ACM-Int. Conference on Multimodal Interface,
Workshop on Multimodal Multiparty Meeting Processing (ICMI-MMMP), pp 9-17, Trento, Oct. 2005.

A Rao-Blackwellized Mixed State Particle Filter for Head Pose Tracking in Meetings

Sileye O. Ba
IDIAP Research Institute
4 Rue du Simplon
Martigny, Switzerland
sileye.ba@idiap.ch

Jean-Marc Odobez
IDIAP Research Institute
4 Rue du Simplon
Martigny, Switzerland
odobez@idiap.ch

ABSTRACT

This paper addresses the problem of head pose estimation in the context of meetings. More precisely, given a video of people involved in a meeting, the goal is to estimate the pose of people's head with respect to the camera, which could ultimately translate into the estimation of the focus-of-attention of people (who is looking at whom or what). To this end, we present a Rao-Blackwellized mixed state particle filter to achieve joint head tracking and pose estimation. Rao-Blackwellizing a particle filter involves splitting the state variables into two sets by marginalizing with respect to some of them, allowing for the exact computation of their posterior probability density function given the samples of the remaining state variables. This splitting and marginalization processes reduce the dimension of the configuration space to be sampled and lead to a more efficient particle filter requiring a lower number of particles to achieve similar tracking performance. To demonstrate this, we conducted experiments on a publicly available database consisting of people engaged in meeting discussions and for which the groundtruth is available thanks to the use of magnetic flock-of-birds sensors. The results from these experiments demonstrated the benefits of the Rao-Blackwellized particle filter model with fewer particles over the plain mixed state particle filter.

1. INTRODUCTION

The automatic analysis of human interaction constitutes a rich research field. In particular, meetings exemplify the multimodal nature of human communication and the complex patterns that emerge from the interaction between multiple people [7]. Besides, in view of the amount of relevant information in meetings suitable for automatic extraction, meeting analysis has attracted attention in fields spanning computer vision, speech processing, human-computer interaction, and information retrieval [13]. In this view, the tracking of people and of their activity is relevant for high-level multimodal tasks that relate to the communicative goal of meetings. Experimental evidence in social psychology has highlighted the role of non-verbal behavior (e.g. gaze and facial expressions) in interactions [9], and the power of speaker turn patterns to capture information about the behavior of a group and its members [7, 9]. Identifying such multimodal behaviors requires reliable people tracking.

In the present work, we focus on the estimation of head pose from video data streams. Head pose estimation is often used as a first step for other higher level tasks such as facial expression recognition or gaze direction estimation. In meetings, head pose can be reasonably used as a proxy for gaze (which usually calls for close views), and can thus be useful for the determination of visual focus-of-attention (FOA) and addressees in conversations. Strictly speaking, visual FOA is defined by eye gaze. However, measuring eye gaze may be invasive or difficult in the presence of low resolution imagery. For this reason, we consider that the visual FOA can be reasonably approximated by the head pose.

Many methods have been proposed to solve the problem of head tracking and pose estimation. The proposed methods can be grossly separated into two groups. The first group considers the problem of head tracking and pose estimation as two separate and independent problems: the head location is found, then processed for pose estimation [2, 13, 10, 15, 17]. The main advantage of this approach is usually a fast processing, as real-time head trackers are available and estimating the head pose from a single location can also be done in real time. However, as a consequence of this approach, head pose estimation processing is highly dependent on the head tracking accuracy. Indeed, it has been shown that head pose estimation is very sensitive to head location [2]. This method does not take advantage of the fact that knowledge about head pose could improve head modeling and thus head tracking accuracy. The second group of methods [3, 6, 14] considers head tracking and pose estimation as a joint process. Following this conception, we proposed in previous works a method relying on a Bayesian formulation coupling the head tracking and pose estimation problems. Our method was based on a mixed state particle filter (MSPF) framework using discrete head pose models, based on texture and color cues, learned from training sets [1]. Using a head pose video dataset with corresponding ground truth acquired by a magnetic field location and orientation tracker (flock of bird), we showed that coupling

International ACM-ICMI Workshop on Multimodal Multiparty Meeting Processing, 2005 Trento, Italy.

head tracking and pose estimation within the mixed-state PF outperformed the approach that first track the head (using a shape and color PF in our experiments) and then estimate the pose (using the same head pose models than in the MSPF case).

In this paper, we propose to use Rao-Blackwellization [4] to improve the performances of our MSPF tracker. Rao-Blackwellization, which corresponds to the marginalization of some of the state variable components, is known to lead to more accurate estimates with a fewer number of particles [4]. It has already been used in [8] for mobile robot localization and in [5] for the tracking of bees. Rao-Blackwellization can be applied to a particle filter when the posterior probability density function (pdf) of some components of the state variable can be computed exactly. This is the case for discrete variables with finite possible values or for a variable with a pdf defined by an analytical expression that can be parameterized by sufficient statistics. As our set of head poses is discrete and finite, we can marginalize the head pose variable in the state and compute the pdf of head poses exactly given the tracking state variables (location ,scale,...). This exact computation step of some variables pdf can be computationally intensive but worthwhile when the reduction of the number of particles compensates for the analytical computation of the pdf. Experiments conducted using our head pose ground truth database demonstrate the improvements resulting from the Rao-Blackwellization of our MSPF.

The remainder of this paper is organized as follows. Section 2 describes the head pose representation and head pose modeling. Section 3 presents the MSPF for head tracking and pose estimation and the derivation of the Rao-Blackwellized particle filter (RBPF). Section 4 describes our evaluation set up and the experiments we conducted to compare the algorithms. Section 5 gives conclusions.

2. HEAD POSE MODELS

2.1 Head Pose Representation

There exist different parameterization of head pose. Here we present two of them which are based on the decomposition into Euler angles (α, β, γ) of the rotation matrix of the head configuration with respect to the camera frame, where α denotes the pan, β the tilt and γ the roll of the head. In the Pointing database representation [1], the rotation axis are rigidly attached to the head. In the PIE representation [11], the rotation axis are those of the camera frame. The Pointing representation leads to more direct interpretable values. However, the PIE representation has a computational advantage: the roll angle corresponds to in-plane rotations. Thus, only poses with varying pan and tilt values need to be modeled, as the head roll can be estimated by applying in-plane rotation to images. Thus, in the following, we will perform the tracking in the PIE angular space.

2.2 Head Pose Modeling

We use the Pointing'04 database to build our head pose models since the discrete set of pan and tilt values available covers a larger range of poses. Texture and color based head pose models are built from all the sample images available for each of the 93 discrete head poses $\theta \in \Theta = \{\theta_j = (\alpha_j, \beta_j, 0), j = 1, ..., N_\Theta\}$. In the Pointing database, there are 15 people per pose. Ground truth image patches are obtained by locating a tight bounding box around the head. Because of the few people in the database, we introduced more variability in the training set by generating virtual training images from the located head images. More precisely, the new training patches were generated by applying small random perturbation to the size, scale, and in-plane rotation of the original images while keeping the cropping bounding box fixed.

2.2.1 Head Pose Texture Model

The head pose texture is represented by the output of three filters: a Gaussian at coarse scale and two Gabor filters at two different scales (finer to coarser). Training patch image are resized to the same reference size 64×64, preprocessed by histogram equalization to reduce the light variations effects, then filtered by each of the above filters. The filter outputs at all locations inside a head mask are concatenated into a single feature vector.
To model the texture of head poses, the feature vectors associated with each head pose $\theta \in \Theta$ are clustered into K clusters using a kmeans algorithm. The cluster centers $e_k^\theta = (e_{k,i}^\theta), k = 1, ..., K$ are taken to be the exemplars of the head pose θ. The diagonal covariance matrix of the features $\sigma_k^\theta = diag(\sigma_{k,i}^\theta)$ inside each cluster is also exploited to define the pose likelihood models. Here, due to the small amount of different people in the training data, we considered only K=2 clusters. Furthermore, the head eccentricity distribution inside each cluster k of a head pose θ is modeled by a Gaussian $p^e_{(\theta,k)}(e)$ where the mean and the standard deviation are learned from the training head image eccentricities, and where the head eccentricity, denoted as e is defined as the ratio of the width over the height of the head,.

Using the above modeling, the likelihood of an input head image, characterized by its extracted features z^{text}, with respect to an exemplars k of a head pose θ is then defined by:

$$p_T(z|k,\theta) = \prod_i \frac{1}{\sigma_{k,i}^\theta} \max(\exp -\frac{1}{2}\left(\frac{z_i^{text} - e_{k,i}^\theta}{\sigma_{k,i}^\theta}\right)^2, T) \quad (1)$$

where $T = \exp -\frac{9}{2}$ is a lower threshold set to reduce the effects of outlier components of the feature vectors. In practice, these likelihood can be of quite different amplitude depending on the pose, and are not discriminative with respect to background clutter. As a consequence, some poses will be more likely than others when given background images, and more importantly, some head pose can produce higher likelihood on background images than other head pose models applied on true foreground head patches having the matched pose. To avoid these effects, we decided to normalize the above likelihood models with the average likelihood of background images, denote by $p_B(k,\theta)$, i.e. we defined the texture likelihood as:

$$p_{text}(z|k,\theta) = \frac{p_T(z|k,\theta)}{p_B(k,\theta)} \quad (2)$$

In practice, this average background likelihood was computed by extracting a set of patches randomly generated from a set of background images.

2.2.2 Head Pose Color Model

To make our head models more robust to background clutter and help the tracking, we learn for each head pose exemplar e_k^θ a face skin color model denoted by M_k^θ using the training images belonging to the cluster of this exemplar. Training images are resized to 64×64, then their pixels are classified as skin or non skin. The skin model M_k^θ is a binary mask in which the value at a given location is 1 when the majority of the training images have this location detected as skin, and 0 otherwise. Additionally we model the distribution of skin pixel values with a Gaussian distribution [16]. Skin colors are modelled in the normalized RG space, and the the parameters of the Gaussian (means and variances), denoted by m_0, are learned using the whole set of training images in the database.

The color likelihood of an input patch image at time t with respect to the k^{th} exemplar of a pose θ is obtained in the following way. Skin pixels are first detected on the 64x64 grid using the skin color distribution model, whose parameters m_t have been obtained in time through standard Maximum A Posteriori techniques, producing this way the skin color mask z_t^{col}. This skin mask is then compared against the model M_k^θ, and we defined the likelihood as:

$$p_{col}(z|k,\theta) \propto \exp -\lambda ||z_t^{col} - M_k^\theta||_1 \quad (3)$$

where λ is a hyper parameter learned from training data, and $||.||_1$ denotes the L_1 norm.

3. JOINT HEAD TRACKING AND POSE ESTIMATION

3.1 MSPF for head Pose Tracking

Particle filtering (PF) implements a recursive Bayesian filter by Monte-Carlo simulations. Let $X_{0:t-1} = \{X_j, j = 0, \ldots, t-1\}$ (resp. $z_{1:t-1} = \{z_j, j = 1, \ldots, t-1\}$) represents the sequence of states (resp. of observations) up to time $t-1$. Furthermore, let $\{X_{0:t-1}^i, w_{t-1}^i\}_{i=1}^{N_s}$ denotes a set of weighted samples that characterizes the pdf $\mathrm{p}(X_{0:t-1}|z_{0:t-1})$, where $\{X_{0:t-1}^i, i = 1, \ldots, N_s\}$ is a set of support points with associated weights w_{t-1}^i. At each time, the samples and weights can be chosen according to the Sequential Importance Sampling (SIS) principle [4]. The principle of SIS to estimate $\mathrm{p}(X_{0:t}|z_{0:t})$ is the following. Assuming that the observations $\{z_t\}$ are independent given the sequence of states and that the state sequence $X_{0:t}$ follows a first-order Markov chain model, the pdf $\mathrm{p}(X_{0:t}|z_{0:t})$ can be written in the recursive way:

$$\mathrm{p}(X_{0:t}|z_{1:t}) = \frac{\mathrm{p}(z_t|X_t)\mathrm{p}(X_t|X_{t-1})\mathrm{p}(X_{0:t-1}|z_{1:t-1})}{\mathrm{p}(z_t|z_{1:t-1})} \quad (4)$$

Furthermore, we assume that $\mathrm{p}(X_{0:t-1}|z_{1:t-1})$, the pdf at time $t-1$, can be approximated by a set of particles according to:

$$\mathrm{p}(X_{0:t-1}|z_{1:t-1}) \approx \sum_{i=1}^{N_s} w_{t-1}^i \delta(X_{0:t-1} - X_{0:t-1}^i) \quad (5)$$

where δ is the Dirac function. In this case, the current pdf is approximated using Equations 5 and 4, and up to the proportionality constant $\mathrm{p}(z_t|z_{1:t-1})$, by:

$$\mathrm{p}(X_{0:t}|z_{1:t}) \approx \mathrm{p}(z_t|X_t) \sum_{i=1}^{N_s} w_{t-1}^i \mathrm{p}(X_t|X_{t-1}^i) \quad (6)$$

1. initialization step: $\forall\ i$ sample X_0^i from $p(X_0)$ and set $t = 1$
2. IS step: $\forall\ i$ sample $\tilde{X}_t^i \sim p(X_t|X_{t-1}^i)$; evaluate $\tilde{w}_t^i = p(z|\tilde{X}_t^i)$
3. selection step: Resample N_s particles $\{X_0^i, w_t^i = \frac{1}{N_s}\}$ from the set $\{\tilde{X}_0^i, \tilde{w}_t^i\}$, set $t = t + 1$, go to Step 2

Figure 1: SIS Algorithm.

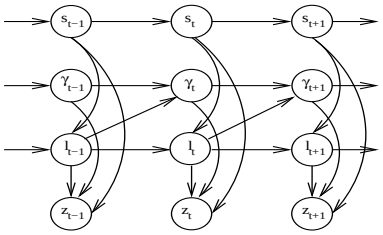

Figure 2: Mixed State Graphical Model.

Using SIS to estimate the pdf $\mathrm{p}(X_{0:t}|z_{1:t})$ consists in drawing N_s samples from the mixture $X_t^i \sim \sum_{i=1}^{N_s} w_{t-1}^i \mathrm{p}(X_t|X_{t-1}^i)$ and computing the particles' weights $w_t^i \propto \mathrm{p}(z_t|X_t^i)$ to compensate for the bias introduced by the sampling. The new particle set $\{X_{0:t}^i, w_t^i\}_{i=1}^{N_s}$ characterizes the pdf $\mathrm{p}(X_{0:t}|z_{0:t})$. Directly applying this scheme leads to sampling degeneracy: all the particles but one have very low weights after a few iterations. To solve the degeneracy problem, an additional resampling step is necessary [4]. Figure 1 displays the standard SIS algorithm.

In order to implement the filter, three elements have to be specified: a state model, a dynamical model and an observation model. Additionally, the filter output needs to be defined.

3.1.1 State Model

The MSPF approach [12], allows to represent jointly in the same state variable discrete variables and continuous variables. In our specific case the state $X = (S, \gamma, l)$ is the conjunction of a discrete index $l = (\theta, k)$ which labels an element of the set of head pose models e_k^θ, while both the discrete variable γ and the continuous variable $S = (x, y, s^x, s^y)$ parameterize the transform $\mathcal{T}_{(S,\gamma)}$ defined by:

$$\mathcal{T}_{(S,\gamma)}u = \begin{pmatrix} s^x & 0 \\ 0 & s^y \end{pmatrix} \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix} u + \begin{pmatrix} x \\ y \end{pmatrix}. \quad (7)$$

which characterizes the image object configuration. (x, y) specifies the translation position of the object in the image plane, (s^x, s^y) denote the width and height scales of the object according to a reference size, and γ specifies the in-plane rotation of the object.

3.1.2 Dynamic Model

The graphical model in Figure 2 describes the dependencies between our variables. The process density on the state sequence is modeled as a first order auto regressive process

$p(X_t|X_{t-1})$. According to the independence assumption in the graphical model, the equation of the process density is:

$$P(X_t|X_{t-1}) = p(S_t|S_{t-1})p(l_t|l_{t-1}, S_t)p(\gamma_t|\gamma_{t-1}, l_{t-1}) \quad (8)$$

The dynamical model of the continuous variable S_t, $p(S_t|S_{t-1})$ is modeled as a classical first order auto regressive process.

Regarding the dynamic of the discrete variable l_t, it is defined by:

$$p(l_t|l_{t-1}, S_t) \propto p(l_t|l_{t-1})p(l_t|S_t) \quad (9)$$

where the two terms are defined in the following way. The second term models the likelihood of a head pose given the tracking state value. Only the head eccentricity has an impact on the head pose. Thus, this term is defined by:

$$p(l_t|S_t) = \frac{p^e_{l_t}(e(S_t))}{\sum_{l'_t} p^e_{l'_t}(e(S_t))} \quad (10)$$

where $p^e_{l_t}$ is the prior on head eccentricity learned from the training data and defined in Section 2.2.1.
Regarding the first term in Eq.9, the exemplar transition process $p(l_t|l_{t-1}) = p(\theta_t, k_t|\theta_{t-1}, k_{t-1})$, it is decomposed as:

$$p(\theta_t, k_t|\theta_{t-1}, k_{t-1}) = p(k_t|\theta_t, k_{t-1}, \theta_{t-1})p(\theta_t|\theta_{t-1}). \quad (11)$$

where the two transition tables appearing in this definition are constructed in the following way. To build the table $p(\theta_t|\theta_{t-1})$, the dynamics of the head poses are first modelled in the continuous space by a Gaussian process, whose parameters are learned from the training sequences of our dataset. This Gaussian process is then used to compute the transition table between the different discrete pose angles.
The second probability table $p(k_t|\theta_t, k_{t-1}, \theta_{t-1})$, which encodes the transition probability between exemplars, is learned using the training set of faces. That is, for different head poses, the exemplars are more related when the same persons were used to build them. When $\theta \neq \theta'$, $p(k|\theta, k', \theta')$ is taken proportional to the number of persons who belong to the class of e^{θ}_k and who are also in the class of $e^{\theta'}_{k'}$. When $\theta = \theta'$, $p(k|\theta, k', \theta')$ is large for $k = k'$ and small otherwise.

Finally, $p(\gamma_t|\gamma_{t-1}, l_t = (k_t, \theta_t))$, the dynamic of the in-plane rotation variable, is also learned using the sequences in the training dataset, and comprises a Gaussian prior on the head roll $p_\Theta(\gamma_t)$. More specifically, the pan tilt space has been divided into nine regions, with pan and tilt ranging from -90 to 90 with a step of 60 degrees. Inside each region, roll transition tables and roll prior are learned from the training data. Hence, the variable l_t acts on the roll dynamic like a switching variable, and this also holds for the prior on the roll value.

3.1.3 Observation Model

The observation likelihood $p(z|X)$, where the observation z are composed of texture and color observations (z^{text}, z^{col}), is defined as follows :

$$p(z|X = (S, \gamma, l)) = p_{text}(z^{text}(S, \gamma)|l)p_{col}(z^{col}(S, \gamma)|l), \quad (12)$$

where we have assumed that these observations were conditionally independent given the state. The texture likelihood p_{text} and the color likelihood p_{col} have been defined in Section 2.
During tracking, the computation of the observations is done as follows. First the image patch associated with the image spatial configuration of the state space, (S, γ), is cropped from the image according to $\mathcal{C}(S, \gamma) = \{\mathcal{T}_{(S,\gamma)}u,\ u \in \mathcal{C}\}$, where $\mathcal{C}$ corresponds to the set of 64x64 locations defined in a reference frame. Then, the texture and color observations are computed using the procedure described in sections 2.2.1 and 2.2.2.

3.1.4 Filter output

We need to define what we use as output of the particle filter. The set of particles defines a pdf over the state space. Thus, we can use as output the expectation value of this pdf, obtained by standard averaging over the particle set. Note that usually, with mixed-state particle filters, averaging over discrete variable is not possible (e.g. if a discrete index represents a person identity). However, in our case, there is no problem since our discrete indices correspond to real Euler angles which can be combined.

At this point, The MSPF for joint head tracking and pose estimation is completely defined. Let us now describe the Rao-Blackwellization of our MSPF.

3.2 Rao-Blackwellizing the MSPF

Rao-Blackwellization can be applied when the filtering pdf of some state model variables can exactly be computed given the samples of the remaining variables. As the exemplar label l is discrete and belongs to a finite set, it fullfils the necessary conditions.
Given the graphical model of our filter (Fig.2), the Rao-Blackwellized particle filter (RBPF) consists of applying the standard SIS algorithm over the tracking variables S and γ while applying an exact filtering step over the exemplar variable l, *given a sample of the tracking variables*. In this way, computing the likelihood of the state can be done using:

$$p(S_{1:t}, \gamma_{1:t}, l_{1:t}|z_{1:t}) = p(l_{1:t}|S_{1:t}, \gamma_{1:t}, z_{1:t})p(S_{1:t}, \gamma_{1:t}|z_{1:t}) \quad (13)$$

In practice, only the sufficient statistics $p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t})$ of the first term in the right hand side is computed and is involved in the SIS steps of the second term. Thus, in the RBPF modeling, the pdf in Equation 13 is represented by a set of particles

$$\{S^i_{1:t}, \gamma^i_{1:t}, \pi^i_t(l_t), w^i_t\}^{N_s}_{i=1} \quad (14)$$

where $\pi^i_t(l_t) = p(l_t|S^i_{1:t}, \gamma^i_{1:t}, z_{1:t})$ is the pdf of the exemplars given a particle and a sequence of measurements, and $w^i_t \propto p(S^i_{1:t}, \gamma^i_{1:t}|z_{1:t})$ is the weight of the tracking state particle. In the following, we detail the methodology to derive the exact steps to compute $\pi^i_t(l_t)$ and the SIS steps to compute w^i_t.

3.2.1 Deriving the Exact Step

The goal here is do derive $p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t})$. As l_t is discrete, this can be done using prediction and update steps similar to those involved in Hidden Markov Model (HMM).
Prediction Step for Variable l:
Given the new samples of S and γ at time t, the prediction

distribution of l_t, $p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1})$ can be evaluated by:

$$p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) = \sum_{l_{t-1}} p(l_t, l_{t-1}|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) \quad (15)$$
$$= \sum_{l_{t-1}} p(l_t|l_{t-1}, S_t) p(l_{t-1}|S_{1:t}, \gamma_{1:t}, z_{1:t-1})$$

Unlike in the standard RBPF, however, the second term $p(l_{t-1}|S_{1:t}, \gamma_{1:t}, z_{1:t-1})$, due to the extra dependency between γ_t and l_{t-1} is not equal to $p(l_{t-1}|S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1})$. However, this term can still be computed. Exploiting the dependency assumptions of the graphical model, we have:

$$p(l_{t-1}|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) = \quad (16)$$

$$\frac{p(\gamma_t|\gamma_{t-1}, l_{t-1}) p(S_t|S_{t-1}) p(l_{t-1}|S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1})}{Z_1(S_t, \gamma_t)} \quad (17)$$

where $Z_1(S_t, \gamma_t) = p(S_t, \gamma_t|S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1})$ the normalization constant of the denominator can easily be computed by suming the numerator w.r.t. l_{t-1}.

Update Step for Variable l:
When new observations z_t are available, the prediction can be updated to obtain our target pdf:

$$p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t}) = \frac{p(z_t|S_t, \gamma_t, l_t) p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1})}{Z_2} \quad (18)$$

where $Z_2 = p(z_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1})$, the normalization constant, can be obtained by suming the numerator with respect to l_t.

3.2.2 Deriving the SIS particle filter steps

The pdf $p(S_{1:t}, \gamma_{1:t}|z_{1:t})$ is approximated using particles whose weight is recursively computed using the standard SIS principle. More precisely, this pdf can be written in a recursive way as in Equation 4. Using the discrete approximation of the pdf at time $t-1$ with the set of particles and weights (see Equation 5), the current pdf $p(S_{1:t}, \gamma_{1:t}|z_{1:t})$ can be approximated (up to the proportionality constant $p(z_t|z_{1:t-1})$) by:

$$p(z_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) \sum_{i=1}^{N_s} w_{t-1}^i p(S_t, \gamma_t|S_{1:t-1}^i, \gamma_{1:t-1}^i, z_{1:t-1}) \quad (19)$$

to which the standard PF steps can be applied. Indeed, the mixture $\sum_{i=1}^{N_s} w_{t-1}^i p(S_t, \gamma_t|S_{1:t-1}^i, \gamma_{1:t-1}^i, z_{1:t-1})$ can be rewritten as:

$$\sum_{i=1}^{N_s} w_{t-1}^i p(S_t|S_{t-1}^i) \sum_{l_{t-1}} \pi_{t-1}^i p(\gamma_t|\gamma_{t-1}, l_{t-1}) \quad (20)$$

which embeds the temporal evolution of the head configurations and allows to draw new (S_t, γ_t) samples. Similarly, the weight of this new samples, defined by the observation likelihood $p(z_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1})$ can be readily obtained from the exact steps computation (cf the computation of the Z_2 constant).

Figure 3 summarizes the steps of the RBPF algorithm with the additional resample step to avoid sampling degeneracy. In the following Section, we describe the experiments we conducted to compare the algorithms and give the results.

1. **initialization step:** $\forall\ i$ sample (S_0^i, γ_0^i) from $p(S_0, \gamma_0)$, and set $\pi_0^i(.)$ uniform and $t = 1$
2. **prediction of new head location configurations:** sample $\tilde{S}_t^i$ and $\tilde{\gamma}_t^i$ from the mixture
$$(\tilde{S}_t^i, \tilde{\gamma}_t^i) \sim p(S_t|S_{t-1}^i) \sum_{l_{t-1}} \pi_{t-1}^i(l_{t-1}) p(\gamma_t|\gamma_{t-1}^i, l_{t-1})$$
3. **head poses distribution of the particles:** compute $\tilde{\pi}_t^i(l_t) = p(l_t|S_{1:t}^i, \gamma_{1:t}^i, z_{1:t})$ using Equations 15 and 18 for all i and l_t
4. **particles weights:** for all i compute the weights $w_t^i = p(z_t|S_{1:t}^i, \gamma_{1:t}^i, z_{1:t-1})$
5. **selection step:** resample N_s particle $\{S_t^i, \gamma_t^i, \pi_t^i(.), w_t^i = \frac{1}{N_s}\}$ from the set $\{\tilde{S}_t^i, \tilde{\gamma}_t^i, \tilde{\pi}_t^i(.), \tilde{w}_t^i\}$, set $t = t+1$ go to step 2

Figure 3: RBPF Algorithm.

3.2.3 RBPF output

At each time step, the filter outputs the mean head pose configuration. For instance, it can be obtained by first computing the head pose mean of each particle, which is given by the average of the exemplars head pose θ_t with respect to the distribution $\pi_t^i(l_t)$. Then the particle head poses are averaged with respect to the distribution of the weights w_t^i to give the head pose output of the RBPF.

4. EXPERIMENTS

4.1 Dataset and Protocol Evaluation

To evaluate head pose technology, we have built a head pose video database of people involved in real situation, where their head poses are continuously annotated using a device called flock-of-bird, a magnetic field 3D location and orientation tracker. The device was well camouflaged behind people's ear. After calibration of the sensor with the camera frame, we can output at each time frame the person's head pose. With this system, we recorded two databases, one in an office environment (not used here) and one in a meeting environment. In the meeting environment, 8 meetings were recorded, each lasting approximatively 8 minutes. In each meeting, two out of four persons had their head poses continuously annotated. The scenario of the meeting was to discuss statements displayed on the projection screen. There were restrictions neither on head motions, nor on head poses. This results in a video database of 16 different people. For our experiments we use half of the people in the training set to learn the parameters of the pose dynamic model, while the remaining half was used as the test set to evaluate the tracking algorithms.

The tracking evaluation protocol is the following. In each one of the 8 meetings of the test set, we selected 1 minute of recording (1500 video frames) for evaluation data. We decided to use only one minute to save machine computation time, as we use a quite slow matlab implementation. In the test dataset pan values ranges from -60 to 60 degree, tilt values from -60 to 15 degrees and roll value from -30 to 30 degrees.
To evaluate tracking performances, we used four error mea-

sures. The one is the error observed on the pointing vector. More precisely, as a head pose defines a vector in the 3D space, the vector indicating where the head is pointing at, the angle between the 3D pointing vectors defined by the head pose GT and the pose estimated by the tracker can be used as pose estimation error measure. This vector depends only on the head pan and tilt values in the Pointing representation. This error measure is well suited for studies on the FOA, where the main concern is to know where the head/person is looking at. However, it gives no information about the roll estimation error. Therefore, in order to possibly have more details about the origins of the errors we will also measure the individual errors made separately on the pan, tilt and roll angles measured in the Pointing representation. For each one of the four error measures, we will compute the mean, standard deviation, and median value of the absolute error values. We use the median value because it is less sensitive to extremal values than the mean, and thus avoids emphasis of situations where the method fails because the tracker is trapped into some local minimum corresponding to a bad localization.

4.2 Results

Experiments were conducted to compare head pose estimation based on the MSPF and the RBPF tracker. The MSPF tracker was run with 200 hundred particles and the RBPF with 100 particles. Except this difference, all the other models/parameters involved in the algorithm where the same (remember that both approaches are based on the same graphical model and involve the setting/learning of the same pdf). In what follows, we first compare the head-pose estimation results, and then discuss the computational cost of both methods.

Head pose estimation performance

As when using half of the number of particles of the MSPF for the RBPF the computational cost of the two systems are equivalent, we compared the performances of a MSPF with 200 hundred particles and the RBPF with 100 particles.

Table 1 shows the pointing vector error for the two methods over our whole test dataset. This table shows that the mean and the median of the pointing vector error is smaller for the RBPF than for the MSPF, though not being statistically significant. This improvment is mainly due to a better exploration of the configuration space of the head poses with the RBPF, as illustrated in Figure 5 which displays sample tracking results of one person of the test set. The first column of this figure presents the results from the MSPF while the second column shows those of the RBPF for the same time instants. Because of a sudden head turn (second row) and the small number of particles, the MSPF lags behind in the exploration of the head pose configuration space, to the contrary of the RBPF approach which nicely follows the head pose. This laging can be diminished by increasing the number of particles by a factor of 3 or 4, but the computational cost of the system will increase at the same time.

To have more details about the errors, Table 2 displays the errors in pan tilt and roll on the whole test dataset. It shows that the errors in pan and roll for both of the methods are smaller than the errors in tilt. This is due to the fact that, even in a perceptive point of view, discriminating between head tilts is more difficult than discriminating between head pan or head roll [2]. For these errors measures also the RBPF is performing better than the MSPF.

	mean	std	median
MSPF	22.5	12.5	20.1
RBPF	20.3	11.3	18.2

Table 1: Mean, standard deviation and median of head pointing vector errors over the test dataset.

	pan			tilt			roll		
	mean	std	med	mean	std	med	mean	std	med
MSPF	10.0	9.6	7.8	19.4	12.7	17.5	11.5	9.9	8.8
RBPF	9.10	8.6	7.0	17.6	12.2	15.8	10.1	9.9	7.5

Table 2: pan, tilt and roll errors statistics over test dataset.

Figure 4 shows the mean of the pan, tilt and roll estimation errors for each person of the test set to study the dependency of the results to individual. For almost all the persons, the RBPF is performing better than the MSPF for head pan and tilt estimation. It is worth noticing in this figure that the improvements due to the Rao-Blackwellisation are more consistent on the marginalized variables (pan and tilt) than on the sampled one (the roll).

Computational cost

In the MSPF algorithm, the most expensive part in terms of computation is the extraction of the texture observation z^{text} for each particle. Filtering patch images with the Gaussian and the two Gabor filters is time consuming. Thus, on the one hand, being able to reduce the number of particles while achieving the same or better tracking performance, as the Rao-Blackwellisation allows us to do, should considerably reduce the tracking cost. On the other hand, the exact marginalization step introduced by the RBPF algorithm has a cost, and involve the computation of the data likelihood for all the exemplars. Given our large observation size (approx. 570) and the number of exemplars (186), this step is not cpu-cost free. Currently, with our matlab implementation, the RPBF algorithm runs approximatively two time faster than the MSPF one. However, the RPBF offers several ways to improve computation efficiency. For instance, space reduction techniques (PCA) could be employed, which would then reduce the likelihood computation.

5. CONCLUSION

In this paper, we have presented a RBPF for joint head tracking and pose estimation. Experiments conducted with labelled head pose ground truth of 8 different people in a meeting room showed that the RPPF tracker is performing better than the MSPF tracker. Results from previous work have already shown that tracking and head pose estimation with the MSPF was performing better than tracking then head pose estimation based on a standard particle filter. Thus, the RBPF has the best performances of these systems. Although our RBPF model performs very well for single person tracking without occlusions, in the future we plan to extend the model to situations with multiple people and possible occlusions.

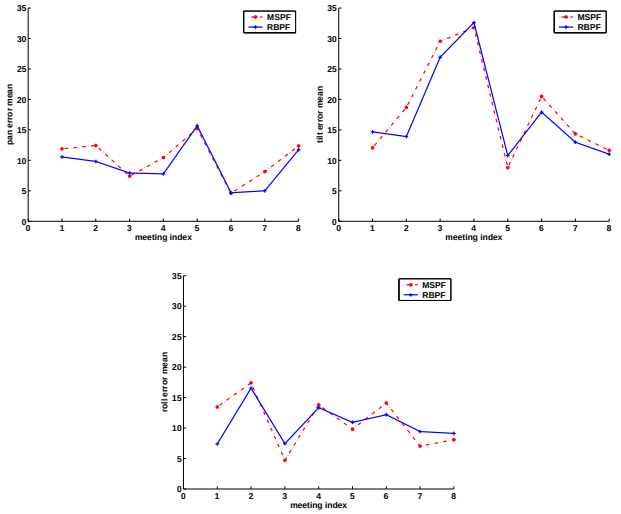

Figure 4: Pan tilt and roll errors over individual meetings.

6. ACKNOWLEDGMENTS

The firts author wants to thank the Swiss National Science Foundation for supporting this work through the National Centre of Competence in Research on "Interactive Multimodal Information Management (IM2)". The second author thanks the the European Union which partly support this work through the 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811, publication 133).

The authors also thank Kevin Smith and Daniel Gatica-Perez for the many fruitful dicussions they had with them.

7. REFERENCES

[1] Prima-pointing head pose database. www-prima.inrialpes.fr/Pointing04/data-face.html.

[2] L. Brown and Y. Tian. A study of coarse head pose estimation. *IEEE Workshop on Motion and Video Computing*, Dec 2002.

[3] T. Cootes and P. Kittipanya-ngam. Comparing variations on the active appearance model algorithm. *BMVC*, 2002.

[4] A. Doucet, S. Godsill, and C. andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 2000.

[5] Z. Khan, T. Balch, and F. Dellaert. A rao-blackwellized particle filter for eigentracking. *CVPR*, 2004.

[6] L. Lu, Z. Zhang, H. Shum, Z. Liu, and H. Chen. Model and exemplar-based robust head pose tracking under occlusion and varying expression. *Proc. of CVPR*, Dec 2001.

[7] J. McGrath. Groups: Interaction and performance. *Prentice-Hall*, 1984.

Figure 5: Sample of tracking failure for MSPF due low number of samples. First column : MSPF; Second column: RBPF.

[8] K. Murphy and S. Russell. Rao-blackwellized particle filtering for dynamic bayesian networks. *in Sequential Monte Carlo Methods in Practice Springer-Verlag*, 2001.

[9] K. Parker. Speaking turns in small group interaction: a context sensitive event sequence model. *Journal of Personality and Social Psychology*, 1988.

[10] R. Rae and H. Ritter. Recognition of human head orientation based on artificial neural networks. *IEEE Trans. on Neural Network*, March 1998.

[11] T. Sim and S. Baker. The cmu pose, illumination, and expression database. *IEEE Trans. on PAMI*, Oct 2003.

[12] K. Toyama and A. Blake. Probabilistic tracking in metric space. *Proc. of ICCV*, Dec 2001.

[13] A. Waibel, M. Bett, F. Metze, K. Ries, T.Schaaf, T. Schultz, H. Soltau, H. Yu, and K. Zechner. Advances in automatic meeting record creation and access. *Proc. ICASSP*, May 2001.

[14] P. Wang and Q. Ji. Multi-view face tracking with factorial and switching hmm. *Workshops on Application of Computer Vision (WACV/MOTION'05),Breckenridge, Colorado*, 2005.

[15] Y. Wu and K. Toyama. Wide range illumination insensitive head orientation estimation. *IEEE Conf. on Automatic Face and Gesture Recognition*, Apr 2001.

[16] J. Yang, W. Lu, and A. Weibel. Skin color modeling and adaptation. *ACCV*, Oct 1998.

[17] L. Zhao, G. Pingai, and I. Carlbom. Real-time head orientation estimation using neural networks. *Proc. of ICIP*, Sept 2002.