Evaluation of Multiple Cues Head Pose Tracking algorithms in Indoor Environments

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

IDIAP–RR 05-05

January 2005

submitted for publication

[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 05-05

Evaluation of Multiple Cues Head Pose Tracking algorithms in Indoor Environments

Sileye O. Ba Jean-Marc odobez

January 2005

submitted for publication

Abstract. Head pose estimation is a research area which has many applications, e.g. in human computer interfaces design or in the analysis of people's focus-of-attention. The paper addresses the issue of head pose estimation, and makes two contributions. First it introduces a database of more than 2 hours of video with head pose annotation involving people engaged in office activities or meeting discussion. The database will be made publicly available. The second is an algorithm which couples tracking and head pose estimation in a mixed-state particle filter. The approach combines the robustness of color-based tracking by exploiting skin head/face models with the localization accuracy of texture-based head models, as demonstrated by the reported experiments.

1 Introduction

The automatic analysis of the gestures, activities and behavior of people constitutes an emerging research field in computer science. It can rely on the extraction of many person-oriented information, such as their localization, the localization of their limbs, or their speaking activity. In particular, the visual focus-of-attention (FOA) plays an important role in the recognition of people activity or the understanding of non-verbal behavior in human interactions. In principle, the FOA should be estimated from a person's gaze. However, in the absence of high-resolution images of faces, which prevents from the analysis of eyes orientation, the head pose can be employed as a surrogate.

A large amount of head pose algorithms have been proposed in the past. However, in most cases, algorithms are evaluated either qualitatively [1] on some sample videos, or quantitatively but on static images (e.g. [2, 3, 1]). There are several exceptions (e.g. [4]), but unfortunately, no data has been made publicly available. Moreover, in many occasions, the recorded sequences involve people performing constrained head motions in front of the camera, a situation which does not reflect the whole variety of natural head attitudes encountered in real environments. In this paper, we introduce a video database with 3D head pose ground-truth that will be made publicly available. The videos depict people engaged in either some office activity, or in a meeting discussion. The ground-truth has been obtained by exploiting the output of magnetic flock-of-birds (FOB) sensors attached to people's head. We believe that the use of common databases is important to evaluate and compare different algorithms, in order to have a better understanding of them, and hope that our database will contribute to such goals.

The second contribution of the paper is an algorithm that performs jointly head tracking and pose estimation, exploiting both texture and skin information. Most of the existing work for head tracking and pose estimation defines the task as two sequential and separate problems: the head is tracked, its location is extracted and *then* used for pose estimation [2, 5, 6, 7, 4, 8, 9]. As a consequence, the estimated head pose totally depends on the tracking accuracy. Indeed, it has been showed in the past [2] that head pose estimation is very sensitive to head location. Hence, the above formulation of the task misses the fact that knowledge about head pose could be used to improve head modeling and thus improve tracking accuracy. Thus, like others [10, 11] before, we recently proposed [1] an algorithm that couples the head tracking and pose estimation problem. The method relies on a Bayesian formulation of the task, which is implemented using a particle filter (PF) approach [12]. The head modeling is achieved by learning discrete head pose models from training sets [2]. In [1], only texture-like features were used. We preferred this approach to the use of 3D head models, since the latter usually require higher resolution head images than those considered in our experiments. Initial results evaluated on some sample sequences using manual ground-truth showed that the algorithm worked better than the track-then-pose paradigm. In this paper, this is confirmed on the much larger database described above. However, these experiments also show that due to the presence of highly textured background in our data (see Fig. 4), the tracker sometimes temporarily locks on the background. To improve its robustness, we propose here to exploit skin masks to model head poses, and during tracking, to automatically build skin maps using a skin color adaptation framework. This way, the approach combines the robustness of standard color trackers [13] with the accuracy of textured-based head modeling.

The paper is organized as follows. Section 2 describes the head pose representation and head modeling. Section 3 presents the probabilistic setting for joint head tracking and pose estimation. Section 4 compares head pose tracking algorithms and Section 5 concludes the paper.

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

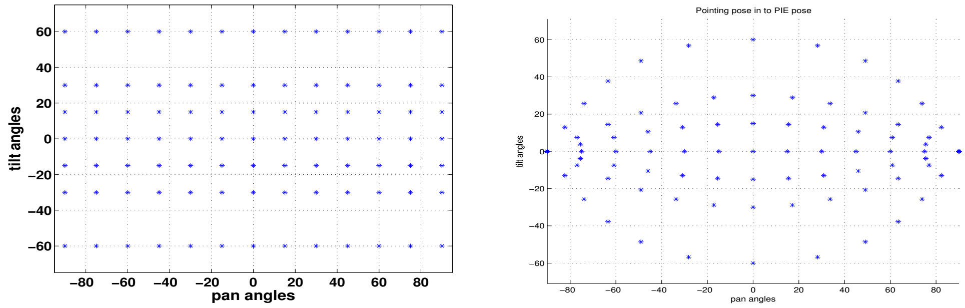

Figure 1: Left: pan-tilt space discretization in the Pointing representation. Right: same discretization in the PIE representation.

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 [3], the rotation axis are rigidly attached to the head. In the PIE representation [14], the rotation axis are those of the camera frame. The Pointing representation leads itself to more natural head-centric values, with direct interpretation. However, the PIE representation has a computational advantage: the roll angle corresponds to in-plane rotations. Thus, only appearances of poses with varying pan and tilt values need to be modeled, as the head roll can be simulated 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 of poses than the one found in other databases (e.g. Ferret, PIE). 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.

2.2.1 Head Pose Texture Model

Head pose texture is modeled by the output of four filters $\Psi_i, 1 = 1, ..., 4$: a Gaussian at coarse scale and 3 Gabor filters at three different scales (finer to coarser). Training image patches are obtained by locating a tight bounding box around the head. The patch image is resized to the same resolution 64 × 64 and preprocessed by histogram equalization to reduce the effect of lighting conditions. Then, patches are filtered by each of the above filters at locations of a grid G inside a head mask. For each filter Ψ_i, the features computed from an image patch $\{f^i_j,\ j \in G\}$ are normalized to give $\tilde{f}^i = \{\tilde{f}^i_j = \frac{f^i_j - m_i}{s_i},\ j \in G\}$, where m_i and s^2_i are the mean and variance of the ith features. This normalization is made to prevent the features of a filter to dominate the other because their values are higher. These features are then concatenated in a single feature vector $z = \{\tilde{f}^i,\ i = 1, 2, 3, 4\}$.

The feature vectors associated with each head pose $\theta \in \Theta$ are clustered into K clusters using a kmeans algorithm. The cluster centers $e^\theta_k = (e^\theta_{k,i}), k = 1, ..., K$ are taken to be the exemplars of the head pose θ. The diagonal covariance matrix of the features $\sigma^\theta_k = diag(\sigma^\theta_{k,i})$ inside each cluster is also exploited to define the pose likelihood models. Here, due to the small amount of 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 texture likelihood of an input image characterized by its extracted features z^T, with respect to an exemplars k of the head pose θ, is given by:

$$p_T(z|k,\theta) = \prod_i \frac{1}{\sigma^\theta_{k,i}} \max(\exp - \frac{1}{2}\left(\frac{z^T_i - e^\theta_{k,i}}{\sigma^\theta_{k,i}}\right)^2, T) \tag{1}$$

where $T = \exp - \frac{9}{2}$ is a lower threshold set to reduce the effects of outlier components of the feature vector.

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^θ_k a face skin color model denoted by M^θ_k using the training images belonging to the cluster of this exemplar. Training images are resized to 64×64, then their pixels are classified interactively as skin or non skin. The skin model M^θ_k 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.

To detect skin pixels at run time, we model the distribution of skin pixel values with a single Gaussian distribution in the normalized (r,g) feature space, as it has been shown in [15] that such a model holds well for people of any race. Thus, the parameters of a general skin color model (means and variances), denoted by m_0, are learned using the whole set of Pointing training images in the database. These parameters are used in the first image of any test sequence, and then adapted through time using standard a Maximum A Posteriori technique, leading to the parameters m_t at time t. The measurements used at time t for adaptation are computed from the image pixels extracted using the estimated mean state of the head (see next section), taking into account both the 2D spatial head localization parameters and the estimated pose, which, through the skin mask, tells which pixels of the head corresponds to the face part.

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 by thresholding the skin likelihood obtained using the skin color distribution model with parameter m_t. The resulting skin mask is then compared against the model M^θ_k, and we defined the likelihood as:

$$p_c(z|k,\theta) \propto \exp -\lambda ||z^C_t - M^\theta_k||_1 \tag{2}$$

where λ is a hyper parameter learned from training data.

3 Head Pose Tracking

3.1 Mixed State Particle Filter

Particle filtering (PF) implements a recursive Bayesian filter by Monte-Carlo simulations. Let $X_{0:t} = \{X_j, j = 0, \ldots, t\}$ (resp. $z_{1:t} = \{z_j, j = 1, \ldots, t\}$) represents the sequence of states (resp. of observations) up to time t. Furthermore, let $\{X^i_{0:t}, w^i_t\}^{N_s}_{i=1}$ denote a set of weighted samples that characterizes $pX_{0:t}|z_{0:t}$ the posterior probability density function (pdf), where $\{X^i_{0:t}, i = 1, \ldots, N_s\}$ is a set of support points with associated weights w^i_t. The samples and weights can be chosen using the Sequential Importance Sampling (SIS) principle [12]. 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, and that the prior distribution $pX_{0:t}$ is employed as proposal, we obtain the following recursive update equation [12] for the weight $w^i_t \propto w^i_{t-1}\; pz_t|X^i_t$. To avoid sampling degeneracy an additional resampling step is necessary [12]. The standard PF is given by :

1. Initialization : $\forall i$, sample $X^i_0 \sim pX_0$; set $t = 1$
2. IS step: $\forall i$ sample $\tilde{X}^i_t \sim pX^i_t|X^i_{t-1}$; evaluate $\tilde{w}^i_t$.

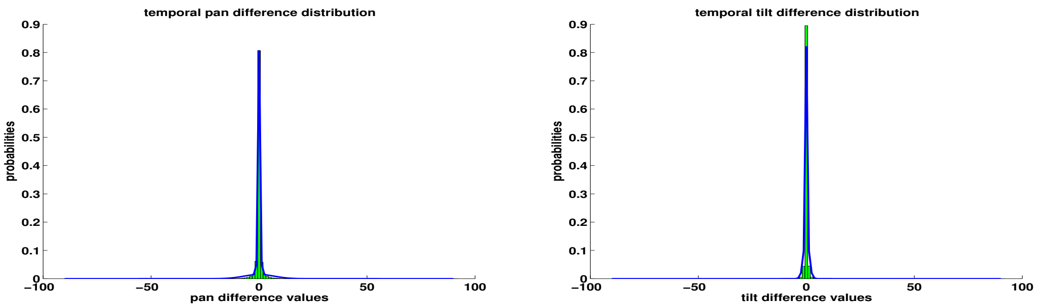

Figure 2: green: histogram of (pan/tilt) differences; blue: fitted GMM to temporal differences

3. Selection: Resample N_s particles $\{X_t^i, w_t^i = \frac{1}{N_s}\}$ from the set $\{\tilde{X}_t^i, \tilde{w}_t^i\}$; set $t = t + 1$; go to step 2.

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

3.2 State Space

The mixed state particle filter approach [16], 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 the continuous variable $S = (x, y, s^x, s^y)$ and discrete variable γ, which together parameterize the 2D spatial transform $\mathcal{T}_{(S,\gamma)}$, and of the discrete index $l = (\theta, k)$ which labels an element of the set of head pose models e_k^θ. The transform $\mathcal{T}_{(S,\gamma)}$ is defined by:

$$\mathcal{T}_{(S,\gamma)} u = \begin{pmatrix} s^x & 0 \\ 0 & s^y \end{pmatrix} \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix} u + \begin{pmatrix} x \\ y \end{pmatrix}. \tag{3}$$

and characterizes the image object configuration. (x, y) specifies the translation, i.e. the 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 angle of the object. The γ parameter was discretized for convenience, though this not a necessity of the approach.

We need to define what we use as output of the particle filter. The set of particle defines a probability density function (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 indexes indeed correspond to real Euler angles which can be combined to produce an average output.

3.3 Dynamical models

The process density on the state sequence is modeled as a second order process $P(X_t|X_{t-1}, X_{t-2})$. [1] We assume that the three components of the states are conditionally independent, and that a head pose at a given time t, l_t, depends only on the head pose at the previous time l_{t-1}. Then the equation of the process density is given by:

$$p(X_t|X_{t-1}, X_{t-2}) = p(S_t|S_{t-1}, S_{t-2}) \times p(l_t|l_{t-1}) \times p(\gamma_t|\gamma_{t-1}, l_{t-1}, l_t) \tag{4}$$

[1] By letting the state corresponds to the augmented state $X_t^\star = (X_t, X_{t-1})$, it is easy to show that we end up with a first order process, as assumed in Section 3.1.

Let us now describe these three terms.

The dynamic of the continuous variable S_t is modeled as a second order auto regressive dynamical model, which includes the prior model on the head eccentricity (see 2.2.1) $p_{r(k,\theta)}(\frac{s^x}{s^y})$. The expression of this process dynamic is given by:

$$p(S_t|S_{t-1},S_{t-2}) \propto \mathcal{N}\left(\left[\begin{array}{c} x_t \\ y_t \\ s_t^x \\ s_t^y \end{array}\right], \left[\begin{array}{c} x_{t-1}+(x_{t-1}-x_{t-2}) \\ y_{t-1}+(y_{t-1}-y_{t-2}) \\ s_{t-1}^x \\ s_{t-1}^y \end{array}\right], \left[\begin{array}{cccc} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_{s^x}^2 & 0 \\ 0 & 0 & 0 & \sigma_{s^y}^2 \end{array}\right]\right) \times p_{r(k,\theta)}(\frac{s^x}{s^y}) \tag{5}$$

where $\mathcal{N}(u,m,\Sigma)$ denotes the density function of a Gaussian with mean m and covariance matrix Σ evaluated at a point u. The diagonal covariance matrix $\Sigma = \text{diag}(\sigma_x^2, \sigma_y^2, \sigma_{s^x}^2, \sigma_{s^y}^2)$ encodes the uncertainty in predicting S_t knowing S_{t-1} and S_{t-2}.

The dynamic of the discrete variable l_t is defined by the transition process $p(l_t|l_{t-1})$:

$$p(l_t|l_{t-1}) = p(\theta_t,k_t|\theta_{t-1},k_{t-1}) = p(k_t|\theta_t,k_{t-1},\theta_{t-1})p(\theta_t|\theta_{t-1}). \tag{6}$$

where the dynamics $p(\theta_t|\theta_{t-1})$ models the transitions between head pose and $p(k_t|\theta_t,k_{t-1},\theta_{t-1})$ models transition between exemplars given the head poses. Transition between poses $p(\theta_t|\theta_{t-1})$ is learned from continuous head pose ground truth. First we assume that for each pose $\theta = (\alpha,\beta)$ the pan component α and the tilt component β are independent leading to $p(\theta_t|\theta_{t-1}) = p(\alpha_t|\alpha_{t-1})p(\beta_t|\beta_{t-1})$. Then the temporal pan and tilt differences $\alpha_t - \alpha_{t-1}$ and $\beta_t - \beta_{t-1}$ are modeled as two Gaussian mixture models (GMM) p_α and p_β in the continuous space. The parameters are obtained by fitting a GMM with two mixture components to the temporal differences. Intuitively one mixture component will model situations when the head remains static, while the other one will account for head pose variations when a person moves his head. Figure 2 displays the fitted mixtures. These Gaussian mixture processes are then used to compute the transition matrix between the discrete poses:

$$p(\theta_t|\theta_{t-1}) \propto p_\alpha(\alpha_t - \alpha_{t-1})p_\beta(\beta_t - \beta_{t-1}) \tag{7}$$

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. We assume that two exemplars of different head poses are more related when the head images of the same persons were used to build them. Let us denote by $\mathcal{L}(\theta,k)$ the set of person's labels used to build the exemplar k of the head pose θ and $|E|$ the number of elements of a set E. When $\theta \neq \theta'$, the transition between the exemplar k' of head pose θ' to the exemplars k of head pose θ is taken to be

$$p(k|\theta,k',\theta') = \frac{|\mathcal{L}(\theta',k')\bigcap\mathcal{L}(\theta,k)|}{|\mathcal{L}(\theta',k')|} \tag{8}$$

and, when $\theta = \theta'$,

$$p(k|\theta,k',\theta) = \left\{ \begin{array}{r} 1-(K-1)\epsilon \text{ if } k=k' \\ \epsilon \text{ otherwise} \end{array} \right. \tag{9}$$

with ϵ taken small.

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 is divided into nine regions $\Theta_i,\ i = 1,...,9$, with pan and tilt ranging from -90 to 90 with a step of 60 degrees. Inside each region Θ_i, a Gaussian distribution $p_{\gamma,i}$ is fitted to the roll temporal differences $\gamma_t - \gamma_{t-1}$ of the training data of head pose with pan tilt in Θ_i. Also a prior distribution on the roll values $p_{\Theta_i}(\gamma)$ is learned by fitting a Gaussian distribution to the roll values of head pose with pan-tilt values in Θ_i. If we define $\Phi : \theta \rightarrow i$ to be the mapping between the pan-tilt space to the indexes of the 9 regions, the roll transition is defined as

$$p(\gamma_t|\gamma_{t-1}, l_t = (k_t,\theta_t)) \propto p_{\gamma,\Phi(\theta_t)}(\gamma_t - \gamma_{t-1})p_{\Theta_{\Phi(\theta_t)}}(\gamma_t) \tag{10}$$

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.

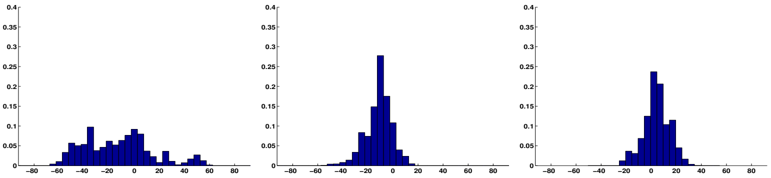

Figure 3: Histograms of pan (left) tilt (center) and roll(right) of the test data values in the Pointing Representation

3.4 observation models

The observation likelihood $p(z|X)$ is defined as follows :

$$p(z|X = (S,\gamma,l)\,) = p_T(z^T(S,\gamma)|l)p_c(z^C(S,\gamma)|l), \tag{11}$$

where the observations z are composed of texture and color observations (z^T, z^C), and we have assumed that these observations where conditionally independent given the state. The texture likelihood p_T and the color likelihood p_c 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.

4 HEAD POSE TRACKING EVALUATION

4.1 Dataset and Protocol Evaluation

We built a head pose video database of people in real situation with their head poses 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 frame with the camera frame, we can for each image output 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 later case, 8 meetings were recorded and each lasted 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 no restrictions on head motions or on head poses. As a result, we obtained a meeting video database of 16 different annotated people.

The tracking evaluation protocol is the following. 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. Figure 3 shows the distribution of the pan, tilt and roll values on the evaluation data in the PIE representation. Because of the scenario used to record data, people have more frequently negative pan values corresponding to them looking at the projection screen located at the right of them (cf Fig. 4). The majority of pan values ranges from -60 to 60 degree. Tilt values range 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. It can be thought of as a vector based on the center of the head and passing trough

	pose vector error	pan errors	tilt errors	roll errors
M1 (texture)	28.1	16.2	22.4	15.1
M1 (texture+color)	28.4	16.6	27.2	13.9
M2 (texture)	32.6	19.0	26.4	16.1
M2 (texture+color)	32.7	19.9	26.2	14.6

Table 1: mean of pose vector, pan, tilt and roll errors for tracking then pose estimation methods (M1 and M2) with texture likelihood or texture+color likelihood for pose estimation

	mean	std	median
M1	28.15	14.6	25.2
M2	32.6	17.7	29.2
M3	23.4	16.6	19.2
M4	21.3	15.2	14.1

Table 2: Mean, standard deviation and median of head pointing vector errors over evaluation data

the nose. It is worth noticing that 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 (GT) and the head pose estimated by the tracker can be used as the first pose estimation error measure. This measure of error will be well suited for studies on the focus of attention, 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 value of the errors. We used the median value because it is less sensitive to extremal values than the mean. Thus, the median value will be less biased by short time period pose estimation errors due to bad head localization. Before describing the experimental results, let us remind that all the error measures are computed in the Pointing representation.

4.2 Experimental Results

Experiments were conducted to compare two classes of trackers. The first class track the head then estimates the pose. In this class we used two methods, an histogram and correlation tracker (M1) [17] and an histogram, correlation and shape tracker (M2) [17]. The principle of estimating the pose with these two methods is the following. At each time t the tracker outputs the head center location and size $\widehat{S}_t = (\hat{x}_t, \hat{y}_t, \hat{s}^x_t, \hat{s}^y_t)$. Then, for all the discrete roll values γ_i, the image patch corresponding to the spatial configuration $(\widehat{S}_t, \gamma_i)$ is extracted and the texture features $z_t(\widehat{S}_t, \gamma)$ computed. Finally, the head pose $(\hat{\theta}_t, \hat{\gamma}_t)$ is estimated by a MAP principle $(\hat{\theta}_t, \hat{\gamma}_t) = \arg\max_{\theta,\gamma} p(z_t(\widehat{S}_t, \gamma)|\theta)$ with:

$$p(z_t(\widehat{S}_t, \gamma)|\theta) = \sum_{k=1}^{K} \pi^{\theta}_k p_T(z_t(\widehat{S}_t, \gamma)|\theta, k) \tag{12}$$

where π^{θ}_k is proportional to the number of images used to build exemplar k. Equation 12 corresponds to modeling a head pose θ as a GMM with it's corresponding exemplars as mixture centers and π^{θ}_k as mixture weights. The likelihood model for tracking then head pose estimation in Equation 12 does not include the color likelihood part of Equation 11. Indeed, experiments we have conducted using a joint texture and color likelihood gave similar results (see Table 1) . Color is helpful when searching for a good head localization, but given the head localization head texture is sufficient for pose determination.

	pan			tilt			roll		
	mean	std	med	mean	std	med	mean	std	med
M1	16.2	13.6	13.1	22.4	15.0	19.1	15.1	12.0	12.5
M2	19.0	17.4	14.2	26.4	17.5	21.5	16.1	12.7	13.4
M3	13.6	14.9	8.3	17.6	13.8	12.8	11.5	10.3	12.9
M4	8.7	9.1	6.2	19.1	15.41	14.0	9.7	7.1	8.6

Table 3: pan, tilt and roll errors statistics over evaluation data (Pointing representation)

The second set of algorithms jointly track head and estimate pose. Two methods were also used in this class. Both methods follow the framework described in Section 3 of this paper. The first tracker (M3) rely on head texture likelihood models only (i.e only $p_T(.)$ of Equation 1 is used in Equation 11) while the second (M4) exploits both texture and color likelihood models.

We ran the four trackers on the test data . Table 2 reports the head pointing vector errors of the four methods over the whole set of evaluation data. The mean and the median errors are smaller for methods M3 and M4. As illustrated in Figure 4, this is due to a better head localization obtained by the methods performing jointly tracking and head pose estimation. Furthermore M4 is surpassing M3 because of the use of the multiple visual cues. More precisely, the Texture cue is very accurate for head pose estimation but is very sensitive to localization accuracy: the texture likelihood function is very peaky. Moreover the texture cue is sometimes distracted by the heavy cluttered background (see Figure 4). The color cue is complementary to the texture cue because it's likelihood is smoother and help in removing most of the ambiguities. According to the head pointing error measure the ranking of the methods from best to worst is M4, M3, M1, and M2.

Table 3 provides the pan, tilt and roll error measures. As for the head pointing errors, the mean and the median of the errors are smaller for methods performing jointly tracking and pose estimation (M3 and M4). The results of Table 3 are showing also that for all the methods, the head pan and head roll estimation are more accurate than the head tilt estimation. This is due to the fact that head tilt estimation is more sensitive to head head localization than head pan estimation, as also reported in [2]. To have more details about the performances of (M4), we computed the mean of the pan tilt and roll error values depending on whether the absolute value of the pan component of the head pose ground truth is lower or higher than 45 degrees. In our test data, the head poses which have a pan value between -45 and 45 degrees represent 79% of the data. The errors are displayed in Table 4. For comparison purposes, this table displays also the results reported in [4] (Wu 01) for a similar experiment. More precisely, the authors used 10 persons to train head models, and used these models to perform head tracking and estimation of a person who was not part of the training set. For this experiment they reported the mean of pan and tilt errors in the Pointing representation. From the results of our tracker (M4) we can conclude that pan estimation is more reliable when the pan value is in the interval $[-45, 45]$. According to the results, our method M4 is performing much better than Wu 01 for pan estimation. For head tilt estimation Wu 01 performs better when pan values are within $[-45, 45]$. A possible explanation is that we have more head tilt variations in our test data. In our test data, the tilt angle are varying from -60 to 15 degrees (see Figure 3). Also for near frontal head pose , head appearances are very similar for different tilt values and are person dependent. When pan values are out of the range $[-45, 45]$ their is a noticeable increase of performance of our method M4 for head tilt estimation and it performs better than Wu 01.

Finally, results on individual people are displayed in Figure 5. The results of this figure show that for all the persons, method M4 estimates the pan and roll with lower errors. Additionnaly they show that there are substantial performance variations across people. This is in good part due to presence or not of a similar looking head in the training set. (e.g. person 5).

	abs(pan of GT)≤ 45			45 < abs(pan of GT)≤ 90		
	pan	tilt	roll	pan	tilt	roll
M4	7.6	20.86	8.05	13.5	11.6	17.1
Wu 01	19.2	12.0	×	33.6	16.3	×

Table 4: mean of pan, tilt and roll errors for abs(pan of GT)≤ 45 and 45 < abs(pan of GT)≤ 90 (Pointing representation)

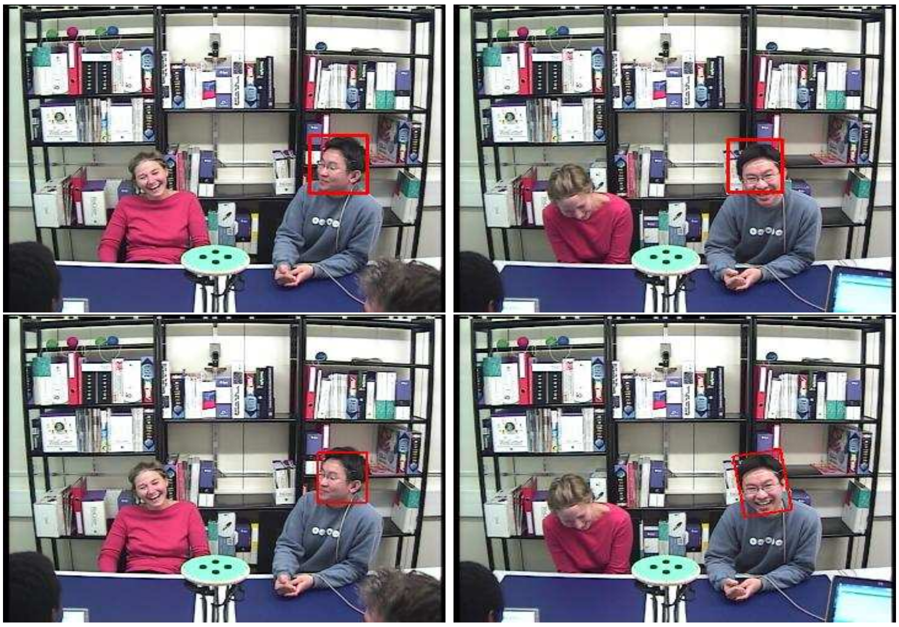

Figure 4: Head localization results for M2 (top row) and M4 (bottom row); left column: frame 571; right column: frame 661

5 Conclusion

In this paper, we described a probabilistic setting for joint head tracking and pose estimation with multiple visual cues. This algorithm was compared to three other algorithms on a set of 8 one minute long annotated real data sequences with a defined protocol of evaluation. The experimental results show that our method outperforms the others for two main reasons. Firstly, the method performs the tracking and pose estimation tasks jointly. Secondly, the use of multiple cues improves head localization.

The test data are part of a larger database which comprises more than two hours of annotated data. This database will be made public, as well as the protocol we followed. We hope that, as people have been working on head pose tracking for many years, such a database will be helpful in allowing for better algorithm evaluation and performance comparison.

References

[1] S.O. Ba. and J.M. Odobez, "A probabilistic framework for joint head tracking and pose estimation," in *Proc. of International Conference Pattern Recognition (ICPR)*, Cambridge, UK, Aug

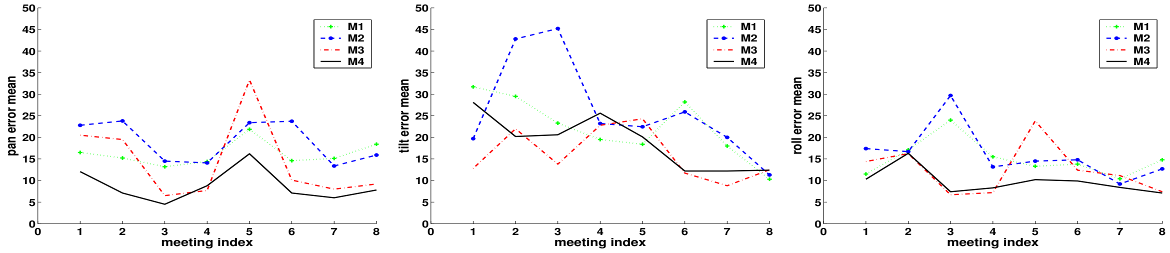

Figure 5: Mean of pan, tilt and roll pose estimation errors for individual meeting evaluation data

2004, pp. 264–267.

[2] L. Brown and Y. Tian, "A study of coarse head pose estimation," in *Proc. of IEEE Workshop on Motion and Video Computing*, Orlando Florida, Dec 2002, pp. 183–191.

[3] "Pointing'04 icpr workshop: Head pose image database," .

[4] Y. Wu and K. Toyama, "Wide range illumination insensitive head orientation estimation," in *Proc. of IEEE Conf. on Automatic Face and Gesture Recognition (AFGR)*, Grenoble France, Apr 2001, pp. 183–188.

[5] B. Kruger, S. Bruns, and G. Sommer, "Efficient head pose estimation with gabor wavelet," in *Proc. of British Machine Vision Conference (BMVC)*, Bristol UK, Sept 2000.

[6] S. Niyogi and W. Freeman, "Example-based head tracking," in *Proc. of Int. Conf. on Automatic Face and Gesture Recognition (ICAFGR)*, Killington, Vermont USA, Oct 1996, pp. 374–377.

[7] R. Rae and H. Ritter, "Recognition of human head orientation based on artificial neural networks," *IEEE Trans. on Neural Network*, vol. 9(2), pp. 257–265, March 1998.

[8] L. Zhao, G. Pingai, and I. Carlbom, "Real-time head orientation estimation using neural networks," in *Proc. of International Conference on Image Processing (ICIP)*, Rochester New York, Sept 2002, pp. 297–300.

[9] R. Stiefelhagen, J. Yang, and A. Waibel, "Estimating focus of attention based on gaze and sound," in *Proc of Workshop on Perceptive User Interface (PUI)*, Florida, USA, Nov 2001, pp. 1–9.

[10] L. Lu, Z. Zhang, H. Shum, Z. Liu, and H. Chen, "Model and exemplar-based robust head pose tracking under occlusion and varying expression," in *Proc. of Conf of Computer Vision and pattern Recognition (CVPR)*, Kauai Marriott, Hawaii USA, Dec 2001.

[11] T. Cootes and P. Kittipanya-ngam, "Comparing variations on the active appearance model algorithm," in *Proc. of British Machine Vision Conference (BMVC)*, Norwich UK, Sept 2002.

[12] A Doucet, S.Godsill, and C. Andrieu, "On sequential monte carlo method for bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, Aug 2000.

[13] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color based probabilistic tracking," in *Proc. of European Conference on Computer Vision (ECCV)*, Copenhagen Danmark, May 2002, pp. 661–675.

[14] T. Sim and S. Baker, "The cmu pose, illumination, and expression database," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 25(12), pp. 1615–1618, Oct 2003.

[15] J. Yang, W. Lu, and A. Weibel, “Skin color modeling and adaptation,” in *Proc. of Asian Conference Computer Vision (ACCV)*, Hong Kong, China, Oct 1998, pp. 687–694.

[16] K. Toyama and A. Blake, “Probabilistic tracking in metric space,” *International Journal of Computer Vision*, vol. 48, pp. 9–19, June 2002.

[17] J.M. Odobez, S.O. Ba., and D Gatica-Perez, “Embedding motion in model-based stochastic tracking,” in *Proc. of International Conference Pattern Recognition (ICPR)*, Cambridge, UK, Aug 2004.