usually computationally inexpensive [10]. Such methods are often equivalent to multiple single-object trackers in parallel. A shortcoming of this approach is that identities and interactions between objects can not be easily modeled in formal (and algorithmic) terms. For these reasons, many adopt a rigorous formulation of the MOT problem using a joint state space [9, 13, 5, 6, 15, 14], where object interactions can be properly defined. However, sampling from a joint state space can quickly become inefficient as the dimension of the space increases.

Recently, finding an efficient MOT sampling method has become a topic of much research. A joint state-space model was proposed to efficiently track a fixed number of interacting objects using a sampling method that combines a PF formulation with MCMC sampling in [6]. This approach addressed the problem of interaction as well, by defining a pairwise Markov Random Field (MRF) prior in the dynamical model, which is more computationally tractable than other methods [9]. The work of [15] proposed a similar formulation with MCMC sampling. Our work generalizes the approach of [6] to handle a variable number of interacting objects using a global non discriminant observation model (instead of the template-based model used in [6]). Specific differences between our work and [6, 15] are detailed in later sections.

This paper is organized as follows: after presenting our approach in Section 2, we evaluate the algorithm in Section 3, and provide some concluding remarks in Section 4.

*This work was supported by the Swiss National Center of Competence in Research on Interactive Multimodal Information Management (IM2), and the EC project Augmented Multi-party Interaction (AMI, pub. AMI-63).

2. Our approach

2.1. Bayesian multi-object tracking with PFs

The Bayesian formulation of the tracking problem is well known. Given a Markov state-space model, with hidden states X_t representing the joint multi-object configuration, and observations Y_t extracted from a scene, the filtering distribution $p(X_t|Y_{1:t})$ of X_t given all observations $Y_{1:t} = (Y_1, ..., Y_t)$ up to time t is recursively computed by

$$p(X_t|Y_{1:t}) = Z^{-1}p(Y_t|X_t) \cdot \int_{X_{t-1}} p(X_t|X_{t-1})p(X_{t-1}|Y_{1:t-1})dX_{t-1}, \quad (1)$$

where $p(X_t|X_{t-1})$ is the dynamical model defining the (predictive) temporal evolution of the multi-object configurations, $p(Y_t|X_t)$ denotes the observation likelihood (which measures how well the observations fit the predictions), and Z is a normalization constant.

PFs approximate Eq. 1 for non-linear, non-Gaussian problems like visual tracking. From the various available formulations to derive the basic PF, we use the one described in [4]. The filtering distribution is represented by a set of particles (weighted configurations) $\{(X_t^{(n)}, w_t^{(n)}), n = 1, ..., N\}$, where $X_t^{(n)}$ and $w_t^{(n)}$ denote the n-th sample and its associated weight at each time-step. Eq. 1 is recursively approximated by

$$p(X_t|Y_{1:t}) \approx Z^{-1}p(Y_t|X_t)\sum_n w_{t-1}^{(n)}p(X_t|X_{t-1}^{(n)}) \quad (2)$$

using importance sampling. Given the particle set from the previous time $\{(X_{t-1}^{(n)}, w_{t-1}^{(n)})\}$, configurations at the current time-step $X_t^{(n)}$ are drawn from a proposal distribution $q(X_t) = \sum_r w_{t-1}^{(r)}p(X_t|X_{t-1}^{(r)})$. The weights are then computed as $w_t^{(n)} \propto p(Y_t|X_t^{(n)})$.

2.2. State-space definition

A state at time t is a multi-object configuration defined by $X_t = \{X_{i,t}, i \in \mathcal{I}_t\}$, where $\mathcal{I}_t$ is the set of object indexes and $m_t = |\mathcal{I}_t|$ denotes the number of objects, ($m_t \in \mathcal{M} = \{0, ..., M\}$, where M is the maximum allowed number of objects and $|\cdot|$ indicates set cardinality). $X_{i,t}$ denotes a single-object configuration and $X_{i,t} = \emptyset$ denotes a zero object configuration. This definition allows the number of objects to vary instead of remain fixed, as in [6]. In this work, $X_{i,t}$ is a continuous vector in a space of transformations $\mathbb{R}^{N_x}$, where $\mathbb{R}^{N_x}$ is a four-dimensional subspace of the affine transformations including horizontal and vertical translation and scaling, $X_{i,t} = (X_{i,t}^{tx}, X_{i,t}^{ty}, X_{i,t}^{sx}, X_{i,t}^{sy})$.

2.3. PF for interacting objects

The simplest multi-object dynamical model assumes a factored representation [13, 5]. However, this assumption does not model any form of interaction. Recently, the work in [6] proposed the introduction of a pairwise Markov Random Field (MRF) prior in the dynamical model [7]. The MRF is defined on an undirected graph, with objects defining the nodes of the graph, and links created at each time-step between pairs of proximate objects. The MRF prior places constraints on each object's state based on the states of its neighbors (Fig. 2). For a fixed set of objects over time, the dynamical model is expressed as

$$p(X_t|X_{t-1}) \propto \prod_{i \in \mathcal{I}_t} p(X_{i,t}|X_{i,t-1})p_0(X_t), \quad (3)$$

where $p(X_{i,t}|X_{i,t-1})$ denotes the dynamics of the i-th object, and the prior $p_0(X_t) = \prod_{ij \in \mathcal{C}} \phi(X_{i,t}, X_{j,t})$ is expressed as a product of pairwise interaction potentials $\phi(X_{i,t}, X_{j,t})$ over $\mathcal{C}$, the set of cliques (i.e., pairs of connected nodes) in the graph. The inclusion of p_0 allows us to model interaction between objects, unlike [15].

2.4. Dynamics for varying number of objects

Our dynamic model p_V, unlike [6], is defined for a variable number of objects

$$p(X_t|X_{t-1}) \propto \prod_{i \in \mathcal{I}_t} p(X_{i,t}|X_{t-1})p_0(X_t) \stackrel{def}{=} p_V(X_t|X_{t-1})p_0(X_t), \quad (4)$$

where we define $p(X_{i,t}|X_{t-1}) = p(X_{i,t}|X_{i,t-1})$ if object i existed in the previous frame ($i \in \mathcal{I}_{t-1}$), $p(X_{i,t}|X_{t-1}) = p(X_{i,t})$ if object i did not exist in the previous frame, and $p(\emptyset|X_{t-1}) = k$ for the zero object case (where k is a constant). With this definition, the Monte Carlo approximation of the filtering distribution in Eq. 2 becomes

$$p(X_t|Y_{1:t}) \approx Z^{-1}p(Y_t|X_t)\prod_{ij \in \mathcal{C}} \phi(X_{i,t}, X_{j,t}) \cdot \sum_n w_{t-1}^{(n)}p_V(X_t|X_{t-1}^{(n)}). \quad (5)$$

It is important to note that the interaction term can moved out of the mixture model defined over all particles [6].

2.5. Trans-dimensional MCMC-based PF

Inference of Eq. 5 with a basic PF is computationally infeasible when tracking several objects due to the inefficiency of importance sampling in high dimensions. Generating particles with good predictions for each object not feasible with a standard PF for more than two or three objects [4]. Schemes like partitioned sampling [9] improve efficiency, but also face limitations.

As an alternative, [6] recently proposed to sample from Eq. 5 with MCMC techniques for the case of a fixed number of objects using a Metropolis-Hastings (MH) sampler at

each time-step [8]. MCMC methods work by generating a sequence of samples from a Markov chain whose stationary distribution corresponds to the target distribution after a sufficiently long run of the sampler (in our case, the filtering distribution), and the discarding of the initial samples generated by the process (the so-called "burn-in" period) [8]. The MH algorithm draws samples from a proposal distribution $q(X^*|X)$, where X and X^* denote the current and proposed configurations, respectively, and accepts the latter with probability (a.k.a. acceptance ratio)

$$\alpha = \min\left(1, \frac{p(X^*)q(X|X^*)}{p(X)q(X^*|X)}\right). \quad (6)$$

To sample for fixed number of objects ($m_t = M\ \forall t$), [6] defined a proposal distribution where at each step in the chain, an object is chosen randomly and its configuration is sampled from the factored dynamic model. The states of all other objects are fixed. By accepting better single-object candidates at each step without discarding good candidates already accepted for other objects, the MH sampler improves the multi-object configuration. Note that, due to the use of MCMC sampling, at each time-step we have a fair set of samples from the true filtering distribution, and so all particle weights are equal to $\frac{1}{N}$ [8].

We generalize this approach to handle a variable number of objects by using trans-dimensional MCMC techniques [3]. This family of algorithms allows for the generation of a Monte Carlo approximation of a distribution defined on a space of variable dimension. The reversible-jump MCMC sampler can be implemented by a MH algorithm, in which a countable set of *moves* Υ (indexed by υ), and its prior $\{p_\upsilon\}$ are first defined, and candidate configurations are sampled from a set of move-specific proposal distributions $\{q_\upsilon(\cdot)\}$. The moves involve reversible jumps across subspaces of different dimension or within the same subspace. We follow the *dimension-matching* strategy described in [3]. We assume that the move-specific proposal distributions are functions of an auxiliary variable U. At each step of the algorithm, a new move υ^* is chosen with probability p_{υ^*}, and a new state X^* is defined by a deterministic function of the current state and a new sample of the auxiliary variable drawn from $q_{\upsilon^*}(U^*)$, $X^* = h(X, U^*)$. The reverse move υ from X^* to X is then computed by sampling U from $q_\upsilon(U)$, with $X = h'(X^*, U)$. The proposed configuration X^* is accepted with probability

$$\alpha = \min\left(1, \frac{p(X^*)}{p(X)}\frac{p_\upsilon}{p_{\upsilon^*}}\frac{q_\upsilon(U)}{q_{\upsilon^*}(U^*)}\left|\frac{\partial(X^*, U)}{\partial(X, U^*)}\right|\right), \quad (7)$$

which includes the Jacobian of the diffeomorphism from (X, U^*) to (X^*, U). In our case, we define $X^* = h(X, U^*) = U^*$, and $X = h'(X^*, U) = U$, so the Jacobian is unity [2]. Though [15] defines a similar *acceptance ratio*, α, the importance of dimension-matching is not mentioned.

For our implementation, we define four types of moves. Two imply jumps across dimensions, and two of fixed dimension:

1. *Birth (b)* of a new object, implying a dimension change from m_t to $m_t + 1$.
2. *Death (d)* of an existing object, implying a dimension change from m_t to $m_t - 1$.
3. *Swap (s)* of the identifiers between two existing objects, remaining in the dimension m_t.
4. *Update (u)* of the parameters of the existing objects, remaining in the dimension m_t.

Once a prior distribution over the moves has been defined ($\{p_b, p_d, p_s, p_u\}$), the RJ-MCMC PF can be summarized by the algorithm in Fig. 1.

`Generate` N samples $\{X_t^{(n)}, w_t^{(n)}\}$ from $\{X_{t-1}^{(n)}, w_{t-1}^{(n)}\}$.

- `Initialize` the MH sampler, by randomly choosing a particle from the subset that share the same object configuration as X_{t-1}^{MAP}, and sampling X from the predictive distribution $\sum_n w_{t-1}^{(n)} p_\upsilon(X_t|X_{t-1}^{(n)})$.
- `MH sampling`. Draw $B + N$ samples, where B is the desired burn-in fraction to be discarded. For each sample,
  - `Choose move`. Sample $\mu \sim U[0, 1]$.
    - if $0 \leq \mu < p_b$, $\upsilon^* = b$.
    - else if $p_b \leq \mu < p_b + p_d$, $\upsilon^* = d$.
    - else if $p_b + p_d \leq \mu < p_b + p_d + p_s$, $\upsilon^* = s$.
    - else $\upsilon^* = u$.
  - `Sample` X^* from the move-specific proposal q_{υ^*}.
  - `Compute acceptance ratio` α.
  - `Accept` the move with probability α.
  - `Add` accepted (X^*) or rejected (X) sample to the set $\{X_t^{(n)}, w_t^{(n)}\}$, $w_t^{(n)} = 1/N$.
- `Compute MAP estimate` X_t^{MAP}.

Figure 1: Trans-dimensional MCMC PF.

In the following subsection, we specify the proposal distribution for each move, and show that the computation of the acceptance ratio can be simplified in each case.

2.6. Move-specific proposal distributions

The proposal distributions should be defined in such a way that they simplify the computation of the acceptance ratio. Otherwise, its direct computation would involve the evaluation Eq. 5; implying a sum over all particles (at great computational cost, not discussed in [15]). We propose to use the predictive term in Eq. 5 in a mixture model formulation, to choose one object (in the case of the birth, death, and update), or an object pair (in the case of swap), to attempt a move. In the first three cases ($\upsilon = \{b, d, u\}$) we

define the proposal as a mixture,

$$q_v(X_t^*) = \sum_i q_v(i) q_v(X_t^*|i), \tag{8}$$

over all the appropriate objects i. To choose a candidate configuration X_t^* from the current configuration X_t, an index i^* is chosen with probability $q_v(i^*)$. A move is attempted on i^*, while the rest of the multi-object configuration is fixed. The mixture components are defined so that

$$q_v(X_t^*|i) = \begin{cases} \frac{1}{N}\sum_n p_V(X_t^*|X_{t-1}^{(n)}) & i = i^* \\ 0 & i \neq i^*. \end{cases} \tag{9}$$

The above general expression can be obtained for each of the moves. Defining the proposal in this manner cancels out all factors involving summations over the particles in the acceptance ratio, and keeps the algorithm computationally efficient.

Birth move. Adding an object i^* implies that $\mathcal{I}_t^* = \mathcal{I}_t \cup \{i^*\}$. In Eq. 8, the mixture components are defined by

$$q_b(X_t^*|i) = \frac{1}{N}\sum_n p(X_{i,t}^*|X_{t-1}^{(n)}) \prod_{l\in\mathcal{I}_t} p(X_{l,t}|X_{t-1}^{(n)}) \delta_{X_{l,t}}(X_{l,t}^*),$$

where $\delta_{X_{l,t}}(X_{l,t}^*) = \delta(X_{l,t}^* - X_{l,t})$. When i^* is the index of the new object, chosen from the available objects, it can be shown that the acceptance ratio for X_t^* is given by

$$\alpha_b = \min\left(1, \frac{p(Y_t|X_t^*)\prod_{j\in\mathcal{C}_{i^*}} \phi(X_{i^*,t}^*, X_{j,t}^*)}{p(Y_t|X_t)} \frac{p_d}{p_b} \frac{q_d(i^*)}{q_b(i^*)}\right). \tag{10}$$

Note that the interaction model for the new object plays an important role in discouraging new births that overlap with an existing object (Fig. 2(b)).

For the new-object pdf $p(X_{i^*,t}^*|X_{t-1}^{(n)})$, there are two cases. If $i^* \in \mathcal{I}_{t-1}^{(n)}$, $X_{i,t}^*$ is sampled from its dynamics. Otherwise, $X_{i,t}^*$ is sampled from its prior distribution, defined by a R-component Gaussian mixture model (GMM). The parameters of the first $R-1$ components are defined to draw samples from the entrance-exit regions in the scene. The last component is set to draw samples from a random configuration.

Death move. This is the reverse move to birth. An object index i^* is chosen with probability $q_d(i^*)$, and object removal is attempted, keeping all the other object configurations unchanged. The mixture components are

$$q_d(X_t^*|i) = \frac{1}{N}\sum_n \prod_{l\in\mathcal{I}_t, l\neq i} p(X_{l,t}|X_{t-1}^{(n)}) \delta_{X_{l,t}}(X_{l,t}^*),$$

The acceptance probability can be simplified to

$$\alpha_d = \min\left(1, \frac{p(Y_t|X_t^*)}{p(Y_t|X_t)\prod_{j\in\mathcal{C}_{i^*}} \phi(X_{i^*,t}, X_{j,t})} \frac{p_b}{p_d} \frac{q_b(i^*)}{q_d(i^*)}\right). \tag{11}$$

Update move. The update move applies dynamics without changing dimension. For a candidate X_t^*, an existing object index i^* is first randomly chosen, and its configuration is sampled from $p(X_{i^*,t}^*|X_{t-1}^{(n^*)})$, using a randomly chosen particle n^* from the previous time, keeping all other configurations unchanged. The acceptance probability is simplified to

$$\alpha_u = \min\left(1, \frac{p(Y_t|X_t^*)\prod_{j\in\mathcal{C}_{i^*}} \phi(X_{i^*,t}^*, X_{j,t}^*)}{p(Y_t|X_t)\prod_{j\in\mathcal{C}_{i^*}} \phi(X_{i^*,t}, X_{j,t})}\right). \tag{12}$$

Swap move. This move involves a pair of objects without changing dimension. The proposal is defined as a mixture model over all object pairs,

$$q_s(X_t^*) = \sum_{i,j} q_s(i,j) q_s(X_t^*|i,j),$$

with components $q_s(X_t^*|i,j)$ which swap the configurations of objects i and j, leaving everything else in the multi-object configuration unchanged. A candidate X_t^* is chosen by selecting a pair of existing object indexes i^*, j^* with probability $p_s(i^*, j^*)$ and swapping their configuration. In this case, the acceptance probability term can be reduced to

$$\alpha_s = \min\left(1, \frac{p(Y_t|X_t^*)}{p(Y_t|X_t)}\right). \tag{13}$$

The definitions for $p_d(i)$, $p_s(i,j)$ and $p_u(i)$ are given in Section 3.3.

2.7. Interaction model

We adopt a simple interaction model that penalizes object overlapping by defining an MRF prior using all the existing objects [6]. This model reduces the likelihood of fitting two trackers to the same object in situations like a brief crossing or people walking in a group. Given $S_t^{X_i}, S_t^{X_j}$, the spatial supports of X_i and X_j, respectively, the interaction potential is defined as,

$$\phi(X_{i,t}, X_{j,t}) \propto \exp\left(-\frac{\lambda_I}{2}(\nu(S_t^{X_i}, S_t^{X_j}) + \rho(S_t^{X_i}, S_t^{X_j})\right) \tag{14}$$

where λ_I is a hyper-parameter, and ν, ρ are the precision and recall measures, which indicate the overlap between $S_t^{X_i}$ and $S_t^{X_j}$ (see next subsection). The argument of the exponential is zero if the objects do not overlap, and minimum when they perfectly match.

2.8. Global observation model

To fairly compare configurations of varying numbers of objects we use a global observation model. We define observations pixel-wise, $Y_t = (Y_{1,t}..., Y_{i,t}, ..., Y_{N_P,t})$, for all N_P pixels in an image. Our approach differs from [15],

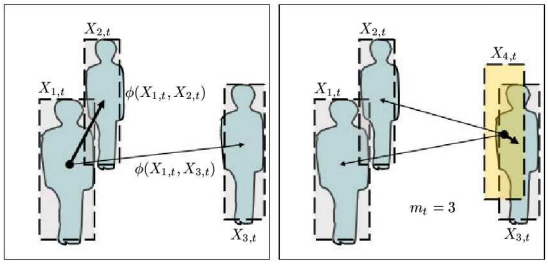

Figure 2: *Interaction potentials*. (Left). A pair-wise MRF is built among all object pairs. Proximate objects (thicker link) influence each other stronger than distant objects. (Right). Object overlapping is penalized, reducing the likelihood of giving birth to an object (yellow box) on a region already occupied by another.

which defines $p(Y_t|X_t)$ over objects and non-objects using products which may vary with the number of objects in the scene. Assuming a fixed camera, we define our model using *binary* and *color* measurements, $Y_t = (Y_t^b, Y_t^c)$. Binary measurements (Y_t^b) are extracted using background subtraction. Color measurements (Y_t^c) are made in HS space. A single pixel observation is thus $Y_{i,t} = (Y_{i,t}^b, Y_{i,t}^c)$, with $Y_{i,t}^b \in \{0, 1\}$ (0 indicates background), and $Y_{i,t}^c \in \mathbb{R}^2$. The multi-object global observation is defined by Bayes' rule as

$$p(Y_t|X_t) = p(Y_t^c|Y_t^b, X_t)p(Y_t^b|X_t). \quad (15)$$

2.8.1. Binary observation model

In order to predict the multi-object configuration and assist in the robust tracking of objects, we introduce a global binary observation model. For an image segmented into foreground and background pixels, the binary observations can be expressed as $Y_t^b = (Y_t^{b,F}, Y_t^{b,B})$ where the foreground and background observation variables are $Y_t^{b,F}$ and $Y_t^{b,B}$, respectively.

This model compares the coverage of foreground/background pixels by the multi-object configurations to learned values. Assuming conditional independence between the foreground and background, we define the binary likelihood as

$$p(Y_t^b|X_t) = p(Y_t^{b,F}|X_t)p(Y_t^{b,B}|X_t). \quad (16)$$

The distributions on the right hand-side are defined over features extracted from the overlap between the support of the binary foreground and background observations $S_t^{Y,b,F}$, $S_t^{Y,b,B}$ and S_t^X, the spatial support of X_t. These features, precision ν and recall ρ, are measures commonly used in information retrieval. Using S_t^X as "ground-truth", precision and recall for the binary foreground pixels $Y_t^{b,F}$ are

$$\nu_t^F = \frac{|S_t^X \cap S_t^{Y,b,F}|}{|S_t^{Y,b,F}|}, \qquad \rho_t^F = \frac{|S_t^X \cap S_t^{Y,b,F}|}{|S_t^X|}. \quad (17)$$

Precision is a measure of how well the foreground is covered by the estimate S_t^X, and recall measures what percentage of the estimate consists of foreground pixels. For the background, ν_t^B and ρ_t^B are defined in a similar manner.

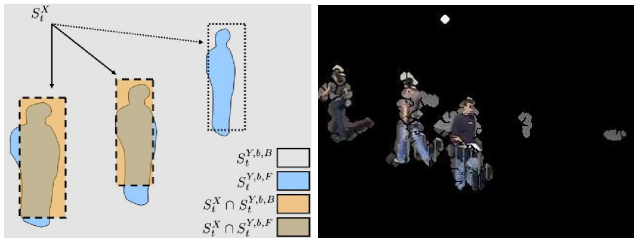

Figure 3: (Left). *Binary Observations.* The image is segmented into foreground and background regions. Precision and recall are computed from the intersection of these regions with the multi-object confi gurations (Right). Background subtraction results with several mislabeled patches of background (eg. shadows).

The distributions $p(Y_t^{b,F}|X_t)$ and $p(Y_t^{b,B}|X_t)$ in Eq. 16 are defined as 2-D GMMs over (ν_t^F,ρ_t^F) and (ν_t^B,ρ_t^B), respectively. To improve discrimination of the model w.r.t. the number of objects, we defined a set of switching 2-D GMM observation likelihood functions for all possible object counts ($m_t \in \mathcal{M}$). Note that, although in strict terms the observations are correlated ($\rho^B + \rho^F = 1$), we treat the observations as independent given the state.

The binary observation model functions as illustrated in Fig. 3. Assume three people are present in the scene, yet the estimated configuration is two objects. For well learned models, the three-object likelihood will be larger than the two-object likelihood ($p_3(Y_t^b|X_t) > p_2(Y_t^b|X_t)$), because $\nu_{t,2}^F$ will not fit the model well (due to foreground pixels not included in the configuration). Covering this area by a third estimate ($\nu_{t,3}^F$, including the dotted box) fits the model better. Binary observations prevent lost/empty estimates, as well. If an estimate fell upon an area devoid of foreground pixels, ν_t^B would not fit the learned model. Thus, configurations with the correct number of objects are rewarded. Binary observations also assist in improving tracking quality, favoring configurations with good coverage.

Looking at Figure 4, we can see how the binary observation model discriminates between objects. Tracking was performed on a dataset with two objects present ($m_t^{GT} = 2$, GT = ground truth). The system was forced to estimate varying multi-object configurations over several experiments ($m_t = 1, .., 4$). Observations ($\nu^F, \rho^F, \nu^B, \rho^B$) for these experiments were recorded and plotted as blue, red, green, and magenta points. Concentric circles represent the GMMs learned for each configuration ($m_t = 1, .., 4$). The two-object data ($m_t = 2$) fits the two-object GMM better than other configurations fits their corresponding models. Thus, for this scenario, the binary observation model attaches the highest likelihood to the multi-object configuration matching the ground truth ($m_t = m_t^{GT} = 2$).

Background subtraction. Because the binary observation model relies on good segmentation, background subtraction must be robust to lighting and appearance changes. We used a standard method [12], employing a pixel-wise model of background appearance. A 2-D GMM color model is learned for each pixel in HS color space using data captured

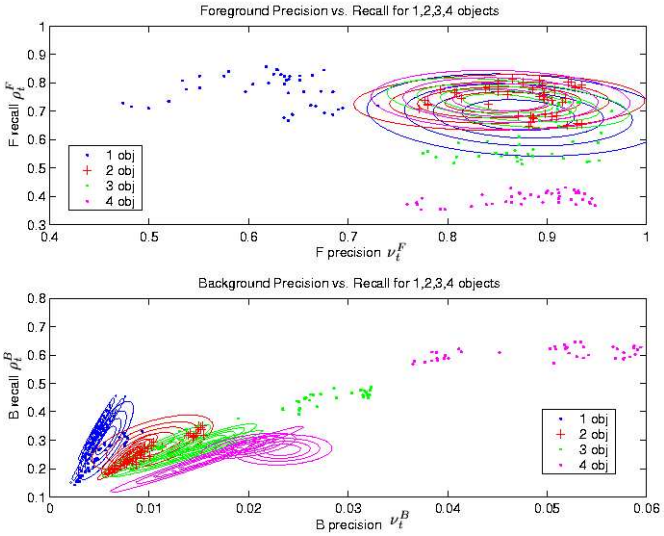

Figure 4: *Modeling multi-object configurations*. (top) Foreground model; (bottom) Background model. In both cases, concentric circles represent GMMs trained for $m_t = 1, ..., 4$ objects. Data points represent observations for multi-object configurations for $m_t = 1 - 4$, using a test sequence that contains ($m_t^{GT} = 2$) objects. See text for details.

at different times of the day. Using this model, each pixel of an image can be classified as foreground/background by likelihood thresholding. The resulting binary image is then subject to morphological filtering to remove foreground blobs of small size. A temporal filter is used to eliminate foreground blobs not connected to foreground blobs in the previous frame. Typical results are shown in Fig. 3.

2.8.2. Color observation model

The binary observation model does not distinguish between objects and failed background subtraction blobs, nor does it make use of foreground color information (i.e. object appearance). To perform these tasks, we define a global color observation model conditioned on the binary observations consisting of foreground and background color observations $Y_t^c = (Y_t^{c,F}, Y_t^{c,B})$. With a conditional independence assumption, the color likelihood is defined as,

$$p(Y_t^c|Y_t^b, X_t) = p(Y_t^{c,F}|Y_t^{b,F}, X_t)p(Y_t^{c,B}|Y_t^{b,B}, X_t). \quad (18)$$

Foreground color observations. The foreground color observations are crucial for maintaining proper object identities through swapping or distraction. The foreground color observation model is defined as a 4D histogram where two dimensions are used to construct 2D HS color histograms for each object [11], one dimension is used to index the objects, and on dimension is used to index spatial components within each object. The histogram is built using only foreground pixels within the estimated multi-object configuration. The color foreground likelihood is defined as $p(Y_t^{c,F}|Y_t^{b,F}, X_t) \propto e^{\lambda_F d_F^2}$, where λ_F is a hyperparameter and d_F is the distance based on the Bhattacharya coefficient [1] between the multi-object color observations and an existing multi-object color model.

Adaptive foreground color model. Every frame, the color foreground observations are compared to an adaptive color foreground model to compute the color foreground likelihood. This model consists of a set of spatial HS histograms for each object, of which one is chosen each frame to be the "current" model. The "current" spatial HS histogram is chosen by a voting system, where each frame the histogram which best matches the observations recieves a vote (and is adapted to the current observations using a running mean).

Background color observations. Background pixels (those not included in S_t^X) are used to build a 2-D HS color histogram. The observation process compares this histogram to a learned model. The background color likelihood can be described as $p(Y_t^{c,B}|Y_t^{b,B}, X_t) \propto e^{\lambda_B d_B^2}$, where λ_B and d_B^2 are defined as in the foreground case. The background color model helps reject configurations with untracked objects (those not covered by an estimate).

Background color model. The background color model is learned by computing the average histogram from background frames prior to the initial frame of the test sequence.

2.9. Computing the MAP estimate

The MAP estimate, X_t^{MAP}, is computed by first determining the best multi-object configuration (whichever is represented most in the Markov Chain), and then computing the mean state vector over samples with that configuration.

3. Results and discussion

3.1 Data

We tested our model on outdoor surveillance data collected over a span of six hours under varying environmental conditions. Several sequences from this raw data were organized into four test sequences: *seq1*, *seq2*, *seq3*, and *seq4*. Each of these sequences consists of one or more people walking alone across the scene, passing each other, meeting at the center of the scene, or walking together across the scene. Details are given in Table 1. Each sequence contains segments collected at different times of day to test robustness to environmental changes (moving shadows and objects in background) and lighting conditions (ranging from bright sunlight to overcast). Image size is 375x300.

	max people per frame	total # people	frames
seq1	1	5	908
seq2	2	8	1015
seq3	3	9	1178
seq4	4	8	714

Table 1: Data sets used for evaluation.

3.2 Learning

The background subtraction model was trained on background images [7421 frames] taken throughout the raw data

set as described in 2.8.1 using five mixture components. Typical background subtraction results can be found in Fig. 3. We trained our binary observation model to discriminate between up to four objects in a scene ($m_t = 0, ..., 4$) using GMMs defined over (ν_t^F,ρ_t^F) [1 mixture component] and (ν_t^B,ρ_t^B) [3 mixture components] for each multi-object configuration. Learning was done using results from an SMC color-based tracker [11] on training sets of different configurations: $m = 1$ [962 frames], $m = 2$ [1223 frames], $m = 3$ [934 frames], $m = 4$ [689 frames]. The resulting model, which is robust to imperfections in background subtraction such as those in Fig. 3, can be seen in Fig. 4.

3.3 Implementation Details

We define a time-varying prior distribution over the MCMC moves depending on the previous state X_{t-1}. The priors for birth ($p_b = [.05, .02]$) and death ($p_d = [.005, .0002]$) moves increase when an object is in an exit region, and the prior for swapping is increased when two objects are within a thresholded distance d_s of each other (.03, .001 otherwise).

Given a death move, an object is selected to be removed as a function of its inverse cubic distance to the nearest exit d_{e_i} $p(i) = \frac{1/d_{e_i}^3}{\sum_{i \in X_t} 1/d_{e_i}^3}$. When a swap move is chosen, the probability that a pair of objects (i, j) will be chosen to swap, $p_s(i, j)$ is a function of the distance between the objects, $d(i, j)$. Specifically, $p(i, j) = \frac{1/d(i,j)^3}{\sum_{i \in X_t} 1/d(i,j)^3}$. When an update move is chosen, $p_u(i)$ is sampled from a uniform distribution over all objects m_t. The motion model uses an AR2 process with variance in image coordinates $\sigma_{tx} = \sigma_{ty} = 3$ and scale $\sigma_{sx} = \sigma_{sy} = 0.008$, whereas [15] uses a Kalman filter. The color foreground models use HS color histograms with three spatial components corresponding to the head, torso, and legs. For our experiments $\lambda_B = \lambda_F = 40$. The MCMC filter discards the first 25% of the samples as the *burn-in*.

3.4 Performance measures

Often times, works in the field of tracking are criticized for not providing an objective evaluation of performance. For this reason, we have defined a set of performance measures to objectively evaluate experimental results using a hand-labeled ground truth. Here, successful tracking is defined as occurring when an estimated object area S_i^X has a non-null intersection with a ground truth area S_i^{GT}. This concept can be used to define the following performance measures:
Configuration error (η_t): a binary value indicating an error in the estimated multi-object configuration. The configuration error measures the trackers ability to correctly predict the number of objects present in the scene while remaining indifferent to the tracking quality. For a single frame, η_t is defined as one if $\sum_i S_{i,t}^{GT} \oplus S_{i,t}^X \geq 1$, and zero otherwise. For example, if two of three objects from the sequence are present at time t ($B_{i,t}^{GT} = \{1, 1, 0\}$, where B is a set of binary values indicating the presence of an object), and only one object is predicted by the estimate ($B_{i,t}^X = \{0, 1, 0\}$), a configuration error occurs ($\eta_t = 1$).
Track state (T_i): a binary variable that indicates the status of the tracking for an object at each frame, defined as one if $S_i^X \cap S_i^{GT} \neq \emptyset$ and zero otherwise.
Tracking success rate (τ_i) : the ratio of correctly tracked frames to total frames an object exists $\tau_i = \frac{\sum_t T_i}{\sum_{t, S_i^{GT} \neq \emptyset} 1}$. Indicates lost tracking or mis-identified objects.
Precision (ν_i): $\nu_i = S_i^X \cap S_i^{GT} / S_i^X$. Measures tracking quality.
Recall (ρ_i): $\rho_i = S_i^X \cap S_i^{GT} / S_i^{GT}$. Measures tracking quality.

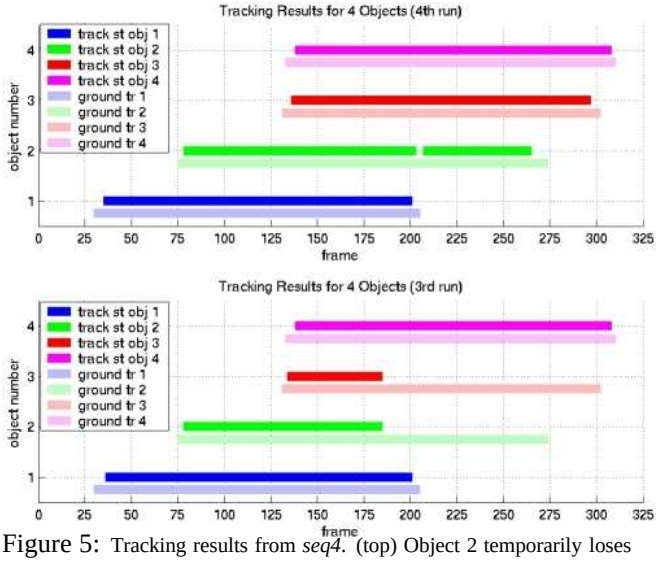

Figure 5: Tracking results from *seq4*. (top) Object 2 temporarily loses tracking (frames 201-206). (bottom) Objects 2 and 3 mistakenly swap identities at frame 185.

3.5 Discussion

The MCMC PF was applied over ten experimental runs for each sequence *seq1*, *seq2*, *seq3*, and *seq4* (videos provided at http://www.idiap.ch/∼smith). Visually, the experimental results were accurate (Fig. 6). The performance measures confirm that our method works effectively. The MCMC PF accurately predicted the multi-object configuration and correctly tracked the up to four objects simultaneously with good quality tracking. The main error sources were: improper swapping (switched identities), delay between the entrance of an object and birth (typically below 5 frames), and the rare accidental birth or premature death. Some of these errors are shown in Fig. 5. In the top figure, object 2 temporarily loses tracking (frames 201-206) when occluded by object 4. In the bottom figure, the identities of objects 2 and 3 were swapped in frame 185, though they were still

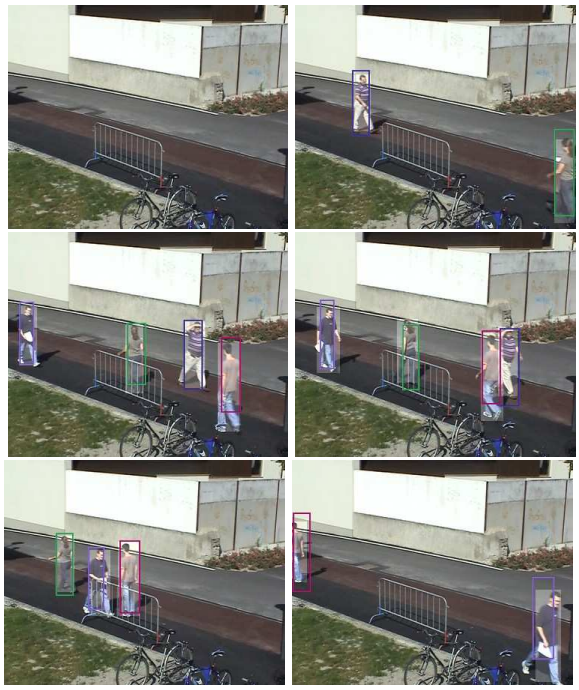

Figure 6: Frames 1,78,155,168,211,295 from an experimental run of *seq4*. Estimated configurations are shown as colored boxes where color corresponds to object ID, and the ground truths as shaded areas.

tracked correctly.

The results in Table 2, show the power of the binary observation model to discriminate between different numbers of objects in the scene. Most of the error in predicting the multi-object configuration (η) can be attributed to the entrance/exit delay, and indeed for *seq1* a mere 0.56% error was caused by other sources. An independent experiment run on *seq4* using only binary observations [see binary.avi] shows that even without color information, the object configuration is correct and tracking quality is very good (though identities are lost).

For *seq1* - *seq4*, the tracking was generally good (τ ranges from .69 to .99 with a median of .88, see Table 3). Results for the first two sequences of *seq2* and the last two of *seq3* suffered slightly due to erroneous swaps caused by objects of similar appearance. The quality of tracking was also high, as precision and recall ranged between $\{.75,.92\}$ and $\{.64,.85\}$ with means of .88 and .76 respectively. Most of the errors of this experiment can be attributed to erroneous swapping. This suggests that the color model and its adaptation scheme could be refined. Also, it is unclear how the likelihood is affected by comparing Bhattachayra distances of different dimension in the color model.

	seq1	*seq2*	*seq3*	*seq4*
η	3.56%	9.98%	13.78%	15.73%
delay error	3%	6%	9%	12%

Table 2: Configuration error rate η and typical error attributed to entrance/exit delays computed over 10 runs for each sequence.

seq		Object Number 1	2	3	4	5	6	7	8	9
	ν	.78	.86	.85	.91	.90				
1	ρ	.88	.85	.72	.74	.79				
	τ	**.97**	**.95**	**.97**	**.98**	**.98**				
	ν	.85	.88	.88	.76	.77	.90	.90	.89	
2	ρ	.79	.76	.75	.64	.77	.66	.75	.73	
	τ	**.87**	**.79**	**.72**	**.88**	**.91**	**.92**	**.83**	**.83**	
	ν	.90	.85	.85	.84	.88	.79	.78	.82	.75
3	ρ	.75	.82	.79	.79	.68	.79	.68	.66	.75
	τ	**.96**	**.93**	**.93**	**.80**	**.86**	**.69**	**.82**	**.81**	**.78**
	ν	.79	.78	.85	.86	.85	.92	.92	.90	
4	ρ	.81	.76	.77	.70	.77	.73	.77	.84	
	τ	**.95**	**.84**	**.88**	**.94**	**.87**	**.93**	**.86**	**.99**	

Table 3: Experimental results for *seq1*, *seq2*, *seq3*, *seq4* averaged over 10 runs of the MCMC PF.

4. Conclusion

In this paper we have presented a Bayesian framework for the fully automatic tracking of a variable number of objects using an interacting trans-dimensional MCMC PF. Results from our implementation of this framework show that it reliably tracks varying numbers of people through a number of real situations. Tracking failures caused by weaknesses in our color model suggest future work could explore methods for refinement. Additional future work could explore other interaction models or examine the handling of more objects within this framework.

References

[1] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proc. IEEE CVPR*, Hilton Head Island, Jun. 2000.

[2] S. Godsill. On the relationship between Markov chain Monte Carlo methods for model uncertainty. *J. Comp. Graph. Stats.*

[3] P. J. Green. Trans-dimensional Markov chain Monte Carlo. In P. J. Green, N. L. Hjort, and S. Richardson, editors, *Highly Structured Stochastic Systems*. Oxford Univ. Press, 2003.

[4] M. Isard. *Visual Motion Analysis by Probabilistic Propagation of Conditional Density*. D.Phil. Thesis, Oxford University, 1998.

[5] M. Isard and J. MacCormick. BRAMBLE: A Bayesian multi-blob tracker. In *Proc. IEEE ICCV*, Vancouver, Jul. 2001.

[6] Z. Khan, T. Balch, and F. Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. In *Proc. ECCV*, Prague, May 2004.

[7] S. Li. *Markov Random Field Modeling in Computer Vision*. Springer, 1995.

[8] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[9] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. In *Proc. IEEE ICCV*.

[10] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe. A boosted particle filter: multitarget detection and tracking. In *Proc. ECCV*, Prague, May 2004.

[11] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based Probabilistic Tracking. In *Proc. ECCV*, Copenhagen, May 2002.

[12] C. Stauffer and E. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. IEEE CVPR*, Ft. Collins, CO,, Jun. 1999.

[13] H. Tao, H. Sawhney, and R. Kumar. A Sampling Algorithm for Detecting and Tracking Multiple Objects. In *Proc. IEEE ICCV Workshop on Vision Algorithms*, Kerkyra, Sep. 1999.

[14] T. Yu and Y. Wu. Collaborative tracking of multiple targets. In *Proc. IEEE CVPR*, Washington, DC, Jun. 2004.

[15] T. Zhao and R. Nevatia. Tracking multiple humans in crowded environment. In *Proc. IEEE CVPR*, Washington DC, Jun. 2004.