IDIAP RESEARCH REPORT

A Probabilistic Framework for Joint Head Tracking and Pose Estimation

Sileye Ba [*] Jean-Marc Odobez [*]

IDIAP–RR 03-78

December 22, 2003

SUBMITTED FOR PUBLICATION

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 − 27 − 721 77 11
fax +41 − 27 − 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[*] IDIAP, Martigny, Switzerland

IDIAP Research Report 03-78

A PROBABILISTIC FRAMEWORK FOR JOINT HEAD TRACKING AND POSE ESTIMATION

Sileye Ba Jean-Marc Odobez

DECEMBER 22, 2003

SUBMITTED FOR PUBLICATION

Abstract. Head Tracking and pose estimation are usually considered as two sequential and separate problems: pose is estimated on the head patch provided by a tracking module. However, precision in head pose estimation is dependent on tracking accuracy which itself could benefit from the head orientation knowledge. Therefore, this work considers head tracking and pose estimation as two coupled problems in a probabilistic setting. Head pose models are learned and incorporated into a mixed-state particle filter framework for joint head tracking and pose estimation. Experimental results on real sequences show the effectiveness of the method in estimating more stable and accurate pose values.

1 Introduction

Head detection and tracking are essential components in video applications related to human behaviour understanding. It is commonly used as a first step before applying algorithms for other higher level tasks, such as face and facial expresion recognition or gaze direction estimation. At the same time, the estimation of the head pose could be useful for behaviour understanding and to improve the higher level tasks.

Many methods have been proposed to estimate head pose [1], [3], [4], [7], [10],[11]. To our knowledge, the previous work consider tracking and head pose estimation as two sequential but independent problems. The principle of these methods is to first track the head to extract its location, and then to estimate head orientation by exploiting this location. As a consequence, the head pose estimation process is very dependent on the accuracy of the tracking since, as reported in [1], head pose is very sensitive to the localization of the extracted head box. At the same time, the knowledge of the head pose could improve the head modeling and thus the accuracy of the tracking. This paper addresses these issues by coupling the tracking and head pose estimation processes in a probabilistic setting. For this purpose, a mixed-state particle filter framework is used [9], where a head spatial configuration (e.g. position and scale) and its pose are represented in a joint state-space model. The joint posterior distribution of the state given the sequence of images is estimated at each instant and propagated to the next time instant using the state dynamic. The pose at a given instant is then obtained by marginalyzing over the spatial configuration part of the state. As a result, in the approach we propose, the spatial configurations leading to a better pose modeling will have a greater impact on the pose result, leading to a more accurate estimation of the pose than with the tracking *then* pose estimation approach. This is supported by experiments performed on several real sequences.

This paper is organized as follows. Section 2 describes our head pose modeling. Section 3 shows the embedding of these pose models in a mixed-state particle filter framework. Section 4 reports results of pose estimation on still images and tracking results on real sequences. Section 5 gives the conclusions.

2 Head Pose Modeling and Estimation

2.1 Head Pose Modeling and Learning

The head poses are defined by a pan angle denoted θ and ranging from -90 to 90 degrees[1]. Allowed values are discretized with a 22.5 degrees step. Training data patches are extracted from head images by locating a tight bounding box around the head. These patch images are resized to the same 64×64 resolution and preprocessed by histogram equalization to reduce the effect of lighting conditions. Four filters, one Gaussian and three rotation invariant Gabor

[1] An additional tilt angle is also considered but is left aside for brievity in the presentation.

a) b)

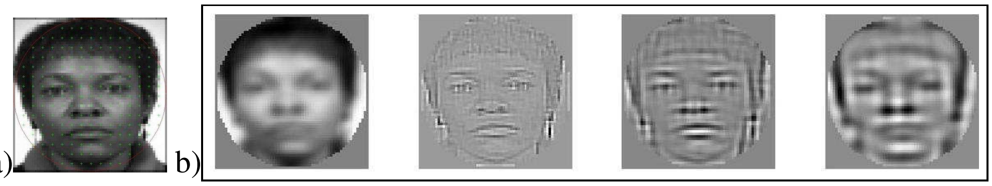

Figure 1: a) Reference grid on a frontal head pose b) The four image features computed from the frontal head pose.

wavelets, are then applied on these patches (Fig. 1). A simple Gabor wavelet is defined by:

$$\psi_{\omega_0,\sigma,\alpha}(x,y) = \exp\left(-\frac{1}{2\sigma^2}\left(x'^2 + y'^2\right)\right)\cos(2\pi\omega_0 x')$$
$$x' = x\cos\alpha - y\sin\alpha \text{ and } y' = x\sin\alpha + y\cos\alpha$$

where ω_0 denotes the angular frequency, σ the scale parameter and α the orientation of the wavelet. A rotation invariant wavelet is obtained by integrating a simple wavelet over the orientation α. The rotation invariant Gabor wavelet we used are defined by the scales $\sigma = 1, 2, 4$ and and angular frequency $w_0 = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$. The resulting images are sampled at 191 points of a grid G regularly located inside a reference disk $\mathcal{C}$ of center $(32.5, 32.5)$ and of radius 31.5 (Fig. 1a).

For each filter Ψ_i, the features computed from an image $\{f^i_j,\ j \in G\}$ 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 represent the mean and variance of the i features, and are given by :

$$m_i = \frac{1}{|G|}\sum_{j\in G} f^i_j \qquad \text{and} \qquad s^2_i = \frac{1}{|G|}\sum_{j\in G} {f^i_j}^2 - m^2_i \tag{1}$$

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 $z = \{\tilde{f}^i,\ i = 1, 2, 3, 4\}$.

To learn the model of a head pose we use the CMU PIE database [8], which contain 68 persons at the needed head poses. For each head pose θ, the feature vectors are clustered in K clusters using a Kmeans algorithms. The K centers of cluster, $e^\theta_k,\ k = 1, ..., K$ are taken to be the models of the head pose. For each head pose the standard deviation of the features σ^θ_k and the normalized number of element of each cluster π^θ_k are kept. The Kmeans procedure was preferred to others modeling methods like Gaussian mixture model because our interest is in modeling representative exemplars of head pose and not directly the probability distribution of the features.

2.2 Head Pose Estimation

The head pose of an input image characterized by its feature z is estimated using the maximum a posteriori principle :

$$\theta^* = \arg\max_\theta p(\theta|z) = \arg\max_\theta \frac{p(z|\theta)p(\theta)}{p(z)}. \tag{2}$$

Assuming for static images that $p(\theta)$ is uniformly distributed, the MAP estimation resume to $\theta^* = \arg\max_\theta p(z|\theta)$. We assume that for each head pose θ the components of the feature vector are independent and can be modeled by a Gaussian mixture having as center the examplars $e_k^\theta,\ \ k = 1, ..., K$, as diagonal covariance matrix $\Sigma^\theta = \text{diag}(\sigma_k^{\theta 2})$ and as probability mixtures π_k^θ. The probability of data given a head pose is modeled by:

$$p(z|\theta) = \sum_{k=1}^{K} \pi_k^\theta p(z|k) \text{ with:} \tag{3}$$

$$p(z|k) = \prod_i \frac{1}{\sigma_{k,i}^\theta} \exp -\frac{1}{2}\left(\frac{z_i - e_{k,i}^\theta}{\sigma_{k,i}^\theta}\right)^2. \tag{4}$$

As components of a feature vector can be outliers, we will also use the saturated Gaussian likelihood:

$$p_T(z|k) = \prod_i \frac{1}{\sigma_{k,i}^\theta} \max\left\{\exp -\frac{1}{2}\left(\frac{z_i - e_{k,i}^\theta}{\sigma_{k,i}^\theta}\right)^2, T\right\}. \tag{5}$$

where $T = \exp^{-3}$ is a lower threshold. This term is usefull to avoid local differences between an examplar and the input image (e.g. in the hair cut) to conduct to a very low likelihood even when the majority of the remaining component features are in good agreement.

3 Joint Tracking and Head Pose Estimation

Head tracking and pose estimation are performed in a probabilistic framework.

3.1 Mixed-State Particle Filter.

Particle filtering (PF) implements a recursive Bayesian filter by Monte-Carlo simulations. Let $X_{0:t} = \{X_l, l = 0, \ldots, 1\}$ (resp. $z_{1:t} = \{z_l, l = 1, \ldots, t\}$) represents the sequence of states (resp. of observations) up to time t. Furthermore, let $\{X_{0:t}^i, w_t^i\}_{i=1}^{N_s}$ denote a set of weighted samples that characterizes the posterior probability density function (pdf) p$(X_{0:t}|z_{0:t})$, where $\{X_{0:t}^i, i = 1, \ldots, N_s\}$ is a set of support points with associated weights w_t^i. The samples and weights can be chosen using the Sequential Importance Sampling (SIS) principle. 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 $\mathrm{p}(X_{0:t})$ is employed as proposal, we obtain the following recursive update equation [2] for the weight:

$$w_t^i \propto w_{t-1}^i \ \mathrm{p}(z_t|X_t^i) \tag{6}$$

To avoid sampling degeneracy an additional resampling step is necessary [2]. The standard PF is given by :

1. Initialisation : $\forall i_{i \in 1:N_s}$, sample $X_0^i \sim \mathrm{p}(X_0)$; set $t = 1$
2. IS step: $\forall i$ sample $\tilde{X}_t^i \sim \mathrm{p}(X_t^i|X_{t-1}^i)$; evaluate $\tilde{w}_t^i$ using (6).
3. Selection: Resample N_s particles $\{X_t^i, w_t^i = \frac{1}{N_s}\}$ from the sample set $\{\tilde{X}_t^i, \tilde{w}_t^i\}$; set $t = t + 1$; go to step 2.

In the mixed state particle filter approach of [9], the state $X = (k, x)$ is the conjunction of a discrete variable k labeling a discrete set of objects models e_k, called exemplars and a continuous variable x specifying the spatial configuration of the object (e.g. position, the size, image rotation). 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 state X is a mixed variable $X = (k, x)$. The discrete variable $k = (\theta, l)$ labels an element of the set of head pose models $\{e_l^\theta, \ \theta, \ l = 1, .., K\}$ built in the previous Section. The continuous variable $x = (t_x, t_y, s_x, s_y)$ is a vector parameterizing the transform $\mathcal{T}_x$ defined by:

$$\mathcal{T}_x u = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} u + \begin{pmatrix} t_x \\ t_y \end{pmatrix}. \tag{7}$$

which characterizes the object configuration, where (t_x, t_y) specifies the translation of the object in the image plane, and (s_x, s_y) the scale of the width and the height of the object according to a reference size.

3.3 Dynamics

The process density on the state sequence $X_t = (k_t, x_t)$ is modeled as a second order autoregressive process $P(X_t|X_{t-1}, X_{t-2})$. We assume that the two components of the states, k_t and x_t, are independent. Also it is assumed that a head pose at a given time t, k_t, depends only on the head pose at the previous time k_{t-1}. Then the equation of the process density is:

$$P(X_t|X_{t-1}, X_{t-2}) = p(k_t|k_{t-1})p(x_t|x_{t-1}, x_{t-2}) \tag{8}$$

The dynamic of the continuous variable x is modeled as a classical second order auto regressive dynamical mode. The dynamic of the discrete variable k, defined by the transition process $p(k_t|k_{t-1}) = p(\theta_t, l_t|\theta_{t-1}, l_{t-1})$:

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

NEP	State of The Art [1]	Gaussian	Sat. Gaussian
1	90%	90%	94%
2	Not Relevant	87.5%	94.8%

Table 1: Recognition rate table for a given number of exemplar per head pose (NEP)

$p(\theta_t|\theta_{t-1})$ is based on the distance between the two head poses. $p(l_t|\theta_t, l_{t-1}, \theta_{t-1})$ is a probability table learned using the training set of faces. More precisely, for different head poses, the examplars are more related when the same persons were used to build them. When $\theta \neq \theta'$ $p(l|\theta, l', \theta')$ is taken proportional to the number of persons who belong to the class of e_l^θ and who are also in the class of $e_{l'}^{\theta'}$. When $\theta = \theta'$, $p(l|\theta, l', \theta')$ is large for $l = l'$ and small otherwise.

3.4 Observation model

Finally, let us define the object likelihood. For each state $X = (k, x)$ the observations are obtained by first extracting an image patch from the image according to $\mathcal{C}(x) = \{\mathcal{T}_x u,\ u \in \mathcal{C}\}$, and then filtering this image patch at the points specified by the grid G with the four filters defined in the previous Section, and concatenating the filtered values in a feature vector $z(x)$. The likelihood $p(z|X) = p(z|k, x)$ is finally modeled by $p(z|k, x) = p_T(z(x)|k)$, p_T referring to Equation 5.

The head pose is then estimated a each time as the mode of the head pose distribution after marginalization over the spatial configuration :

$$\theta_t^\star = \arg\max_\theta \sum_{i/\theta_t^i=\theta} w_t^i \tag{10}$$

4 Results

4.1 Head Pose Estimation Results

To test the efficiency of the pose modeling we used the 68 persons of PIE database and their head pose. For the first experiments we use the same setup than [1]. The 34 first persons were selected and their head poses used to train the head pose models. The half remaining were used to test the models. Table 1 shows the recognition rates when the number of examplars per head pose are 1 and 2. This table shows that smoothing the likelihood is indeed very useful, helping in reducing the effect of outlier feature components. Besides, Table 2 displays the confusion matrix of the recognition. It shows that estimation errors occur in general between close head poses. These errors are still acceptable, there is not a total mismatch of different profile views.

To further study the effect of the number of examplars, we included in the database 72 persons of the FERET database [6], leading to a total of 140 persons. Then, 70 persons were

	90	67.5	45	22.5	0	-22.5	-45	-67.5	-90
90	100	0	0	0	0	0	0	0	0
67.5	14.7	85.3	0	0	0	0	0	0	0
45	0	0	100	0	0	0	0	0	0
22.5	0	0	5.9	94.1	0	0	0	0	0
0	0	0	0	0	100	0	0	0	0
-22.5	0	0	0	0	5.9	94.1	0	0	0
-45	0	0	0	0	0	5.9	94.1	0	0
-67.5	0	0	0	0	0	0	3.1	91	5.9
-90	0	0	0	0	0	0	0	0	100

Table 2: Confusion matrix for NEP=2 and saturated Gaussian likelihood

	Gaussian	Sat. Gaussian
NEP	3	4
Av. of R.R.	67.2%	70.5%
St.dev. of RR	2.4	2.3

Table 3: Best recognition rates for PIE+FERET

ramdomly selected and their head pose used to train the models, and the half remaining used to test the models. We ran this set up 100 times and computed the average and standard deviation of recognition rates for NEP=1,2,3,4. Table 3 gives the best results that were achieved with NEP=3 for the Gaussian likelihood and NEP=4 for the Saturated Gaussian likelihood. These results show that more exemplars improve the recognition and that the saturated Gaussian likelihood is still doing better than the Gaussian likelihood (this indeed true for all NEP).

4.2 Tracking Results

The tracking algorithm described previously was tested on several video sequences. None of the tracked persons belonged to the training database. The positions of the camera for the video sequences used to test the algorithm were different than those used in the training database. Also the illumination condition were very different in the training database and in our test video sequences. Despite this mismatch between training and test sets, the tracking was correctly done and the estimation of the head pose was visually very satisfying. Fig. 2 shows tracking results of a typical sequence. Videos of more tracking results are available on the web.

We conducted experiments to compare our method to the traditional sequential head tracking *then* pose estimation approach. We used a color-based state-of-the-art particle filter tracker described in [5], which provides at each time a patch image corresponding to the head. The patch image is processed as described in Section 2 to extract the features, then

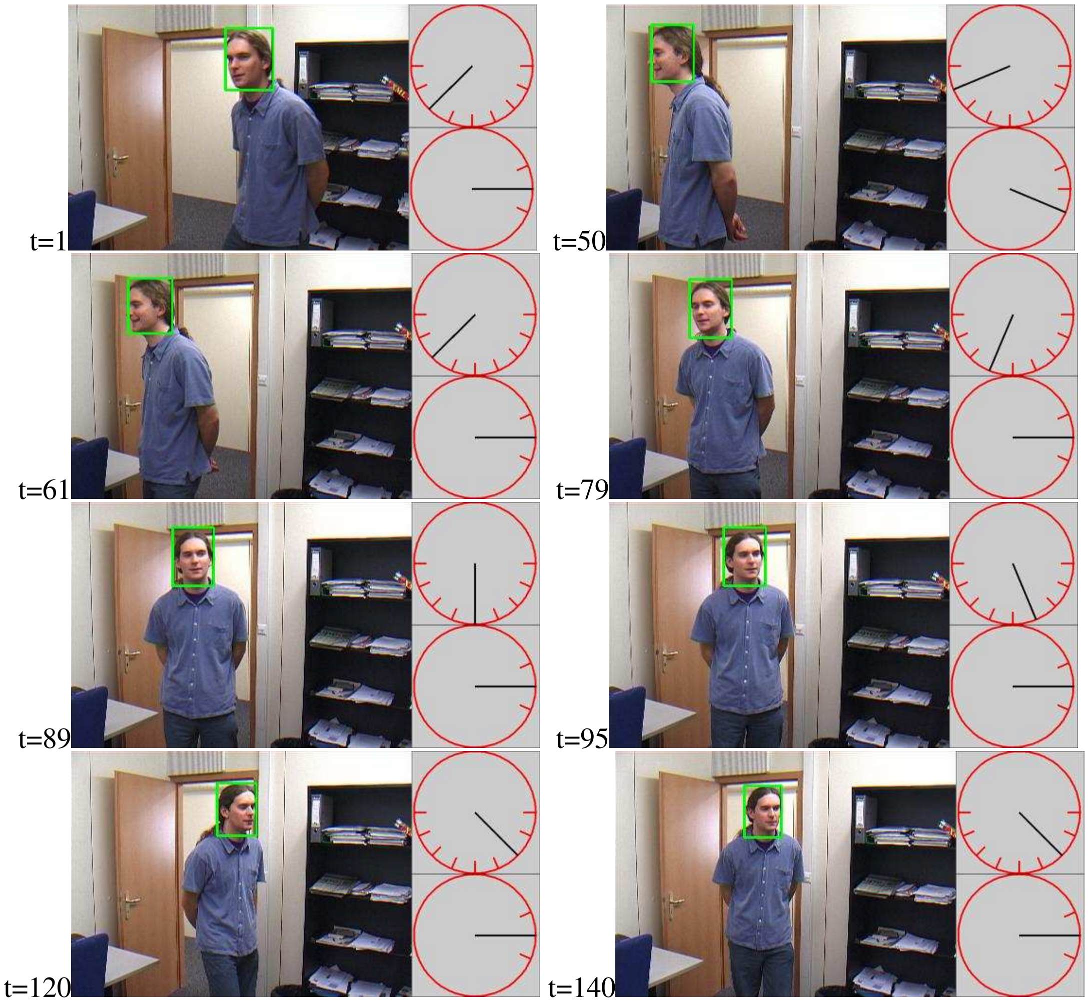

Figure 2: Tracking and head pose estimation results. First clock: pan angle; second clock: tilt angle.

compared to the exemplars using Equation 5 for pose evaluation. For the sequence of Figure 2 we generated head orientation ground truth by manually extracting a tight bounding box around the head and applying the pose estimation method. At each time t, the surface of the ground truth box is denoted $GS(t)$. We ran the two trackers that output at each time a box containing the head with surface $TS(t)$ and an estimated head pose. If at each time $JS(t)$ is the joint surface between the ground truth and the tracker, we choose to measure the tracking error by $e(t) = \frac{1}{2}\left(\frac{GS(t)-JS(t)}{GS(t)} + \frac{TS(t)-JS(t)}{TS(t)}\right)$. This error is 0 when tracking is perfect, and 1 when it totally fails. Figure 3 shows tracking errors for the two methods and the estimated head orientations. The results in Fig. 3, left, shows that our method leads to smaller tracking errors in average. The color tracker, on the other side, can be confused by similar color in the background. This results in an over-estimate of the head patch size (cf. the two error peaks at the end of the sequence), which in turn results in pose estimation failures (Fig. 3, right),

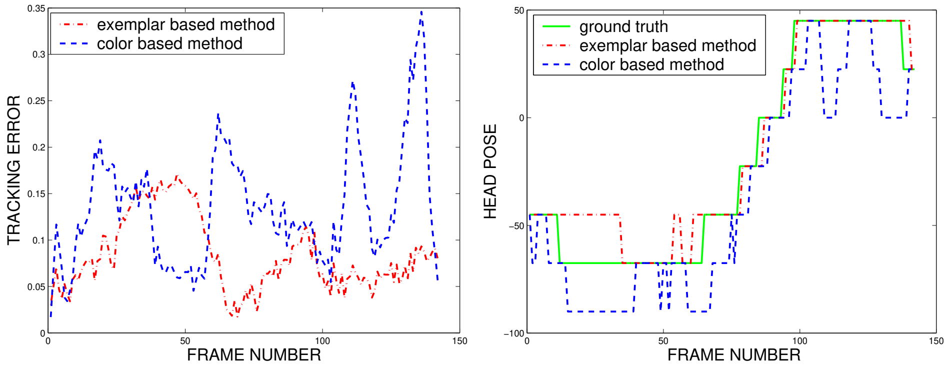

Figure 3: Left: spatial configuration errors. Right: Pan head orientation estimation.

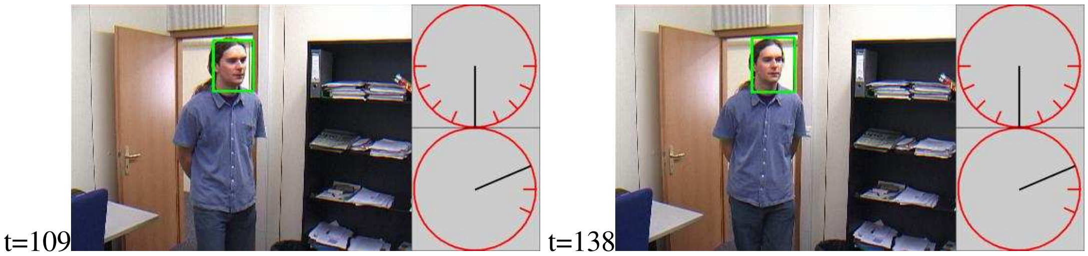

Figure 4: Sample of head pose estimation failure due to bad head location.

as illustrated by Fig. 4.

5 Conclusion

We described in this paper a joint head tracking *and* pose estimation algorithm. The novelty of the approach lie in the coupling of the tracking and head pose estimation processes. This coupling is handled in a probabilistic framework within a mixed state particle filter framework. By implicitly allowing to test multiple head configurations, it reduces the sensitivity of the pose estimation process on the tracking accuracy, a drawback of methods that perform head tracking *then* pose estimation in a sequential manner, and results in more stable and accurate pose estimates.

References

[1] L. Brown and Y. Tian. A study of coarse head pose estimation. *IEEE Workshop on Motion and Video Computing*, pages 125–130, Dec. 2002.

[2] A. Doucet. On sequential monte carlo method for bayesian filtering. Technical report, University of Cambridge, 1998.

[3] B. Kruger, S. Bruns, and G. Sommer. Efficient head pose estimation with gabor wavelet. *Proc. of 11th British Machine Vision Conference*, pages 11–14, Sept. 2000.

[4] S. Niyogi and W. Freeman. Example-based head tracking. *Proc. Int. Conf. on Auto. Face and Gesture Rec.*, Oct. 1996.

[5] P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color based probabilistic tracking. *European Conference on Computer Vision*, pages 661–675, 2002.

[6] P. Phillips, P. R. H. Moon, and S. Rizvi. The feret evaluation methodology for face recognition algorithms. *IEEE Trans. on Pat. Anal. and Machine Intelligence*, 22(10), Oct. 2000.

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

[8] T. Sim and S. Baker. The cmu pose, illumination, and expression database. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 50, Oct. 2003.

[9] K. Toyama and A. Blake. Probabilistic tracking in metric space. *Proc. 7th Int. Conf. on Computer Vision*, Dec. 2001.

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

[11] L. Zhao, G. Pingai, and I. Carlbom. Real-time head orientation estimation using neural networks. *Proc. Int. Conf. on Image Processing*, Sept. 2002.