Driver’s head pose estimation using a hierarchical classification on an effective feature space

Ali Ghaffari¹, Mahdieh Rezvan¹, Alireza Khodayari², Seyed Hossein Sadati² and Afra Vahidi-Shams¹

Abstract
This paper introduces a direct accountability technique for the real-time estimation of the head’s position and orientation. Unlike existing works which rely on feature extraction either in the image domain or in three-dimensional space, our proposed approach based on a hierarchical expert classification will estimate the head pose for real-time applications, specifically for driver assistance systems. We proposed a method for feature extraction which aims at keeping only the angle-related features, independent of identity. We then used a hierarchical algorithm for classification which divides the whole space into smaller spaces. Moreover, we made a performance study aiming at evaluating the accuracy of the proposed approach. Experimental results show that the proposed features are able to describe the pose of the driver face image successfully, with a performance in agreement with the actual poses. This approach is not only useful for automobile safety and security but also can help severely handicapped people and can be used in collision prevention systems and other driving applications.

Keywords
Head pose estimation, human–machine interface, intelligent automotive systems, driver behaviors surveillance

Date received: 13 August 2011; accepted 14 February 2012

Introduction
Techniques related to the detection and recognition of the human face play important roles in many applications such as the human–computer interface, video surveillance, and more recently, human–robot interactions. Head poses are certainly quite an important indicator of a person’s focus of attention. Determining a face pose in a driving application is a complex problem for which numerous methods have been devised.¹ ² A historical study done on the subject for driver assistance systems is in fact an indication of how rapidly these systems are being developed. Several different techniques have been proposed for vision pose estimation in recent years.³–⁶ It is noted that one of the main application domains is in fact the so-called advanced driver assistance.

In machine vision applications, head pose estimation is normally related to the state of a human head orientation. In addition to biological appearance, an ideal head pose estimator, as with any other facial vision techniques, must be invariant against many factors. Such factors could include physical phenomena such as the facial expression, camera distortion, lighting, and the presence of accessories such as hats and glasses.

In machine vision, the detection of facial features and tracking in video sequences are indeed two of the main challenging problems.⁷ In a general sequence, detection of the face in a frame is normally followed by an automatic extraction of facial features in video sequences. Among the many approaches proposed to detect faces in static scenes are principal component analysis,⁸,⁹ and clustering and neural nets.¹⁰ Although these techniques may result in good detection accuracy rates, they are usually computationally expensive, in
addition to their inability to offer a real-time performance.

In recent years, there have been various approaches in the literature for head pose estimation. The approaches proposed in the literature can be generally classified into two main categories, namely the three-dimensional (3D) and two-dimensional (2D) head pose estimation approaches. Another method of categorizing estimation techniques relates to the approach used to estimate the head position. These methods can be described in the following main categories.

- **Appearance Template Methods** make a comparison of a group of image samples, each taken at a discrete pose, so as to find the most similar view.
- **Detector Array Methods** focus on the training of an ensemble of head detectors, each corresponding to a certain pose, assigning a discrete pose to the detector having the greatest support.
- **Nonlinear Regression Methods** make use of nonlinear interpolation to devise a functional mapping from the image or feature space to a measurement space for the head pose.
- **Manifold Embedding Methods** models with low dimension manifolds representing the continuous changes in head pose. When a new image arrives, it can be embedded into these manifolds, then used for embedding the template matching or interpolation.
- **Flexible Models** which set a non-rigid (flexible) model onto the facial geometry of every individual inside the image plane. The head pose is then estimated by proper comparisons made at the feature levels or by using the model parameters.
- **Geometric Methods** make use of the feature locations as the nose tip, mouth, and eyes in order to find the pose from their relative configuration.
- **Tracking Methods** detect the possible head changes using the movement observed from one frame to the next.
- **Hybrid Methods** make use of a combination of one or more of the previous methods to remedy the limitations inherent in any single approach.

It is noted that for a proper design in our method, the system should be able to make head pose estimations using a single camera. The system should also be able to give the proper estimates for a continuous range of head orientations in the more important directions in driving, namely, the yaw and pitch, within reasonable time (approximately 9 m/s or faster). In driving situations, the orientation of the head in roll is not as important as that in pitch and yaw; therefore, we introduce our method for the estimation of the head position in just yaw and pitch and disregard the roll. Finally, the performance of the proposed method must be invariant of the driver as well as the operating conditions. What we are interested in is the development of full intelligent interfaces, serving as the first step in enabling computers to interact with people.

The remainder of this paper is organized as follows. This Introduction has reviewed some techniques related to the present work. The next section presents a description for the methodology of the work. In this section, the complete details of the face detection, normalization, classification, and feature extraction methodology are described. The proposed methodology and the results obtained are presented in the third section. The last section presents the conclusions and scope for future studies.

**Detailed design and implementation**

The main purpose of our study is to provide estimations for head pose parameters using images obtained from video sequences. In other words, we track the head pose configuration in two directions over time. In this section, we propose an approach that creates a feature space from the video images containing the angle...
only related features. We then introduce our hierarchical method for classification. Figure 1 shows the algorithm flowchart.

Our framework consists of these stages: 1. Image enhancement; 2. Face detection; 3. Features extraction; 4. Pose estimation using classification. These steps are illustrated in the following sections.

**Image enhancement algorithm**

The light level condition is the major restriction in our work. What is required is that the lighting for the images used must in fact be normal. If, therefore, the faces to be detected happen to be too bright or too dark, we increase the robustness against the light variations by applying a proper light compensation/correction preprocessing technique. We use the ‘Retinex’ which is a well-known algorithm for image enhancement. The Retinex is aimed at obtaining the balance between the human vision and machine vision system in addition to providing color constancy.\(^{15}\) The initial idea behind Retinex was introduced by Land\(^{16}\) as a model of lightness and color perception of human vision. It is evident that Retinex is not only a model, but can also serve as an algorithm used for image enhancement. After Land, Jobson and his colleagues\(^{17}\) defined a single-scale Retinex (SSR), which is an implementation of what became known as the center/surround Retinex. The single-scale Retinex is given by the relation

\[
R_i(x,y) = \log I_i(x,y) - \log[F(x,y) \times I_i(x,y)]
\]

where \(I_i(x,y)\) is the image distribution in the \(i\)th color band, \(F(x,y)\) is the normalized surround function given as

\[
\int F(x,y)dx\,dy = 1
\]

It is noted that the image distribution is in fact the product of a scene’s reflectance and illumination, i.e.

\[
I_i(x,y) = S_i(x,y)r_i(x,y)
\]

where \(S_i(x,y)\) is the spatial distribution of illumination and \(r_i(x,y)\) is the distribution of scene reflectance. There exist a variety of surround functions. We use the Gaussian formula with absolute parameter \(c\) as

\[
F(x,y) = \exp\left(-r^2 + c^2\right)
\]

Depending on the scale used, SSR can either provide dynamic range compression (small scale) or tonal rendition (large scale). Superposition of weighted difference scale SSR provides a balance between these two effects resulting in a multi-scale Retinex (MSR) given by

\[
R_{MSR} = \sum_{k=1}^{N} \omega_k R_{nk}
\]

where \(N\) is the number of scales, \(R_{nk}\) is the \(n\)th component of the \(k\)th scale. An immediate question about

**Face detection**

Edge detection is used here followed by filling holes so as to separate objects in each frame and find the face-like regions. At this stage, it so happens that there may exist several face-like regions in the image block. In such cases, we select the area with the maximum pixels as the face region. With 1200 test images of 15 subjects, background complexities, and lighting conditions, the rate for the correct face detection is found to be 86% in this work. It is worth mentioning that the proposed system does not guarantee whether the identified face region is accurate enough for the next stages.

**Features extraction**

A face detector has been utilized for the detection of the facial region in the earlier stage. At this point now, the edge detection is performed on the images in four directions, followed by a dilation process on the four images and an averaging process was then done for the final images obtained. The resulting final image is the image belonging to the feature space, a space termed as the pose feature space. This pose is what is used for the
ultimate classification. Sample feature spaces are shown in Figure 2 for a number of subjects with different poses.

**Dimension reduction by principal component analysis (PCA)**

The next stages in our method are prototype fabrication, the definition of a calibration method, and then study of a tracking framework for the driver’s head pose. We normalize feature vectors from previous steps and calculate the average of training images to create the prototype for each face pose. Then we use PCA to reduce the dimensions of the obtained vectors. Afterwards, features are stored in the new dataset so that the features of new frame can be compared by one nearest neighbor classifier.

PCA\textsuperscript{20,21} is a general method used in an image processing system. With the \( N \) sample of images \( \{x_1, x_2, ..., x_N\} \) in an \( n \)-dimensional space \( \mathbb{R}^n \) used as the training set, PCA performs a linear mapping or transformation \( W^T \), where \( W \in \mathbb{R}^{n \times m} \) is a matrix with orthonormal columns. The mapping is done from an \( \mathbb{R}^n \) space using the original image to an \( \mathbb{R}^m \) feature space, in which \( m < n \). The coordinates of the new image vectors, known as the feature vectors, are given by

\[
Y_k = W^T x_k, \quad k = 1, 2, \ldots, N
\]

The total scatter matrix \( S_T \) is defined as

\[
S_T = \sum_{k=1}^{N} (X_k - \mu)(X_k - \mu)^T
\]

where \( \mu \in \mathbb{R}^n \) is the mean of all images.

Once the linear transformation \( W^T \) is performed, the transformed feature vectors \( \{y_1, y_2, ..., y_N\} \) have the scatter \( W^T S_T W \). The PCA aims at maximizing the projection \( W_{opt} \) of the determinant of the total scatter matrix of the projected images. The \( W_{opt} \) matrix is given by the relation

\[
W_{opt} = \text{argmax}_{W} |W^T S_T W| = [w_1 w_2 \ldots w_m]
\]

where \( \{w_i | i = 1, 2, \ldots, m\} \) is the \( m \) eigenvectors in \( \mathbb{R}^n \) of \( S_T \) associated with the largest \( m \) eigenvalues. These eigenvectors represent the original image vectors, which are known as the principal components or the eigenfaces in the relevant literature.

**Classification**

The purpose of this section is to perform pattern recognition, i.e. the recognition and classification of the image based on its describing features. The process of pose estimation will be completed at the end of this stage. In this article, neural networks of the multilayer perceptron (MLP) type are used for classification.

**Experimental set-up and results**

We implement our feature space for head pose estimation and present the results on the public head pose database (Pointing’04 database\textsuperscript{19}).

**Dataset**

The Pointing’04 database consists of 15 sets of images, with and without wearing glasses and with various skin colors. Each set contains two series of 93 images of the same person, and the 93 head poses are determined by the yaw and pitch angles, which vary from \(-90^\circ \) to \(+90^\circ \). One subject with various head poses is shown in Figure 3.

**Cropping based on nose-tip method**

We first crop each image based on the nose-tip method. Background influence is generally avoided. Therefore, to reduce such influence, the cropping process for the image patches is not performed by centering on the nose-tips. Rather, the nose-tip position is shifted from the center of the cropped image based on the known head poses in order to remove the background to a high degree.\textsuperscript{22} As shown in Figure 4, by specifying the nose-tip location \( (x_n, y_n) \) and the ground truth yaw and pitch angle \( (\theta, \phi) \), the center of the image patch (of size \( W \times H \)) is determined by

\[
x_c = x_n + p \frac{W \phi}{180}
\]

\[
y_c = y_n + p \frac{H \phi}{180}
\]

where \( p \) denotes the percentage of shift from the center.
Edge detection

The face-cropped images must be changed so that the image is independent of the features related to the identity, keeping only the features related to the angle. Such an image encompassing only the face features related to the angles constitutes our feature space. The edge detection process is performed by convolving the image with an edge operator (a 2D filter known as mask). In the present study, edge detection is performed in four directions on the images by using the horizontal Sobel operator, vertical Sobel operator, Robert's operator in the 45° direction, and Robert's operator in the 135° direction.

Edge dilation on edge detected images

It is noted that the edges can change depending on various environmental conditions such as lighting and partial head displacement in different images of the same angle, hence making different results in the edge detection. With such considerations, a method is needed to dilate the edges in the edge detected image. For this purpose, a random window of size $6 \times 6$ is selected from the parts of images which include the edge information and it is convolved with the images resulting from each edge detector. This process is repeated several times by choosing different windows from different parts of the image. The obtained images are then averaged together. This results in the dilation of the image edges. Finally, the four images resulting from all the operators after dilation are put together and averaged pixel by pixel.

Classification

The process of pattern recognition is performed for the classification. Considering the large number of classes in the case of head pose estimation on the Pointing’04 database, in the following experiments the total problem space is divided into a number of smaller spaces in order to increase the accuracy of the estimation.

Experiment 1: The first experiment is performed on the images that contain only the yaw changes with 13 angles, with the pitch angles unchanged. The purpose of this experiment is to review the ability of the method presented in this article to separate the extracted features when only the yaw classification is considered. The test includes seven stages, each corresponding to one of the pitch angles. Each stage contained 13 angles, making the problem a thirteen-class type.

To solve this thirteen-class problem by the proposed method, it is necessary to train the classifier by making use of the images of eight subjects as the training set and the images of seven subjects as the test set, all selected randomly. The experiment was repeated with 8, 10, and 12 training samples. Figure 5 shows the result of this experiment obtained from 20 runs.

Experiment 2: In order to evaluate the efficiency of our feature extraction technique in pitch only, in the next experiment the images differing only in their pitch angles were used, keeping yaw angles unchanged. The simulation is done in 13 stages, each corresponding to one of the yaw angles with seven pitch angles. The problem is then of a seven-class type in this case.

The number of training and test images selected for this experiment is eight and seven, respectively, all selected randomly. Figure 6 shows the results of this experiment performed in 20 runs for 8, 10, and 12 training image samples.

Experiment 3: This experiment concerns the head pose estimation with the proposed method as applied on the whole database space with a different classification. In this experiment, the images are classified by two
separate classifiers in both the yaw and pitch directions. These two classifier experts perform the classification process in parallel on the feature space created from the images.

The two expert systems are 7-class and 13-class classifiers responsible for the pitch and yaw directions, respectively. The end result is the presented angle for the test image. The result of this experiment for 20 runs shows a 36.73% correct estimation rate for the images of 8 subjects used as the training set and 7 subjects used as the test set, as depicted in Table 1. This result shows that the classification method used in experiment 3 offers better estimations for the yaw than the pitch angles, although overall, this method outperforms the existing methods, as shown in Table 1.

Hierarchical classification algorithm

This is an experiment intended to examine a different methodology for the classification, setting grounds to assess our method on a set of images that includes 72 angles of the Pointing’04 database.

Considering the large number of classes in the case of head poses estimation on the Pointing’04 database, we proposed a hierarchical classification algorithm so as to increase the accuracy of the estimation. The hierarchical classification is an algorithm based on the well-known ‘divide and conquer (D&C)’ algorithm, as described elsewhere.23 In computer science, divide and conquer is an important algorithm design paradigm based on multi-branched recursion. This algorithm works by recursively breaking down a problem into two or more sub-problems of the same (or related) type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem.23 Therefore, in this method the whole space is divided into smaller ones in the three steps below.

First, the problem is separated into two two-class problems for the up–down and right–left cases. An
An expert is used to classify each (with a 95% correct classification rate). By applying these two classifiers, the primary space is partitioned into four parts. Each of these four areas is then looked at as a left–right two-class problem by classifiers of a second-level classifier expert type, resulting in an approximately 95% correct rate. In the final level then, to estimate the angle, each of the eight regions created is considered as a nine-class problem, classified by classifiers of a third-level expert type. This hierarchical procedure is depicted in Figure 7.

Application of this hierarchical method converts the complex space of the original problem into smaller problems so that the MLP classification results will increase the performance from a very low rate (about 15%) with one whole space to an acceptable
level with the divided space. Figure 8 illustrates the ‘zoning’ procedure.

In this experiment, like the previous ones, the images of eight subjects from the total 15 are used for training of the classifier, while the remaining 7 are used as the test images, all selected randomly. Figure 9 illustrates the results of this experiment obtained in 20 runs.

Table 1 shows the performance evaluation and comparison of the model in experiment 3 and the hierarchical model with those presented in the literature. The performance is assessed by the mean error in both the pitch and yaw directions. In addition, the classification rates are calculated in the yaw and pitch. The classification rates were also calculated by disregarding the estimation error of up to $15^\circ$.

The results of Table 1 illustrate that the hierarchical method outperforms the method of experiment 3. It is also observed that the results of our proposed method are considerably better than those of the PNMF, NMF, PCA, HOSVD, LAAM, and NN-based methods reported in the literature.

Although we reported the results with the MLP, we tested our method using the RBF as well. The results show that the estimation rate in the classification method is independent of the type of neural net used.

**Experiment for real-time video**

In order to demonstrate the feasibility and applicability of the method proposed in this study, we performed an experiment to evaluate the performance of our method in the estimation of the head poses from the images in a video format database. This experiment shows that the performance of our method is quite feasible within the acceptable time frame required for image processing.

The time needed by the algorithm is directly related to the image quality. The minimum and maximum processing times needed in our work were observed to be about 20 ms and 34 ms, respectively, with an average of about 28 ms, a value less than the average required processing time of 30 ms reported in the literature.

We also examined our proposed method in real driving situations with different drivers. The experiments we performed involved a certain vehicular testbed including a monocular camera mounted on the dashboard of the car in front of the driver. Furthermore, we designed a graphic user interface in Matlab to see the functionality of the model simultaneously.

It must be mentioned that a fair evaluation of the results in a real-world application requires certain devices such as calibrated cameras or accurate simulators, as indeed done elsewhere. In addition, as stated by some authors, certain notion of the ground truth for the driver’s head pose is ideally needed for the validation of the system. Without a thorough and detailed understanding and knowledge of the true intentions of a driver in a driving situation, it is quite difficult to obtain an actual ground truth. In the absence of such a sound ground truth, we have made use of some actual driving data with a scaled frame applied on the program graphical user interface (GUI) for several hundred frames of the video sequences. This procedure, in effect, resulted in a ground setting which could be used as our ground truth in our ‘real-world’ driving experiment.

Figure 10 shows some of the drivers who participated in the experiment to examine the ‘real-world’ applicability of our proposed method.

Figure 11 shows the real-time execution of the computer program written based on the proposed algorithm for the estimation of the head pose.
Conclusion

In this work, an effective algorithm for real-time head pose estimation based on a novel feature space and a hierarchical classification technique was proposed. This method is useful for driver assistance systems and video-based surveillance systems. While the real-time speed is a critical requirement in such applications, it was shown that our model performs fast with high accuracy. The main focus of this article is whether the driver is paying adequate attention to the driving, so that in the event he is not, some vocal warning can be issued to alert the driver. When the head direction is deviated for too long, the proposed algorithm provides a judgment to this effect.

There are a number of advantages associated with the proposed technique. First, the method of feature extraction in the image domain is independent of identity, keeping only the features related to the angles. Furthermore, the technique can handle cases in which the feature extraction method does not require the appearance of facial components on the sequences so clearly. Thus, our algorithm can detect those incomplete faces resulted from great occlusions and large orientations. Second, the hierarchical technique offers a stable performance for all orientations in the head pose space, requiring only one camera for each position, in contrast with those systems requiring more cameras with certain locations for the driver head pose estimation. In essence, the proposed methodology can be considered as a part of an advanced driver assistant system. It is also useful for collision prevention systems, severely handicapped people, and other driving applications.

The main restriction in using the proposed approach is the high similarity between head positions in each of the eight regions of our zoning procedure, an area open to further study in our future work. In addition we plan to add some means of verification to be used in security systems.

Acknowledgements

The authors wish to express their thanks to ICPR’04 and the Face and Gesture Recognition Working group for providing the Pointing04 dataset which is used in this research.

References