IDIAP RESEARCH REPORT

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

Sileye O. Ba [a] Jean-Marc Odobez [a]

IDIAP–RR 05-35

April 2005

PUBLISHED IN
ACM ICMI Workshop on Multimodal Multiparty Meeting Processing (MMMP), 2005

[a] IDIAP Research Institute

IDIAP Research Institute **www.idiap.ch**
Rue du Simplon 4 P.O. Box 592 1920 Martigny – Switzerland
Tel: +41 27 721 77 11 Fax: +41 27 721 77 12 Email: info@idiap.ch

IDIAP Research Report 05-35

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

Sileye O. Ba Jean-Marc Odobez

April 2005

published in
ACM ICMI Workshop on Multimodal Multiparty Meeting Processing (MMMP), 2005

Abstract. This paper presents a Rao-Blackwellized mixed state particle filter for joint head tracking and pose estimation. Rao-Blackwellizing a particle filter consists of marginalizing some of the variables of the state space in order to exactly compute their posterior probability density function. Marginalizing variables reduces the dimension of the configuration space and makes the particle filter more efficient and requires a lower number of particles. Experiments were conducted on our head pose ground truth video database consisting of people engaged in meeting discussions. Results from these experiments demonstrated benefits of the Rao-Blackwellized particle filter model with fewer particles over the mixed state particle filter model.

1 Introduction

Behavior modeling is an emerging research field. Researchers have investigated how to model behaviors such as human interaction, human computer interaction and visual focus of attention (FOA), which is our main concern. 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, visual FOA may be approximated by head pose [11]. Thus, head pose estimation can be viewed as a preprocessing step for visual FOA estimation.

Many methods have been proposed to solve the problem of head tracking and pose estimation. The proposed methods can be 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, 6, 12, 9, 15, 17]. As a consequence, head pose estimation processing is dependent on head tracking accuracy. 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, 7, 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 head tracking and pose estimation with the MSPF outperforms tracking the head then estimating the pose.

In this paper, we propose to use Rao-Blackwellization [4] to improve the performances of our MSPF tracker. The Rao-Blackwellization of a mixed state, corresponding to the marginalization of some of the components of the state variable, 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 EigenTracking 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 a discrete variable with finite possible values or for a variable with a pdf defined by an analytical expression. 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. Exactly computing the pdf of some variables 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.

To the best of our knowledge, this Rao-Blackwellized tracking and head pose modeling is new. The modeling proposed in [14] is similar but, in their modeling, two probabilistic frameworks are mixed (an HMM framework and a particle filter framework). The advantage of our methodology is that it is embedded in a single probabilistic framework.

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 [10], 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:

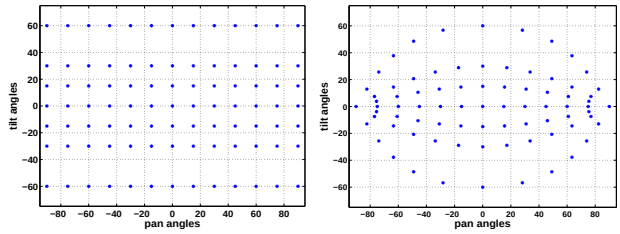

Figure 1: Left: pan-tilt space discretization in the Pointing representation. Right: same discretization in the PIE representation.

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, 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 poses. The left plot of Figure 1 shows the discretization that was used in building the Pointing database, while the right plot displays the same head poses in the PIE representation. While the discretization is regular in Pointing, this is no longer true in the PIE representation. Texture and color based head pose models are built from all the sample images available for each of the 93 discrete head poses $\theta \in \Theta = \{\theta_j = (\alpha_j, \beta_l, 0), j = 1, ..., 93\}$. 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. We sampled randomly head images consisting of very small shifted version with slight variation in size of located ground truth head images.

2.2.1 Head Pose Texture Model

Head pose texture is modeled by the output of four filters: a Gaussian at coarse scale and 3 Gabor filters at three 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. 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, by defining the head eccentricity as the ratio of the width over the height of the head, the head eccentricity distribution inside each cluster k of a head pose θ is modeled by a Gaussian $p_{r(\theta,k)}$ where the mean and the standard deviation are learned from the training head image eccentricities.

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 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. Since this likelihood have different amplitude of values depending on the exemplars, and is not discriminative with respect to background clutter, we normalized it by learned background likelihood answers. From a background image, we extracted patch images of random location and size, and computed their likelihood with respect to the exemplars. For each exemplar k of a head

pose θ, the mean of the background likelihoods $p_B(k,\theta)$ is used as normalization value. Finally, the texture likelihood with respect to an exemplar of an input patch image is defined by:

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

2.2.2 Head Pose Color Model

To make our head models more robust to background clutter, 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:

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

where λ is a hyper parameter learned from training data.

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. 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, 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:

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

The pdf at time $t-1$, $\mathrm{p}(X_{0:t-1}|z_{1:t-1})$, can be approximated with the set of particles by:

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

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

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

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 $w_t^i \propto \mathrm{p}(z_t|X_t^i)$. The new particles set

1. initialization step: $\forall\, i$ sample $X_0^i \sim p(X_0)$ 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 2: SIS Algorithm.

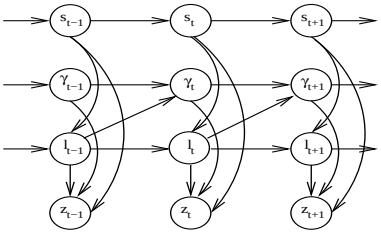

Figure 3: Mixed State Graphical Model.

$\{X_{0:t}^i, w_t^i\}_{i=1}^{N_s}$ 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 2 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.

3.1.1 State Model

The MSPF approach [13], 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:

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

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.
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.

3.1.2 Dynamic Model

The graphical model in Figure 3 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:

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

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

We defined the dynamic of the discrete variable l_t by $p(l_t|l_{t-1}, S_t) = p(l_t|l_{t-1})p(l_t|S_t)$. Let us define $p(l_t|S_t)$. Bayes rule gives $p(l_t|S_t) = \frac{p(S_t|l_t)p(l_t)}{p(S_t)}$. Assuming that $p(S_t)$ and $p(l_t)$ are uniformly distributed makes $p(l_t|S_t)$ is proportional to $p(S_t|l_t)$. We assumed also that $p(S_t|l_t)$ is characterized by the prior on the head eccentricity $p(S_t|l_t = (k_t, \theta_t)) = p_{r(k,\theta)}(\frac{s^x}{s^y})$ learned with the head pose models in Section 2.2.1. Thus, $p(l_t|l_{t-1}, S_t)$ is proportional to $p(l_t|l_{t-1})p_{r(k,\theta)}(\frac{s^x}{s^y})$.
The transition process $p(l_t|l_{t-1}) = p(\theta_t, l_t|\theta_{t-1}, l_{t-1})$ can be written:

$$p(\theta_t, k_t|\theta_{t-1}, k_{t-1}) = p(k_t|\theta_t, k_{t-1}, \theta_{t-1})p(\theta_t|\theta_{t-1}). \tag{9}$$

where the dynamics $p(\theta_t|\theta_{t-1})$ is modelled as a Gaussian process in the continuous space, and Gaussian parameters are learned from the training sequences of our dataset. This Gaussian process is then used to compute the transition matrix between the different discrete pose angles. The 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_k^θ and who are also in the class of $e_{k'}^{\theta'}$. Thus, 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 $p(z|X)$ is defined as follows :

$$p(z|X = (S, \gamma, l)) = p_{text}(z^{text}(S, \gamma)|l)p_{col}(z^{col}(S, \gamma)|l), \tag{10}$$

where the observations z are composed of texture and color observations (z^{text}, z^{col}), and 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.
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.

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

3.2 Rao-Blackwellizing the MSPF

Rao-Blackwellization can be applied when the pdf of some of the variables in the state model X can exactly computed. The exemplars label l is discrete and its set of possible values finite, its pdf can be exactly computed.

In the MSPF algorithm described previously, 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 three Gabor filters is time consuming. Reducing the number of particles by half will make the computational cost of the MSPF and the RBPF equivalent. The computation time devoted in the MSPF to the texture observations extraction for $2 \times N_s$ particles is the same than the computation time devoted in the RBPF to extract observations for N_s particles and and exactly compute the head pose exemplars pdf. Thus, Rao-Blackwellization is worthwile.

At each time our particle filters output the mean of the state X_t with respect to the pdf $p(X_{1:t}|z_{1:t})$. This mean is the same than the mean of the state X_t taken with respect to the pdf $p(S_{1:t}, \gamma_{1:t}, l_t|z_{1:t})$.

Rao-Blackwellization will be applied to this distribution which can be factorised:

$$p(S_{1:t}, \gamma_{1:t}, l_t|z_{1:t}) = p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t})p(S_{1:t}, \gamma_{1:t}|z_{1:t}) \tag{11}$$

$p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t})$ will be computed exactly and $p(S_{1:t}, \gamma_{1:t}|z_{1:t})$ approximated with SIS. In our RBPF modeling, the pdf in Equation 11 is represented by a set of particles $\{S_t^i, \gamma_t^i, \pi_t^i(l_t), w_t^i\}_{i=1}^{N_s}$ where $\pi_t^i(l_t) = p(l_t|S_{1:t}^i, \gamma_{1:t}^i, z_{1:t})$ is the pdf of the exemplars given a particle and a sequence of measurements. In the following, we give the equations to derive the exact step and the SIS step.

3.2.1 Deriving the Exact Step

We would like a formula to exactly compute the distribution $p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t})$. This is done with the prediction and update step.

Prediction Step for Variable l:

The prediction distribution $p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1})$ can be written:

$$\begin{aligned} p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) &= \sum_{l_{t-1}} p(l_t, l_{t-1}|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) \\ &= \sum_{l_{t-1}} p(l_t|l_{t-1}, S_t)p(l_{t-1}|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) \end{aligned} \tag{12}$$

Unlike the standard RBPF the 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 $p(l_{t-1}|S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1})$. This term can be computed as follows:

$$\begin{aligned} p(l_{t-1}|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) &= \frac{p(S_t, \gamma_t|l_{t-1}, S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1})p(l_{t-1}|S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1})}{p(S_t, \gamma_t|S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1})} \\ &= \frac{p(\gamma_t|\gamma_{t-1}, l_{t-1})p(S_t|S_{t-1})p(l_{t-1}|S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1})}{p(S_t, \gamma_t|S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1})} \end{aligned} \tag{13}$$

where the normalization constant is given by:

$$p(S_t, \gamma_t|S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1}) = p(S_t|S_{t-1}) \sum_{l_{t-1}} p(\gamma_t|\gamma_{t-1}, l_{t-1})p(l_{t-1}|S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1}) \tag{14}$$

Update Step for Variable l:

When new observations z_t are available the distribution $p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t})$ can be updated with the following equation:

$$\begin{aligned} p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t}) &= \frac{p(z_t|l_t, S_{1:t}, \gamma_{1:t}, z_{1:t-1})p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1})}{p(z_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1})} \\ &= \frac{p(z_t|S_t, \gamma_t, l_t)p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1})}{p(z_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1})} \end{aligned} \tag{15}$$

with normalization constant given by:

$$p(z_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) = \sum_{l_t} p(z_t|S_t, \gamma_t, l_t)p(l_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) \tag{16}$$

3.2.2 Deriving the Importance Sampling Step

The pdf $p(S_{1:t}, \gamma_{1:t}|z_{1:t})$ is computed with SIS. This pdf can be written:

$$p(S_{1:t}, \gamma_{1:t}|z_{1:t}) = \frac{p(z_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1})p(S_t, \gamma_t|S_{1:t-1}, \gamma_{1:t-1}, z_{1:t-1})}{p(z_t|z_{1:t-1})} p(S_{1:t-1}, \gamma_{1:t-1}|z_{1:t-1}) \tag{17}$$

1. **initialization step:** $\forall\ i\ (S_0^i, \gamma_0^i) \sim p(S_0, \gamma_0)$ set $\pi_0^i(l_0)$ uniform $\forall\ l_0$, set $t = 1$
2. **prediction step for S and γ:** sample $\tilde{S}_t^i$ and $\tilde{\gamma}_t^i$, $i = 1, ..., N_s$ 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. **prediction step for variable l:**
   - $\forall\ i$ compute $Z_1(i, l_{t-1}) = p(\tilde{S}_t^i|S_{t-1}^i)\pi_{t-1}^i(l_{t-1})p(\tilde{\gamma}_t^i|\gamma_{t-1}^i, l_{t-1})$
   - $\forall\ i$ compute $p(l_{t-1}|S_{1:t}^i, \gamma_{1:t}^i, z_{1:t-1}) = \frac{Z_1(i,l_{t-1})}{Z_1(i)}$ with $Z_1(i) = \sum_{l_{t-1}} Z_1(i, l_{t-1})$
   - $\forall\ i, \forall\ l_t$ compute $p(l_t|S_{1:t}^i, \gamma_{1:t}^i, z_{1:t-1}) = \sum_{l_{t-1}} p(l_t|l_{t-1}, \tilde{S}_t^i) p(l_{t-1}|S_{1:t}^i, \gamma_{1:t}^i, z_{1:t-1})$
4. **update step for variable l:**
   - $\forall\ i, \forall\ l_t$ compute $Z_2(i, z_t, l_t) = p(z_t|\tilde{S}_t^i, \tilde{\gamma}_t^i, l_t) p(l_t|S_{1:t}^i, \gamma_{1:t}^i, z_{1:t-1})$
   - $\forall\ i$ compute $\tilde{\pi}_t^i(l_t) = \frac{Z_2(i,z_t,l_t)}{Z_2(i,z_t)}$ with $Z_2(i, z_t) = \sum_{l_t} Z_2(i, z_t, l_t)$
5. **update step for variable S and γ:** $\forall\ i\ w_t^i = Z_2(i, z_t)$
6. **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 4: RBPF Algorithm.

The pdf at the previous time $t-1$ is estimated with the set of particles by:

$$p(S_{1:t-1}, \gamma_{1:t-1}|z_{1:t-1}) \approx \sum_{i=1}^{N_s} w_{t-1}^i \delta((S_{1:t-1}, \gamma_{1:t-1}) - (S_{1:t-1}^i, \gamma_{1:t-1}^i)) \tag{18}$$

thus, the current pdf can be approximated up to the proportionality constant $p(z_t|z_{1:t-1})$ by:

$$\begin{aligned} p(S_{1:t}, \gamma_{1:t}|z_{1:t}) &\approx p(z_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) \sum_{i=1}^{N_s} p(S_t, \gamma_t|S_{1:t-1}^i, \gamma_{1:t-1}^i, z_{1:t-1}) \\ &\approx p(z_t|S_{1:t}, \gamma_{1:t}, z_{1:t-1}) \sum_{i=1}^{N_s} p(S_t|S_{t-1}^i) \sum_{l_{t-1}} \pi_{t-1}^i(l_{t-1}) p(\gamma_t|\gamma_{t-1}^i, l_{t-1}) \end{aligned} \tag{19}$$

At each time step, the RBPF head pose output is obtained by first computing the head pose estimated by each particle. The head pose of each particle i is given by the average of the exemplars head pose θ_t with respect to the distribution $\pi_t^i(l_t)$. Then particles head poses are averaged with respect to the distribution of the weights w_t^i to give the head pose output of the RBPF. Figure 4 summarized 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.

4 Experiments

4.1 Dataset and Protocol Evaluation

To evaluate our algorithms, we used meeting recordings from our head pose video database in which people had their head poses continuously annotated using a magnetic field 3D location and orientation tracker (flock of bird).

	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.

For our experiments we use half of the persons of the meeting database as train set to train pose dynamic model and the half remaining persons as test set to evaluate the tracking algorithms. In each one of the recording of the 8 persons 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 of our algorithms. 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.

In this paragraph, we define the head pose estimation error measures used to evaluate tracking performances. A head pose defines a vector in the 3D space, the vector indicating where the head is pointing at. In the Pointing representation, this vector depends only on the head pan and tilt angles. The angle between the 3D pointing vector defined by the head pose ground truth and the head pose estimated by the tracker can be used as a first pose estimation error measure. 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. In order to 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 used the median value because it is less sensitive to extremal values than the mean.

4.2 Results

Experiments were conducted to compare head pose estimation based on the MSPF and the RBPF racker. 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. This is due to a better exploration of the configuration space with the RBPF as shown in Figure 6 which displays sample tracking results of one person of the test set. The first row of this figure are the results from the MSPF while the second row are results of the RBPF for the same time instants. Because of a sudden head turn and the small number of particles the MSPF explores the head spatial configuration's space with more difficulties. The configuration space's exploration could always be improved by increasing the number of particles 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.

Figure 5 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 visible on the marginalized variables (pan and tilt) although globally the improvements are noticeable for the head roll (Table 2).

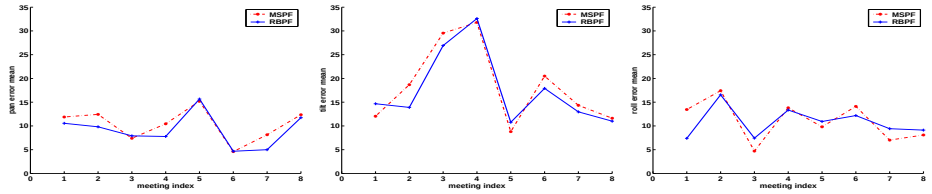

Figure 5: Pan tilt and roll errors over individual meetings.

Figure 6: Sample of tracking failure for MSPF due low number of samples. First row : MSPF; Second row RBPF.

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.

References

[1] Pointing'04 icpr workshop: Head pose image database. http://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] B. Kruger, S. Bruns, and G. Sommer. Efficient head pose estimation with gabor wavelet. *Proc. of BMVC*, Sept 2000.

[7] 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.

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

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

[11] R. Stiefelhagen, J. Yang, and A. Waibel. Estimating focus of attention based on gaze and sound. *Workshop on Perceptive User Interfaces (PUI'01), Orlando, Florida*, 2001.

[12] R. Stiefelhagen and J. Zhu. Head orientation and gaze direction in meetings. *Conference on Human Factors in Computing Systems, Minneapolis, Minnesota, USA*, 2002.

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