Machine Vision and Applications (2012) 23:479–490
DOI 10.1007/s00138-012-0410-z

ORIGINAL PAPER

Multi-camera head pose estimation

Rafael Muñoz-Salinas · E. Yeguas-Bolivar · A. Saffiotti · R. Medina-Carnicer

Received: 24 March 2010 / Revised: 26 April 2011 / Accepted: 19 January 2012 / Published online: 21 February 2012

Abstract Estimating people's head pose is an important problem, for which many solutions have been proposed. Most existing solutions are based on the use of a single camera and assume that the head is confined in a relatively small region of space. If we need to estimate unintrusively the head pose of persons in a large environment, however, we need to use several cameras to cover the monitored area. In this work, we propose a novel solution to the multi-camera head pose estimation problem that exploits the additional amount of information that provides multi-camera configurations. Our approach uses the probability estimates produced by multi-class support vector machines to calculate the probability distribution of the head pose. The distributions produced by the cameras are fused, resulting in a more precise estimate than the one provided individually. We report experimental results that confirm that the fused distribution provides higher accuracy than the individual classifiers and a high robustness against errors.

Keywords Head pose · Multiple views · Support vector machines · People tracking

R. Muñoz-Salinas (✉) · E. Yeguas-Bolivar · R. Medina-Carnicer
Department of Computing and Numerical Analysis,
University of Córdoba, 14071 Córdoba, Spain
e-mail: rmsalinas@uco.es

E. Yeguas-Bolivar
e-mail: in1yeboe@uco.es

R. Medina-Carnicer
e-mail: rmedina@uco.es

A. Saffiotti
AASS Mobile Robotics Laboratory, University of Örebro,
70182 Örebro, Sweden
e-mail: asaffio@aass.oru.se

1 Introduction

Head pose estimation is an important problem with applications in several fields, such as ambient intelligence and human–computer interaction. It has been an active topic of research in the last decade or so, which has led to the development of a large number of solutions. These can be roughly divided into two categories: 2D view-based and 3D model-based approaches. Examples of the former include the use of support vector machines (SVM) [16,22], PCA [32], Kernel PCA (KPCA) [8], independent subspace analysis [33], Gabor filters [41,42] and networks [19], active appearance models (AAM) [11], shape-from-shading [9] and 2D geometric heuristics [23], amongst others. Model-based approaches can be tackled from a concise mathematical formulation [15], or if real-time analysis is required, by simplifying the problem using affine transformations [13]. Examples of this class of approaches include the use of vanishing points [39,40], different shape models such as planar [4], cylindrical [7], ellipsoidal [2,10] and deformable ones [21]. In some cases, the problem is tackled from a tracking point of view that can be solved by fusing multiple cues [31] or using infrared light to detect the pupils [17]. In others, range information is used to improve the results [24,29].

Despite the impressive advances in this field, most of the existing works employ a single camera for estimating the head pose. A single camera configuration, however, is not a feasible solution if we need to observe people's behaviour in large environments [25,27,28]. In these applications, multiple cameras are needed to cover the monitored area, and techniques able to exploit the additional amount of information are therefore preferable. The presence of multiple cameras can also be exploited to provide a greater degree of robustness, since failure of one camera can in principle be compensated using the others.

Few solutions for head pose estimation using multiple cameras have been reported in the literature. One of them is proposed by Zhang et al. [43]. The authors employ a FloatBoost detector to classify head poses into only five categories and fuse the classifications from each camera using a naive Bayesian network. This approach has a main limitation in the limited range of head poses covered.

Canton-Ferer et al. [6] propose a model-based approach. They apply a skin colour filter to all images, and then fit a ellipsoid to the skin colour patches found. The face region is then located by finding the 3D centroid of the skin colour patches. A problem with this approach is the strong reliance on the skin colour filter. If this filter fails in one of the cameras, producing either an over- or an undersegmentation, the centroid calculation is heavily affected. Another problem is that all views are required to calculate the head pose, so the method is very sensitive to occlusion. We claim that it would be preferable to produce a hypothesis from each camera that has the head in view, and then fuse the available hypotheses. In this way, the failure of one camera would not strongly affect the final performance. The approach that we present in this paper follows this road.

Brunelli and Lanz [20] tackle the problem using a Monte Carlo filter that examines colour and gradient information. Instead of using a classifier, a colour model of the body and head is created from which head pose is estimated via interpolation. The multi-view problem is then gracefully tackled in the Bayesian context by jointly considering evidences of the multiple cameras. Although the reported accuracy is low, compared to other approaches on the same data set, the main advantage of their method is the lack of training. Similarly, the work of Ba and Odobez [1] presents a head pose estimator based on a particle filter. The main weakness of their approach is that the head estimation is strongly based on the skin detection, which might run the risk of making the method environment dependent.

Probably, the most sophisticated approach is the one presented by Voit and colleagues [36–38]. In their last works, they propose a model that combines view-based approaches, tracking and a method to determine the camera reliability. Aiming at determining the head pose in a video sequence, they employ a particle filter that tracks the head. At each possible location, two neural networks (NN) (one for pan and another one for tilt) are employed to estimate the head pose. The networks are trained to output a probability distribution of the estimated angle that can be merged with the estimates of other cameras. Additionally, since the image patches might not be properly centred at the head location, an external analysis using histograms of gradient (HOG) is applied. The HOG provides a confidence value indicating whether the head is in the center of the image path. The value is employed as a confidence factor when fusing multiple views by assigning less relevance to unaligned views. Nevertheless, the work has two main limitations. First, it uses a very large input feature vector (2,048 inputs), which might easily result in over-fitting. Second, there are some known limitations in the use of neural networks, like the need to choose an appropriate topology and the use of a training method that might fall into local minima.

In this paper, we propose a new approach to the multi-view head pose estimation problem which is based on multi-class support vector machines (MSVM). The head images are preprocessed using PCA, and an MSVM classifier is trained on a discretisation of the angular space. Our contribution is twofold. First, we propose a head pose estimator that uses the probability estimates produced by an MSVM. In particular, we show how we can use the voting probability distribution (vpd) to improve the head pose estimate, rather than simply picking the most voted class. Second, we show how the above head pose estimator can be extended to deal with the multiple-camera case, by fusing information from multiple cameras to create a consensus estimate. As we show in the experiments, the fused distribution provides higher accuracy than the individual classifiers, as well as high robustness against errors.

The rest of this paper is organised as follows. Section 2 introduces the principles of SVM. Section 3 formulates the problem and Sect. 4 explains the proposed solution. Section 5 presents the experiments and the results. The data set employed for this work is publicly available for evaluation. Finally, Sect. 6 draws some conclusions.

2 Support vector machines

Support vector machines are maximum margin classifiers that appear as a consequence of the research on *structural risk minimisation* [35]. They map input vectors to a higher dimensional space where a maximal separating hyperplane is constructed. Suppose that we are given with the training data

$$(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \{\pm 1\},$$

where $x_i \in \mathcal{R}^d$ represent the patterns and y_i the labels.

Let us define a kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}$ that measures the similarity of two patterns and a mapping function $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ that translates the patterns to a higher dimensional feature space. Then, the kernel is defined as $k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j) \rangle$. The problem of finding the optimal hyperplane $\mathbf{w}$ that separates the classes is solved by the following minimisation problem:

$$\min \ \frac{1}{2}||\mathbf{w}||^2$$

subject to

$$y_i(\langle \mathbf{w}, \Phi(x_i) \rangle + b) \geq 1, \ i = 1, \ldots, m.$$

Making use of the Lagrangian dual, the optimisation problem can be transformed into:

$$L(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j),$$

subject to

$$0 \leq \alpha_i \leq C, \ i = 1, \ldots, m,$$
$$\sum_{i=1}^{m} \alpha_i y_i = 0,$$

where α_i are the Lagrangian multipliers and C is its upper bound. The optimal decision function is then expressed as:

$$f(x) = sgn\left(\sum_{i=1}^{m} y_i \alpha_i k(x, x_i) + b \right).$$

2.1 Regression vector machines

SVMs can also be applied for regression problems, thus leading to the regression vector machines (RVM). Let us consider a set of data points $\{(x_1, z_1), \ldots, (x_m, z_m)\}$ such that $z \in \mathcal{R}^1$ is the desired output. Then, the support vector regression can be solved by maximising

$$\begin{aligned} &-\frac{1}{2} \sum_{i=1}^{l} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) k(x_i, x_j) \\ &-\epsilon \sum_{i=1}^{l} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{l} z_i (\alpha_i^* - \alpha_i) \end{aligned} \tag{1}$$

subject to

$$\sum_{i=1}^{l} (\alpha_i^* - \alpha_i) = 0,$$
$$0 \leq \alpha_i^*, \alpha_i \leq C,$$

providing the solution:

$$f(x) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) k(x, x_i) + b \tag{2}$$

From the point of view of the problem addressed in this work, to estimate people's head pose from camera data, RVM seems to provide a very interesting solution. However, we would also like to fuse the information coming from the different cameras to improve the robustness of the classifier. In that sense, the RVM formulation has a serious limitation in that it can only deal with one input. Thus, we explore a novel solution based on the use of MSVM, and of the probability estimates that they provide.

2.2 MSVM with probability estimates

The SVM is a two-class classifier, but many problems involve k classes. The multi-class problem can be tackled by a combination of multiple binary classifiers to create a multi-class support vector machine classifier (MSVM). A popular approach to do this is the "one-against-one" method [18] that employs $k(k-1)/2$ pairwise classifiers, each of which has training data from two different classes. When classifying, all the binary classifiers are evaluated and a voting scheme employed, i.e. the class receiving the maximum number of votes is assumed to be the correct one.

Selecting the most voted class as the correct one might exclude relevant information about the nature of the patterns to be classified. Instead, information about the vpd can be effectively employed to improve the results in certain problems. A very convenient approach for obtaining the vpd has been proposed by Wu and Trivedia [34], and it can be summarised as follows.

Given k classes, the goal is to estimate the probability of pattern x belonging to each class c:

$$p_c = p(y = c \mid x), c = 1, \ldots, k.$$

For that purpose, it is necessary to estimate first the pairwise class probabilities

$$r_{ab} \approx p(y = c \mid x, \ y = a \ or \ y = b),$$

such that $r_{ab} + r_{ba} = 1$. This can be done as proposed by Lin et al. [14]:

$$r_{ab} \approx \frac{1}{1 + e^{A\hat{f} + B}},$$

where A and B are estimated by minimising the negative log-likelihood function from training data and their corresponding decision values $\hat{f}$. For further information, the reader is referred to the above paper [14].

Given the pairwise class probabilities, the problem of estimating the probability distribution

$$\mathbf{p} = \{p_c\} \ \forall c,$$

turn into the following minimisation problem:

$$\underset{\mathbf{p}}{\text{argmax}} = \frac{1}{2} \sum_{a=1}^{k} \sum_{b:b \neq a} (r_{ab} p_a - r_{ba} p_b)^2$$

subject to

$$\sum_{a=1}^{k} p_a = 1, \ p_i \geq 0, \forall a.$$

After operating on this equation, the problem is reformulated as

$$\underset{\mathbf{p}}{\text{argmin}} = \frac{1}{2} \mathbf{p}^T Q \mathbf{p},$$

where

$$Q_{ab} = \begin{cases} \sum_{s:s\neq a} r_{sa}^2 & \text{if } a = b \\ -r_{ba}r_{ab} & if\ a \neq b \end{cases}$$

This is a linear-equality-constrained convex quadratic programming problem that can be solved by finding the scalar b such that:

$$\begin{bmatrix} Q & \mathbf{i} \\ \mathbf{i}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix},$$

where b is the Lagrangian multiplier of the equality constraint $\sum_{a=1}^{k} p_a = 1$, $\mathbf{i}$ is the vector of dimension $k \times 1$ of all ones, and $\mathbf{0}$ a vector of the same dimension but with all zeros. In that way, we can obtain the probability of each class based on the votes received. This information is employed in our method to achieve a higher accuracy when recovering the head pose as explained below.

3 Problem formulation

Our goal is to estimate the pan (horizontal) and tilt (vertical) angles of the head pose from a set of cameras. We make two assumptions in this work: first, the pan and tilt angles are independent, therefore we apply our estimation technique separately for each angle; second, all the cameras are equally reliable. The latter assumption can easily be relaxed.

We consider a set V of cameras such that all of them share the same global coordinate system, established by calibration, and all cameras see the person's head. We assume that the 3D location of each camera is known, and that the 3D location of the person's head is also known. The former can be obtained by camera calibration, while the latter can be computed from the projection of the head in the cameras.

More precisely, let us denote by

$$x = \{x^v \mid v = 1, \ldots, V\} \tag{3}$$

the set of patterns (images of the head) extracted by the cameras, and by

$$\theta(x) = \{\theta(x^v) \in [l_l, l_u] \mid v = 1, \ldots, V\} \tag{4}$$

their corresponding angles relative to the cameras. The parameters l_l and l_u represent the upper and lower limit of the angular space, respectively. Two things are worth noting here. First, all the patterns in x correspond to the same head seen from different points of view in the same time instant. Second, $\theta(x^v)$ is the angle relative to the v-th camera. What this means in practice is that one unique classifier can be trained with the patterns of all cameras, and that this classifier is not camera specific.

Finally, let us denote by

$$\theta^g \in [l_l, l_u] \tag{5}$$

the true head pose of the person in the global coordinate system.

The angles $\theta(x^v)$ and θ^g represent continuous values. To use MSVMs that rely on discrete labels, we must then discretise the angular space into k equally distributed intervals. We shall represent by

$$\mathcal{C} = \{c_j \mid j = 1, \ldots, k\}, \tag{6}$$

the centres of these intervals, which are calculated by:

$$\begin{aligned} c_j &= l_l + \frac{(l_u - l_l)}{2k} + \frac{(l_u - l_l)(j-1)}{k} \\ &= l_l + \frac{(l_u - l_l)(2j-1)}{2k}. \end{aligned} \tag{7}$$

The interval centres in $\mathcal{C}$ define the set of classes of our problem. Then, patterns are assigned to the nearest class c_j defining the labels

$$y = \{y^v \in \mathcal{C} \mid v = 1, \ldots, V\}, \tag{8}$$

where

$$y^v = \underset{j}{\operatorname{argmin}}\, d(c_j, \theta(x^v)), \tag{9}$$

represents the element of $\mathcal{C}$ minimising the angular distance d to the angle of the pattern $\theta(x^v)$. Intuitively, what we have done is to turn a regression problem into a classification problem by discretisation of the target angle. Our final aim, however, is to estimate the continuous value of a global angle θ^g. In the next section, we shall show how we can achieve this by fusing, in an adequate way, the classification results obtained from each camera.

4 Proposed solution

Our approach to estimate θ^g is based on a two-step process. In the first step, the individual classification result of each camera is transformed into a probability distribution over angles, with reference to the global coordinate system. In the second step, the results from all individual cameras are fused into a combined probability distribution. The angle with maximum combined probability is then delivered as the final estimate generated by this process.

The first step consists in providing an angle estimate for each individual camera. Instead of using the most voted class as the best angle estimate, we propose deriving a finer estimation by employing the vpd of the classes (obtained as explained in Sect. 2.2). For that purpose, let us denote by

$$\mathbf{p^v} = \{p^v(c_i) \mid i = 1, \ldots, k\}, \tag{10}$$

the probability distribution of the pattern x^v over the classes in $\mathcal{C}$. The probability distribution in Eq. 10 is given with reference to the camera coordinate system. In order to fuse

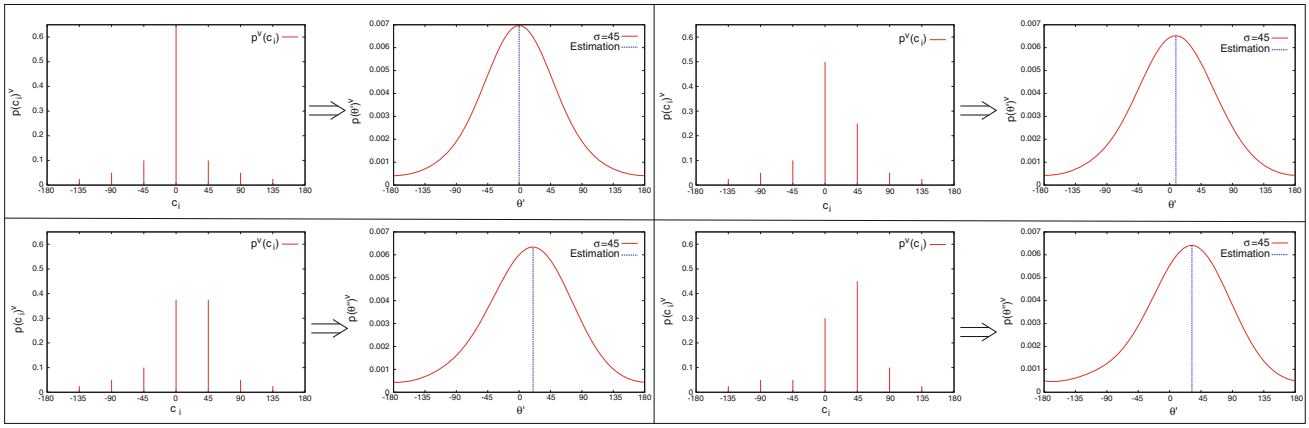

Fig. 1 Example of probability distributions of the angles (*right* graphs of the boxes) obtained from voting probability estimates (*left* graphs)

information from different cameras, it is preferable to translate the angles to the global coordinate system. Let us denote by

$$\theta^g(c_i, v), \tag{11}$$

the angle, in the global coordinate system, corresponding to the class c_i when observed by camera v. Please note that c_i refers to different global angles depending on cameras (since cameras are placed at different locations), i.e.

$$\theta^g(c_i, v_1) \neq \theta^g(c_i, v_2)\ \forall v_1 \neq v_2.$$

Let us define a finer discretisation of the angular space:

$$\theta' = \left\{ \theta'_i = l_l + \frac{(l_u - l_l)(2j-1)}{2k'} \mid i = 1, \ldots, k' \right\}, \tag{12}$$

such that $k' \gg k$. Our approach consists in mapping the probability distribution of Eq. 10 onto this new distribution by interpolation using Gaussian functions as:

$$p^v(\theta') = \frac{1}{k} \sum_k e^{-d(\theta^g(c_i,v),\theta')^2/2\sigma^2} p^v(c_i). \tag{13}$$

The function d represents angular distance, and the parameter σ is the variance of the Gaussian function. The basic idea of Eq. 13 is that angles near class c_j are considered to have a probability proportional to $p^v(c_i)$. Finally, the most probable angle $p^v(\theta')$ is obtained by:

$$\hat{\theta}^v = \underset{\theta}{\operatorname{argmax}}\ p^v(\theta'). \tag{14}$$

A clearer idea of this first step can be obtained from Fig. 1. The figure shows the results of Eq. 13 (right graphs of the boxes) for four different vpd (left graphs). A blue line indicates the best estimate $\hat{\theta}^v$. The proposed method can be employed to obtain a finer estimation of the angle by considering the vpd, as long as this is a faithful representation of the real probability.

Note that one could use a simpler approach to estimate the angle, by calculating the mean of the $\theta^g(c_i, v)$ weighted by their respective probabilities. This would implicitly assume that the underlying probability distribution is Gaussian, which may not be true. Equation 13 allows us to have arbitrary distributions, including multi-modal ones, which is especially interesting in the next step when the information from multiple cameras are fused.

The resulting angle $\hat{\theta}^v$ provides the best estimate of the angle for a single camera. Our next step, then, is to fuse the information obtained from all the available cameras to obtain a more robust and precise angle estimation. To do so, we fuse the cameras vpds so as to create a new one:

$$p^g(\theta') = \frac{1}{V} \sum_v p^v(\theta'). \tag{15}$$

The new distribution $p^g(\theta')$ fuses the information from all cameras considering all of them equally reliable. Finally, we determine

$$\hat{\theta}^g = \underset{\theta}{\operatorname{argmax}}\ p(\theta')^g. \tag{16}$$

as the angle that best explains the observations from all the cameras.

The assumption of equal reliability can be relaxed by actuating on the σ parameter of Eq. 13. For cameras with low confidence, the σ value can be increased so as to smooth the output function, thus reducing the final contribution of the camera to the estimated fused distribution.

Figure 2 summarises the whole pose estimation process. The three-dimensional head location must be first determined. While this problem is out of the scope of this paper, state of the art tracking approaches exist to solve it: in our experiments, we used the approach proposed Muñoz et al. [26,27] for head localisation. Knowing the head location, a three-dimensional cylinder of dimensions 0.5×0.5 m is placed around the head, and it is projected on the camera images. The cylinder projects in rectangular regions that might be of different sizes in each camera. This is due to the different distances of the cameras to the person, or to the

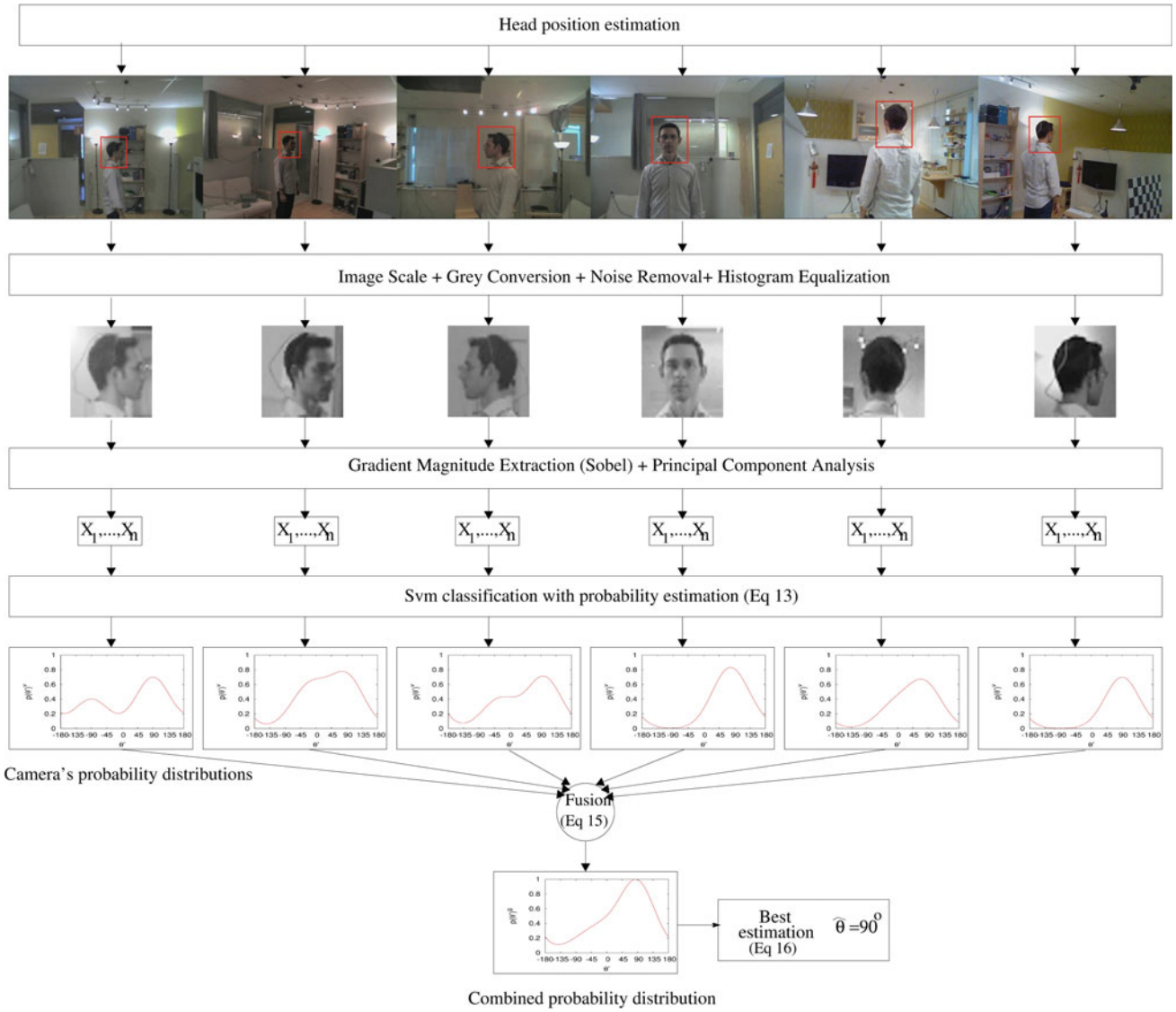

Fig. 2 General scheme of the proposed solution. Head images are extracted and then preprocessed. The n first principal components are the input for an SVM classifier. A probability distribution of the head angles is obtained for each camera and then fused to obtain the best estimation

different optical properties of the cameras employed. Hence, we resize the image head patches to be all of the same size (40×40 pix in our work). Next, the images are grey-scaled and de-noised using a Gaussian kernel. Afterwards, the histograms of the images are equalised so as to compensate for irregular lightning.

As usual in view-based approaches, it is preferable to employ gradient information than grey-scale information. Therefore, we compute the gradient magnitude using the Sobel operator, and we then submit the resulting image to a PCA [12] reduction. The input vectors x_v to the classifier are obtained by selecting the n first principal components and the output used in Eq. 13 to obtain the corresponding pose probability distributions. Finally, these distributions are fused into a single one from which the best estimate is obtained.

5 Experimental results

We now explain the experiments carried out to test and validate the proposed approach. To perform validation, we have created a data set of head images in different poses using six cameras simultaneously. This data set has been obtained in the PEIS-Home [30], a testbed apartment used for research in robotic technology for elderly care.

The apartment has been equipped with six usb cameras placed approximately at 1.4 m height surrounding the subject being recorded. Cameras are configured to record at a frame rate of 6 Hz at 640×480 pix. We employed such a low frame rate because of bandwidth limitations of the usb port. The working space was reduced (16 m^2), but, on average, the distance from the cameras to the person was 2 m.

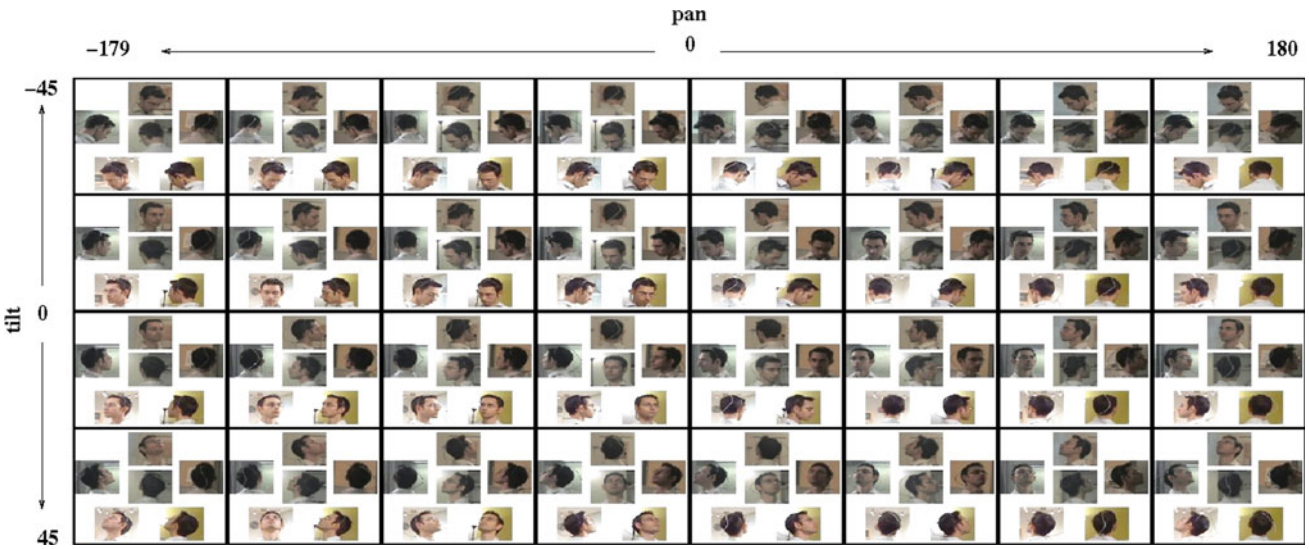

Fig. 3 Patterns from the data set recoded by six cameras simultaneously. The images show the range of angles covered by the subjects

While for the farthest cameras, the image resolution of the head is 60×60 pix, for the nearer cameras it is 120×120 pix. In any case, all images are resized to 40×40 pix.

The top row of Fig. 2 shows images captured by the cameras while creating the data set for one of the subjects. Calibration was performed using the OpenCv library [5], and using a Pholemus tracker sensor mounted on the subject's heads to obtain the ground truth.

A total of ten different persons participated in the experiments. They were instructed to stand and rotate in place so as to cover 360° in pan in intervals of 30°. For covering the tilt angles, the persons were instructed to look up and down covering 90°.

The data set consists of a training and a test set of 9,442 and 4,604 patterns, respectively. Each set contains approximately the same number of patterns for each person. The patterns for each person in the sets are equally distributed among the covered angles: $[-180°, 180°]$ for the pan angle, and $[-45°, 45°]$ for tilt. Figure 3 shows images for one of the subjects in the data set. Each box contains the images captured by the six cameras simultaneously.

For evaluating purposes, the data set is publicly available at http://www.uco.es/grupos/ava/node/25.

The report of the experiments is organised as follows. First, Sect. 5.1 presents an analysis of the problem in terms of regression so as to be able to compare it with our approach. Second, Sect. 5.2 analyses the results of the MSVM without considering probability distributions. Third, Sect. 5.3 shows the results of our head pose estimator using probability distributions at camera level, i.e. without considering fusing the results. Fourth, Sect. 5.4 shows the results obtained when information from multiple cameras are fused. Finally, Sect. 5.5 compares our approach with the neural network approach presented in [38].

5.1 Results with RVM

To compare our approach with the regression one, we trained a pair of RVMs: one for each angle. Training was performed using five-cross validation to estimate the best parameters for the classifiers. In this work, we have employed a radial basis function as kernel [3]. It is defined as

$$k(x_i, x_j) = exp(-\gamma ||x_i - x_j||^2),$$

where $\gamma > 0$ is the kernel parameter.

The best results were obtained for a total for $n = 100$ principal components, $C = 64$ and $\gamma = 0.25$. We set the parameter $\epsilon = 1°$ in all the experiments. The root mean-squared error (RMSE) obtained for pan is 57, 40° and 10, 08° for tilt.

While the error obtained for tilt could be considered appropriate for certain applications, the error in the estimation of pan is very high. Below, the results with our method are presented.

5.2 Results of the MSVM training

For the MSVM classifiers, training was also performed using five-cross validation and using radial basis functions. The best results were obtained for a total for $n = 100$ principal components with the parameters $C = 32$ and $\gamma = 0.03125$.

With these parameters, we trained several classifiers for different angular resolutions. For the pan angle, we tested three possible angle discretisations, namely $k = \{8, 16, 32\}$, thus obtaining resolutions of $45°, 22.5°$ and $11.25°$,

Table 1 Classification results of the different classifiers trained

	Pan $k = 8$	Pan $k = 16$	Pan $k = 32$	Tilt $k = 5$	Tilt $k = 9$
Success (%)	90.11	85.05	73.04	77.86	57.6

respectively. For tilt, we tested $k = \{5, 9\}$ that provide resolutions of 18° and 10°. In each case, the classifier was trained using the images from all the cameras so that the resulting classifiers are not camera specific. The results of the classifiers on the test set are summarised in Table 1.

As it can be seen, the success decreases as the resolution increases. However, the analysis of the confusion matrices (see Fig. 4) reveals that, in most of the cases, errors occur in the neighbour classes. This means that as the resolution increases, there is a larger number of patterns that lay in the margins of the classifiers, thus increasing the likelihood of misclassification in the neighbour classes.

The results of Table 1 show the success in terms of classification rate. However, our goal is to recover the real valued angle. For that purpose, we calculate the error between the ground truth angle (provided by the Pholemus tracker) and the angle of the class given by the classifier. The RMSE are shown in Table 2 along with 95% confidence intervals. The first row shows the RMSE (expressed in degrees) of all the patterns of the training set. The second row represents the RMSE only for these patterns that resulted in a misclassification. Finally, the third row shows the results only for the correctly classified patterns.

As can be seen, the RMSE obtained is rather low in general. As expected, there is a reduction in the error as k increases, but only up to a certain limit. For pan, the limit seems to be in $k = 16$ since when $k = 32$, the increment in the misclassification rate seems to compromise the overall RMSE. For tilt, very good results are obtained even for $k = 5$.

5.3 Results using vpd

The results reported in the previous section are based exclusively on the most voted classifier. This section shows the results obtained using the method proposed in Sect. 4 for obtaining the angle from the vpd (Eqs. 13, 14). As explained in Sect. 4, the method depends on the parameter σ that represents the standard deviation of the Gaussian function. For each classifier reported in the previous section, we have calculated RMSE for the values of $\sigma = \{0, 15, 30, \ldots, 180\}$. The results obtained are summarised in Table 3 for the best configuration of this parameter, denoted σ^* in the Table.

The results obtained show that the proposed approach based on the vpd is an effective way to reduce the error in this problem. Although there is general error reduction in all cases, it is specially relevant in the erroneous patterns. We believe that this is because, in most of the cases, the misclassified patterns lays in the margin of the classifiers, so that they

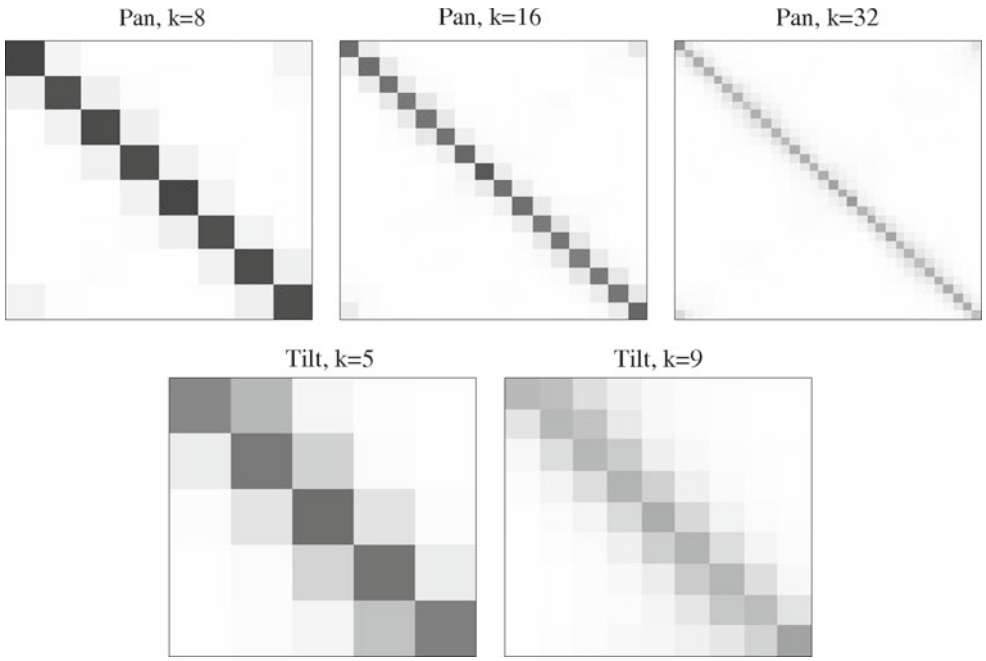

Fig. 4 Confusion matrices for the different classifiers trained

Table 2 RMSE in degrees of the classifiers for the angles of the classes

	Pan $k = 8$	Pan $k = 16$	Pan $k = 32$	Tilt $k = 5$	Tilt $k = 9$
All	14.11 ± 0.48	9.56 ± 0.51	9.56 ± 0.58	7.64 ± 0.19	6.46 ± 0.19
Error	43.47 ± 3.47	33.05 ± 2.76	23.69 ± 1.94	16.89 ± 0.47	11.48 ± 0.33
Correct	10.89 ± 0.21	5.43 ± 0.11	3.59 ± 0.08	5.06 ± 0.06	2.78 ± 0.06

Table 3 Error in the determination of the angles when vpd are employed

	Pan $k = 8$	Pan $k = 16$	Pan $k = 32$	Tilt $k = 5$	Tilt $k = 9$
(σ^*)	(45)	(45)	(45)	(30)	(15)
All	10.92 ± 0.48	8.49 ± 0.48	8.32 ± 0.54	5.95 ± 0.19	5.52 ± 0.19
Error	29.72 ± 3.27	26.09 ± 2.76	19.62 ± 1.83	10.49 ± 0.51	8.46 ± 0.31
Correct	8.56 ± 0.20	5.15 ± 0.10	3.66 ± 0.09	4.27 ± 0.13	3.48 ± 0.12

Table 4 Errors obtained by our proposal as the number of fused cameras increases

N_{cams}	Pan $k = 8, \sigma^* = 45$	Pan $k = 16, \sigma^* = 45$	Pan $k = 32, \sigma^* = 15$	Tilt $k = 5, \sigma^* = 15$	Tilt $k = 9, \sigma^* = 15$
1	10.92 ± 0.48	8.49 ± 0.48	8.32 ± 0.54	5.95 ± 0.54	5.52 ± 0.14
2	8.55 ± 0.22	6.01 ± 0.23	5.70 ± 0.28	4.72 ± 0.48	4.56 ± 0.42
3	7.02 ± 0.14	4.84 ± 0.16	4.57 ± 0.20	4.23 ± 0.29	4.19 ± 0.12
4	6.14 ± 0.14	4.19 ± 0.16	3.93 ± 0.22	4.00 ± 0.21	4.01 ± 0.31
5	5.65 ± 0.19	3.90 ± 0.25	3.57 ± 0.31	3.86 ± 0.86	3.90 ± 0.13
6	5.23 ± 0.35	3.38 ± 0.43	3.24 ± 0.73	3.76 ± 0.35	3.81 ± 0.46

are confused with a neighbour class. Thus, the problem can be alleviated by considering the probability of the neighbour by the interpolation method proposed.

5.4 Results fusing multiple views

The two previous sections have shown the results of estimating the person's head pose from a single camera. This section shows the results obtained when information from multiple cameras are fused using Eqs. 15 and 16.

Table 4 shows the error obtained when the data from the six cameras are fused using our proposal. The first column indicates the number of cameras employed for the fusion. The second row shows the average RMSE obtained by fusing the results of all possible combinations of two cameras in the test set. The same rationale is applied to the rest of the columns for an increasing number of cameras. At the top of each column, the angle resolution employed along with the σ^* that produced the best fusion results is indicated.

As can be observed, the proposed method is able to reduce the error in the angle by fusing information from multiples cameras. Indeed, the reduction is more pronounced when using few cameras. As the number of cameras increase, the improvement becomes smaller.

To better analyse the behaviour of the proposed method, we have evaluated the error reduction as a function of the number of cameras in two different experiments. In the first experiment, we focused only on those situations in which all the cameras obtained correct classifications. For these cases, we evaluated the reduction obtained as the number of fused cameras increased. This experiment aims predicting the upper bound of the method's performance for a variable number of cameras. In the second experiment, we focused only on those situations in which at least one of the cameras obtained a misclassification. For these cases, we fused the result of the erroneous camera with all possible correct combinations of the rest of the cameras. This experiment aims to examine the method's ability to overcome camera errors.

The results of the first experiment is shown in Table 5. The table contains in each column the RMSE for each one of the configurations analysed. Rows represent the number of cameras employed to obtain the results. Please note that the results for more than one camera have been obtained by averaging all the possible combinations of fusion between the cameras. The results obtained show that the greatest error reduction takes place when combining information from two cameras. Above this number, a smaller reduction is observed.

The results of the second experiment are shown in Table 6. The N_{cams} parameter indicates the total number of cameras

Table 5 Error in the determination of the angles when data from an increasing number of correct cameras are fused

N_{cams}	Pan $k = 8$	Pan $k = 16$	Pan $k = 32$	Tilt $k = 5$	Tilt $k = 9$
1	8.56 ± 0.20	5.15 ± 0.10	3.66 ± 0.09	4.27 ± 0.13	3.48 ± 0.12
2	6.78 ± 0.08	3.57 ± 0.06	3.43 ± 0.07	3.46 ± 0.07	3.13 ± 0.15
3	5.71 ± 0.08	2.98 ± 0.05	2.76 ± 0.06	3.06 ± 0.07	2.90 ± 0.14
4	5.09 ± 0.08	2.62 ± 0.05	2.34 ± 0.06	2.83 ± 0.07	2.75 ± 0.17
5	4.99 ± 0.11	2.41 ± 0.07	2.11 ± 0.09	2.67 ± 0.12	2.70 ± 0.16
6	4.11 ± 0.26	2.29 ± 0.16	1.91 ± 0.09	2.60 ± 0.29	2.61 ± 0.15

Table 6 Error in the determination of the angles when data from the six cameras are fused

N_{cams}	N_{corr}	Pan $k = 8$	Pan $k = 16$	Pan $k = 32$	Tilt $k = 5$	Tilt $k = 9$
1	0	29.72 ± 3.27	26.09 ± 2.76	19.62 ± 1.83	10.49 ± 0.51	8.46 ± 0.31
2	1	11.93 ± 0.77	8.48 ± 0.51	6.58 ± 0.40	6.51 ± 0.24	5.55 ± 0.13
3	2	7.20 ± 0.15	4.83 ± 0.21	4.05 ± 0.16	5.04 ± 0.10	4.47 ± 0.18
4	3	6.02 ± 0.43	4.00 ± 0.10	3.04 ± 0.13	4.77 ± 0.09	3.96 ± 0.10
5	4	5.21 ± 0.32	3.79 ± 0.35	2.65 ± 0.12	4.54 ± 0.12	3.57 ± 0.18
6	5	4.66 ± 0.30	3.69 ± 0.21	2.62 ± 0.12	4.53 ± 0.13	3.38 ± 0.43

employed in the fusion, while the second column N_{corr} indicates how many of these cameras produced correct results. Therefore, the first row shows the results of these cameras that produced misclassifications, i.e. these are the same results presented in the row labelled "Error" of Table 3. The second row of Table 6 shows the results obtained when a misclassification is fused with a correct classification. Similarly, the third row represents the angle estimated when a misclassified results is fused with all possible combinations of two correct results of the rest of the cameras. A similar rationale is applied to the rest of the rows. The results presented are the average results of combining the misclassified camera with all possible combinations of the correct cameras.

The results in Table 6 shows a clear improvement in the angle estimated as soon as a misclassification is fused with a correct classified pattern. In general, it can be observed that even when two cameras are employed (second row), the results are much better than these obtained by the individual classifiers reported in Table 2. So, it can be considered that the proposed method is an effective way of fusing the outputs from the multiple cameras.

5.5 Comparison with neural networks

In this section, we aim at comparing our approach with the neural network method presented in [38]. Although their method also includes a method for head tracking in video sequences, at the core is an NN as classifier. Our approach works in a similar way to the one proposed by [38], i.e. both produces probability distribution about the head pose that can be fused with multiple views. As a consequence, both classifiers are easily interchangable. The point is then to decide which one produces better results.

The work of Voit et al. assumes independence between pan and tilt angles, thus employing a separate NN for each angle. The network input consists in a total of 2,048 features. Half of the inputs are obtained by concatenating the resampled image (at 32×32 pixels) after equalising its histogram (for light correction). The rest of the inputs correspond to the gradient magnitude of the image applying a Sobel operator. The middle layer of the NN comprises 100 hidden neurons and the desired output is the probability distribution of the measured head orientation, i.e. the target values correspond to a discretised Gaussian distribution centred at the head orientation. As in our work, the output distributions obtained across multiple cameras are fused considering the relative position of the camera and the subject. The result is a fused distribution from which the most likely angle is obtained as the best estimation.

In order to achieve a fair comparison between the two techniques, we have employed the same discretisation for pan ($k = \{8, 16, 32\}$) and tilt ($k = \{5, 9\}$). In other words, for the pan angles, we have trained NNs with three different outputs, namely 8, 16 and 32 outputs. The same rationale is applied for tilt angles. Likewise, we have tried the values $\sigma = \{15, 30, 45\}$ for the Gaussian distribution that defines the output.

The results obtained are shown in Table 7, which present the errors as the number of cameras fused increase. All possible combinations of cameras haven been tested for each case, and the average values with with 95% confidence intervals are shown in the table. The leftmost column indicates

Table 7 Errors obtained by the neural networks as the number of cameras fused increases

	Pan $k = 8, \sigma^* = 30$	Pan $k = 16, \sigma^* = 15$	Pan $k = 32, \sigma^* = 30$	Tilt $k = 5, \sigma^* = 15$	Tilt $k = 9, \sigma^* = 15$
1	12.05 ± 0.67	14.45 ± 0.85	14.21 ± 0.02	6.32 ± 0.34	7.30 ± 0.74
2	8.05 ± 0.14	9.05 ± 0.44	9.32 ± 0.82	5.13 ± 0.38	6.05 ± 0.92
3	6.69 ± 0.74	7.20 ± 0.68	7.43 ± 0.67	4.63 ± 0.99	5.58 ± 0.61
4	5.87 ± 0.02	6.27 ± 0.76	6.39 ± 0.66	4.37 ± 0.84	5.36 ± 0.46
5	5.28 ± 0.56	5.35 ± 0.13	5.67 ± 0.16	4.23 ± 0.74	5.24 ± 0.39
6	4.90 ± 0.23	4.99 ± 0.71	5.22 ± 0.37	4.14 ± 0.65	5.17 ± 0.31

the number of cameras fused and each column presents the results for a different discretisation of the output. The table contains only the results for the σ^* of the Gaussian distribution that achieved the best results.

The results obtained by the NN approach can be contrasted with those of our method presented in Table 4. It can be observed that the NN approach obtains slightly worse results than the method proposed in this work.

6 Conclusions

This paper has proposed a novel approach to the problem of head pose estimation using multiple cameras. The key point of our approach is to employ the vpd of multi-class support vector machine classifiers to derive a probability distribution of the head pose angles. The proposed approach outperforms the result of both support vector and regression vector machines. In addition, in our approach information from multiple views can be fused to produce a more precise and more robust estimate. The experiments conducted demonstrate that the proposed fusion approach is able to overcome classification errors from individual cameras by using the information from the remaining cameras.

There are three important assumptions that underlay our approach: first, the position of all cameras is known with respect to a common global frame of reference; second, the position of the head is known with respect to the same frame; third, all cameras are equally reliable. While state of the art techniques exist to cope with the first two assumptions, relaxing the last assumption is something which is part of our future work. We consider that fuzzy logic can be employed to improve the final voting scheme by adding information about the cameras' confidence.

References

1. Ba, S.O., Odobez, J.-M.: From camera head pose to 3d global room head pose using multiple camera views. Lect. Notes Comput Sci **4625**, 276–286 (2008)
2. Basu, S., Essa, I., Pentland, A.: Motion regularization for model-based head tracking. In: International Conference on Pattern Recognition, pp. 611–616 (1996)
3. Bishop, C.M.: Pattern recognition and machine learning. Springer, New York (2006)
4. Black, M., Yacoob, Y.: Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In: International Conference on Computer Vision, pp. 374–381 (1995)
5. Bradski, G., Kaehler, A.: Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly, Sebastopol, CA (2008)
6. Canton-Ferrer, C., Segura, C., Casas, J.R., Pardas, M., Hernando, J.: Audiovisual head orientation estimation with particle filtering in multisensor scenarios. EURASIP J. Adv. Signal Process. **4625**, 1–12 (2008)
7. Cascia, M.L., Sclaroff, S., Athitsos, V.: Fast, reliable head tracking under varying illumination: an approach based on registration of textured-mapped 3d models. Pattern Anal. Mach. Intell. **22**, 322–336 (2000)
8. Chen, L., Zhang, L., Hu, Y., Li, M., Zhang, H.: Head pose estimation using fisher manifold learning. In: IEEE International Workshop on Analysis and Modeling of Faces and Gestures, pp. 203–207 (2003)
9. Choi, K.N., Worthington, P.L., Hancock, E.R.: Estimating facial pose using shape-from-shading. Pattern Recogn. Lett. **23**, 533–548 (2002)
10. Choia, S., Kim, D.: Robust head tracking using 3d ellipsoidal head model in particle filter. Pattern Recognit. **41**, 2901–2915 (2008)
11. Cootes, T.F., Wheeler, G.V., Walker, K.N., Taylor, C.J.: View-based active appearance models. Image Vis. Comput. **20**, 657–664 (2002)
12. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press, New York (1990)
13. Gee, A., Cipolla, R.: Fast visual tracking by temporal consensus. Image Vis. Comput. **14**, 105–114 (1996)
14. Weng, R.C., Lin, H.T., Lin, C.J.: A note on platt's probabilistic outputs for support vector machines. Mach. Learn. **68**, 267–276 (2007)
15. Horprasert, T., Yacoob, Y., Davis, L.S.: Computing 3-D orientation from a monocular image sequence. In: IEEE Conference on automatic face and gesture recognition, pp. 242–247 (1996)
16. Huang, J., Shao, X., Wechsler, H.: Face pose estimation using support vector machines. In: International Conference on pattern recognition, pp. 154–156 (1998)
17. Ji, Q.: 3D face pose estimation and tracking from a monocular camera. Image Vis. Comput. **20**, 499–511 (2002)
18. Knerr, S., Personnaz, L., Dreyfus, G.: Single-layer learning revisited: a stepwise procedure for building and training a neural network. In: Neurocomputing: Algorithms, Architectures and Applications. NATO ASI Series. Springer, Heidelberg (1990)
19. Krüger, V., Sommer, G.: Gabor wavelet networks for efficient head posenext term estimation. Image Vis. Comput. **20**, 665–672 (2002)

20. Lanz, O., Brunelli, R.: Joint bayesian tracking of head location and pose from low-resolution video. Lect. Notes Comput. Sci. **4625**, 287–296 (2008)
21. Lee, M.W., Ranganath, S.: Pose-invariant face recognition using a 3D deformable model. Pattern Recognit. **36**, 1835–1846 (2003)
22. Li, Y., Gong, S., Liddell, H.: Support vector regression and classification based multi-view face detection and recognition. In: IEEE International Conference on Automatic Face and Gesture Recognition, pp. 300–305 (2000)
23. Lin, C., Fan, K.-C.: Pose classification of human faces by weighting mask function approach. Pattern Recognit. Lett. **24**, 1857–1869 (2003)
24. Malassiotis, S., Strintzis, M.G.: Robust real-time 3d head posenext term estimation from range data. Pattern Recognit. **38**, 1153–1165 (2005)
25. Muñoz Salinas, R.: A bayesian plan-view map based approach for multiple-person detection and tracking. Pattern Recognit. **41**, 3665–3676 (2008)
26. Muñoz Salinas, R., García-Silvente, M., Medina-Carnicer, R.: Adaptive multi-modal stereo people tracking without background modelling. J Vis Commun Image Represent **19**, 75–91 (2008)
27. Muñoz Salinas, R., Medina-Carnicer, R., Madrid-Cuevas, F.J., Carmona-Poyato, A.: Multi-camera people tracking using evidential filters. Int J Approx Reason **50**, 732–749 (2009)
28. Muñoz Salinas, R., Medina-Carnicer, R., Madrid-Cuevas, F.J., Carmona-Poyato, A.: People detection and tracking with multiple stereo cameras using particle filters. J Vis Commun Image Represent **20**, 339–350 (2009)
29. Rajwade, A., Levine, M.D.: Facial pose from 3d data. Image. Vis. Comput. **24**, 849–856 (2006)
30. Saffiotti, A., Broxvall, M.: PEIS ecologies: Ambient intelligence meets autonomous robotics. In: Proceedings of the International Conference on Smart Objects and Ambient Intelligence (sOc-EUSAI), pp. 275–280, Grenoble (2005)
31. Sherrah, J., Gong, S.: Fusion of perceptual cues for robust tracking of head posenext term and position. Pattern Recognit. **34**, 1565–1572 (2001)
32. Srinivasan, S., Boyer, K.L.: Head pose estimation using view based eigenspaces. In: 16th International Conference on Pattern Recognition, pp. 302–305 (2002)
33. Zhang, H.J., Cheng, Q.S., Li, S.Z., Peng, X.H.: Multi-view face pose estimation based on supervised isa learning. In: IEEE International Conference Automatic Face and Gesture Recognition, pp. 100–105 (2002)
34. Weng, R.C., Wu, T., Lin, C.: Probability estimates for multi-class classification by pairwise coupling. J. Mach. Learn. Res. **5**, 975–1005 (2004)
35. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
36. Voit, M., Nickel, K., Stiefelhagen, R.: Multi-view head pose estimation using neural networks. In: 2nd Canadian Conference on Computer and Robot Vision, pp. 347–352 (2005)
37. Voit, M., Nickel, K., Stiefelhagen, R.: Head pose estimation in single- and multi-view environments—results on the clear'07 benchmarks. Lect. Notes Comput. Sci. **4625**, 307–316 (2009)
38. Voit, M., Stiefelhagen, R.: A system for probabilistic joint 3d head tracking and pose estimation in low-resolution, multi-view environments. Lect. Notes Comput. Sci. **5815**, 415–424 (2009)
39. Wang, J.-G., Sung, E.: Pose determination of human faces by using vanishing points. Pattern Recognit. **34**, 2427–2445 (2001)
40. Wanga, J.-G., Sungb, E.: Em enhancement of 3d head posen estimated by point at infinity. Image Vis. Comput. **25**, 1864–1874 (2007)
41. Wu, J., Trivedia, M.M.: A two-stage head pose estimation framework and evaluation. Pattern Recognit. **41**, 1138–1158 (2008)
42. Fradet L., Wei, Y., Tan, T.: Head pose estimation using gabor eigenspace modeling. In: IEEE International Conference on image processing, pp. 281–284 (2002)
43. Zhang, Z., Hu, Y., Liu, M., Huang, T.: Head pose estimation in seminar room using multi view face detectors. In: CLEAR Evaluation and Workshop, pp. 281–290 (2006)

Author Biographies

Rafael Muñoz-Salinas received the Bachelor degree in Computer Science from Granada's University (Spain) and the Ph.D. degree from Granada's University (Spain), in 2006. Since 2006 he has been working with the Department of Computing and Numerical Analysis of Cordoba University; currently, he is lecturer. His research is focused mainly on Mobile Robotics, Human-Robot Interaction, Artificial Vision and Soft Computing techniques applied to Robotics.

E. Yeguas-Bolivar received the Bachelor degree in Computer Science from Granada University (Spain) and the Ph.D. degree from Granada University (Spain), in 2009. Since 2006 he has been working with the Department of Computing and Numerical Analysis of Cordoba University. Currently, he is assistant professor. His research is focused mainly on Geometric Constraint Solving, Virtual Reality and Soft Computing techniques applied to Computer Vision.

A. Saffiotti received the M.S. degree in computer science from the University of Pisa, Pisa, Italy, and the Ph.D. degree in applied science from the Université Libre de Bruxelles, Brussels, Belgium. He is currently a Full Professor of computer science at the Center for Applied Autonomous Sensor Systems, Department of Technology, Örebro University, Örebro, Sweden, where he has been heading the AASS Cognitive Robotic Systems Laboratory since 1998. He has published more than 120 papers in international journals and conferences, and coedited the book Fuzzy Logic Techniques for Autonomous Vehicle Navigation (Springer, 2001). His research interests encompass artificial intelligence, autonomous robotics and soft computing.

R. Medina-Carnicer received the Bachelor degree in Mathematics from University of Sevilla (Spain). He received the Ph.D. in Computer Science from the Polytechnic University of Madrid (Spain) in 1992. Since 1993 he has been a lecturer of Computer Vision in Cordoba University (Spain). His research is focused on Edge detection, 3D Vision and Pattern Recognition.