LEARNING LARGE MARGIN LIKELIHOODS FOR REALTIME HEAD POSE TRACKING

Elisa Ricci, Jean-Marc Odobez

Idiap Research Institute,
Centre du Parc, Rue Marconi 19, CH-1920, Martigny, Switzerland
{*elisa.ricci,jean-marc.odobez*}@*idiap.ch*

ABSTRACT

We consider the problem of head tracking and pose estimation in realtime from low resolution images. Tracking and pose recognition are treated as two coupled problems in a probabilistic framework: a template-based algorithm with multiple pose-specific reference models is used to determine jointly the position and the scale of the target and its head pose. Target representation is based on Histograms of Oriented Gradients (HOG): descriptors which are at the same time robust under varying illumination, fast to compute and discriminative with respect to pose. To improve pose recognition accuracy, we define the likelihood as a parameterized function and we propose to learn it from training data with a new discriminative approach based on the large-margin paradigm. The performance of the learning algorithm and the tracking are evaluated on public images and video databases.

Index Terms— realtime tracking, particle filter, head pose estimation, discriminative learning.

1. INTRODUCTION

The problem of head pose estimation has attracted the attention of several researchers mainly due to the large amount of applications such as human computer interaction or visual focus of attention recognition [1]. Using a monocular camera to robustly track a head and estimate its orientation is a challenging task especially due to the loss of depth information. Moreover very often only faces at low resolution are available and 3D modeling techniques cannot be used. An additional difficulty arises if, as required by several applications, tracking and head pose estimation must be performed in realtime.

Head tracking and pose estimation can be realized with the cascade of a system which extracts the location of the face and a classifier which determines the pose of the localized face. A better alternative, proposed in [2], consists in modeling tracking and pose recognition as two paired tasks in a single framework. In this way the tracking robustness is improved by defining a pose-specific observation model while the pose estimation accuracy is increased due to a better localization of the target.

Following this principle, in this paper a new algorithm for joint head tracking and pose estimation is proposed. As in [2], we focus our attention on the challenging task of pose estimation for faces at low resolution but, in contrast to [2], our system runs in realtime. We model head pose recognition as a discrete state estimation problem. We learn offline multiple pose-specific templates and we use them in a mixed state particle filter (MSPF) which computes the position and the scale of the head and simultaneously determines its orientation. Realtime is achieved mainly thanks to an effective target representation based on multilevel HOG [3]: features which allows good pose classification and are extremely fast to compute due to integral histograms [4]. To our knowledge this is the first time that HOGs are employed for head pose recognition. A similar descriptor has been proposed in [5] but in this case no multilevel representation neither integral images are used.

A main novelty of the paper concerns the definition of the likelihood, i.e. the function which measures the compatibility between the current observation and the reference models of a specific pose. We propose to express it as a function of a set of parameters and to learn them offline in a way such that the similarity between two images is imposed to be high if the poses are close and large otherwise. To this aim we introduce a new discriminative algorithm which improves significantly pose estimation accuracy.

2. HEAD POSE REPRESENTATION

We consider the head orientation θ as described by 3 angles, pan and tilt (to represent out-of-plane rotations, i.e. respectively the horizontal and the vertical inclination of the face) and roll (for in-plane rotations). We discretize the space of all possible orientations into $\Theta = 273$ poses: 13 possible values for pan, 7 for tilt and 3 for roll. Pan and tilt angles varies in the range $\pm 90^o$, while roll in $\pm 15^o$.

For the purpose of tracking we consider multiple reference models for each pose and we denote by $\mathcal{R}_\theta^k$ the k-th reference of pose θ. We use 91 poses of the PRIMA-POINTING head pose database [6] to build the set of templates corre-

This work has been supported by the IST European projects FP6 AMIDA (Augmented Multi-Party Interaction with Distance Access) and FP7 TA2 (Together Anywhere, Together Anytime).

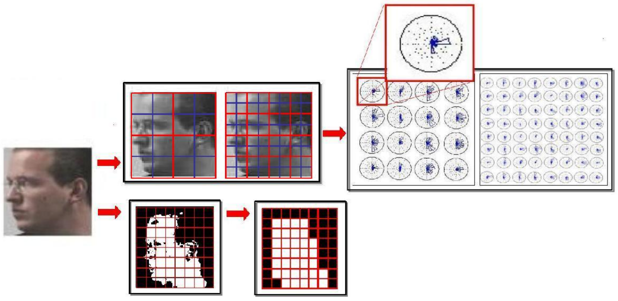

Fig. 1. Original image, multilevel HOGs (cells in blue, blocks in red) and skin mask

sponding to roll equal to 0^o. By rotating them we also obtain the reference images for roll $\pm 15^o$. For each pose we use 15 images of different individuals. In this way we can alleviate the problems due to the large variations of models corresponding to the same pose. These variations can be due to image cropping or alignment or to the appearances of people. We rescale all the images to 64×64 pixels and we compute the feature vectors $\mathbf{r}_\theta^k$ associated to $\mathcal{R}_\theta^k$. We adopt two types of features in order to discriminate between different head orientations: texture features and color features (Fig. 1).

Texture features. We use multi level HOG descriptors as texture features: we partition the image into 2×2 (first level) and 4×4 (second level) non overlapping blocks of 2×2 cells and compute the histograms of gradient orientation on each cell. More specifically, we first convert the color image to grayscale. Then the horizontal and vertical image gradient are computed by a 1-D centered mask and used to calculate the magnitude and the orientation of the gradient. As suggested by [3] we employ unsigned orientation of the image gradient. For each cell we construct a 1D histogram quantizing the gradient orientation of all the pixels into 8 bins weighting the contribute of each pixel by the gradient magnitude. This can be done by integral histograms. Histograms are then normalized locally i.e. considering all the cells in the same block. The final HOG descriptor is obtained by the concatenation of the small histograms. Fig. 1 give a example of an image and its HOG representation.

Color features. Color features are also extracted. Skin colors are modeled in the normalized RG space and a gaussian color model is learned from a training set of skin patches. This model is employed to classify pixels of an image as *skin/not skin*. The resulting binary image is divided into 8×8 cells non overlapped. Finally a binary mask (Fig. 1) is constructed: for each cell if the majority of pixels corresponds to *skin* the output associated to the cell is 1, otherwise it is 0. This computation is done efficiently with integral images.

3. TRACKING AND POSE ESTIMATION

Let us denote by $\mathbf{s}_t$ the hidden state which represent the object configuration and by $\mathbf{o}_t$ the associated observation extracted from the image at time t. In bayesian tracking the sequence of hidden parameters $\mathbf{s}_{1:t}$ is estimated based on the observed data $\mathbf{o}_{1:t}$. All bayesian estimates of $\mathbf{s}_t$ follow from the posterior distribution $p(\mathbf{s}_t|\mathbf{o}_{1:t})$. In the most common situations of non-linearity and multimodality a practical approach is to approximate $p(\mathbf{s}_t|\mathbf{o}_{1:t})$ is by a set of samples (the particles) each one associated with a weight which indicates its "quality". This approximation is recursively updated using a particle filer algorithm e.g. `CONDENSATION`. In `CONDENSATION` two phases can be distinguished: prediction and update. During the prediction each particle is modified according to a state model adding some random noise. In the update phase, each particles weight is updated based on the new observed data. A resampling procedure provides the elimination of particles with small weights and the replication of those with larger weights. In the following, we describe the main features of our particle filter.

State space. We choose a rectangular box as head tracking region described by the vector $\mathbf{s} = (t_x, t_y, s_x, e_y, \theta, k)$. In practice we consider a MSPF [7] since $\mathbf{s}$ contains both continuous variables ($x = (t_x, t_y, s_x, e_y)$ to indicate head location and size) and discrete variables (θ, k representing respectively the pose and the k-th reference model of pose θ).

Dynamical model. We assume continues states components independent from discrete ones. Standard autoregessive models are chosen to describe the dynamics of the translation components (t_x, t_y), the scale s_x and excentricity e_y. For the discrete variables θ and k, we define two probability tables ($p(\theta_t|\theta_{t-1})$ and $p(k_t|k_{t-1}, \theta_t, \theta_{t-1})$) and we learn them from opportune training sequences: $p(\theta_t|\theta_{t-1})$ is based on the distance between adjacent poses and $p(k_t|k_{t-1}, \theta_t, \theta_{t-1})$ on the difference between images in the training set (for the same pose images of similar appearances are preferred, for nearby poses images of the same individual are considered).

Observation model. An observation $\mathbf{o} = (\mathbf{o}^{tex}, \mathbf{o}^{col})$ is composed by texture and skin color features computed on each image. Under the assumption that the features being used are independent, the overall likelihood $p(\mathbf{o}_t|\mathbf{s}_t)$ is the product of the likelihoods of the separate cues:

$$p(\mathbf{o}_t|\mathbf{s}_t) = p^{col}(\mathbf{o}_t^{col}|\mathbf{s}_t)p^{tex}(\mathbf{o}_t^{tex}|\mathbf{s}_t)$$

The texture likelihood $p^{tex}(\mathbf{o}_t^{tex}|\mathbf{s}_t)$ is obtained by:

$$p^{tex}(\mathbf{o}_t^{tex}|\mathbf{s}_t) = e^{-\lambda_T D_{\boldsymbol{W}}(\mathbf{o}_t^{tex}, \mathbf{r}_{\theta_t}^{k_t tex})}$$

where λ_T is a user define constant and the distance $D_{\boldsymbol{W}}(\mathbf{o}, \mathbf{r}_\theta^k)$ is a linear function of some parameter vector $\boldsymbol{w}_\theta^k \in R^M$ e.g. $D_{\boldsymbol{W}}(\mathbf{o}, \mathbf{r}_\theta^k) = {\boldsymbol{w}_\theta^k}^T \boldsymbol{d}_\theta^k(\mathbf{o})$. The vector $\boldsymbol{d}_\theta^k(\mathbf{o})$ contains the concatenation of elementary distances between features of an observation $\mathbf{o}$ and the corresponding features in the reference model $\mathbf{r}_\theta^k$. Arbitrary distances can be used as elementary distances: in our experiments we used χ_2 distances between histograms of corresponding HOG cells. In the following

Section we describe the algorithm we used to learn the set of weighted vectors $\boldsymbol{w}_\theta^k$. It is worth noting that in general a parameterized likelihood function can be expensive to evaluate. However since we assume the set of all possible poses to be known in advance, the reference models $\mathbf{r}_\theta^k$ and the weight vectors $\boldsymbol{w}_\theta^k$ are precomputed off-line. Moreover the features vectors entirely rely on integral images. Therefore the run-time complexity is drastically reduced.

The color likelihood $p^{col}(\mathbf{o}_t^{col}|\mathbf{s}_t)$ is obtained by computing the L_1 distance between the features associated to the k_t reference model of pose θ_t ($\mathbf{r}_{\theta_t}^{k_t col}$) and the features computed on the current particle ($\mathbf{o}^{col}$), i.e.:

$$p^{col}(\mathbf{o}_t^{col}|\mathbf{s}_t) = e^{-\lambda_C D_{L_1}(\mathbf{o}^{col}, \mathbf{r}_{\theta_t}^{k_t col})}$$

where λ_C is an appropriate constant. Note that also for color we could have employed a weighted distance function. However the purpose of learning the weights is to increase pose recognition accuracy and pose information is mainly provided by texture features. On the other hand color features only contain a rough information about pose but are very useful to improve tracking localization.

Filter output. The estimate of the variables of interest (in this case the mean of the distribution) is obtained by averaging over the set of particles. Note that for poses (which are represented by discrete variables) this is still possible since in practice they correspond to real-value angles.

4. LEARNING THE LIKELIHOOD

In this Section we denote by $\mathbf{o}_i$ the set of texture features associate to an image and by y_i a label that indicates the head orientation. We assume that we have a training set $\mathcal{T} = \{(\mathbf{o}_1, y_1), (\mathbf{o}_2, y_2), \dots, (\mathbf{o}_\ell, y_\ell)\}$ of pairs of images $\mathbf{o}_i$ with associated poses y_i. We select a set of training points as reference models (i.e. $\mathbf{r}_{y_i}^k = \mathbf{o}_i$). This set can correspond to the entire training set or be a subset of representative datapoints selected by a-priori knowledge or by clustering techniques as K-means. We propose to learn the distance functions $D_{\boldsymbol{W}}$ in order to impose that exemplars $\mathbf{o}_i$ associated to pose y_i should be closer to all reference models of the same pose $\mathbf{r}_{y_i}^k$ and separated at least by a margin of 1 from reference models $\mathbf{r}_{\theta'}^{k'}$ of different pose ($\theta' \neq y_i$). In formulas:

$$\min_{y_i \neq \theta', k'} \boldsymbol{w}_{\theta'}^{k'T} \boldsymbol{d}_{\theta'}^{k'}(\mathbf{o}_i) - \max_k \boldsymbol{w}_{y_i}^{k\;T} \boldsymbol{d}_{y_i}^k(\mathbf{o}_i) \geq 1 \;\; \forall \mathbf{o}_i$$

In other words we impose that for each image $\mathbf{o}_i$ the difference between the minimal distance from references of different poses and the maximal distance from references of the same pose should be larger than one. We define $\boldsymbol{W} = [\boldsymbol{w}_1^1 \dots \boldsymbol{w}_1^{K_1} \dots \boldsymbol{w}_\Theta^1 \dots \;\boldsymbol{w}_\Theta^{K_\Theta}]^T$, and $\boldsymbol{\delta}(\mathbf{o}_i, \theta, k) = [0 \;\dots \boldsymbol{d}_\theta^k(\mathbf{o}_i) \;\dots 0]^T$ i.e. all the entries are 0 except the part corresponding to pose θ and reference k where they are set to $\boldsymbol{d}_\theta^k(\mathbf{o}_i)$. It is easy to verify that all the constraints above can be rewritten in the form:

$$\min_{y_i \neq \theta', k'} \boldsymbol{W}^T \boldsymbol{\delta}(\mathbf{o}_i, \theta', k') - \max_k \boldsymbol{W}^T \boldsymbol{\delta}(\mathbf{o}_i, y_i, k) \geq 1 \;\; \forall \mathbf{o}_i$$

Table 1. Average error in degrees with CLEAR06 setup. Numbers in parenthesis correspond to all weights set to 1.

	THIS PAPER	BA	VOIT	TU	GOURIER
PAN	9.1 (13.7)	11	12.3	14.1	10.3
TILT	10.5 (14.2)	11.5	12.7	14.9	15.9

The task we are interested in is to find the weight vector $\boldsymbol{W}$ such that all constraints are satisfied. A simple approach is to employ the large margin principle as in Support Vector method [8] and to choose the vector $\boldsymbol{W}$ with minimum norm. The resulting optimization problem is:

$$\min_{\boldsymbol{W} \geq 0, \xi_j \geq 0} \;\; \tfrac{1}{2}||\boldsymbol{W}||^2 + C \textstyle\sum_{j=1}^{\ell} \xi_j$$

$$\text{s.t.} \; \min_{y_i \neq \theta', k'} \boldsymbol{W}^T \boldsymbol{\delta}(\mathbf{o}_i, \theta', k') - \max_k \boldsymbol{W}^T \boldsymbol{\delta}(\mathbf{o}_i, y_i, k) \geq 1 - \xi_j$$

where we have introduced slack variables ξ_j to allow the problem to be solved in non-separable cases. The parameter C control the trade-off between regularization and violation of the margin. Note that the constraint $\boldsymbol{W} \geq 0$ has been introduced to impose that the learned distance function should be valid i.e. always positive. We solve this problem by an efficient iterative algorithm based on stochastic gradient descent which is a variation of the optimization strategy recently proposed in [9] and we do not describe here for lack of space.

To our knowledge, in the context of head pose recognition we are the first to suggest to learn distance functions and to employ them in the likelihood of a MSPF. Among previous works on distance learning the most similar to our algorithm is the one proposed in [10] to improve nearest neighbor (NN) classification accuracy in the context of object classification. However the method in [10] employs a constraint for each possible triplet $(\mathbf{x}_i, \mathbf{r}_{yi}^k, \mathbf{r}_{\theta'}^{k'})$ and it is solved by a dual optimization method. Therefore in large multiclass problems its computational cost becomes prohibitive (e.g. in our experiments we would have about 20 million constraints). On the contrary using our approach the number of constraints is linear in the size of the training set ℓ and a primal solver is used, therefore it scales much better.

5. RESULTS AND DISCUSSION

We first show that with our distance learning algorithm head pose estimation is greatly improved. We consider static images of 93 poses in the PRIMA-POINTING database. The experimental setup is the same one as for the CLEAR evaluation workshop 2006 (*http://isl.ira.uka.de/clear06/*). Images are split in two sets: the first series is used as training set, the second as test set. Faces in the images are cropped automatically by a skin color model and rescaled into 64×64

Table 2. Pose estimation errors (degrees) for person left (L) and right (R) in the IDIAP Head pose database.

	1L	1R	2L	2R	3L	3R	mean
pan	16.9	11.2	16.6	12.1	11.3	7.2	12.5 (15.8)
tilt	8.4	5.7	7.1	13.1	11.5	5.1	8.5 (11.3)
roll	6.9	9.6	11.7	8.4	9.8	5.1	8.5 (9.6)

pixels. Histogram equalization is performed to reduce the effect of lighting condition. In the distance learning algorithm we use $K = 15$ reference models per pose. Once the training is terminated, classification is performed with 1-NN classifier. As shown in Table 1 our method achieves better accuracy than state-of-the approaches (the numbers in the table are taken from Table II in[1] and correspond to all the methods evaluated with the same protocol). Between parenthesis the errors of a NN classifier without previous distance learning (all weights set to 1) are indicated. This demonstrates that multilevel HOGs are effective descriptors: data are clustered with respect to pose and classification performance are already good. Moreover with distance learning the accuracy is significantly improved.

Finally we show the validity of our approach for joint tracking and pose estimation. A qualitative analysis of several videos demonstrate that the system provides satisfactory results both in terms of head localization and pose estimation. Some examples of videos can be found at $www.idiap.ch/\sim odobez/icip2009.html$.

To quantify the performance of the tracker in terms of head pose estimation accuracy we use the IDIAP Head pose database (*www.idiap.ch/HeadPoseDatabase*). The same protocol and the same performance measures described in [11] are adopted for conducting experiments. Fig.2 illustrates tracking results on a typical sequence. The first row shows two cases where both tracking and pose estimation are accurate, while in the second row two examples of failure due to occlusion are depicted. Note that in this case the head orientation is wrongly estimated but the face is still localized correctly despite the cluttered background. The pose estimation errors corresponding to $K = 5$ reference models per pose are shown in Table 2. It is evident that using our large margin learning approach the estimation accuracy significantly improves with respect to the baseline (no distance learning) reported between parenthesis. This is somehow expected because the algorithm favors better discrimination between different poses. Moreover with our approach we learn several distances solving a single optimization problem, therefore the distances (and then the likelihoods) are comparable i.e. interpretable on an absolute scale. On the other hand in previous template-based approaches (e.g.[2]) multiple reference models are learned independently. This introduces normalization problems since different likelihoods can be not comparable.

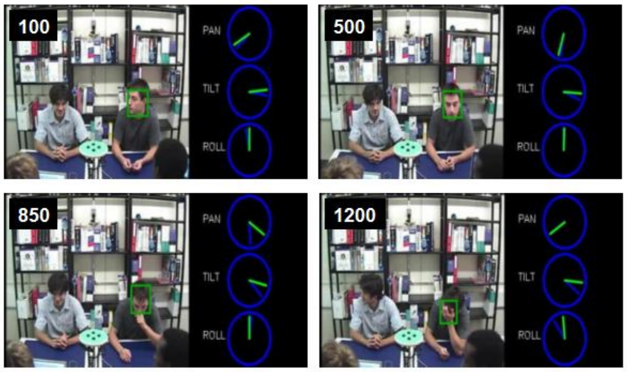

Fig. 2. Head tracking and pose estimation results

Comparing our results with those reported in [1] and in particular with the best method [11] we see that we achieve higher performance in term of tilt and roll estimation while the pan recognition is less accurate. Moreover our tracker runs close to realtime (at about 20fps) while the system in [11] is very slow (about 2fps) due to the likelihood computation (which heavily relies on particles resizing, histogram equalization and Gabor filters) and to Rao-Blackwellization. From the analysis of the output videos we observe that the major cause of pose estimation errors is probably the fact that we do not model large in-plane rotations since in these cases it is difficult to compute features with integral images. We leave this as topic of further research.

6. REFERENCES

[1] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: A survey," in *IEEE Trans on PAMI*, April 2008.

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

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[4] F. Porikli, "Integral histogram: A fast way to extract higtograms in cartesian spaces," in *CVPR*, 2005.

[5] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation," in *Proc. IEEE Conf. Intelligent Transportation Systems*, 2007, pp. 709–714.

[6] N. Gourier, D. Hall, and J. L. Crowley, "Estimating face orientation from robust detection of salient facial features," in *Int. Work. on Visual Observation of Deictic Gestures*, 2004.

[7] K. Toyama and A. Blake, "Probabilistic tracking in a metric space.," in *ICCV*, 2001, pp. 50–59.

[8] V.N. Vapnik, "The nature of statistical learning," in *Springer, 2nd edition*, 1998.

[9] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm," in *ICML*, 2007.

[10] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning globally-consistent local distance functions for shape-based image retrieval and classification," in *ICCV*, 2007.

[11] S. Ba and J.-M. Odobez, "A probabilistic head pose tracking evaluation in single and multiple camera setups," in *CLEAR Evaluation and Workshop*, 2007.