Recognising Behaviours of Multiple People with Hierarchical Probabilistic Model and Statistical Data Association

Nam Nguyen, Svetha Venkatesh
Curtin University of Technology, GPO Box U1987, Perth
Western Australia, {nguyentn,svetha}@cs.curtin.edu.au

Hung Bui
Artificial Intelligence Center, SRI International, 333 Ravenswood Ave
Menlo Park, CA 94025, USA, bui@ai.sri.com

Abstract

Recognising behaviours of multiple people, especially high-level behaviours, is an important task in surveillance systems. When the reliable assignment of people to the set of observations is unavailable, this task becomes complicated. To solve this task, we present an approach, in which the hierarchical hidden Markov model (HHMM) is used for modeling the behaviour of each person and the joint probabilistic data association filters (JPDAF) is applied for data association. The main contributions of this paper lie in the integration of multiple HHMMs for recognising high-level behaviours of multiple people and the construction of the Rao-Blackwellised particle filters (RBPF) for approximate inference. Preliminary experimental results in a real environment show the robustness of our integrated method in behaviour recognition and its advantage over the use of Kalman filter in tracking people.

1 Introduction

Building smart surveillance systems has attracted much interest recently because of their numerous applications [1, 9, 12]. Recognising people behaviours, especially high-level behaviours, is a fundamental problem in many systems. This task is challenging because of noisy data from cameras and complex pattern of the high-level behaviours.

Much research has focused on recognising the high-level behaviour of a single person [9, 13, 14, 15]. Hierarchical probabilistic models such as the stochastic context free grammar (SCFG) [9], the abstract hidden Markov model (AHMM) [4], and the hierarchical hidden Markov model (HHMM) [3, 7] have been used recently to model the high-level behaviour and deal with uncertainty. Liao *et al.* [12] use the AHMM in a surveillance system in which GPS sensors are deployed to recognise a user's daily activities in large and complex environments. The AHMM is used to represent the activity hierarchy and the expectation and maximization (EM) algorithm is applied to learn the model's parameters. Nguyen *et al.* [14] use the HHMM to recognise a set of complex activities in indoor environments.

The problem of recognising behaviours of multiple people is more complicated. Usually, the reliable assignment of people to the set of observations is unavailable. We have

BMVC 2006 doi:10.5244/C.20.126

a data association problem. An efficient method to resolve this problem is the joint probabilistic data association filter (JPDAF) [2, 5]. However, the restriction of the JPDAF is the underlying Gaussian assumption, which has been relaxed in recent approaches that integrate particle filters with the JPDAF [10, 16, 17, 18]. This method has been applied with great success in non-Gaussian and non-linear dynamic processes. Work of note includes Schulz *et al.*'s [16], which uses particle filters to represent the target states and then applies the JPDAF directly to the sample set of particle filters. The algorithm is implemented in a mobile robot to track people in indoor environments. The Markov chain Monte Carlo (MCMC) can be used to generate samples from the large discrete space of the assignments of targets to measurements, reducing the computational cost of the tracking algorithm [11, 17].

Most research so far has not tackled the problem of recognising the high-level behaviours of multiple people in a unified probabilistic framework. Wilson and Atkeson [19] propose a system for simultaneous tracking and recognising behaviours of multiple people. However, their model is flat and cannot easily be extended to model high-level activities.

We propose an integrated approach for tracking and recognising high-level behaviours of multiple people. We consider primitive and complex behaviours, where primitive behaviour is a single action such as moving from one landmark towards another landmark, while complex behaviour is a sequence of primitive behaviours. Modeling the primitive and complex behaviours requires a hierarchical model. We use the HHMM [3, 7] $-$ an extension of the hidden Markov model $-$ in our framework because there are efficient learning and inference algorithms in this hierarchical model. We construct a unified graphical model, which we call the HHMM-JPDAF, to incorporate a set of HHMMs with data association. Further, we present a Rao-Blackwellised particle filter (RBPF) algorithm that efficiently computes the filtering distribution of the model at each time. We present the experimental results to demonstrate the robustness of our integrated method in behaviour recognition and its advantages over the use of the Kalman filters in tracking people.

The novelty of this paper is two-fold: 1) we propose an integrated graphical model $-$ the HHMM-JPDAF $-$ to track and recognise behaviours of multiple people and 2) we describe an efficient algorithm for approximate inference. Our work goes beyond the work of Wilson and Atkeson [19] by providing a framework for recognising more expressive classes of behaviours. While the behaviour recognition in Wilson and Atkeson's work is limited to whether or not a person is moving, our work deals with a hierarchy of primitive and complex behaviours.

The paper is organised as follows: Section 2 describes the HHMM and its use in behaviour recognition of a single person. Section 3 discusses the HHMM-JPDAF for tracking and recognising behaviours of multiple people. The system implementation and experimental results in a real environment are presented in Section 4, followed by concluding remarks in Section 5.

2 The HHMM for behaviour recognition

2.1 The HHMM

The hierarchical hidden Markov model (HHMM) [3, 7] is an extension of the hidden Markov model (HMM) to include a hierarchy of hidden states. A HHMM is defined by a tuple $< \zeta, \mathscr{Y}, \theta >$, where ζ is the topological structure, $\mathscr{Y}$ is the observation alphabet, and θ is the parameter of the model. The topology ζ specifies the depth of the model, the state space at each level, and the parent-child relationship between two consecutive levels. States at the lowest level are called *production* states and states at higher levels are called *abstract* states. At each level, an *end* state is introduced to signal when the control of activation is returned to the state at the higher level. Only *production* states emit observations. A representation of the HHMM as a dynamic Bayesian network (DBN) is provided in [3].

2.2 Recognising primitive and complex behaviours of a person

The *primitive* behaviour represents a person's action of going from one specific landmark to another specific landmark in the environment. For example, consider an environment that has four landmarks − *door*, *cupboard*, *fridge*, and *dining_table* − we can define the following primitive behaviours: (1) *door_to_cupboard*, (2) *cupboard_to_fridge*, (3) *fridge_to_dining_table*, and (4) *dining_table_to_cupboard*. The *complex* behaviour is defined from a set of primitive behaviours. A complex behaviour can be refined into different sequences of the primitive behaviours. For example, the sequence of primitive behaviours − (1), (2), (3), and (4) − can belong to the complex behaviour *have_meal*.

The HHMM for recognising the primitive and complex behaviours of a single person is discussed in Nguyen *et al.* [14]. A three-level HHMM is used for modeling the behaviour hierarchy. The complex behaviour, primitive behaviour and discrete position of a person are mapped into the top, middle and bottom levels of the HHMM, respectively. The parameters for the HHMM can be learned from a set of training sequences using the asymmetric inside-outside (AIO) [3] or junction tree algorithm [8]. The filtering distribution of the HHMM for each new observation arrival can be computed by a RBPF algorithm as in [14].

3 Recognising behaviours of multiple people

We consider the problem of recognising the primitive and complex behaviours of K people. We assume that at each time a person generates at most one observation. An observation can be noise and a person can generate no observation. Because the reliable assignment of people to observations is unavailable, we have a data association problem. We propose the HHMM-JPDAF − an extension of the HHMM that corporates the JPDAF − for tracking and behaviour recognition. A RBPF algorithm is adapted for the HHMM-JPDAF to provide an efficient approximate inference algorithm.

3.1 The HHMM-JPDAF

We use K HHMMs to model the behaviours of K people in the environment. Integrating the K HHMMs with the assignment of people to observations, we have a HHMM-JPDAF model. The representation of the HHMM-JPDAF as a DBN is shown in Figure 1.

In Figure 1, $u_t = (u_t^1, \ldots, u_t^K)$, $v_t = (v_t^1, \ldots, v_t^K)$ and $x_t = (x_t^1, \ldots, x_t^K)$ are the complex behaviours, primitive behaviours, and positions of the K people, respectively. $e_t = (e_t^1, \ldots, e_t^K)$ is the end status of the primitive behaviours. e_t^k represents whether the primitive behaviour v_t^k terminates or not. The set of observations at time t is $o_t = (o_t^1, \ldots, o_t^{m_t})$, where m_t is the number of observations. We assume that the position x_t and observation o_t are discrete. We also do not consider the problem of recognising a sequence of complex behaviours, thus a single complex behaviour is assumed to last from time $t = 1$ to $t = T$.

The assignment of K people to observations at time t is $\theta_t = (\theta_{1,t}, \ldots, \theta_{K,t})$, where $\theta_{k,t} \in \{0, \ldots, m_t\}$. If $\theta_{k,t} \neq 0$, the observation $o_t^{\theta_{k,t}}$ originates from person k. Otherwise

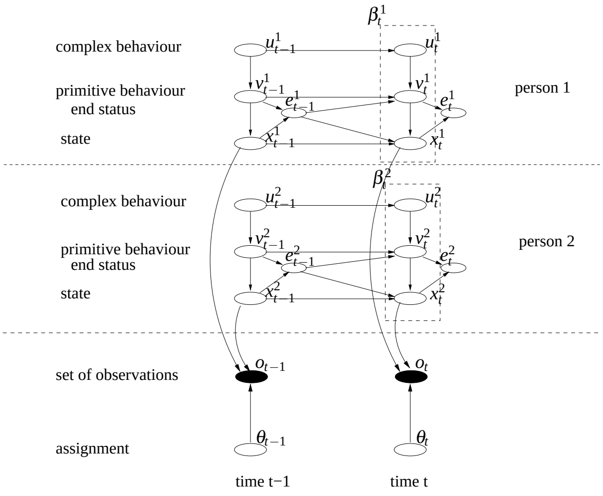


Figure 1: The DBN representation of the HHMM-JPDAF when there are two HHMMs.

person k has no observation at time t. $\theta_{k_1,t} \neq \theta_{k_2,t}$ if $k_1 \neq k_2$, $\theta_{k_1,t} \neq 0$, and $\theta_{k_2,t} \neq 0$. For example, we have two people – that is, person 1 and person 2 – and three observations $o_t = (o_t^1, o_t^2, o_t^3)$. $\theta_t = (2,0)$ means that person 1 generates the observation o_t^2 at time t, person 2 generates no observation, and the observations o_t^1 and o_t^3 are noise.

Let $\delta(\theta_t)$ denote the vector of detected people: $\delta(\theta_t) = (\delta^1(\theta_t), \ldots, \delta^K(\theta_t))$. $\delta^k(\theta_t) = 1$ if person k has a corresponding observation – that is, $\theta_{k,t} \neq 0$ – otherwise $\delta^k(\theta_t) = 0$. Let $\omega(\theta_t) = \{j \mid o_t^j \text{ is a false observation}\}$ and $\phi(\theta_t)$ denote the number of false observations. Given the assignment θ_t, then $\delta(\theta_t)$ and $\phi(\theta_t)$ are completely defined. For example, if $\theta_t = (2,0)$, then $\delta(\theta_t) = (1,0)$, $\omega(\theta_t) = \{1,3\}$, and $\phi(\theta_t) = 2$.

In the case that the assignments up to time t – that is, $\tilde{\theta}_t = (\theta_1, \ldots, \theta_t)$ – are given, the HHMM-JPDAF can be separated into K HHMMs and the sequence of observations corresponding to each HHMM is completely defined. Thus, the exact inference algorithms in the HHMM such as AIO [3] or junction tree algorithm [8] can be applied to estimate the current filtering distribution of each HHMM.

3.2 The RBPF in the HHMM-JPDAF

Let $\beta_t = \Pr(u_t, v_t, x_t \mid \tilde{o}_t)$ denote the belief state of the HHMM-JPDAF given the observations up to time t. Let β_t^k denote the belief state of each person k $(1 \leq k \leq K)$ – that is, $\beta_t^k = \Pr(u_t^k, v_t^k, x_t^k \mid \tilde{o}_t)$. For tracking and recognising people behaviours at a specific time t, we need to compute the belief state β_t. However, exact methods to compute β_t are intractable because:

$$\beta_t = \Pr(u_t, v_t, x_t \mid \tilde{o}_t) = \sum_{\tilde{\theta}_t} \Pr(u_t, v_t, x_t \mid \tilde{\theta}_t, \tilde{o}_t) \times \Pr(\tilde{\theta}_t \mid \tilde{o}_t)$$

and the number of possible values of $\tilde{\theta}_t$ is large when t increases. Thus, we need an approximate inference algorithm such as the RBPF [6] to compute β_t. We represent β_t by a set of particles and select $r_t = (\theta_t, e_t)$ as the Rao-Blackwellised (RB) variable. With each particle i, the RBPF samples the RB variable $r_t^{(i)} = (\theta_t^{(i)}, e_t^{(i)})$ and updates the belief state corresponding to that particle – that is, $\beta_t^{(i)}$ – using exact inference. The RBPF in the HHMM-JPDAF is shown in Algorithm 1, which is detailed below.

Algorithm 1 The RBPF algorithm in the HHMM-JPDAF.

```
Input   $S_{t-1} = \{< \beta^{(i)}_{t-1}, \theta^{(i)}_{t-1}, e^{(i)}_{t-1}, w^{(i)}_{t-1} > \mid i = 1, \ldots, N\}$, observation $o_t$
Begin
  /* sampling step */
  For each sample $i = 1, \ldots, N$
    Update the weight $w^{(i)}_{t-1} = w^{(i)}_{t-1} \times \Pr(o_t \mid \tilde{\theta}^{(i)}_{t-1}, \tilde{e}^{(i)}_{t-1}, \tilde{o}_{t-1})$
    Sample $\theta^{(i)}_t$ and $e^{(i)}_t$ from $\Pr(\theta^{(i)}_t, e^{(i)}_t \mid \tilde{\theta}^{(i)}_{t-1}, \tilde{e}^{(i)}_{t-1}, \tilde{o}_t)$
  /* re-sampling step*/
  Normalise the weight $w^{(i)}_{t-1} = w^{(i)}_{t-1} / \sum_{i=1}^{N} w^{(i)}_{t-1}$
  Re-sample the sample set according to $w^{(i)}_{t-1}$
  /* Exact step */
  For each sample $i = 1, \ldots, N$
    Compute $\beta^{(i)}_t$ using exact inference in the HHMM
    Set the weight $w^{(i)}_t = \frac{1}{N}$
  Compute $\beta_t \approx \frac{1}{N} \sum_{i=1}^{N} \beta^{(i)}_t$
End
```

A set of particles $S_t = \{< \beta^{(i)}_t, e^{(i)}_t, \theta^{(i)}_t, w^{(i)}_t > \mid i = 1, \ldots, N\}$ is maintained at each time t, where $w^{(i)}_t$ is the weight of each particle. The belief state β_t of the HHMM-JPDAF is obtained from the set of particles S_t.

Assume that the set of particles at time $t-1$, that is, S_{t-1}, is known, the set of particles at time t – that is, S_t – is computed as follows:

Updating weights. The weight $w^{(i)}_{t-1}$ is updated as:

$$w^{(i)}_{t-1} = w^{(i)}_{t-1} \times \Pr(o_t \mid \tilde{r}^{(i)}_{t-1}, \tilde{o}_{t-1}) = w^{(i)}_{t-1} \times \Pr(o_t \mid \tilde{\theta}^{(i)}_{t-1}, \tilde{e}^{(i)}_{t-1}, \tilde{o}_{t-1}) \qquad (1)$$

where $\tilde{r}^{(i)}_{t-1} = (\tilde{\theta}^{(i)}_{t-1}, \tilde{e}^{(i)}_{t-1})$, and $\tilde{e}^{(i)}_{t-1}$ and $\tilde{o}_{t-1}$ are the end nodes and observations up to time $t-1$, respectively. From now on, the upper indice (i) is omitted for simplicity. We have:

$$\Pr(o_t \mid \tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_{t-1}) = \sum_{\theta_t} (\Pr(o_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_{t-1}) \times \Pr(\theta_t \mid \tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_{t-1})) \qquad (2)$$

The probability $\Pr(o_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_{t-1})$ is computed as follows. Note that, $\{o^1_t, \ldots, o^{m_t}_t\} = \{o^j_t \mid o^j_t \text{ is not noise}\} \cup \{o^j_t \mid o^j_t \text{ is noise}\} = \{o^{\theta_{k,t}}_t \mid \theta_{k,t} \neq 0, k = 1, \ldots, K\} \cup \{o^j_t \mid j \in \omega(\theta_t)\}$. Thus, $\Pr(o_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_{t-1})$ can be factorised as:

$$\begin{aligned}
\Pr(o_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_{t-1}) &= \prod_{k=1, \theta_{k,t} \neq 0}^{K} \Pr(o^{\theta_{k,t}}_t \mid \tilde{\theta}_{k,t-1}, \tilde{e}^k_{t-1}, \tilde{o}_{t-1}) \\
&\quad \times \prod_{j=1, j \in \omega(\theta_t)}^{m_t} \Pr(o^j_t \text{ is noise}) \\
&= \prod_{k=1, \theta_{k,t} \neq 0}^{K} \Pr(o^{\theta_{k,t}}_t \mid \tilde{\theta}_{k,t-1}, \tilde{e}^k_{t-1}, \tilde{o}_{t-1}) \times V^{\phi(\theta_t)} \qquad (3)
\end{aligned}$$

where V is the probability that an observation is noise. $\Pr(o_t^{\theta_{k,t}} \mid \tilde{\theta}_{k,t-1}, \tilde{e}_{t-1}^k, \tilde{o}_{t-1})$ is factorised as:

$$\Pr(o_t^{\theta_{k,t}} \mid \tilde{\theta}_{k,t-1}, \tilde{e}_{t-1}^k, \tilde{o}_{t-1}) = \sum_{x_t^k} (\Pr(o_t^{\theta_{k,t}} \mid x_t^k) \times \Pr(x_t^k \mid \tilde{\theta}_{k,t-1}, \tilde{e}_{t-1}^k, \tilde{o}_{t-1})) \tag{4}$$

To compute the probability $\Pr(x_t^k \mid \tilde{\theta}_{k,t-1}, \tilde{e}_{t-1}^k, \tilde{o}_{t-1})$, we first obtain the belief state β_t^k by projecting β_{t-1}^k from time $t-1$ to t with the value of the end node e_{t-1}^k available. Then, we marginalise β_t^k over $\{u_t^k, v_t^k\}$ to obtain the probability $\Pr(x_t^k \mid \tilde{\theta}_{k,t-1}, \tilde{e}_{t-1}^k, \tilde{o}_{t-1})$. These steps are carried out in the HHMM corresponding to person k in a similar manner as in [4].

According to the DBN representation of the HHMM-JPDAF, θ_t and $\{\tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_{t-1}\}$ are independent when the observation o_t is unknown (see Figure 1). Thus,

$$\Pr(\theta_t \mid \tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_{t-1}) = \Pr(\theta_t) \tag{5}$$

Note that, given the assignment θ_t, $\delta(\theta_t)$ and $\phi(\theta_t)$ are completely defined. Thus, $\Pr(\theta_t)$ can be computed as:

$$\begin{aligned} \Pr(\theta_t) &= \Pr(\theta_t, \delta(\theta_t), \phi(\theta_t)) \\ &= \Pr(\theta_t \mid \delta(\theta_t), \phi(\theta_t)) \times \Pr(\delta(\theta_t), \phi(\theta_t)) \\ &= \Pr(\theta_t \mid \delta(\theta_t), \phi(\theta_t)) \times \prod_{k=1}^{K} (P_D^{\delta^k(\theta_t)} \times (1-P_D)^{1-\delta^k(\theta_t)}) \times \mu(\phi(\theta_t)) \end{aligned} \tag{6}$$

where P_D is the probability that person k is detected, $\delta^k(\theta_t)$ is the k^{th} element of the vector of detected people $\delta(\theta_t)$, and $\mu(\phi(\theta_t))$ is the probability that the number of false observations at time t is $\phi(\theta_t)$. Assuming that there is a uniform distribution over the set of the assignments θ_t given $\delta(\theta_t)$ and $\phi(\theta_t)$, we have: $\Pr(\theta_t \mid \delta(\theta_t), \phi(\theta_t)) = \frac{\phi(\theta_t)!}{m_t!}$. From (5) and (6), we can obtain the probability $\Pr(\theta_t \mid \tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_{t-1})$.

After obtaining the probabilities $\Pr(o_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_{t-1})$ and $\Pr(\theta_t \mid \tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_{t-1})$, we sum the product of these two probabilities over all possible assignments θ_t as in (2), then compute the weight $w_{t-1}^{(i)}$ from (1). In the case that the number of the assignments θ_t is large, the Markov Chain Monte Carlo (MCMC) method can be applied for sampling the assignment θ_t as in [17] to reduce the computation cost.

Sampling the RB variable. The RB variable $r_t = (\theta_t, e_t)$ is sampled from $\Pr(r_t \mid \tilde{r}_{t-1}, \tilde{o}_t)$, which can be factorised as:

$$\begin{aligned} \Pr(r_t \mid \tilde{r}_{t-1}, \tilde{o}_t) &= \Pr(\theta_t, e_t \mid \tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_t) \\ &= \Pr(e_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_t) \times \Pr(\theta_t \mid \tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_t) \end{aligned} \tag{7}$$

We first sample θ_t from $\Pr(\theta_t \mid \tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_t)$, then sample e_t from $\Pr(e_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_t)$.

$$\begin{aligned} \Pr(\theta_t \mid \tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_t) &\propto \Pr(\theta_t, o_t \mid \tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_{t-1}) \\ &= \Pr(o_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_{t-1}) \times \Pr(\theta_t \mid \tilde{\theta}_{t-1}, \tilde{e}_{t-1}, \tilde{o}_{t-1}) \\ &= \Pr(o_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_{t-1}) \times \Pr(\theta_t) \end{aligned} \tag{8}$$

Methods to compute $\Pr(o_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_{t-1})$ and $\Pr(\theta_t)$ have been discussed in the step of updating the weight $w_{t-1}^{(i)}$. Thus, we can sample the assignment θ_t from (8).

The probability $\Pr(e_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_t)$ – which is used to sample e_t – can be factorised as:

$$\begin{aligned} \Pr(e_t \mid \tilde{\theta}_t, \tilde{e}_{t-1}, \tilde{o}_t) &= \prod_{k=1,\theta_{k,t}\neq 0}^{K} \Pr(e_t^k \mid o_t^{\theta_{k,t}}, \tilde{\theta}_{k,t-1}, \tilde{e}_{t-1}^k, \tilde{o}_{t-1}) \\ &\quad \times \prod_{k=1,\theta_{k,t}=0}^{K} \Pr(e_t^k \mid \tilde{\theta}_{k,t-1}, \tilde{e}_{t-1}^k, \tilde{o}_{t-1}) \end{aligned} \tag{9}$$

We sample e_t^k , where $\theta_{k,t} \neq 0$, from $\Pr(e_t^k \mid o_t^{\theta_{k,t}}, \tilde{\theta}_{k,t-1}, \tilde{e}_{t-1}^k, \tilde{o}_{t-1})$ as follows. We first project the belief state β_{t-1}^k from time $t-1$ to t with the value of e_{t-1}^k available to obtain the belief state β_t^k. Then, we absorb the observation $o_t^{\theta_{k,t}}$ into β_t^k and sample the values of v_t^k and x_t^k from β_t^k. The end node e_t^k is sampled from the probability $\Pr(e_t^k \mid v_t^k, x_t^k)$.

We sample e_t^k, where $\theta_{k,t} = 0$, from $\Pr(e_t^k \mid \tilde{\theta}_{k,t-1}, \tilde{e}_{t-1}^k, \tilde{o}_{t-1})$ in a similar manner but without absorbing the observation value.

Re-sampling and exact step. We re-sample the set of particle filters S_{t-1} according to the weights $w_{t-1}^{(i)}$. In the exact step, we need to compute the belief state of each person k at time t – that is, β_t^k. We first obtain the corresponding observation $o_t^{\theta_{k,t}}$ of each person k (if person k generates an observation). The belief state β_t^k is computed by projecting the belief state β_{t-1}^k from time $t-1$ to t, then absorbing the observation $o_t^{\theta_{k,t}}$ and the end node e_t^k. These steps are carried out by using exact inference algorithms in the HHMM. In the case that person k has no corresponding observation, the absorbing observation step is skipped.

4 Experimental results

4.1 Implementation

We set up the system to recognise the primitive and complex behaviours in an environment as shown in Figure 2. The special landmarks in the environment are the *door*, *TV_chair*, *fridge*, *stove*, *cupboard*, and *dining_table*. We use a top-down camera to obtain the current image of the environment. A segmentation algorithm is used to extract motion blobs from the image, which are considered to be the observations of people in the environment. The features used to infer the people behaviours are the coordinates of the centroid of the motion blob. We do not use color in tracking because the color of a motion blob varies significantly from time to time. The environment is divided into a grid of discrete states, that are numbered $1, 2, \ldots, 96$. Each state is a square region in the image. The observation model is computed from a set of 1600 pairs (observation, groundtruth), that are collected manually.

We define 13 primitive behaviours and three complex behaviours in the environment. The primitive behaviours are:

(1) *door_to_cupboard*
(2) *cupboard_to_dining_table*
(3) *dining_table_to_cupboard*
(4) *dining_table_to_fridge*
(5) *fridge_to_dining_table*
(6) *door_to_TV_chair*
(7) *TV_chair_to_stove*
(8) *stove_to_TV_ chair*
(9) *door_to_stove*
(10) *stove_to_fridge*
(11) *fridge_to_stove*
(12) *stove_to_cupboard*
(13) *cupboard_to_stove*

The structure of the complex behaviours are shown in Figure 2. To learn the parameters for the primitive and complex behaviours, we obtain ten training sequences for primitive behaviours and five sequences for complex behaviours. The primitive behaviours and complex behaviours are learned from these training sequences in a similar manner as in [14].

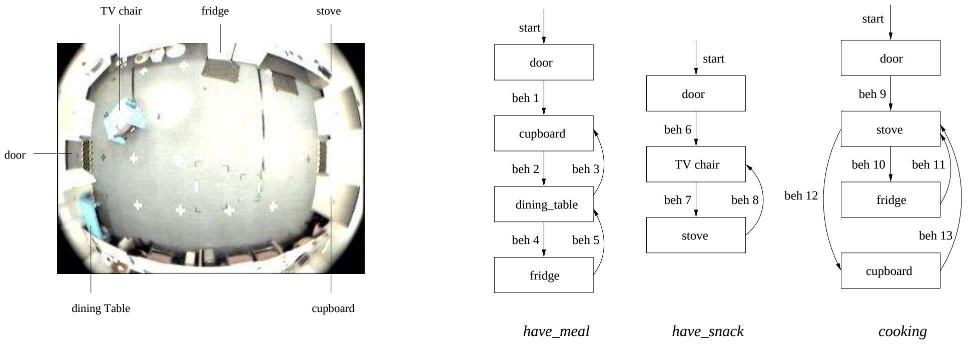


Figure 2: The room viewed from the top-down camera and the complex behaviours *have_meal*, *have_snack* and *cooking.*

4.2 Behaviour recognition results

We run the system to track and recognise people behaviours in real scenarios. We evaluate the performance of the system by considering the *winning complex behaviour* and the *correct duration*. The *winning complex behaviour* of each person in a scenario is defined as the complex behaviour that is assigned the highest probability at the end of the scenario. The system recognises the complex behaviour of a person correctly if the winning complex behaviour matches the groundtruth. The *correct duration* is defined as the total of the time periods, in which the primitive behaviour assigned the highest probability matches the groundtruth, over the length of the scenario. The correct duration shows the performance of the system in recognising the primitive behaviour.

We consider 12 scenarios. Each scenario has two people and each executes a specific complex behaviour. Table 1 shows the results of recognising the behaviour of each person. Compared with the groundtruth, the system recognises correctly the complex behaviour executed by each person in all scenarios. The average correct duration in all scenarios is 79%, showing that the system is able to recognise the primitive behaviours reliably. We also compare the position of each person estimated by the system with the groundtruth. The *position error* is the mean of the distance between the centroid of the person state and the groundtruth. The average position error of each person in each scenario is shown in Table 1. The average position error in all scenarios is $0.42 \times size\_of\_state$, showing that the system can track multiple people reliably.

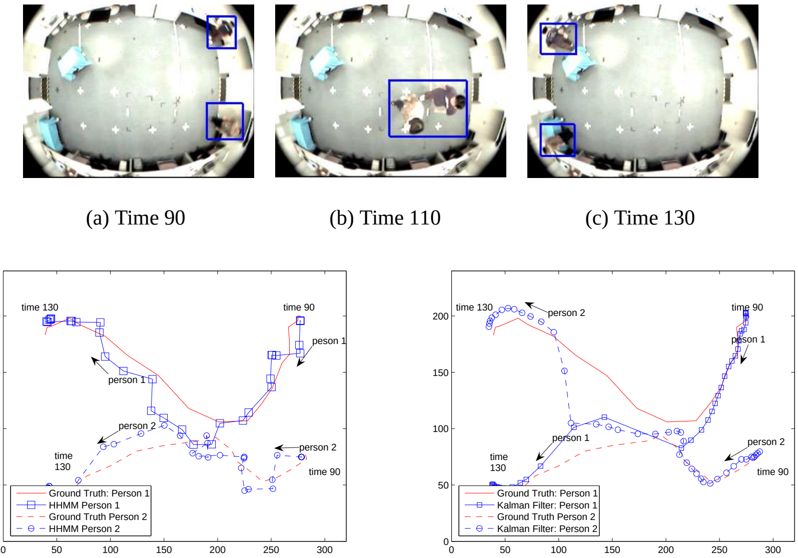
4.3 Compare the HHMM-JPDAF with the Kalman filter

We use the multiple Kalman filters and the JPDAF to track people in a similar manner as in [2]. Then we compare the results with the use of the HHMM-JPDAF.

Consider Scenario 3 from time 90 to 130. At time 90, person 1 and person 2 are at the top-right and bottom-right corners of the room, respectively (Figure 3(a)). Two people walk towards each other and person 1 is occluded by person 2 at time 110 (Figure 3(b)). Then, person 1 changes the direction and heads to the top-left corner of the room. Person 2 also changes direction and heads to the bottom-left corner of the room. We obtain the trajectory of each person by taking the mean of the centroid of the person state at each time. Figure 3(d) shows the trajectories of person 1 and person 2 tracked by the HHMM-JPDAF compared with the groundtruth. The HHMM-JPDAF can track person 1 and person 2

Scenario	Winning complex behaviour		Correct duration		Average position error (unit = size of state)	
	Person 1	Person 2	Person 1	Person 2	Person 1	Person 2
1	*have_meal*	*have_snack*	88%	97%	0.30	0.32
2	*cooking*	*have_meal*	76%	40%	0.66	0.89
3	*have_snack*	*have_meal*	99%	79%	0.25	0.28
4	*cooking*	*have_meal*	75%	77%	0.32	0.37
5	*have_meal*	*cooking*	90%	81%	0.29	0.23
6	*have_snack*	*cooking*	96%	75%	0.31	0.26
7	*have_meal*	*have_meal*	46%	74%	0.28	0.31
8	*have_meal*	*have_snack*	80%	96%	0.31	0.35
9	*have_snack*	*have_meal*	97%	94%	0.38	0.35
10	*cooking*	*have_meal*	42%	63%	1.13	1.33
11	*have_snack*	*have_meal*	94%	94%	0.30	0.27
12	*have_meal*	*have_meal*	74%	75%	0.34	0.33
Average			79%		0.42	

Table 1: The *winning complex behaviour*, the *correct duration* and the average *position error* in the 12 scenarios.



(a) Time 90

(b) Time 110

(c) Time 130

(d) HHMM-JPDF vs. groundtruth

(e) Kalman filter vs. groundtruth

Figure 3: The tracking results from time 90 to 130 in Scenario 3.

properly even when person 1 is occluded by person 2 and then the two persons change their directions. In contrast, the Kalman filter mislabels them in this case (Figure 3(e)). Under the same circumstance, the HHMM-JPDAF tracks the two people better than the Kalman filter. That is because the HHMM-JPDAF can use information about the behaviours of the two people to solve the labelling confusion.

5 Conclusion

We have presented the HHMM-JPDAF − which is an integrated framework of multiple hierarchical hidden Markov models (HHMM) and data association − to recognise high-level behaviours of multiple people. The HHMM is used for modeling the primitive and complex behaviours of each person, while the joint probabilistic data association filters (JPDAF) deal with data association. A Rao-Blackwellised particle filter (RBPF) algorithm is adapted for the HHMM-JPDAF as an efficient approximate inference method. Experimental results in a real environment show that the system is able to recognise primitive and complex behaviours reliably. The results also demonstrate that, in some scenarios, the HHMM-JPDAF outperforms the Kalman filter in tracking people.

References

[1] D. Ayers and M. Shah. Monitoring human behavior from video taken in an office environment. *Image and Vision Computing*, 19(12):833–846, October 2001.

[2] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, New York, date 1988.

[3] H. Bui, D. Phung, and S. Venkatesh. Hierarchical hidden Markov models with general state hierarchy. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence*, pages 324–329, San Jose, California, 2004.

[4] H. Bui, S. Venkatesh, and G. West. Policy recognition in the abstract hidden Markov model. *Journal of Artficial Intelligence Research*, 17:451–499, 2002.

[5] I. J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.

[6] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence*, 2000.

[7] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.

[8] C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263, 1996.

[9] Y. Ivanov and A. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(8):852–872, August 2000.

[10] Z. Khan, T. Balch, and F. Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In *European Conference on Computer Vision (ECCV)*, pages 279–290, 2004.

[11] Z. Khan, T. R. Balch, and F. Dellaert. Multitarget tracking with split and merged measurements. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, pages 605–610, San Diego, CA, 2005.

[12] L. Liao, D. Fox, and H. Kautz. Learning and inferring transportation routines. In *Proceedings of the National Conference on Artificial Intelligence(AAAI-04)*, 2004.

[13] K. Murphy and M. Pashkin. Linear time inference in hierarchical HMMs. In *NIPS-2001*, 2001.

[14] N. Nguyen, D. Phung, H. Bui, and S. Venkatesh. Learning and detecting activities from movement trajectories using the hierarchical hidden markov model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, volume 2, pages 955–960, San Diego, CA, 2005.

[15] V. Pavlovic, J. M. Rehg, and J. MacCormick. Learning switching linear models of human motion. In *Advances in Neural Information Processing Systems (NIPS)*, pages 981–987, 2000.

[16] D. Schulz, W. Burgard, D. Fox, and A. B. Cremers. Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2001.

[17] D. Schulz, D. Fox, and J. Hightower. People tracking with anonymous and id-sensors using rao-blackwellised particle filters. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

[18] J. Vermaak, S. J. Godsill, and P. Perez. Monte carlo filtering for multi-target tracking and data association. *IEEE Transaction on Aerospace and Electronic Systems*, 41(1):309–332, 2005.

[19] D. H. Wilson and C. Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. In *Third International Conference on Pervasive Computing*, pages 62–79, 2005.