# 3D Head Pose Estimation in Monocular Video Sequences Using Deformable Surfaces and Radial Basis Functions

Michail Krinidis[†], Nikos Nikolaidis[†] and Ioannis Pitas[†]

*Abstract*— **This paper presents a novel approach for estimating $3D$ head pose in single-view video sequences. Following initialization by a face detector, a tracking technique that utilizes a $3D$ deformable surface model to approximate the facial image intensity is used to track the face in the video sequence. Head pose estimation is performed by using a feature vector which is a byproduct of the equations that govern the deformation of the surface model used in the tracking. The afore-mentioned vector is used as input in a Radial Basis Function (RBF) interpolation network in order to estimate the $3D$ head pose. The proposed method was applied to IDIAP head pose estimation database. The obtained results show that the method can estimate the head direction vector with very good accuracy.**

*Index Terms*— **Head pose estimation, $3D$ deformable models, Radial Basis Function Interpolation.**

1

## I. Introduction

Head pose estimation in video sequences is a frequently encountered task in many computer vision applications. In video surveillance [1], head pose combined with prior knowledge about the world, enables the analysis of person motion and intentions. Head pose is also indicative for the focus of attention of people, a fact that is very important in human-computer interaction [2]. Head pose can moreover be used in navigation of $3D$ games [3], in visual communications [4], $3D$ face reconstruction [5], etc. Head pose estimation is also used as a preprocessing step in face detection [6], face recognition [7] and facial expression analysis [8], since these tasks are very sensitive to even minor head rotations. Thus, the exact knowledge of the face pose is an essential problem which can boost the performance of such applications. A number of head pose estimation algorithms [9], [10] operate on stereoscopic sequences. However, stereoscopic information might not be available in the above-mentioned applications. As a result, research on single-view head pose estimation has been on the rise during the last years.

The basic challenge in head pose estimation from single-view videos is to derive fast algorithms that do not require extensive preprocessing of the video sequence. Low-resolution images, image clutter, partial occlusions, unconstrained motion, varying illumination conditions and complex background can mislead the head pose estimation procedure. Depending on the way the face is treated, existing methods can be broadly divided in three categories:

- approaches based on facial features,
- model-based algorithms,
- appearance-based algorithms.

A comparison of existing head pose estimation algorithms is given in [11], [12], [13].

The use of the spatial arrangement of important facial features for face pose estimation has been investigated by many researchers [14], [15], [16], [17]. In these approaches, the $3D$ face structure is exploited along with *a priori* anthropometric information in order to define the head pose. The elliptic shape of the face and the ratio of the major and minor axes of this ellipse, the mouth-nose region geometry, the line connecting the eye centers, the line connecting the mouth corners and the face symmetry are some of the geometric features used to estimate the $3D$ head pose. In [17], five facial features, i.e., the eye centers, the mouth centers and the nose are localized within the detected face region. A weighting strategy is applied after the detection of the facial features, so as to estimate the final location of the five components more accurately. The face pose is estimated by exploiting a metric which is based on comparing the location of the acquired facial features with the corresponding locations on a frontal pose. In [14], the location of facial feature points is combined with color information in order to estimate the $3D$ face pose. The skin and the hair region of the face is extracted, based on a perceptually uniform color system. Facial feature detection is performed on the face region and the bounding boxes of eyes, eyebrows, mouth and nose are defined. Then, corner detection is applied on these bounding boxes. The left-most and the right-most corner are selected as the feature points of each facial feature. The $3D$ head pose is inferred from both the facial features and the skin and hair region of the face. This category of algorithms has a major disadvantage: their performance depends on the successful detection of facial features which remains a difficult problem, especially in non-frontal faces.

In the last few years many efforts have been spent on model-based head pose estimation algorithms [18], [19], [20]. The basic idea in this category of methods is to use an *a priori* known $3D$ face model which is mapped onto the $2D$

[†]Aristotle University of Thessaloniki, Department of Informatics, Box 451, 54124 Thessaloniki, Greece, *email: {mkrinidi, nikolaid, pitas}@ aiia.csd.auth.gr* Fax/Tel ++ 30 231 099 63 04, http://www.aiia.csd.auth.gr/.

images. Once $2D$-$3D$ correspondences are found between the input data and the face model, conventional pose estimation techniques are exploited to provide the $3D$ face pose. The main problem in these algorithms is to find in a robust way characteristic facial features that can be used to define the best mapping of the $3D$ model to the $2D$ face images. In [20], the $3D$ model is a textured triangular mesh. The similarity between the rendered (projected) model and the input facial image is evaluated through an appropriate metric. The pose of the model that gives the best match is the estimated head pose. In [19], a cubic explicit polynomial in $3D$ is used to morph a generic face into the specific face structure using as input multiple views. The estimation of the head structure and pose is achieved through the iterative minimization of a metric based on the distance map (constructed by using a vector-valued Euclidean distance function).

Appearance-based approaches [21], [22], [23], achieve satisfactory results even with low-resolution head images. In these approaches, instead of using facial landmarks or face models, the whole image of the face is used for pose classification. In [23], a neural network-based approach for $2D$ head pose estimation from low-resolution facial images captured by a panoramic camera is presented. A multi-layer perceptron is trained for each pose angle (pan and tilt) by feeding it with preprocessed facial images derived from a face detection algorithm. The preprocessing consists of either histogram normalization or edge detection. In [12], [21], an algorithm that couples head tracking and pose estimation in a mixed state particle filter framework is introduced. The method relies on a Bayesian formulation and the goal is to estimate the pose by learning discrete head poses from training sets. Texture and color features of the face regions were used as input to the particle filter. Two different variants were tested in [12]. The first tracks the head and then estimates the head pose and the second jointly tracks the head and estimates the $3D$ head pose. Support Vector Regression (SVR) [24], i.e., Support Vector Machines where the output domain contains continuous real values has been also used in appearance-based approaches. In [25], [26], two Sobel operators (horizontal and vertical) were used to preprocess the training images and the two filtered images were combined together. Principal Component Analysis (PCA) is then performed on the filtered image in order to reduce the dimensionality of the training examples (facial images of known pose angles). SVM regression was utilized in order to construct two pose estimators, for the tilt and yaw angles. The input to SVM was the PCA vectors and the output was the estimated face angle. Since the final aim of the paper was multi-view face detection, these angles were subsequently used for choosing the appropriate face detector among a set of detectors designed to operate on a different view angle interval. In [27], SVR is used to estimate head pose from range images. A three-level discrete wavelet transform is applied on all the training range images and the LL sub-band (which accentuates pose-specific details, suppresses individual facial details, and is relatively invariant to facial expressions) is used as input to two support vector machines that are trained using labelled examples to estimate the tilt and yaw angles.

The single-view $3D$ head pose estimation approach pro-posed in this paper belongs to the appearance-based methods. The method utilizes the deformable intensity surface approach proposed in [28], [29] for image matching. According to his approach, an image is represented as a $3D$ surface in the so-called XYI-space by combining its spatial (XY) and intensity (I) components. A deformable surface model, whose deformation equation is solved through modal analysis, is subsequently used to approximate this surface. Modal analysis is a standard engineering technique that has been introduced in the field of computer vision and image analysis in [30]. Modal analysis allows effective computations and provides closed form solutions of the deformation process and has been used in a variety of different applications for solving model deformations, i.e. for analyzing non-rigid object motion [31], for the alignment of serially acquired slices [32], for multimodal brain image analysis [33], segmentation of $2D$ objects [34], image compression [35] and $2D$ object tracking [36].

In our case, such a deformable intensity surface is used to approximate, in the XYI-space, image regions depicting faces. The generalized displacement vector, which is an intermediate step of the deformation process, is subsequently used in a novel way i.e. for both tracking the head and estimating its $3D$ pose in monocular video sequences. Similarly to [36], the tracking procedure is based on measuring and matching from frame to frame the generalized displacement vector of a deformable model placed on the face. The generalized displacement vector is also used to train three RBF interpolation networks into estimating the pan, tilt and roll angles of the head, with respect to the camera image plane. The tilt and the pan angles represent the vertical and the horizontal inclination of the face, whereas the roll angle represents the rotation of the head on the image plane (Figure 1). The proposed algorithm was tested on the IDIAP head pose database [12] which consists of video sequences that were acquired in natural environments and contain large rotations of the face. The database includes head pose ground truth information. The results show that the proposed algorithm can estimate the $3D$ orientation vector of the face with an average error of 4 degrees.
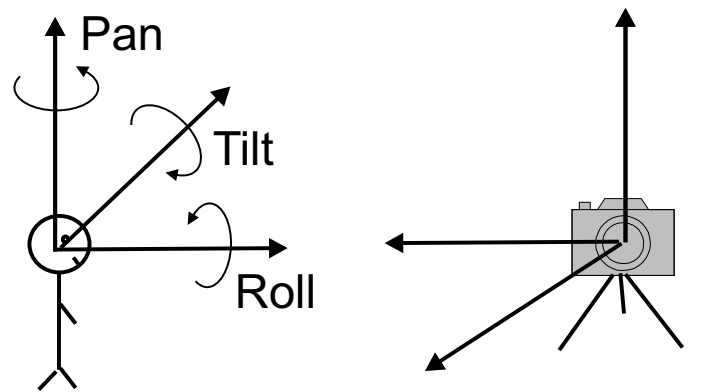


Fig. 1. *Pan, tilt and roll head pose angles.*

The remainder of the paper is organized as follows. In

Section II, a brief description of the deformation procedure in the XYI-space is presented. The tracking algorithm and the derivation of the feature vector used for pose estimation are introduced in Section III. In Section IV, Radial Basis Function interpolation is reviewed and its use for pose estimation is explained. The performance of the proposed technique is studied in Section V. Finally, conclusions are drawn in Section VI.

## II. A FACIAL IMAGE DEFORMABLE MODEL

In this section, the physics-based deformable surface model that is used along with modal analysis to approximate image regions depicting faces in the XYI-space will be briefly reviewed. As already mentioned in Section I this approach has been introduced in [28], [29] and has been used in our case with small modifications, described in this Section. The novelty of our approach lies in the utilization of the so-called generalized displacement vector, involved in the modal analysis, for tracking the face and estimating the pose angles, as will be described in Section III.

According to [28], [29] an image can be represented as an intensity surface $(x, y, I(x, y))$ by combining its intensity $I(x, y)$ and spatial $(x, y)$ components (Figure 2). The corresponding space is called the XYI space and a deformable mesh model is used to approximate this surface. Modal analysis [30] is used to solve the deformation equations.
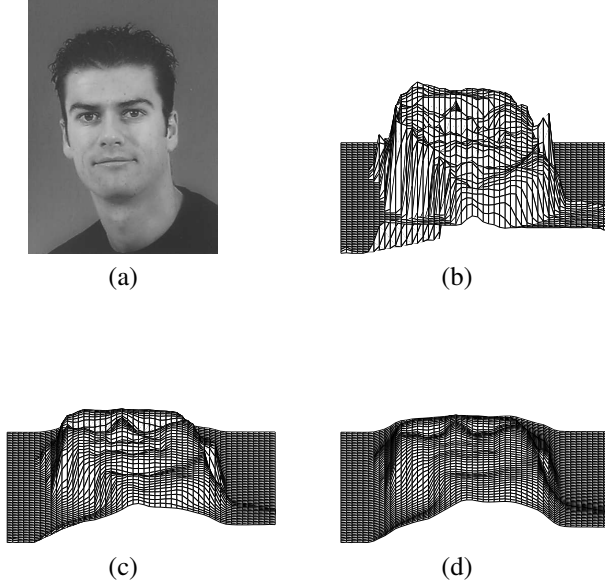


Fig. 2. *(a) Facial image, (b) intensity surface representation of the image, (c) deformed model approximating the intensity surface, (d) deformed model approximating the intensity surface (only 25 % of the coefficients were used in the deformation procedure).*

The deformable surface model consists of a uniform quadrilateral mesh of $N = N_h \times N_w$ nodes, as illustrated in Figure 3. In this section, we assume that $N_h$, $N_w$ are equal to the image region height and width (in pixels) respectively, so that each image pixel corresponds to one mesh node. Each node is assumed to have a mass $m$ and is connected to its neighbors

with perfect identical springs of stiffness $k$ having natural length $l_0$ and damping coefficient $c$. Under the influence of internal and external forces, the mass-spring system deforms to a $3D$ mesh representation of the image intensity surface, as can be seen in Figure 2c.
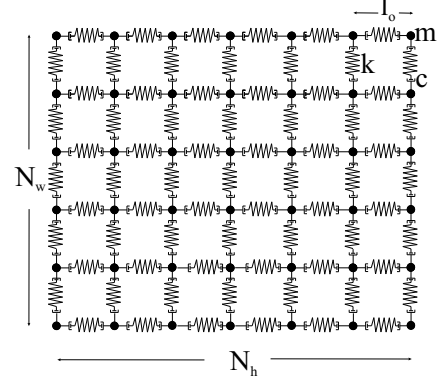


Fig. 3. *Quadrilateral surface (mesh) model.*

In our case, the initial and the final deformable surface states are known. The initial state is the initial (planar) model configuration and the final state is the image intensity surface, shown in Figure 2b. Therefore, it can be assumed that a constant force load $\mathbf{f}$ is applied to the surface model [33]. Since we are not interested in the deformation dynamics, we can deal with the static problem formulation:

$$\mathbf{Ku} = \mathbf{f}. \tag{1}$$

where $\mathbf{K}$ is the $N \times N$ stiffness matrix, $\mathbf{f} = [\mathbf{f}_1, \ldots, \mathbf{f}_N]^T$ is the $N \times 3$ vector whose elements are the $N$ 3D external force vectors applied to the model and $\mathbf{u}$ is the $N \times 3$ nodal displacements vector given by:

$$\mathbf{u} = [\mathbf{u}_1, \ldots, \mathbf{u}_i, \ldots, \mathbf{u}_N]^T, \tag{2}$$

where $\mathbf{u}_i = [u_{i,x}, u_{i,y}, u_{i,z}]$ is the displacement of the $i$-th node.

Instead of finding directly the equilibrium solution of (1), one can transform it by a basis change [30]:

$$\mathbf{u} = \mathbf{\Phi}\tilde{\mathbf{u}} = \sum_{i=1}^{N=N_h N_w} \phi_i \tilde{\mathbf{u}}_i, \tag{3}$$

where $\tilde{\mathbf{u}}$ is referred to as the *generalized displacement* vector, $\tilde{\mathbf{u}}_i$ is the $i$-th component of $\tilde{\mathbf{u}}$ and $\mathbf{\Phi}$ is a matrix of order $N$, whose columns are the eigenvectors $\phi_i$ of the generalized eigenproblem:

$$\mathbf{K}\phi_i = \omega_i^2 \mathbf{M}\phi_i, \tag{4}$$

where $\mathbf{M}$ is the mass matrix of the model. The $i$-th eigenvector $\phi_i$, i.e., the $i$-th column of $\mathbf{\Phi}$ is also called the $i$-th *vibration mode* and $\omega_i$ is the corresponding eigenvalue (also called *vibration frequency*). Equation (3) is known as *modal superposition equation*.

In practice, we wish to approximate nodal displacements $\mathbf{u}$ by $\hat{\mathbf{u}}$, which is the truncated sum of the $N'$ low-frequency vibration modes, where $N' < N$:

$$\mathbf{u} \approx \hat{\mathbf{u}} = \sum_{i=1}^{N'} \phi_i \tilde{\mathbf{u}}_i. \tag{5}$$

The eigenvectors $\phi_i$, $i = 1, \ldots, N'$, form the *reduced modal basis* of the system. This is the major advantage of modal analysis: it is solved in a subspace corresponding to the $N'$ truncated low-frequency vibration modes of the deformable structure [31], [33], [30]. The number of vibration modes $N'$ retained in the surface description is chosen so as to obtain a compact but adequately accurate deformable surface representation. A typical *a priori* value for $N'$, covering many types of standard deformations, is equal to one quarter of the total number of the vibration modes.

A significant advantage of this formulation, in the full as well as in the truncated modal space, is that the vibration modes (eigenvectors) $\phi_i$ and the frequencies (eigenvalues) $\omega_i$ of a plane topology have an explicit formulation [31] and they do not have to be computed using eigen-decomposition techniques:

$$\omega^2(j, j') = \frac{4k}{m} \left[ \sin^2 \left( \frac{\pi j}{2N_h} \right) + \sin^2 \left( \frac{\pi j'}{2N_w} \right) \right], \tag{6}$$

$$\phi_{n,n'}(j, j') = \cos \frac{\pi j (2n - 1)}{N_h} \cos \frac{\pi j' (2n' - 1)}{N_w}, \tag{7}$$

where $j = 0, 1, \ldots, N_h - 1$, $j' = 0, 1, \ldots, N_w - 1$, $n = 1, 2, \ldots, N_h$, $n' = 1, 2, \ldots, N_w$, $\omega^2(j, j') = \omega_{j\,N_w + j'}^2$, $\phi_{n,n'}(j, j')$ is the $(n, n')$-th element of matrix $\phi(j, j')$, where $\phi(j, j') = \phi_{j\,N_w + j'}$.

In the modal space, (1) can be written as:

$$\tilde{\mathbf{K}} \tilde{\mathbf{u}} = \tilde{\mathbf{f}}, \tag{8}$$

where $\tilde{\mathbf{K}} = \mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi}$ and $\tilde{\mathbf{f}} = \mathbf{\Phi}^T \mathbf{f}$, $\mathbf{f}$ being the external force vector. Hence, by using (3), (6) and (7), equation (8) is simplified to $3N$ scalar equations:

$$\omega_i^2 \tilde{u}_{i,j} = \tilde{f}_{i,j}, \tag{9}$$

where $j = x, y, z$ and $\tilde{u}_{i,j}$ is the $j$-th component of the $i$-th vector of $\tilde{\mathbf{u}}$.

In our case, the components of the forces in $\mathbf{f}$ along the $x$ and $y$ axes are taken to be equal to zero, i.e. $f_{i,x} = f_{i,y} = 0$. On the other hand, the components of these forces along the $z$ (intensity) axis are taken to be proportional to the Euclidean distance between the point $(x, y, I(x, y))$ of the intensity surface and the corresponding model node position in its initial configuration $(x, y, 0)$, i.e., equal to the intensity $I(x, y)$ of pixel $(x, y)$: $f_{(x-1)N_w + y, z} = f(x, y) = I(x, y)$, where $f_{(x-1)N_w + y, z}$ is the component along the $z$ axis of the $(x - 1)N_w + y$-th element $\mathbf{f}_{(x-1)N_w + y}$ of vector $\mathbf{f}$. Moreover, the model is not allowed to deform along the x and y axes i.e. $\tilde{u}_{i,x} = \tilde{u}_{i,y} = 0$. Hence, the $N \times 3$ generalized displacement vector can be simplified by ignoring these components and constructing a $N$-dimensional vector that contains only the $z$ components: $\tilde{\mathbf{u}} = [\tilde{u}_1, \ldots, \tilde{u}_i, \ldots, \tilde{u}_{N_h N_w}]^T =$ $[\tilde{u}_{1,z}, \ldots, \tilde{u}_{i,z}, \ldots, \tilde{u}_{N_h N_w, z}]^T$. By using (3), (6), (7) and (9) along with the force values mentioned above, one can explicitly compute $\tilde{u}_k$ as follows:

$$\tilde{u}_{(i-1)N_w + j} = \frac{\sum_{n=1}^{N_h} \sum_{n'=1}^{N_w} I(n, n') \phi_{n,n'}(i, j)}{(1 + \omega^2(i, j)) \sqrt{\sum_{n=1}^{N_h} \sum_{n'=1}^{N_w} \phi_{n,n'}^2(i, j)}}. \tag{10}$$

It should be noted that the deformable model achieves only an approximation of the intensity surface of the target image.

For the problem at hand, facial areas of the video sequence are described in terms of the vibrations of an initial model. Figure 2 illustrates the approximation by a deformable surface model (Figures 2b,c) of the intensity surface (Figure 2b) of the $2D$ image of a facial image shown in Figure 2a. The size of the model (in nodes) that was used to parameterize the image surface was equal to the image size (in pixels).

The generalized displacement vector $\tilde{\mathbf{u}}$ of equations (8), (10) is exploited, as will be shown in the following sections, in order to track facial regions on $2D$ images and estimate the $3D$ head pose. A flow diagram of the proposed algorithm is shown in Figure 4. The details of the algorithm will be provided in the following sections.

## III. FACE TRACKING AND DERIVATION OF THE POSE FEATURE VECTOR

A real-time frontal face detection algorithm [37] is applied on the first image of the video sequence in order to initialize the face tracking and pose estimation procedure. The face detection scheme is based on simple features that are reminiscent of Haar basis functions [38]. These features were extended in [37] to further reduce the number of false alarms. The output of the face detection procedure is a window around the face center, i.e., around the nose, that tightly encloses the face area.

Subsequently, a region based tracking approach similar to the one proposed in [36] is used to track the central face region. The main difference between the tracking algorithm used here and the one introduced in [36], is that the latter aims at tracking feature points (by utilizing information from the surrounding region), while the former is adapted to track regions. Additional information for the tracking algorithm along with numerous experimental results can be found in [36]. The following assumptions are adopted by this algorithm:

- The face window (bounding box) is of constant size, i.e., the person does not move significantly towards or away from the camera. This is a realistic assumption for most applications that require head pose estimation (e.g. human-computer interaction, face recognition, gaze estimation, etc.).
- A part of the human face is always visible (images depicting the back of the head are not handled) and no occlusions occur. Both these assumptions are also realistic for most applications. If needed the tracking algorithm can be enhanced with occlusion handling mechanisms.

Region-based tracking is performed by applying the deformable model described in the previous section on a small window $W$ (e.g. one of dimensions $20 \times 20$ pixels) around
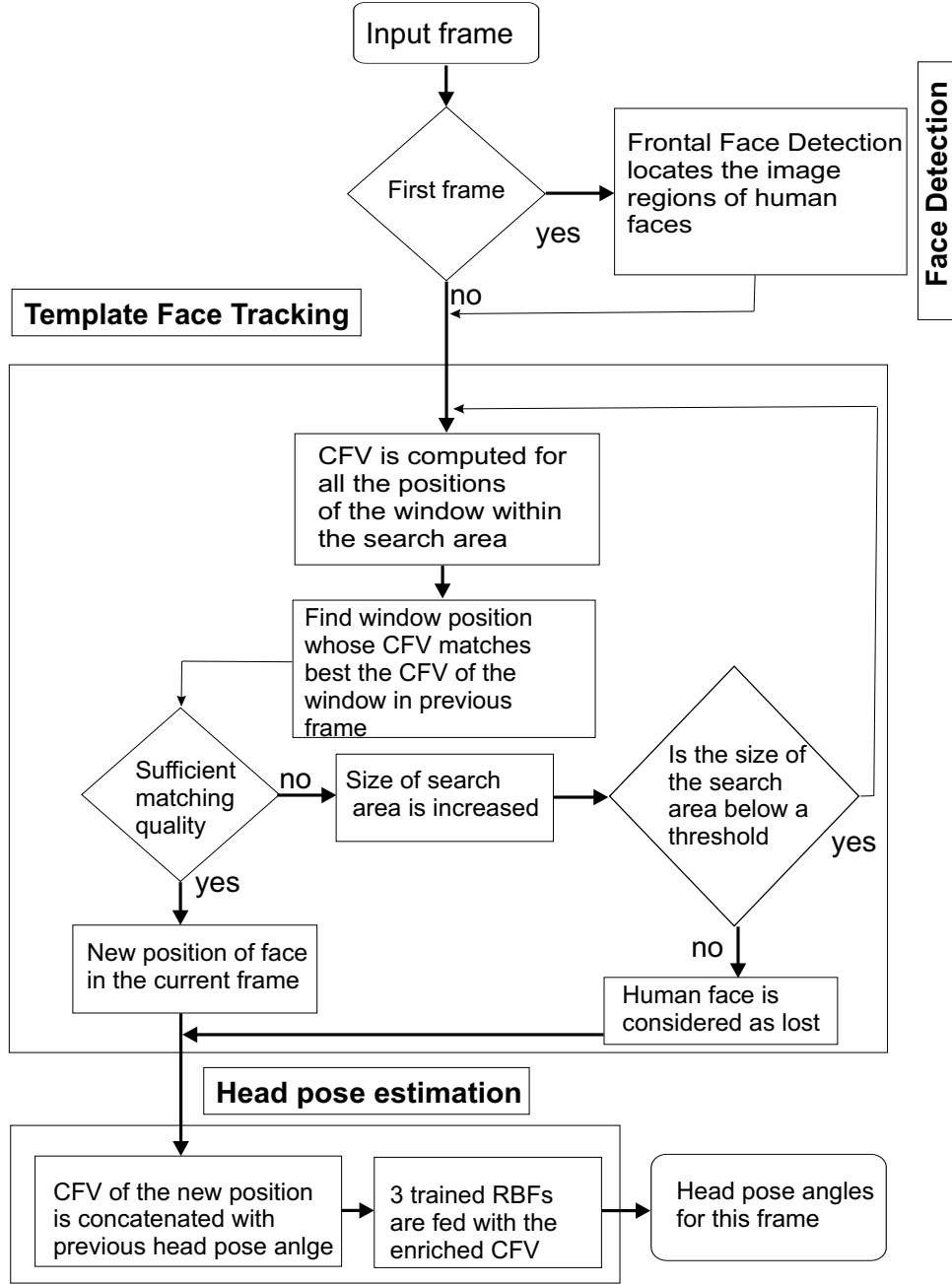
Fig. 4. *Flow diagram of the proposed head pose estimation algorithm.*

the face center $\mathbf{p}^t = (x, y)$ and evaluating the generalized displacement vector $\tilde{\mathbf{u}}^t(x, y)$ of equation (8) for this window:

$$\tilde{\mathbf{u}}^t(x, y) = [\tilde{u}_1^t(x, y), \tilde{u}_2^t(x, y), \ldots, \tilde{u}_{N_H N_W}^t(x, y)]^T, \quad (11)$$

where $t$ is the time instance and $N_H$, $N_W$ are the height and width of the deformable surface model (equal to the dimensions of the window). We will call vector $\tilde{\mathbf{u}}^t(x, y)$ the *characteristic feature vector* (CFV). It has been shown [36] that this vector is the combination of the output of various line and edge detection masks applied on the tracking window. As a result, tracking by matching the CFV along images is sufficiently robust to illumination changes.

In order to find the position $\mathbf{p}^{t+1} = (x', y')$ of the center

of the $N_H \times N_W$ face window $W$ in the next frame $I_{t+1}$, the algorithm computes the CFV $\tilde{\mathbf{u}}^{t+1}(k, l)$ for all windows whose center $(k, l)$ lie within a search region $R$ with height $N_{H\,reg}$ and width $N_{W\,reg}$, centered at coordinates $(x, y)$ in image $I_{t+1}$. The new location of the center of $W$ is found as the location $(x', y')$ in the search region whose CFV is closer to that of $\mathbf{p}^t$ in the current frame. More specifically:

$$\mathbf{p}^{t+1} = (x', y') \longrightarrow \arg\min_{kl}(\|\tilde{\mathbf{u}}^{t+1}(k, l) - \tilde{\mathbf{u}}^t(x, y)\|), \quad (12)$$

where $k \in \{x - \frac{N_{H\,reg} - 1}{2}, \ldots, x, \ldots, x + \frac{N_{H\,reg} - 1}{2}\}$ and $l \in \{y - \frac{N_{W\,reg} - 1}{2}, \ldots, y, \ldots, y + \frac{N_{W\,reg} - 1}{2}\}$ and $\|\cdot\|$ denotes the Euclidean distance.

The choice of the Euclidean distance was based on a set of experiments which aimed at providing results for the performance of the proposed tracking algorithm when different measures are used in order to select the next position of the face. The Euclidean distance was found to perform better than both the normalized correlation and $|S_{x,y}^t - S_{k,l}^{t+1}|$, where $S_{x,y}^t$ is given by:

$$S_{x,y}^t = \sum_{i=1}^{N_H N_W} \left| \tilde{u}_i^t(x,y) \right|. \tag{13}$$

The experiment is described in detail in Section V.

Since the motion characteristics of the face to be tracked might change over time, i.e. the face can speed up or slow down at certain video frames, the algorithm uses a search region $R$ of variable size. For each video frame, the algorithm tries to locate the new center of the face window $W$ using initially a small search region (e.g. $7 \times 7$). However, if for the best candidate position the Euclidean distance (equation (12)) is above a certain threshold, the algorithm increases the search region size, trying to find a better match (a match corresponding to a matching error below the threshold) in the larger search area. If this is again not feasible, the size increase continues up to a certain maximum region size ($R = 23$). Some tracking results on the IDIAP database are presented in Figure 5.

In addition to its use for tracking, the CFV of the face window $W$ is used for deriving the head pose. The CFV contains information about the central visible face region, i.e. the region around the nose in the frontal images or the cheek in profile images (Figure 5), since its elements are related to the displacements of the deformable surface model which approximates the intensity surface in this area. As the face/head changes orientation in the $3D$ space, its projection on the image ($2D$ space) changes. Thus, the fixed-size window $W$ includes the central part of the face in different perspective views (Figure 5). Hence, this information can be used to derive the orientation of the face. The characteristics of the deformable surface model used in the experimental setup, were set so that the model is a rigid one. Thus, the final state of the deformable surface was a smoothed version of the face intensity surface, in order to be insensitive to clutter, differences between faces of persons and varying illumination conditions. By utilizing the truncated space of the modal analysis, one can reduce the size of the CFV feature length to $25\%$ of its original size (in our case to 100 elements, down from $20 \times 20 = 400$ elements), without losing significant information. The information contained in the CFV was used along with appropriately trained RBF interpolation networks to derive pose information, as will be described in the next Section.

## IV. RADIAL BASIS FUNCTION INTERPOLATION

The design of an interpolation system can be seen as a surface fitting problem in a high-dimensional space [39]. In such a situation, learning is equivalent to finding a smooth surface, which interpolates (or approximates) the training data.

Radial Basis Functions have been used in our case for this purpose. RBFs were chosen for the following reasons:
- they have a simple structure,
- they have the property of "best approximation",
- they have the property of best learning and reduced sensitivity to the order of presentation of training data.

Let us assume a set of $N$ $K$-dimensional vectors $\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \ldots & \mathbf{x}_N \end{bmatrix}$, $\mathbf{x}_i \in \mathbb{R}^K$ and a set of scalar values $\begin{bmatrix} l_1 & l_2 \ldots & l_N \end{bmatrix}$, $l_i \in \mathbb{R}$, which correspond to samples of an unknown function $f : \mathbb{R}^K \to \mathbb{R}$, i.e. $f(\mathbf{x}_i) = l_i$. The RBF interpolation solves the problem of finding a smooth function $\hat{f}(\mathbf{x})$ that satisfies the following relation:

$$\hat{f}(\mathbf{x}_i) = l_i, \quad i = 1, \ldots, N, \tag{14}$$

i.e. interpolates $(\mathbf{x}_i, l_i)$ and hopefully approximates sufficiently well $f(\mathbf{x})$ elsewhere.

Radial basis function interpolation is defined as:

$$f(\mathbf{x}) = \sum_{1}^{N} w_i R_i(\mathbf{x}), \tag{15}$$

where $R_i$ are the chosen radial basis functions and $w_i$ are linear weights used to combine the RBF.

In our case, isotropic Gaussian radial basis functions were used:

$$R(\|\mathbf{x} - \mathbf{u}\|^2) = \exp(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{u}\|^2\|). \tag{16}$$

The parameters to be learnt in such an RBF interpolation are the weights $w_i$ in (15), the means $\mathbf{u}^i$ and the standard deviations $\sigma_i$ of the RBF $R_i$. Many approaches for training an RBF interpolation network have been proposed [39]. According to one of them, the vectors $\mathbf{u}^i$ can be chosen to be equal to the vectors $\mathbf{x}_i$, i.e. $\mathbf{u}^i = \mathbf{x}_i$, so that each RBF is centered at one training sample $\mathbf{x}_i$. The standard deviation for all the Gaussian basis functions can be set equal to [39]:

$$\sigma = \frac{d}{\sqrt{2N}}, \tag{17}$$

where $N$ is the number of the training data and $d$ is the maximum Euclidean distance between any two training samples $\mathbf{x}_i$, $\mathbf{x}_j$. This choice of $\sigma$ ensures that the RBFs are neither too flat nor too peaky and that they will "fill" the space sufficiently well. Once the RBF have been fully defined, a straightforward procedure for solving for the weights $w_i$ in (15), so that the network satisfies (14) is to use the *pseudoinverse method* [40] on the following linear system:

$$\mathbf{R}\mathbf{w} = \mathbf{l} \tag{18}$$

where the element $r_{ji}$ of $\mathbf{R}$ is equal to $R_i(\|\mathbf{x}_j - \mathbf{x}_i\|^2)$, vectors $\mathbf{w}$ and $\mathbf{l}$ contain the weights $w_i$ and the scalar values $l_i$ respectively and $j, i = 1, 2, \ldots, N$. Obviously, this linear system has the following solution:

$$\mathbf{w} = \mathbf{R}^+\mathbf{l}, \tag{19}$$

where $\mathbf{R}^+$ is the pseudoinverse of $\mathbf{R}$.

Radial basis function interpolation was used in our case, to estimate values for the three pose angles (pan, tilt, roll) of frame $I^t$ by using as input the CFV $\tilde{\mathbf{u}}^t$ of the corresponding
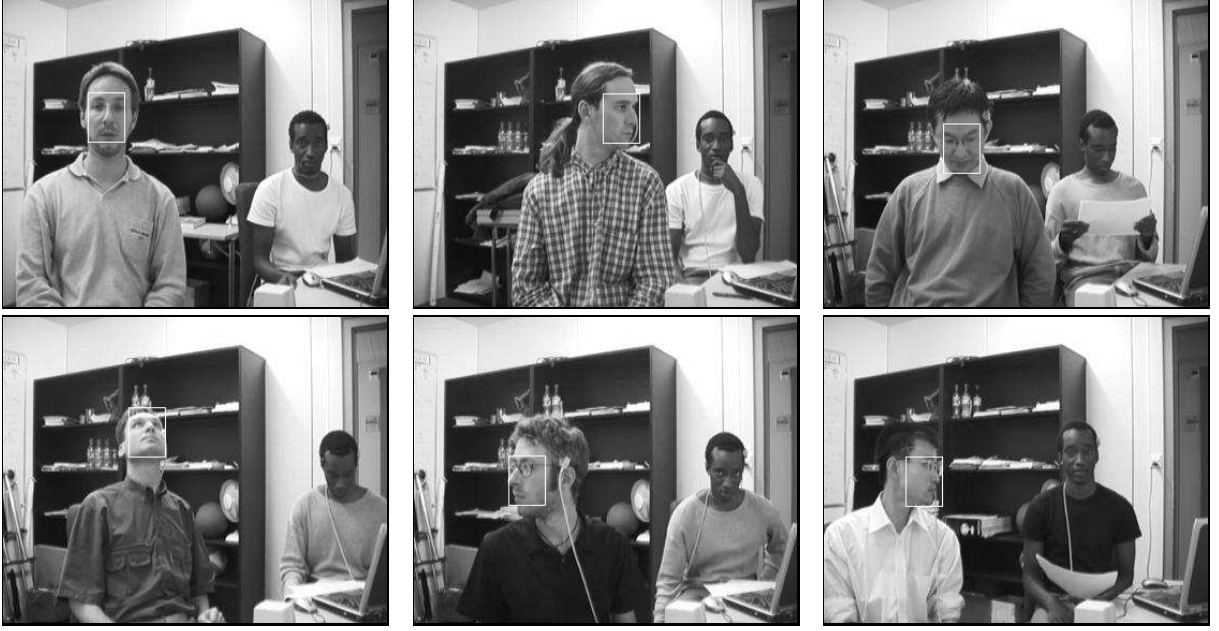
Fig. 5. *Face tracking results obtained in video sequences depicting persons in various poses.*

frame. Thus, the input vector in the training and testing procedure was the CFV of the area over the face that was derived by the tracking algorithm, whereas the output was the pose angle value. More specifically, three RBF networks, each handling a different angle (pan, tilt, roll) of the face pose, were used. The value range of each of these parameters varied from $[-90\dots90]$ for pan, $[-60\dots60]$ for tilt and $[-30\dots30]$ for roll. During the training of the RBF system, i.e. during the evaluation of the mean and variance of the RBFs and the weights $w_i$, a set of CFVs $\tilde{\mathbf{u}}^t$ along with the corresponding pose angle derived from the ground truth, was used as input. This set was a subset of the CFV-pose angle pairs, derived from the training sequences and the ground truth. In accordance to the training procedure described above, the network consisted of as many RBFs as the training samples $\tilde{\mathbf{u}}^t$. In other words, the number of RBFs was equal to the number of frames of the training sequences. To perform testing, an unlabelled feature vector $\mathbf{x}' = \tilde{\mathbf{u}}^{t'}$ is used as an input. The trained RBF system that handles a certain pose angle, interpolates the test data in the surface derived from the training data and estimates the pose angle value.

In order to increase the performance of the system in terms of pose angles estimation accuracy, the input vectors of the RBF interpolation were expanded to consist of a concatenation of the CFV $\tilde{\mathbf{u}}^t$ with the pose angle of the previous frame. More specifically, the RBF interpolation networks were fed at time instant $t$ with vectors of the form:

$$[\tilde{\mathbf{u}}^t | \theta_{t-1}]^T \qquad (20)$$

where $\theta_{t-1}$ denotes the pan, tilt or roll angle in the previous frame. During training the ground truth pose angles were used, whereas during testing, the estimates from the application of the system in the previous frame were inserted. The use of the pose angle of the previous frame into the input vectors introduces the notion of coherence. Indeed, if videos of sufficient frame rate are available and the head movements are smooth, the pose angles in the current frame will not differ significantly from those in the previous frame. Thus, by including this information the system can cope more efficiently with tracking errors. On the other hand, if abrupt head motions occur, the pose angles of the previous frame would be uncorrelated to those of the current frame and this information will affect the result in a negative way. However, such movements are rather infrequent and the user can hope that the tracking will succeed in following the target and thus, will provide correct values for the CFV part of the input vector.

The head pose for the first frame during the testing procedure was assumed to be known. More specifically, it was assumed that on the first frame the person under examination is looking straight ahead with no rotations in any of the three axes, so that all three angles (pan, roll, tilt) are equal to zero. This scenario is not unrealistic, since in many applications the face is in a frontal/neutral pose at the beginning of the video sequence and many algorithms that are applied on facial images adopt the same assumption. Moreover, the face detector used to initialize the tracking and head pose estimation, is a frontal face detector and thus, this assumption is necessary for its proper operation.

## V. Performance Evaluation

### A. Evaluation Setup and Dataset

Experiments were conducted to evaluate the performance of the proposed algorithm in video sequences recorded under realistic conditions.

The aim of the first set of experiments was two-fold: to evaluate the performance of the tracking part of the algorithm and find a suitable metric for measuring the similarity of

*Precision and recall of the region tracking algorithm when using Euclidean distance, normalized correlation and absolute difference of $S_{x,y}^t$ (13) in order to match CFVs between consecutive frames.*

| Metric | P | R |
|---|---|---|
| Euclidean distance | **96.9** | **95.2** |
| Normalized correlation | 89.7 | 85.3 |
| $|S_{x,y}^t - S_{k,l}^{t+1}|$ | 96.1 | 94.6 |

CFVs. The performance of the algorithm was evaluated at the object level, i.e., on the basis of whether the entire object under examination was correctly tracked or not. True positives ($TP$), false positives ($FP$) and false negatives ($FN$) were obtained using manually extracted ground truth data. Cases where the bounding box of the tracked face region contained more than 70% of the face, were considered as $TP$. When more than 30% of the bounding box consisted of background area, the frame was considered as contributing to $FP$. Situations where the tracking algorithm loses the target (i.e., it tracked the background), were considered as contributing to $FN$. Based on these numbers, the well-known precision ($P = \frac{TP}{TP+FP}$) and recall measures ($R = \frac{TP}{TP+FN}$) were calculated. The algorithm was tested in 5 indoor and outdoor video sequences (5600 frames) with motions frequently encountered in tracking situations, such as translation and rotation. The studio (indoor) sequences depict one person moving on a predefined or random trajectory, under either optimal (uniform lighting) or suboptimal (hard shadows and bright/dark areas) lighting conditions created using the studio's lighting equipment. The outdoor video sequences depict one person moving on a random trajectory under realistic conditions. The results for the 3 different metrics, namely the Euclidean distance, the normalized correlation and the absolute difference of $S_{x,y}^t$ in (13) are summarized in Table I. One can see that the Euclidean distance achieves the best performance. In addition, these experimental results provide quantitative evidence regarding the satisfactory performance of the tracking algorithm. Since the tracking algorithm is not the focus of this paper and the algorithm used here resembles the one proposed in [36], no additional experiments were performed towards this direction. Readers are encouraged to consult [36] for more tracking experiments with various types of motion and lighting conditions.

In the second set of experiments, the proposed method was used to estimate the $3D$ head pose on parts of the IDIAP video database [12]. This database was selected because it contains ground truth data and because the video sequences were recorded under realistic conditions and depict realistic situations. Each of the 23 video clips lasts from 4 to 10 minutes and has a resolution of $360 \times 288$ pixels, 25 frames per second. There are two parts of the database. The one depicts an office environment and the other a meeting environment. The main scenario of the IDIAP database is that subjects act as in their daily life in their office or in a meeting. In total, 16 different subjects participate in the video database. The database contains head pose ground truth in the form of pan, tilt and roll angles (i.e. Euler angles with respect to the camera coordinate system) for each frame of the video sequences.

Eleven of the video sequences were used for training the RBF interpolation network and the rest for testing the system. In total, 9500 frames were used to train the system. Thus, an equal amount of RBFs were utilized. For each RBF, one needs to store its center (in our case a 100-dimensional vector for the input and one value for the output) the standard deviation (same for all RBFs) and the corresponding weight. The parameters of the deformable surface model were defined so as to give a smooth representation of the face intensity surface, i.e. a ratio $\frac{k}{m} = 10$ ($k$ being the stiffness of the springs and $m$ the mass of the nodes) was used. Thus, the final state of the deformable surface was a smoothed version of the face intensity surface, in order to be insensitive to clutter, differences between faces of persons and varying lighting conditions.

One minute, i.e. 1500 frames, were selected from each of the 12 test video sequences. The selection of the part of each video sequence that was used in the experiments was not random. We tried to choose 1500 frames depicting rich person activity, covering as many head pose angles as possible. Face detection was performed in the first frame of the selected part of each video sequence and the tracking technique described in Section III was applied to the rest of the frames in order to track the position of the face. For each frame, the CFV was calculated in order to be further used either for training or testing the RBF network. The tracking results showed that the proposed tracking algorithm offers satisfying results. However, in some frames the tracking algorithm can lose part of the target, i.e. the face rectangle might contain part of the background. This can happen either when the movement of the face is abrupt and extreme or in instances that do not involve such movements. In the first case, the head pose of the previous frame does not help in the estimation of the pose angle in the current frame and the algorithm might provide estimates with significant error (e.g. $8^o$). In the second case, the head pose of the previous frame can improve the estimation in the current frame. In both cases, the tracking algorithm usually recovers within a few frames.

The metrics used for the evaluation of the proposed algorithm was the error (absolute difference) in degrees between the ground truth and the estimated value for pan (PE), tilt (TE) and roll (RE) angles. Moreover, the head pose defines a vector in the $3D$ space which indicates where the head is pointing at. The angle (DE) between the pointing vector defined by the head pose ground truth and the pose estimated by the proposed system was used as a pose estimation error measure, as proposed in [12]. DE is given by the following equation:

$$DE = \frac{180}{\pi} \ acos \sum_{i=1}^{3} (v_g(i) \ * \ v_e(i)), \qquad (21)$$

where $acos$ is the inverse cosine, $v_g(i)$ and $v_e(i)$ are the $i$-th components of the pointing vectors $\mathbf{v}_g$ and $\mathbf{v}_e$ respectively, which are constructed from the ground truth data and the estimated pose respectively, as follows [12]:

$$\mathbf{v}_q = [\sin(\theta_p), -\sin(\theta_t) * \cos(\theta_p), \cos(\theta_t) * \cos(\theta_p)], \quad (22)$$

where $q = \{g, e\}$, $\theta_p$ and $\theta_t$ are the pan and tilt angles of the face. Because this vector depends only on the pan and

tilt values, another error metric was also used. This metric denoted by AE, is the angle between the unit length vectors $\mathbf{a}_g$, $\mathbf{a}_e$ defined by rotating the "neutral" direction vector $\mathbf{a}_n = [1,0,0]^T$ by the estimated and ground truth pan, tilt and roll angles $\theta_p, \theta_t, \theta_r$ as follows:

$$\mathbf{a} = \mathbf{R}_{\theta_p} \mathbf{R}_{\theta_t} \mathbf{R}_{\theta_r} \mathbf{a}_n, \tag{23}$$

where $\mathbf{R}_{\theta_p}$, $\mathbf{R}_{\theta_t}$ and $\mathbf{R}_{\theta_r}$ are the rotation matrices for the pan, tilt and roll axes defined in Table II.

### B. Experimental Results

Figure 6 depicts the estimated pan, tilt and roll angles along with the corresponding ground truth values over time for three of the test video sequences. Moreover, the absolute error in degrees between the three estimated angles and the ground truth values, namely the values of PE, TE, RE over time, are presented in Figures 7. One can see that the proposed algorithm can estimate the $3D$ head pose with very good accuracy. In certain cases, when the movement of the face is extreme and sudden, or in large rotation angles, the error is bigger than usual. The angle AE between the $3D$ direction vector defined by the estimated angles and the direction vector defined by the ground truth is shown in Figure 8. For example, the absolute error averaged over all frames for the first test video sequence (Figure 8a) is $1.76$ degrees and its variance is $2.69$ degrees.

Tables III, IV and V present the mean, variance and median values (with respect to time) of the aforementioned errors for five of the test video sequences. Average values over all video sequences are also provided. One can see that the average value (over all sequences) of AE is $3.20$ degrees and the average value of variance is $3.31$ degrees. The average values for the other error metrics are also very small. Thus it is evident that the proposed system can accurately estimate the $3D$ face pose in a video sequence.

The values of PE, TE, RE, DE errors obtained by the best algorithm in [12] are reported in Table VI for comparison purposes. One can see, that the proposed algorithm achieves far better results than the method in [12]. For example, the mean error in the pointing vector (DE) for the proposed algorithm is $3.8^o$, a large improvement over the error of $21.3^o$ achieved by [12].

In terms of computational complexity, $0.2$ seconds per frame are required for the tracking procedure and $0.05$ seconds per frame for the pose estimation procedure on an Intel Pentium $4$ ($3.01$ GHz) processor PC with $1.5$GB of RAM. Thus the algorithm requires in total $0.25$ seconds per frame in this modest hardware configuration and without any code optimization.

## VI. CONCLUSION

A $3D$ head pose estimation algorithm based on the use of a parameterized $3D$ physics-based deformable model was proposed in this paper. In this approach, the intensity surface of the facial area is represented by a $3D$ physics-based deformable model. We have shown how to tailor the model deformation equations to efficiently track the human face in

TABLE VI

*Mean, variance and median values for the PE, TE, RE and DE errors (in degrees) achieved on the IDIAP database by the best algorithm presented in [12].*

|          | PE  | TE    | RE  | DE   |
|----------|-----|-------|-----|------|
| mean     | 8.7 | 19.1  | 9.7 | 21.3 |
| variance | 9.1 | 15.41 | 7.1 | 15.2 |
| median   | 6.2 | 14.0  | 8.6 | 14.1 |

a video sequence and concurrently feed three properly trained RBF interpolation networks that estimate the pan, tilt and roll angles of the face. Results obtained on the IDIAP database show that the proposed method produces accurate results.

Future work includes the use of SVM regression instead of RBF interpolation networks in order to estimate the $3D$ face pose. Additionally, we plan to test our system on an even larger data set and explore the influence of the adopted classifier into our method's performance.

## REFERENCES

[1] M. Doi and Y. Aoki, "Real-time video surveillance system using omni-directional image sensor and controllable camera," in *Proceedings of SPIE, Real-Time Imaging VII*, vol. 5012, April 2003, pp. 1–9.

[2] U. Weidenbacher, G. Layher, P. Bayerl, and H. Neumann, "Detection of head pose and gaze direction for human-computer interaction," in *Perception and Interactive Technologies, Springer Berlin / Heidelberg*, vol. 4021/2006, June 2006, pp. 9–19.

[3] P. Lu, X. Zeng, X. Huang, and Y. Wang, "Navigation in 3d game by markov model based head pose estimating," in *Proceedings of the Third International Conference on Image and Graphics (ICIG'04)*, December 2004, pp. 493–496.

[4] B. Yip and J. Jin, "Pose determination and viewpoint determination of human head in video conferencing based on head movement," in *Proceedings of the 10th International Multimedia Modelling Conference*, Brisbane, Australia, January 2004, pp. 130–135.

[5] ——, "3d reconstruction of a human face with monocular camera based on head movement," in *Proceedings of the Pan-Sydney area workshop on Visual information processing (VIP 2003)*, Darlinghurst, Australia, 2003, pp. 99 – 103.

[6] J. Ng and S. Gong, "Multi-view face detection and pose estimation using a composite support vector machine across the view sphere," in *IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, Corfu, Greece, September 1999, pp. 14 – 21.

[7] H. Song, U. Yang, J. Kim, and K. Sohn, "A 3d head pose estimation for face recognition," in *Proceedings of the IASTED International Conference on Signal and Image Processing (SIP 2003)*, Honolulu, USA, August 13-15 2003, pp. 255–260.

[8] C. Chien, Y. Chang, and Y. Chen, "Facial expression analysis under various head poses," in *Proceedings of Third IEEE Pacific Rim Conference on Multimedia*, vol. 2532/2002, Taiwan, December 16-18 2002, pp. 199–212.

[9] E. Seemann, K. Nickel, and R. Stiefelhagen, "Head pose estimation using stereo vision for human-robot interaction," in *Proc. of the Sixth International Conference on Automatic Face and Gesture Recognition (AFGR04)*, Seoul, Korea, May 17-19 2004, pp. 626–631.

[10] R. Yang and Z. Zhang, "Model-based head pose tracking with stereo vision," in *Proc. of the Sixth International Conference on Automatic Face and Gesture Recognition (AFGR02)*, D.C., USA, May 2002, pp. 255–260.

[11] L. M. Brown and Y.-L. Tian, "Comparative study of coarse head pose estimation," in *IEEE Workshop on Motion and Video Computing*, December 2002, pp. 125– 130.

[12] S. Ba and J.-M. Odobez, "Evaluation of multiple cue head pose estimation algorithms in natural environments," in *IEEE International Conference on Multimedia and Expo (ICME)*, Amsterdam, July 2005, pp. 1330–1333.

TABLE II

*Rotation matrices for the pan, tilt and roll axes.*

$$\mathbf{R}_{\theta_p} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_p) & -\sin(\theta_p) \\ 0 & \sin(\theta_p) & \cos(\theta_p) \end{bmatrix} \quad \mathbf{R}_{\theta_t} = \begin{bmatrix} \cos(\theta_t) & 0 & \sin(\theta_t) \\ 0 & 1 & 0 \\ -\sin(\theta_t) & 0 & \cos(\theta_t) \end{bmatrix} \quad \mathbf{R}_{\theta_r} = \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) & 0 \\ \sin(\theta_r) & \cos(\theta_r) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$
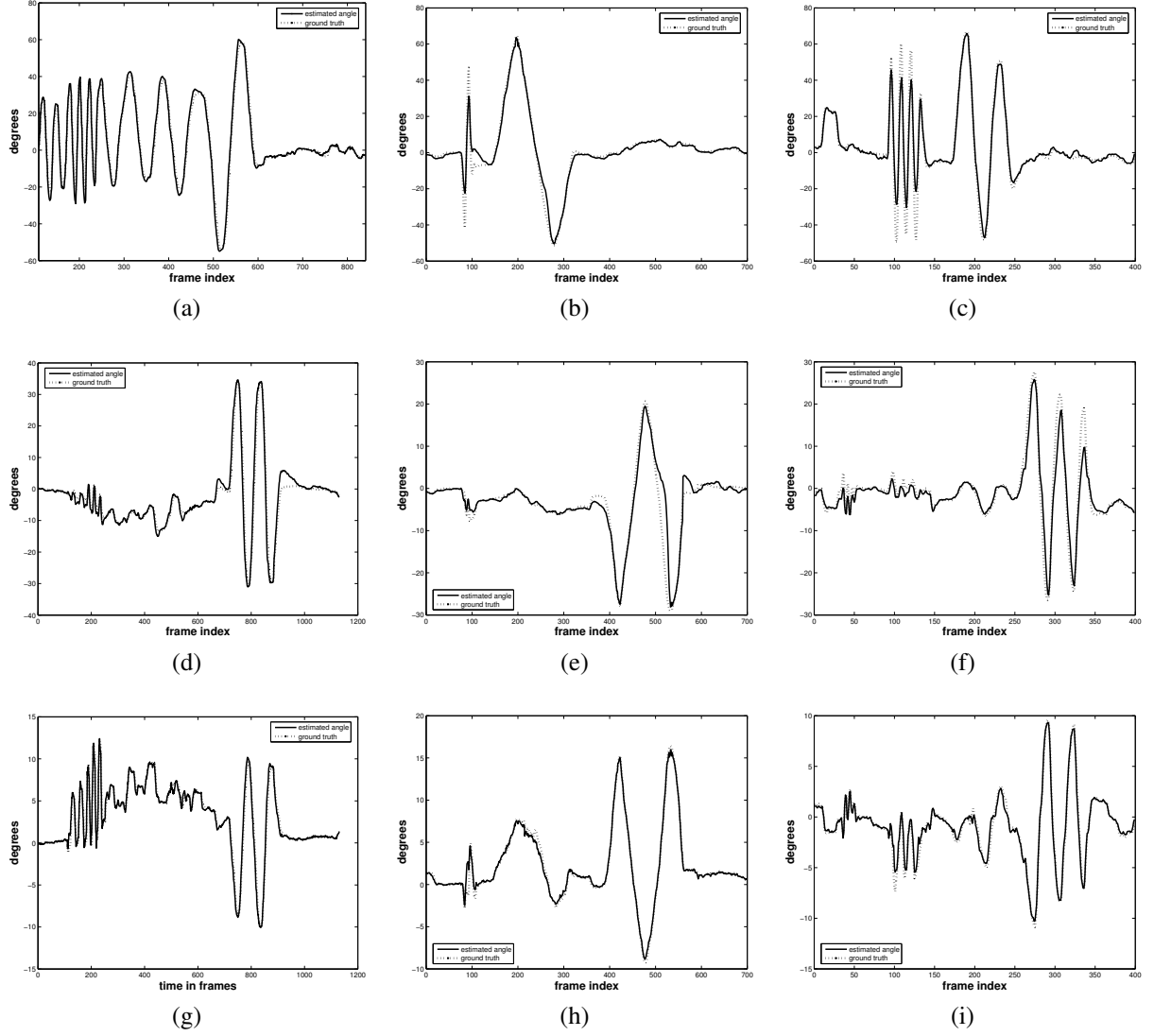
(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Fig. 6.   *The angles (in degrees) estimated by the proposed system and the corresponding ground truth values for parts of three test video sequences that contain significant head motion: **(a)-(c)** pan angles, **(d)-(f)** tilt angles and **(g)-(i)** roll angles.*

TABLE III

*Mean values for the PE, TE, RE, DE and AE errors (in degrees) achieved on the IDIAP database by the proposed algorithm.*

| Test Videos | Mean | | | | |
|---|---|---|---|---|---|
| | PE | TE | RE | DE | AE |
| sequence 3 | 1.2665 | 1.0143 | 0.3245 | 2.2453 | 2.0133 |
| sequence 5 | 3.2214 | 4.3564 | 0.7980 | 3.9876 | 3.4331 |
| sequence 7 | 2.1234 | 1.4612 | 0.4125 | 3.0945 | 2.5503 |
| sequence 10 | 3.6889 | 3.5433 | 0.8764 | 4.3321 | 3.1902 |
| sequence 12 | 0.7650 | 2.3767 | 0.2984 | 5.6501 | 4.5774 |
| average values (all sequences) | 2.2114 | 2.5902 | 0.5328 | 3.8712 | 3.1997 |

[13] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Accepted for future publication.

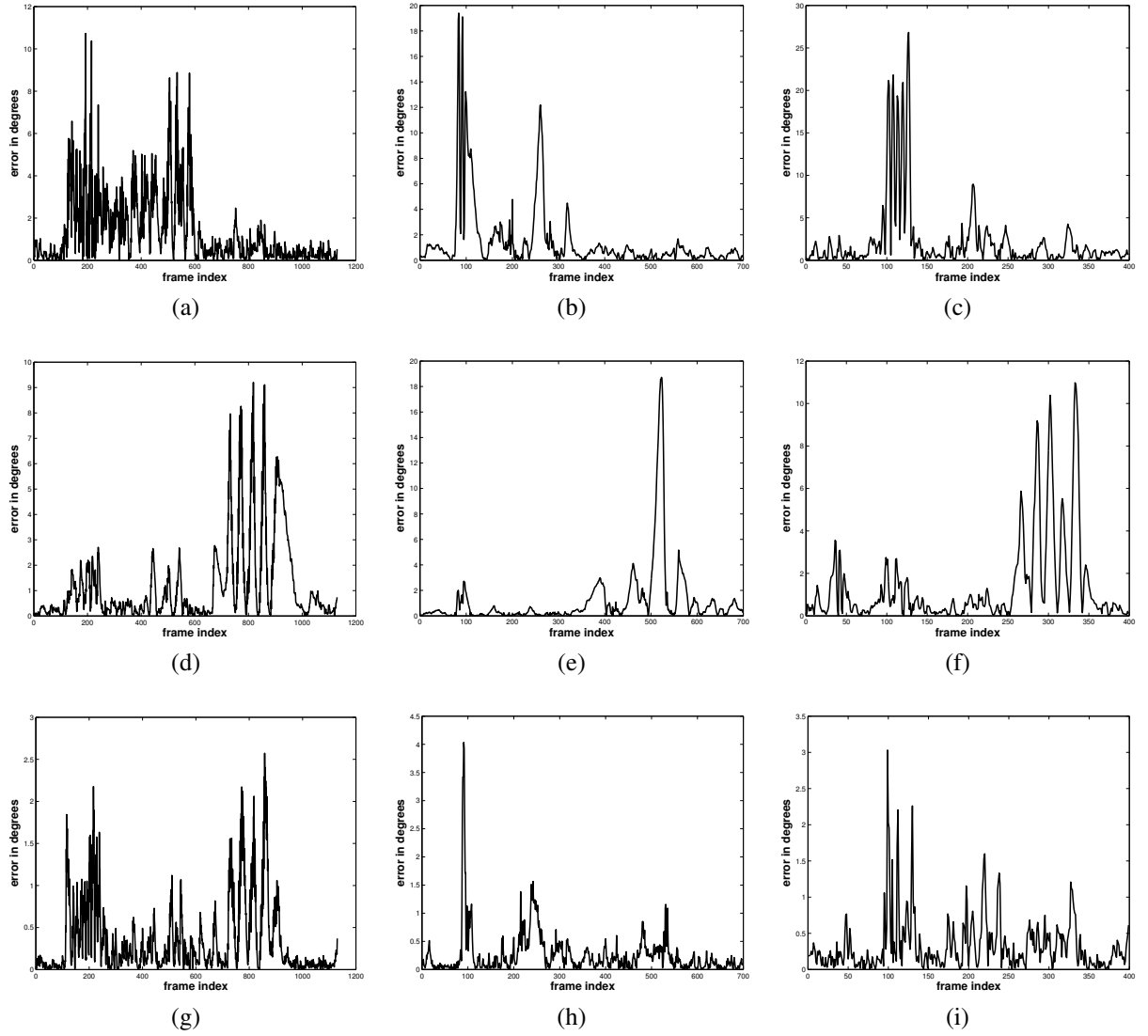[14] Q. Chen, T. Shimada, H. Wu, and T. Shioyama, "Head pose estimation

Fig. 7. *The absolute error in degrees between the angles estimated by the proposed system and the ground truth angles for parts of three test video sequences that contain significant head motion: (a)-(c) pan angles, (d)-(f) tilt angles and (g)-(i) roll angles.*
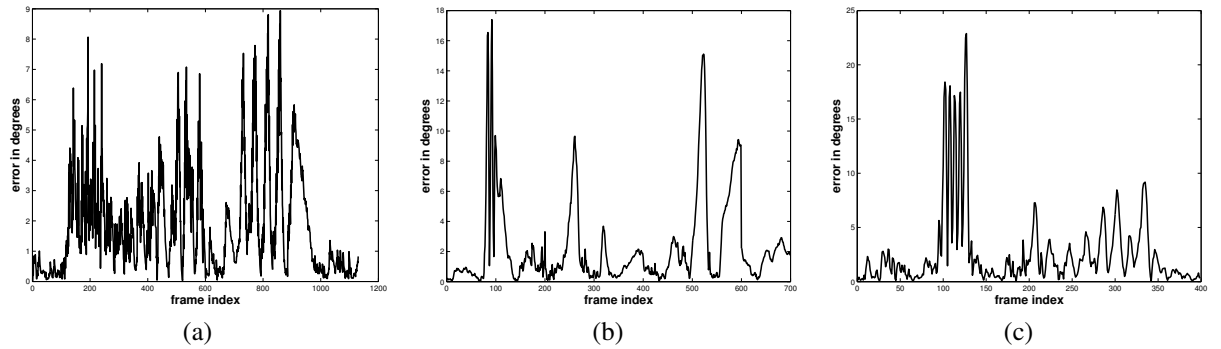


Fig. 8. *The angle AE in degrees between the 3D direction vector of the angles estimated by the proposed system and the direction vector of the ground truth angles for parts of three test video sequences that contain significant head motion.*

using both color and feature information," in *15th International Conference on Pattern Recognition (ICPR'00)*, vol. II, Barcelona, Spain,

September 2000, pp. 842–845.

[15] T. Huang, A. Bruckstein, R. Holt, and A. Netravali, "Uniqueness of 3d

TABLE IV

*Variance values for the PE, TE, RE, DE and AE errors (in degrees) achieved on the IDIAP database by the proposed algorithm.*

| Test Videos | Variance | | | | |
|---|---|---|---|---|---|
| | PE | TE | RE | DE | AE |
| sequence 3 | 2.6689 | 2.4335 | 0.1878 | 1.9520 | 1.6902 |
| sequence 5 | 15.2355 | 7.2235 | 1.7895 | 3.5699 | 3.8987 |
| sequence 7 | 16.0236 | 4.3125 | 0.2123 | 4.3154 | 3.5246 |
| sequence 10 | 15.0012 | 14.8789 | 0.5889 | 4.2103 | 3.5842 |
| sequence 12 | 6.1872 | 16.7901 | 0.4261 | 5.4566 | 4.6678 |
| average values (all sequences) | 11.0012 | 8.0329 | 0.5547 | 3.6398 | 3.3145 |

TABLE V

*Median values for the PE, TE, RE, DE and AE errors (in degrees) achieved on the IDIAP database by the proposed algorithm.*

| Test Videos | Median | | | | |
|---|---|---|---|---|---|
| | PE | TE | RE | DE | AE |
| sequence 3 | 0.4578 | 0.5366 | 0.1456 | 1.4582 | 1.3606 |
| sequence 5 | 2.3258 | 4.1245 | 0.2471 | 5.4213 | 5.2031 |
| sequence 7 | 0.8474 | 0.3245 | 0.8099 | 1.5541 | 1.7880 |
| sequence 10 | 2.5241 | 2.5897 | 0.5013 | 4.0110 | 3.5963 |
| sequence 12 | 0.9987 | 2.1146 | 0.2896 | 4.3201 | 3.0158 |
| average values (all sequences) | 1.0245 | 1.8873 | 0.3314 | 3.3660 | 2.7021 |

pose under weak perspective: A geometrical proof," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 12, pp. 1220–1221, December 1995.

[16] Z. Liu and Z. Zhang, "Robust head motion computation by taking advantage of physical properties," in *IEEE Workshop on Human Motion*, Los Alamitos, CA, USA, December 2000, pp. 73–77.

[17] Y. Hu, L. Chen, Y. Zhou, and H. Zhang, "Estimating face pose by facial asymmetry and geometry," in *IEEE Proceedings of the Sixth International Conference on Automatic Face and Gesture Recognition (FGR 2004)*, Seoul, Korea, May 2004, pp. 651– 656.

[18] M. L. Cascia and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 4, pp. 322–336, April 2000.

[19] C. Zhang and F. S. Cohen, "3-d face structure extraction and recognition from images using 3-d morphing and distance mapping," *IEEE Transactions on Image Processing*, vol. 11, no. 11, pp. 1249–1259, November 2002.

[20] B. Kwolek, "Model based facial pose tracking using a particle filter," in *IEEE Proceedings of the Geometric Modeling and Imaging New Trends (GMAI'06)*, July 2006, pp. 203– 208.

[21] S. O. Ba and J. Odobez, "A probabilistic framework for joint head tracking and pose estimation," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR04)*, vol. 4, 2004, pp. 264 – 267.

[22] S. Z. Li, X. Lu, X. Hou, X. Peng, and Q. Cheng, "Learning multiview face subspaces and facial pose estimation using independent component analysis," *IEEE Transactions on Image Processing*, vol. 14, no. 6, pp. 705–712, June 2005.

[23] R. Stiefelhagen, Y. Jie, and A. Waibel, "Simultaneous tracking of head poses in a panoramic view," in *Proceedings of 15th International Conference on Pattern Recognition, 2000*, vol. 3, Barcelona, Spain, September 2000, pp. 722–725.

[24] V. Vapnik, *The nature of statistical learning theory. Springer Verlag*, 1995.

[25] Y. Li, S. Gong, J. Sherrah, and H. Liddell, "Support vector machine based multi-view face detection and recognition," Image and Vision Computing, *vol. 1, no. 5, p. 413427, May 2004.

[26] Y. Li, S. Gong, and H. Liddell, "Support vector regression and classification based multi-view face detection and recognition," in IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France, March 2004, pp. 300–305.

[27] A. Rajwade and M. Levine, "Facial pose from 3d data," Image and Vision Computing, *vol. 24, no. 8, pp. 849–856, August 2006.

[28] B. M. C. Nastar and A. Pentland, "Generalized image matching: Statistical learning of physically-based deformations," in 4th European Conference on Computer Vision (ECCV'96), *vol. 1, Cambridge, UK, April 1996, pp. 589–598.

[29] B. Moghaddam, C. Nastar, and A. Pentland, "A bayesian similarity measure for direct image matching," in International Conference on Pattern Recognition (ICPR 1996), *Vienna, Austria, August 1996, pp. 350–358.

[30] A. Pentland and S. Sclaroff, "Closed-form solutions for physically based shape modeling and recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, *vol. 13, no. 7, pp. 715–729, July 1991.

[31] C. Nastar and N. Ayache, "Frequency-based nonrigid motion analysis: Application to four dimensional medical images," IEEE Transactions on Pattern Analysis and Machine Intelligence, *vol. 18, no. 11, pp. 1069–1079, November 1996.

[32] S. Krinidis, C. Nikou, and I. Pitas, "Reconstruction of serially acquired slices using physics-based modelling," IEEE Transactions on Information Technology in Biomedicine, *vol. 7, no. 4, pp. 394–403, December 2003.

[33] C. Nikou, G. Bueno, F. Heitz, and J. Armspach, "A joint physics-based statistical deformable model for multimodal brain image analysis," IEEE Transactions on Medical Imaging, *vol. 20, no. 10, pp. 1026–1037, October 2001.

[34] C. Nastar and N. Ayache, "Fast segmentation, tracking, and analysis of deformable objects," in Proceedings of the Fourth International Conference on Computer Vision (ICCV'93), *Berlin, Germany, May 1993, pp. 11–14.

[35] M. Krinidis, N. Nikolaidis, and I. Pitas, "The discrete modal transform and its application to lossy image compression," Signal Processing: Image Communication, *vol. 22, no. 5, pp. 480–504, June 2007.

[36] ——, "2d feature point selection and tracking using 3d physics-based deformable surfaces," IEEE Transactions on Circuits and Systems for Video Technology, *vol. 17, no. 7, pp. 876 – 888, July 2007.

[37] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in IEEE International Conference on Image Processing (ICIP02), *Rochester, New York, USA, September 2002, pp. 900–903.

[38] P. Viola and M. J. Jones, "Robust real-time object detection," Cambridge Research Laboratory, Tech. Rep. 01, 2001.

[39] S. Haykin, Neural Networks: A comprehensive Foundation, 2nd Edition. *Englewood Cliffs: Macmillan Publishing Company, 1998.

[40] D. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," Complex Systems, *vol. 2, pp. 321–355, 1998.

**Michail Krinidis** received the B.S. degree from the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece, in 2002. He is currently pursuing a Ph.D. degree in the same department while also serving as a teaching assistant. His current research interests lie in the areas of 2D tracking, face detection and 3D head pose estimation in image sequences.

**Nikos Nikolaidis** received the Diploma of Electrical Engineering and the Ph.D. degree in electrical engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1991 and 1997, respectively. From 1992 to 1996, he was Teaching Assistant at the Departments of Electrical Engineering and Informatics at the Aristotle University of Thessaloniki. From 1998 to 2002, he was a Postdoctoral Researcher and Teaching Assistant at the Department of Informatics, Aristotle University of Thessaloniki, where he is currently an Assistant Professor. He is the co-author of the book 3-D Image Processing Algorithms (New York: Wiley, 2000). He has co-authored 11 book chapters, 27 journal papers, and 84 conference papers. His research interests include computer graphics, image and video processing and analysis, computer vision, copyright protection of multimedia, and 3-D image processing. Dr. Nikolaidis currently serves as Associate Editor for the International Journal of Innovative Computing Information and Control, the International Journal of Innovative Computing Information and Control Express Letters and the EURASIP Journal on Image and Video Processing.

**Ioannis Pitas** received the Diploma of Electrical Engineering in 1980 and the PhD degree in Electrical Engineering in 1985 both from the Aristotle University of Thessaloniki, Greece. Since 1994, he has been a Professor at the Department of Informatics, Aristotle University of Thessaloniki. From 1980 to 1993 he served as Scientific Assistant, Lecturer, Assistant Professor, and Associate Professor in the Department of Electrical and Computer Engineering at the same University. He served as a Visiting Research Associate or Visiting Assistant Professor at several Universities. He has published 153 journal papers, 400 conference papers and contributed in 22 books in his areas of interest and edited or co-authored another 5. He has also been an invited speaker and/or member of the program committee of several scientific conferences and workshops. In the past he served as Associate Editor or co-Editor of four international journals and General or Technical Chair of three international conferences. His current interests are in the areas of digital image and video processing and analysis, multidimensional signal processing, watermarking and computer vision.