EVALUATION OF MULTIPLE CUE HEAD POSE ESTIMATION ALGORITHMS IN NATURAL ENVIRONEMENTS

Silèye O. Ba and Jean-Marc Odobez

IDIAP Research Institute

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 is 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 behaviour 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 behaviour 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 environements. In this paper, we introduce a video database with 3D head pose ground-truth which is publicly available at http://mmm.idiap.ch/HeadPoseDatabase/. 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, 4, 5]. 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 [6, 7] 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 [8]. 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. 3), 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 [9] with the accuracy of textured-based head modeling.

The paper is organized as follows. Section 2 describe 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 parameterizations 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 [3], the rotation axes are rigidly attached to the head. In the PIE representation [10], the rotation axes are those of the camera frame. The Pointing representation leads to more direct interpretable values. However, the PIE representation has a computational advantage: the roll angle corresponds to in-plane rotations. Thus, only poses with varying pan and tilt values need to be modeled, as the head roll can be estimated by applying in-plane rotation to images. Thus, 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 authors want to thank the Swiss National Science Foundation for supporting this work through the National Centre of Competence in Research on "Interactive Multimodal Information Management (IM2)

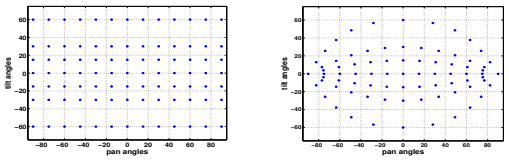

Fig. 1. Left: pan-tilt space discretization in the Pointing representation. Right: same discretization in the PIE representation.

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: 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 and resizing it to a reference size of 64×64. Then, patches are filtered by each of the above filters, and the filter outputs at subsampled locations inside a head mask are concatenated into a single feature vector of dimension 736. 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. We chosed $K = 2$ because experiments we conducted (see [1]) showed that with still images head pose recognition rates for K=2 were better than for K=1. Together with the head pose models, 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 of learned from the training head image eccentricities.

The texture likelihood with respect to an exemplars k of the head pose θ of an input image characterized by its extracted features z^{text} is given by:

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

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

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 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. Additionally we model the distribution of skin pixel values with a Gaussian distribution [11]. Skin colors are modelled in the normalized RG space, and 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^{col}_{t}. This skin mask is then compared against the model M^{θ}_{k}, and we defined the likelihood as:

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

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 $\mathrm{p}(X_{0:t}|z_{0:t})$ 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 [8]. 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 [8] for the weight $w^i_t \propto w^i_{t-1}\ \mathrm{p}(z_t|X^i_t)$. To avoid sampling degeneracy an additional resampling step is necessary [8]. The standard PF is given by :

1. Initialization : $\forall i$, sample $X^i_0 \sim \mathrm{p}(X_0)$; set $t = 1$
2. IS step: $\forall i$ sample $\tilde{X}^i_t \sim \mathrm{p}(X^i_t|X^i_{t-1})$; evaluate $\tilde{w}^i_t$.
3. Selection: Resample N_s particles $\{X^i_t, w^i_t = \frac{1}{N_s}\}$ from the set $\{\tilde{X}^i_t, \tilde{w}^i_t\}$; 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 Model

The mixed state particle filter approach [12], allows to represent jointly in the same state variable discrete variables and continuous variables. In our specific case the state $X = (S, \gamma, l)$ is the conjunction of a discrete index $l = (\theta, k)$ which labels an element of the set of head pose models e^{θ}_{k}, 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:

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

which characterizes the image object configuration. γ specifies the in-plane rotation of the object, (x, y) specifies the translation position) of the object in the image plane, and (s^x, s^y) denote the width and height scales of the object according to a reference size. 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 indices indeed correspond to real Euler angles which can be combined.

3.3. Dynamical Model

The process density on the state sequence is modeled as a second order process $P(X_t|X_{t-1}, X_{t-2})$. We assume that the three components of the states are conditionnaly 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}. The equation of the process density is:

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

The dynamics 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 dynamics of the discrete variable l_t is defined by the transition process $p(l_t|l_{t-1}) = p(\theta_t, k_t|\theta_{t-1}, k_{t-1})$:

$$p(\theta_t, k_t|\theta_{t-1}, k_{t-1}) = p(k_t|\theta_t, k_{t-1}, \theta_{t-1})p(\theta_t|\theta_{t-1}). \quad (5)$$

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 examplars, 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.

3.4. Observation model

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

$$p(z \mid X = (S, \gamma, l)) = p_T(z^{text}(S, \gamma)|l)p_c(z^{col}(S, \gamma)|l), \quad (6)$$

where the observations z are composed of texture and color observations (z^{text}, z^{col}), 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 2.
The computation of the observations is done as follows. First the image patch associated with the image spatial configuration of the state space, (x, 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 the flock of bird, a magnetic field 3D location and orientation tracker. The device was well camouflaged behind people's ear. After calibration of the sensor with camera frame, we can output at each time frame the person's head pose. With this system, we recorded two databases, one in an office environment (not used here) and one in a meeting environment. In the meeting environment, 8 meetings were recorded each lasting approximatively 8 minutes. In each meeting, two out of four persons had their head poses continuously annotated. The scenario of the meeting was to discuss

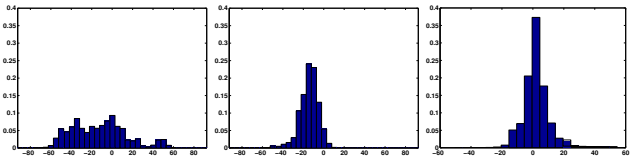

Fig. 2. Histograms of pan tilt and roll values of the evaluation data

	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 1. Mean, standard deviation and median of head pointing vector errors over evaluation data

statements displayed on the projection screen. There were restrictions neither on head motions, nor on head poses. This results in a video database of 16 different people. For our experiments we use half of the database as train set to train pose dynamic model and the half remaining as test set to evaluate the tracking algorithms.

The tracking evaluation protocol is the following. In each one of the 8 meetings of the test set, we selected 1 minute of recording (1500 video frames) for evaluation data. We decided to use only one minute to save machine computation time, as we use a quite slow matlab implementation. Figure 2 shows the distribution of the pan, tilt and roll values on the evaluation data. Because of the scenario used to record data, people often have negative pan values corresponding to looking at the projection screen. But the pan values range from -60 to 60 degree. Tilt values range from -60 to 15 degrees and roll value from -30 to 30 degrees. To evaluate tracking performances, we used four error measures. The three first measures are the errors in pan, tilt and roll angle expressed in the Pointing representation, i.e. the absolute difference between the pan, tilt and roll of the ground truth (GT) and the tracker estimation. Also, as a head pose defines a vector in the 3D space, the vector indicating where the head is pointing at, the angle between the 3D pointing vectors defined by the head pose GT and the pose estimated by the tracker can be used as pose estimation error measure. This vector depends only on the head pan and tilt values in the Pointing representation.

4.2. Experiments 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) [13] and an histogram, correlation and shape tracker (M2) [13]. 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 while the second (M4) exploits both texture and color likelihood models.

We ran the four trackers on the test data . Table 1 reports the head pointing vector errors of the four methods. The mean and the median errors are smaller for methods M3 and M4. As illustrated in Figure 3, 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 and is sometimes distracted by the heavy cluttered background. The color cue is complementary to the texture cue because it helps 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 2 provides the pan, tilt and roll error measures. As for

	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 2. pan, tilt and roll errors statistics over evaluation data (Pointing representation)

	abs(pan of GT)$\leq$ 45			45 $<$ abs(pan of GT)$\leq$ 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	$\times$	33.6	16.3	$\times$

Table 3. mean of pan, tilt and roll errors for abs(pan of GT)$\leq$ 45 and 45 $<$ abs(pan of GT)$\leq$ 90 (Pointing representation)

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 2 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 give in Table 3 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. For comparison purposes, this table displays also the results reported in [4] (Wu 01) for a similar experimental setup. 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 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. Also for near frontal head pose, head appearances are very similar for different tilt values. 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 4. 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).

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. Although our algorithm 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.
Our data are part of a larger database which comprises more than two hours of annotated data. This database is publicly available at http://mmm.idiap.ch/HeadPoseDatabase, 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.

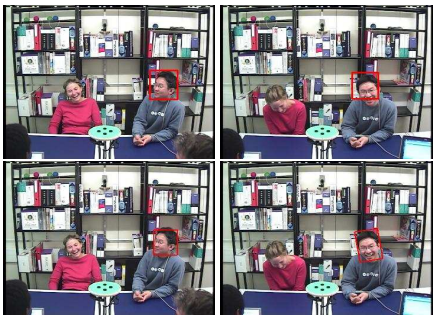

Fig. 3. Head localization results for M2 (top row) and M4 (bottom row); left column: frame 571; right column: frame 661

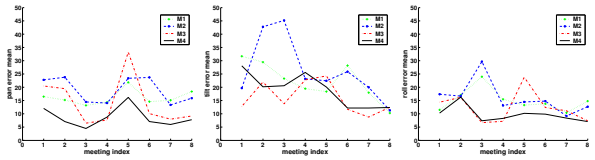

Fig. 4. Mean of pan, tilt and roll pose estimation errors for individual meeting evaluation data

6. REFERENCES

[1] S.O. Ba. and J.M. Odobez, "A probabilistic framework for joint head tracking and pose estimation," *ICPR*, Aug 2004.

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

[3] "Pointing'04 icpr workshop: Head pose image database," http://www-prima.inrialpes.fr/Pointing04/data-face.html.

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

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

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

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

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

[9] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color based probabilistic tracking," *ECCV*, 2002.

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

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

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

[13] J.M. Odobez, S.O. Ba., and D Gatica-Perez, "Embedding motion likelihood for tracking with particle filters," *BMVC*, Sept 2003.