Three-Dimensional Face Pose Detection and Tracking Using Monocular Videos: Tool and Application

Fadi Dornaika and Bogdan Raducanu

***Abstract*—Recently, we have proposed a real-time tracker that simultaneously tracks the 3-D head pose and facial actions in monocular video sequences that can be provided by low quality cameras. This paper has two main contributions. First, we propose an automatic 3-D face pose initialization scheme for the real-time tracker by adopting a 2-D face detector and an eigenface system. Second, we use the proposed methods—the initialization and tracking—for enhancing the human–machine interaction functionality of an AIBO robot. More precisely, we show how the orientation of the robot's camera (or any active vision system) can be controlled through the estimation of the user's head pose. Applications based on head-pose imitation such as telepresence, virtual reality, and video games can directly exploit the proposed techniques. Experiments on real videos confirm the robustness and usefulness of the proposed methods.**

***Index Terms*—AIBO robot, face detection, human–computer interaction (HCI), real-time 3-D head-pose tracking, 3-D head-pose estimation.**

I. Introduction

THE ABILITY to detect and track human heads and facial features in video sequences is useful in a great number of applications, such as human–computer interaction (HCI) and gesture recognition [4], [25]. Vision-based tracking systems represent an attractive solution since vision sensors are not an invasive technology. To this end, many systems and methods have been developed. Of particular interest are vision-based markerless head and/or face trackers. Since these trackers do not require any artificial markers to be placed on the face, comfortable and natural movements can be achieved. On the other hand, building robust and real-time markerless trackers for head and facial features is a difficult task due to the high variability of the face and the facial features in videos.

In general, there are two main approaches for head-pose estimation: feature- and view-based (holistic) approaches. Feature-based approaches refer to the extraction of salient facial characteristics (eyes, nose, and mouth) which are used to compute the head pose based on their spatial relations. In the past, many feature-based approaches have been proposed (e.g., [7], [14], [17], and [26]).

The main drawback of the feature-based approaches is that it could happen that not all the points are visible during a video sequence. Moreover, reliable feature tracking is often difficult, since minor changes between frames can lead to very different segmentation in consecutive frames—the drift problem.

A solution to overcome the drawbacks of feature-based approaches is given by view-based approaches (appearance-based approaches), which try to analyze the whole facial appearance in order to infer the 3-D head pose. To overcome the problem of appearance changes, recent works on faces adopted statistical facial textures. For example, active appearance models (AAMs) received a lot of attention recently. The AAMs have been proposed as a powerful tool for analyzing facial images [6]. Deterministic and statistical appearance-based tracking methods have been proposed [5], [9], [15]. These methods can successfully tackle the image variability and drift problems by using deterministic or statistical models for the global appearance of a special object class: the face.

A few algorithms exist, which attempt to track both the head and the facial features in real time, e.g., [9] and [15]. These works have addressed the combined head and facial feature tracking using the AAM principles. However, [9] and [15] require tedious learning stages that should be performed beforehand and should be repeated whenever the imaging conditions change. Recently, we have developed a head and facial feature tracking method based on online appearance models (OAMs) [10]. Unlike the AAMs, the OAMs offer a lot of flexibility and efficiency since they do not require any facial texture model that should be computed beforehand. Instead, the texture model is built online from the tracked sequence.

This paper extends our previous work [10] in two directions. First, we propose a method for the automatic initialization of the 3-D head pose. In [10], the initialization is performed manually. Second, we use the tracker in a human–robot interaction application in which the gaze of a robotic vision sensor is controlled by the user's gaze. The proposed scheme for estimating and tracking the 3-D head-pose methods is automatic. The remainder of this paper is organized as follows. Section II briefly describes the deformable 3-D face model that we use to create shape-free facial patches from input images. Section III states the problem we propose to solve.

Manuscript received August 14, 2007; revised January 18, 2008 and March 25, 2008. First published March 24, 2009; current version published July 17, 2009. This work was supported in part by MEC Grant TIN2006-15308-C02, by the CONSOLIDER-INGENIO 2010 under Grant CSD2007-00018, Spain, and by the Ramon y Cajal research program. This paper was recommended by Associate Editor Q. Zhu.

F. Dornaika is with the Institut Géographique National, 94165 Saint-Mandé, France (e-mail: fadi.dornaika@ign.fr).

B. Raducanu is with the Computer Vision Center, 08193 Barcelona, Spain (e-mail: bogdan@cvc.uab.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2008.2009566

Section IV describes the proposed automatic 3-D head-pose initialization from one single image. Section V presents the real-time face and facial action tracker. Section VI describes the proposed human–machine interaction application that is based on controlling a robotics camera gaze through the use of the tracked user's head orientation. Section VII concludes this paper.

II. Modeling Faces

A deformable 3-D model. In this paper, we use the 3-D face model *Candide* [2]. This 3-D deformable wire frame model was first developed for the purpose of model-based image coding and computer animation. The 3-D shape of this wire frame model is directly recorded in coordinate form. It is given by the coordinates of the 3-D vertices $\mathbf{P}_i$, with $i = 1, \ldots, n$, where n is the number of vertices. Thus, the shape up to a global scale can be fully described by the $3n$-vector $\mathbf{g}$, which is the concatenation of the 3-D coordinates of all vertices $\mathbf{P}_i$. The vector $\mathbf{g}$ is written as

$$\mathbf{g} = \mathbf{g}_s + \mathbf{A}\boldsymbol{\tau_a} \tag{1}$$

where $\mathbf{g}_s$ is the static shape of the model, $\boldsymbol{\tau_a}$ is the animation control vector, and the columns of $\mathbf{A}$ are the animation units (AUs). The static shape is constant for a given person. In this paper, we use six modes for the facial AU matrix $\mathbf{A}$. We have chosen the following AUs: lower lip depressor, lip stretcher, lip corner depressor, upper lip raiser, eyebrow lowerer, and outer eyebrow raiser. These AUs are enough to cover most common facial animations. Moreover, they are essential for conveying emotions.

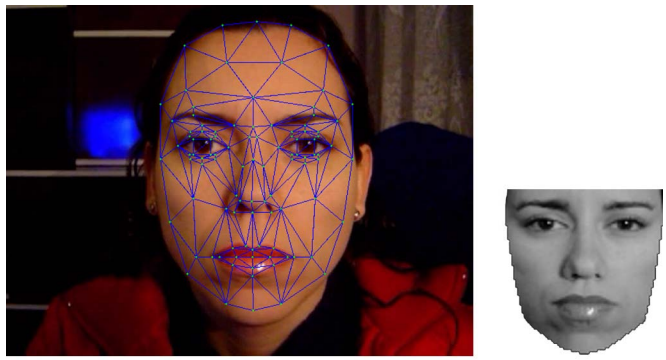
In (1), the 3-D shape is expressed in a local coordinate system. However, one should relate the 3-D coordinates to the image coordinate system. To this end, we adopt the weak perspective projection model. We neglect the perspective effects since the depth variation of the face can be considered as small compared to its absolute depth. Thus, the state of the 3-D wire frame model is given by the 3-D head-pose parameters (three rotations and three translations) and the internal face animation control vector $\boldsymbol{\tau_a}$. This is given by the 12-D vector $\mathbf{b}$

$$\mathbf{b} = [\theta_x, \quad \theta_y, \quad \theta_z, \quad t_x, \quad t_y, \quad t_z, \quad \boldsymbol{\tau_a}^{\mathrm{T}}]^{\mathrm{T}}. \tag{2}$$

Note that, if only the aspect ratio of the camera is known, then the component t_z is replaced by a scale factor having the same mapping role between 3-D and 2-D. In this case, the state vector is given by

$$\mathbf{b} = [\theta_x, \quad \theta_y, \quad \theta_z, \quad t_x, \quad t_y, \quad s, \quad \boldsymbol{\tau_a}^{\mathrm{T}}]^{\mathrm{T}}. \tag{3}$$

Shape-free facial patches. A facial patch is represented as a shape-free image (geometrically normalized raw brightness image). The geometry of this image is obtained by projecting the static shape $\mathbf{g}_s$ (neutral shape) using a centered frontal 3-D pose onto an image with a given resolution. The texture of this geometrically normalized image is obtained by texture mapping from the triangular 2-D mesh in the input image (see Fig. 1) using a piecewise affine transform $\mathcal{W}$ (see [2] for more details). The warping process applied to an input image $\mathbf{y}$ is denoted by



(a) (b)

Fig. 1. (a) Input image with correct adaptation. (b) Corresponding shape-free facial patch.

$$\mathbf{x}(\mathbf{b}) = \mathcal{W}(\mathbf{y}, \mathbf{b}) \tag{4}$$

where $\mathbf{x}$ denotes the shape-free patch, and $\mathbf{b}$ denotes the geometrical parameters. Several resolution levels can be chosen for the shape-free patches. Regarding photometric transformations, zero-mean unit-variance normalization is used to partially compensate for contrast variations.

III. Problem Formulation

Given a video sequence depicting a moving head/face, we would like to recover, for each frame, the 3-D head pose and the facial actions encoded by the control vector $\boldsymbol{\tau_a}$. In other words, we would like to estimate the vector $\mathbf{b}_t$ (3) at time t given all the observed data until time t, denoted as $\mathbf{y}_{1:t} \equiv \{\mathbf{y}_1, \ldots, \mathbf{y}_t\}$. In a tracking context, the model parameters associated with the current frame will be carried over to the next frame.

For each input frame $\mathbf{y}_t$, the observation is the shape-free facial patch associated with the geometric parameters $\mathbf{b}_t$. We use the HAT symbol for the tracked parameters and patches. For a given frame t, $\hat{\mathbf{b}}_t$ represents the computed geometric parameters and $\hat{\mathbf{x}}_t$ the corresponding shape-free patch, i.e.,

$$\hat{\mathbf{x}}_t = \mathbf{x}(\hat{\mathbf{b}}_t) = \mathcal{W}(\mathbf{y}_t, \hat{\mathbf{b}}_t). \tag{5}$$

The estimation of the initial parameters $\hat{\mathbf{b}}_1$ corresponding to the first frame will be described in Section IV—3-D head-pose initialization. The estimation of the current parameters $\hat{\mathbf{b}}_t$ from the previous ones $\hat{\mathbf{b}}_{t-1}$ and the sequence of images will be presented in Section V—simultaneous head and facial action tracking.

Fig. 2 shows our proposed 3-D face tracker. The initialization part relies on a 2-D face detector and a statistical facial texture. The tracking part relies on image registration based on the principles of OAMs. One can notice that the initialization part uses a statistical facial texture model, whereas the tracking part uses an adaptive appearance model.

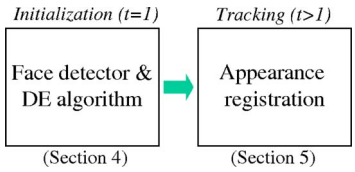


Fig. 2. Fully automatic 3-D face and facial feature tracker.

IV. 3-D Head-Pose Initialization

As can be seen, the tracker requires the knowledge of the state vector (the 3-D head-pose parameters and the facial actions) associated with the *first* frame in the monocular video sequence. Note that even in the case where the static shape of the user's face model is known inferring its 3-D pose (head-pose parameters) with respect to the camera using a single image is a challenging task since there is no correspondence between the 3-D wire frame model and the raw image. Previous works adopted a simple scheme where the user is asked to align his/her head position and orientation with respect to the camera such that the actual 3-D head-pose becomes equal to a predefined head pose. The alignment can be controlled and assessed by the projection of some facial features (nose tip, eye corners) on their predefined locations corresponding to the predefined 3-D head pose.

In this paper, we relax the use of a predefined 3-D head pose in order to get a very flexible 3-D face tracker. In order to compute the 3-D head-pose parameters associated with the first frame, we will use a statistical facial texture model which is built offline. The 3-D head-pose parameters are then estimated by minimizing the distance between the input image texture and a learned face space—eigenface system. Reaching a global minimum can be achieved through the use of the differential evolution (DE) algorithm. In the current implementation, we assume that the first frame in the video captures a face with a neutral configuration. Note that this assumption is very realistic since the neutral state is usually the user's emotion state. Therefore, the state vector will reduce to six parameters describing the 3-D head pose, i.e., $\mathbf{b}_t = [\theta_x, \theta_y, \theta_z, t_x, t_y, s, \mathbf{0}^{\mathrm{T}}]^{\mathrm{T}} = [\mathbf{h}^{\mathrm{T}}, \mathbf{0}^{\mathrm{T}}]^{\mathrm{T}}$. The vector $\mathbf{h}$ encodes the six 3-D head-pose parameters.

The use of a statistical facial texture model has been used in order to track the head parameters in monocular video sequences (e.g., [2]). However, in [2], recovering the 3-D head pose for the first frame is preformed manually. In this paper, those parameters are automatically estimated, whereas most of the proposed tracking methods use a manual initialization.

A. Backgrounds and Statistical Facial Texture

To build a statistical facial texture model, we use our appearance-based tracker [10] (outlined in Section V). This tracker provides the time-varying 3-D head pose and facial actions together with the corresponding shape-free patches $\hat{\mathbf{x}}$. Using these training patches, one can easily build a statistical facial texture model. We assume that we have K shape-free patches. Applying a classical principal component analysis (PCA) on the training patches, we can compute the mean and the principal modes of variation. Thus, the parameters of the facial texture model will be given by the average texture $\overline{\mathbf{x}}$ and the principal texture modes encoded by the matrix $\mathbf{X}$. The columns of $\mathbf{X}$ represent the principal modes, and their size is d. The number of principal modes is set such that their corresponding variation is equal to a high percentage of the total variation. Note that each training patch has undergone zero-mean unit-variance normalization. In general, the total number of training facial patches K is not equal to d.

If the model instance $\mathbf{h}$ is a good fit to the input image (i.e., the 3-D mesh is aligned with the actual 3-D head pose), then the residual error between the shape-free patch $\mathbf{x}$ and its projection onto the PCA space $\widetilde{\mathbf{x}}$ is small since the remapped texture will be consistent with the statistical model of a face texture. Thus, a reliable measure of the goodness of any fit $\mathbf{h}$ can be given by the norm of the associated residual image between the shape-free patch and its PCA approximation

$$e(\mathbf{h}) = \|\mathbf{r}\|^2 = \|\mathbf{x}(\mathbf{h}) - \widetilde{\mathbf{x}}(\mathbf{h})\|^2 . \tag{6}$$

The projection of the texture $\mathbf{x}(\mathbf{h})$ onto the space spanned by the texture modes is given by

$$\widetilde{\mathbf{x}}(\mathbf{h}) = \overline{\mathbf{x}} + \mathbf{X}\mathbf{X}^{\mathrm{T}} (\mathbf{x}(\mathbf{h}) - \overline{\mathbf{x}}) .$$

In the literature, the error (6) is known under the name of the reconstruction error or distance from feature space (DFFS).

Fig. 3 shows the principle of the technique. Fig. 3(a) shows a good model adaptation. Both the input image and the corresponding shape-free patch are shown. In this case, the residual error (6) corresponds to a minimum. Fig. 3(b) shows a bad model adaptation. In this case, the error (6) does not correspond to the minimum. Thus, the basic idea is to estimate the 3-D head-pose parameters, i.e., the vector $\mathbf{h}$, such that the associated shape-free patch will be as close as possible to a facial texture.

The general scheme adopted for building a facial texture model (PCA space parameters) is to use training images belonging to several persons. However, since we are interested in a specific-user interface application, we will adopt a more flexible scheme. Within this scheme, the system stores data about several subjects. The static shape $\mathbf{g_s}$, as well as the facial texture model (PCA space parameters) of each subject, is stored. In other words, the static shapes and the statistical texture models (estimated offline) are labeled by the subject's identity.

The static shape of every subject or its low dimensional representation (provided by *Candide* model) is estimated offline either manually or automatically [1]. The statistical texture model of every subject is computed as follows. The subject is asked to perform head movements together with some facial expressions. The appearance-based tracker [10] is then run to get the training shape-free facial patches associated with each image in the training video (the first frame in this video is manually adapted). This manual initialization is carried out using an interactive graphical interface that displays the first/current frame in the training video together with a 2-D projection of the *Candide* model. This interface includes three menus for setting the following three kinds of parameters: 1) the person specific shape ($\mathbf{g_s}$); 2) the facial actions (the vector $\boldsymbol{\tau_a}$); and 3) the 3-D head pose (the vector $\mathbf{h}$). The user can control

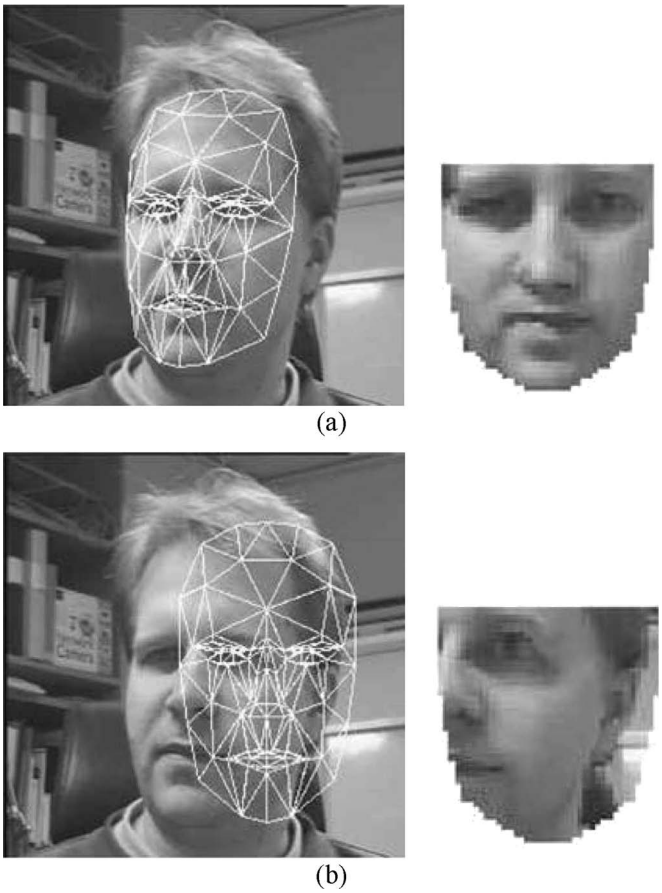


Fig. 3. Example of model fitting. (a) Corresponds to correct 3-D head-pose parameters $\mathbf{h}$. (b) Corresponds to bad 3-D head-pose parameters $\mathbf{h}$.

every parameter and immediately see the corresponding alignment between the projected 3-D mesh and the actual face. The good initial parameters are those ones who give the best alignment.

B. 3-D Head-Pose Initialization Using a 2-D Face Detector and the DE Algorithm

Our proposed initialization is shown in Fig. 4. It proceeds as follows. First, a 2-D face detector is invoked. This provides the 2-D rectangular window bounding the face. Second, the identity of the detected face is recognized using a simple classifier. Here, we use the linear discriminant analysis followed by the nearest neighbor classifier. More sophisticated recognition algorithms can also be used [3], [28]. Once the identity of the detected face is known, the corresponding static shape and the corresponding PCA space parameters (mean and principle modes) of the recognized subject are then used in order to recover the 3-D head pose—initializing the 3-D head pose. Recall that the static shape is constant for a given person.

As we have mentioned earlier, the initial 3-D head-pose parameters $\mathbf{h}$ are recovered by minimizing the residual error (6)

$$\mathbf{h} = \arg\min_{\mathbf{h}} e(\mathbf{h}).$$

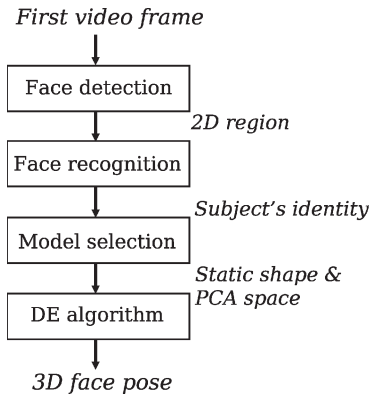


Fig. 4. Proposed automatic initialization for subject-specific HCI applications.

Fig. 5. Mean texture associated with 500 training images.

To this end, we use the DE algorithm [8], [23] in order to minimize the error (6) with respect to the 3-D head-pose parameters. The DE algorithm is a practical approach to global numerical optimization that is easy to implement, reliable, and fast [19].

This is carried out using generations of solutions—population. The population of the first generation is randomly chosen around a rough solution $\mathbf{h}^\star$. Thus, the first population is centered on a solution formed by $\mathbf{h}^\star = (0, 0, 0, t_x^\star, t_y^\star, s^\star)^{\mathrm{T}}$. The 2-D translation $(t_x^\star, t_y^\star)$ is set to the center of the rectangle found by Viola and Jones face detector [24]. The scale $s^\star$ is directly related to the size of the detected rectangle.

The optimization adopted by the DE algorithm is based on a population of N solution candidates $\mathbf{h}_{n,i}$ $(n = 1, \ldots, N)$ at iteration (generation) i, where each candidate has six components. Initially, the solution candidates are randomly generated within the provided intervals of the search space. The population improves by generating new solutions iteratively for each candidate. New solution for every population member $\mathbf{h}_{n,i}$ for the iteration step $i + 1$ is determined in two steps

$$\mathbf{v}_{n,i+1} = \mathbf{h}_{\mathrm{best}} + f\,(\mathbf{h}_{a,i} - \mathbf{h}_{b,i} + \mathbf{h}_{c,i} - \mathbf{h}_{d,i}) \tag{7}$$

$$\mathbf{z}_{n,i+1} = C(\mathbf{h}_{n,i}, \mathbf{v}_{n,i+1}) \tag{8}$$

where $\mathbf{h}_{a,i}$, $\mathbf{h}_{b,i}$, $\mathbf{h}_{c,i}$, and $\mathbf{h}_{d,i}$ are the randomly selected four population vectors, $\mathbf{h}_{\mathrm{best}}$ is the best-so-far solution, f is a weighting scalar, $\mathbf{v}_{n,i+1}$ is a displaced version of $\mathbf{h}_{\mathrm{best}}$, and $C()$ is a crossover operator that copies coordinates from both $\mathbf{h}_{n,i}$ and $\mathbf{v}_{n,i+1}$ to the trial solution $\mathbf{z}_{n,i+1}$. Only if the trial solution $\mathbf{z}_{n,i+1}$ proves to have lower cost (6) it replaces the population member $\mathbf{h}_{n,i}$ $(\mathbf{h}_{n,i+1} \longleftarrow \mathbf{z}_{n,i+1})$; otherwise, the current population member is used in the next generation $i + 1$ $(\mathbf{h}_{n,i+1} \longleftarrow \mathbf{h}_{n,i})$. The scalar f can be fixed or set

Fig. 6. Automatic 3-D head-pose initialization. (Left column) Four unseen images together with the 2-D face detector results. (Right column) Corresponding estimated 3-D head-pose using the DE algorithm.

to a random variable belonging to the interval [0.5, 1.5]. In our implementation, we use a uniform random variable for obtaining fast convergence [8].

We stress the fact that the use of the face detector can be relaxed on the expense of a very large range for the solutions belonging to the first population.

C. Results

Fig. 5 shows the average texture obtained with a sequence of 500 training patches. In this example, we found that the first ten principal modes correspond to 87% of the total variation associated with the training images. The first 20 principal modes correspond to 93% of the total variation. Fig. 6 shows the automatic 3-D head-pose initialization associated with four unseen images. The number of principal modes was set to 20. The left column displays the original image together with the 2-D face detector results. The right column displays the corresponding estimated 3-D head pose using the DE algorithm. The 3-D mesh is projected according to the estimated 3-D head pose. As can be seen, even though the head was not in a frontal view, the corresponding 3-D pose parameters are correctly estimated using the DE algorithm. Recall that the recovered parameters are richer than those provided by the 2-D

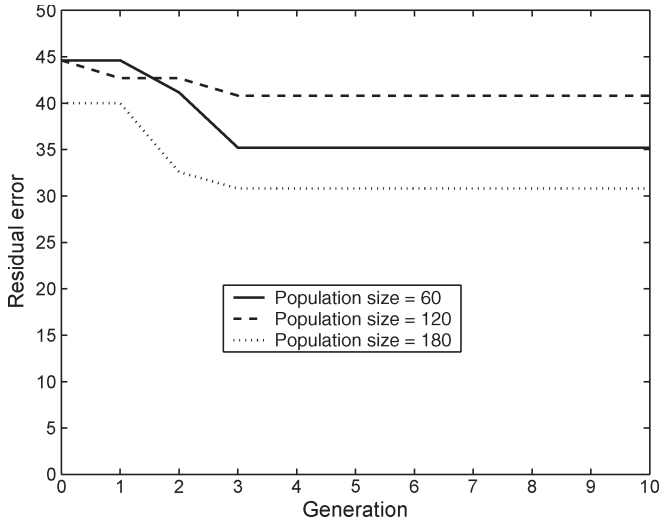


Fig. 7. Evolution of the best error obtained by the DE algorithm associated with the image shown in Fig. 6 (top). The DE algorithm was run with three population sizes: $N = 60$, $N = 120$, and $N = 180$.

face detector since they depict the 3-D pose of the head with respect to the camera.

Fig. 7 shows the evolution of the best error obtained by the DE algorithm associated with the image shown in Fig. 6 (top). The DE algorithm was run with three population sizes: $N = 60$, $N = 120$, and $N = 180$. Fig. 8 shows the estimated 3-D head pose obtained at convergence when the population size is set to 60, 120, and 180 (from left to right). As can be seen, the best fitting results were obtained when the population size was 180.

The CPU time associated with the automatic initialization ranges from 1 s to a few seconds. This computing time depends on many factors such as the number of generations and the number of texture modes used. It is worth noting that this initialization does not prohibit the real-time performance of the 3-D face tracker (presented in Section V) since this initialization is performed only on the first video frame, as the subject lets the system captures the 3-D pose of his/her head.

In addition to the qualitative evaluation of the fitting solution, we used an automatic evaluation based on the obtained DFFS. For each subject-specific PCA space, two classes of distances were built: 1) distances corresponding to a good fit and 2) distances corresponding to a bad fit. First, several training images are manually fitted. The corresponding DFFS provides the positive examples (good fit class). Second, using the same set of training images, the fit is significantly perturbed, and the corresponding DFFS provides the negative examples (bad fit class). At running time, once the DE algorithm converged, the obtained DFFS is compared with the learned ones. The class of the current fit is chosen by the nearest neighbor rule.

V. Simultaneous Head and Facial Action Tracking

In the previous section, we have addressed the initialization problem, i.e., the estimation of the state vector (the 3-D head pose) for the first video frame. In this section, we will describe the tracking process, i.e., the estimation of the state vector (the 3-D head pose and the facial actions) for every subsequent video frame. Certainly, one can use the same initialization process for estimating the state vector for every frame in the video. However, using this scheme has the following three major disadvantages: 1) It cannot run in real time; 2) the 2-D face detector may fail when the face undergoes significant out-of-plane movements; and 3) the statistical facial texture model is fixed in the sense that it does not take into account the possible appearance changes during the whole video sequence. For these reasons, we will use our tracker based on OAMs—described in [10]. This appearance-based tracker aims at computing the 3-D head pose and the facial actions, i.e., the vector $\mathbf{b}$, by minimizing a distance between the incoming warped frame and the current *shape-free* appearance of the face. This minimization is carried out using a gradient descent method. The statistics of the *shape-free* appearance, as well as the gradient matrix, are updated every frame. This scheme leads to a fast and robust tracking algorithm. This tracker is shown in Fig. 9.

In that work, the initial 3-D head-pose parameters were manually provided. On a 3.2-GHz PC, a nonoptimized C code of the approach computes the 3-D head pose and the facial actions in 50 ms.

1) Remark: While the OAM and AAM tracking methods share some points (for instance, both methods are based on appearance models), their pros and cons are different. The tracking method based on the principles of AAM requires a training data set for computing an eigenface system among other things [1], [9]. Thus, the statistical facial texture model is fixed. While AAM-based methods are promising with respect to some aspects, they are depending on the imaging conditions under which the learning is performed. Thus, by changing these conditions, one should repeat the whole learning process, which can be very tedious.

On the other hand, the OAM method builds the texture model online by combining the tracked facial textures obtained from the first to the current frame. In this regard, the OAM method gives more flexibility than the AAM method. Moreover, the tracking based on the OAM method is, in general, faster than the one based on the AAM method.

For these reasons, we did not use the AAM-based tracking, although the 3-D face pose initialization used a statistical facial texture model.

VI. Application: Human–Robot Interaction Through Head Movement Imitation

As computers start to become more and more pervasive in our daily life, new perceptual interfaces must be developed. Compared with traditional interaction tools (based on command line, GUI, etc.), these interfaces are expected to exploit natural ways of communication used by humans (facial movement, hand and body gestures, lip reading, etc.). However, at the same time, they can also be more ambiguous and error-prone than classical interfaces. For this reason, the design of such an interface is very important to guarantee the system's robustness. Faces play a major role in any HCI system, because they represent a rich source of information. Faces are the main

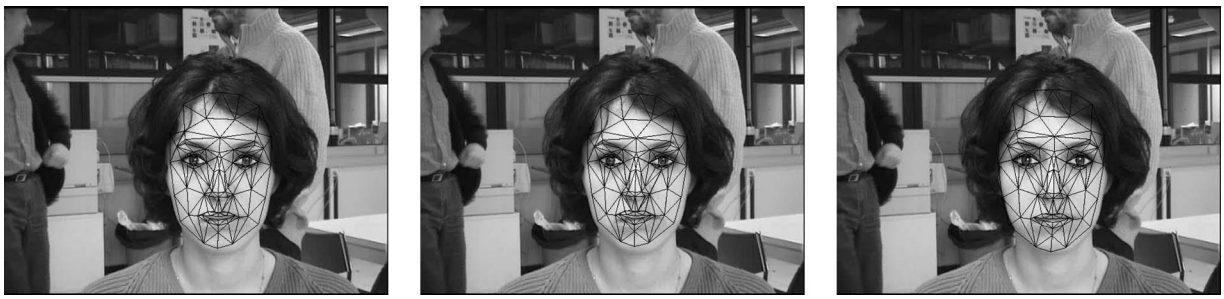


Fig. 8. Automatic 3-D head-pose initialization using the DE algorithm associated with the image shown in Fig. 6 (top). The estimated 3-D head pose obtained at convergence when the population size is set to (from left to right) 60, 120, and 180.

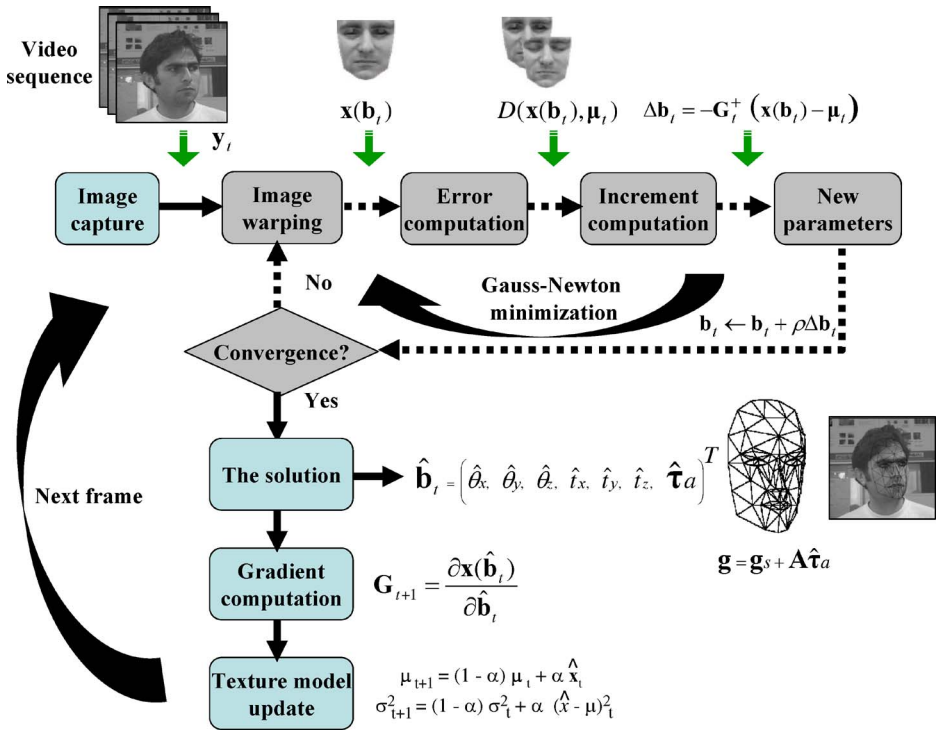


Fig. 9. Three-dimensional face and facial feature tracker using OAM.

cue that humans use for person detection/identification. Aside from this, faces are the main gateway to express our feelings and emotional states. Being able to estimate 3-D head pose in real time, we can get a clue about the user's intentions. Since head orientation is related to a person's gaze direction, this helps to know the objects or persons which become his/her focus of attention. In the context of this work, we make the assumption that head orientation is a reliable indicator of the direction of someone's attention. A more subtle difference between head pose and gaze direction is beyond the scope of our analysis.

As the practical applications of perceptual interface based on 3-D head-pose estimation, we could mention telepresence, "smart" objects which become aware of human's attention, virtual reality, video games, and human–robot interaction. The current trend in robotics is represented by social-oriented robots, i.e., robots which are enabled with perceptual capabilities in order to make communication with humans more natural [12], [13], [18], [20], [27]. In [12], the authors propose a humanoid robot able to learn tasks by imitating human demonstrators.

Monitoring the user's head orientation can be of great utility in assessing interest level in meetings, since human gaze is a precursor to shared attention [11], [21], [22]. In [13], the authors propose a novel approach to recognize head nod and shake. First, head poses are detected by multiview mode, and then, hidden Markov models are used as head gesture statistic inference model for gesture recognition. In [16], the author introduces a developmental learning strategy in which an infantlike robot first has the experiences of visually tracking a human face.

Our application is intended to show how a change in the user's gaze direction can be imitated by a robot's camera, based on the proposed methods. The experimental setup is shown in Fig. 10. The input to the system consists of a video stream capturing the user's face from a fixed camera. The user's head pose is estimated by our tracker (Sections IV and V). Then, the corresponding pitch and yaw angles of the head are encoded and

Fig. 10. Experimental setup.

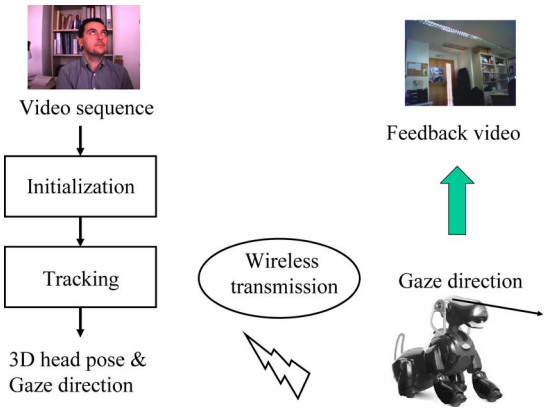


Fig. 11. Data flow for a user's gaze imitation using a monocular camera. The user's gaze direction is continuously controlling the gaze of the AIBO's camera, which is capturing the remote scene.

sent to the robot using a wireless network. Without any loss of generality, we used in our experiments the Sony's AIBO robot, which has the advantage of being particularly designed for an HCI. Thus, our application can be considered as a natural extension of AIBO's built-in behaviors. The orientation of the robot's head is updated online according to the desired direction imposed by the user's head pose. Due to the motor response, there is a very small delay between the desired orientation and the robot's response. The inertia associated with the motors is most visible when a change in direction has to be performed. Fig. 11 shows the data flow between the real-time 3-D head pose tracker and AIBO. Fig. 12 shows the results of head movement imitation associated with a 691-frame sequence. In this video, the person looks around without any restriction. Only eight frames are shown in the figure. The left column displays the user's head pose, and the right column shows the corresponding snapshot of the scene as seen by the AIBO's camera. This experiment can be downloaded from the following address: http://www.cvc.uab.es/~ bogdan/AIBO-Bogdan.avi. There are two more other video demonstrations available at the following addresses: http://www.cvc.uab.es/~ bogdan/AIBO-Angel.avi and http://www.cvc.uab.es/~ bogdan/AIBO-Soumya.avi.

In general, the user and the robot can be very distant from each other. This case corresponds to a telepresence application, where the user is exploring remote (dangerous or inaccessible)

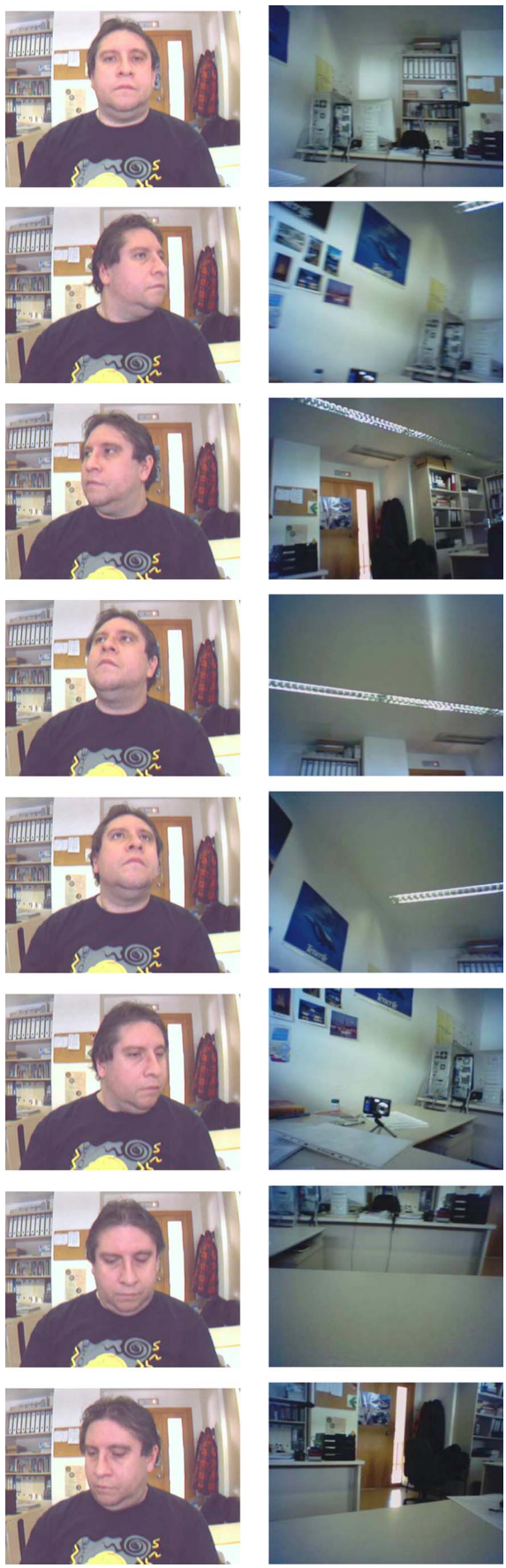


Fig. 12. (Left column) Some input images from the original video. (Right column) Corresponding snapshot acquired by the controlled robot camera.

spots by only changing his gaze direction. This is the case, for instance, for rescue robots, which can be sent to areas affected by earthquakes or fires in order to perform an exhaustive search of the environment to find survivors. Note that, if the user should simultaneously control the remote camera and observe the captured scene, then scaling factors should be used with the tracked head movements so that the subject can still observe his monitor as he controls the camera's movements.

The use of a special display device that visualizes the scene as it is viewed by the remote camera will boost the telepresence feeling. If we assume that the orientation of the fixed camera is aligned with that of the robot camera (in its reference position), then its gaze direction will be equivalent to that of the user.

VII. Conclusion

This paper described two main contributions. First, we proposed an automatic 3-D head-pose initialization scheme for a real-time appearance-based tracker by adopting a 2-D face detector and an eigenface system. Second, we used the proposed methods—the initialization and tracking—for extending the human–machine interaction functionality of the AIBO robot. More precisely, we show that the gaze of any active vision system can be controlled through the estimated direction of the user's gaze. Applications such as telepresence, virtual reality can directly use the proposed techniques. Future work may investigate the extension of the initialization technique such that it will be able to recover the 3-D face shape, the 3-D head pose, and the facial actions from one single image. Moreover, the use of all degrees of freedom as a tool for controlling the robot movements and actions will be investigated.

Our proposed initialization method combines a 2-D face detector having a very high detection rate with a method based on the use of a statistical texture model. The success percentage of the proposed method is thus related to the success rate of both methodologies. Since the initial video frame is depicting a near-frontal face imaged with enough resolution and the global numerical optimization algorithm (the DE algorithm) is used under controlled conditions (no significant changes in imaging conditions), we are expecting that the percentage of success of the proposed initialization method to be very high. The main cause of possible failure will be due to significant changes in imaging conditions, which are common to all methods based on learned statistical texture models.

Acknowledgment

The authors would like to thank Dr. F. Davoine from CNRS, Compiegne, France, for providing some training video sequences.

References

[1] J. Ahlberg, "An active model for facial feature tracking," *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 6, pp. 566–571, Jun. 2002.

[2] J. Ahlberg, "Model-based coding: Extraction, coding, and evaluation of face model parameters," Ph.D. dissertation, Linköping Univ., Linköping, Sweden, Sep. 2002. No. 761.

[3] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell, "Face recognition with image sets using manifold density divergence," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 581–588.

[4] A. Bakhtari and B. Benhabib, "An active vision system for multitarget surveillance in dynamic environments," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 190–198, Feb. 2007.

[5] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 4, pp. 322–336, Apr. 2000.

[6] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.

[7] M. D. Cordea, E. M. Petriu, N. D. Georganas, D. C. Petriu, and T. E. Whalen, "3D head pose recovery for interactive virtual reality avatars," in *Proc. IEEE Instrum. Meas. Technol. Conf.*, 2001, pp. 72–77.

[8] S. Das, A. Konar, and U. Chakraborty, "Two improved differential evolution schemes for faster global search," in *Proc. Genetic Evol. Comput.*, 2005, pp. 991–998.

[9] F. Dornaika and J. Ahlberg, "Fast and reliable active appearance model search for 3-D face tracking," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 4, pp. 1838–1853, Aug. 2004.

[10] F. Dornaika and F. Davoine, "On appearance based face and facial action tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 9, pp. 1107–1124, Sep. 2006.

[11] M. W. Hoffman, D. B. Grimes, A. P. Shon, and R. Rao, "A probabilistic model of gaze imitation and shared attention," *Neural Netw.*, vol. 19, no. 3, pp. 299–310, Apr. 2006.

[12] M. Lopes and J. Santos-Victor, "A developmental roadmap for learning by imitation in robots," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 2, pp. 308–321, Apr. 2007.

[13] P. Lu, M. Zhang, X. Zhu, and Y. Wang, "Head nod and shake recognition based on multi-view model and hidden Markov model," in *Proc. IEEE Conf. Comput. Graph., Imag. Vis.: New Trends*, 2005, pp. 61–64.

[14] Y. Matsumoto and A. Zelinsky, "An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement," in *Proc. IEEE Conf. Autom. Face Gesture Recog.*, 2000, pp. 499–505.

[15] I. Matthews and S. Baker, "Active appearance models revisited," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 135–164, Nov. 2004.

[16] Y. Nagai, "Joint attention development in infant-like robot based on head movement imitation," in *Proc. 3rd Int. Symp. Imitation Animals Artifacts*, 2005, pp. 87–96.

[17] Y. Nakanishi, T. Fujii, K. Kiatjima, Y. Sato, and H. Koike, "Vision-based face tracking system for large displays," in *Proc. UbiComp,* G. Borriello and L. E. Holmquist, Eds., 2002, vol. 2498, pp. 152–159.

[18] K. Nickel and R. Stiefelhagen, "Detection and tracking of 3D-pointing gestures for human–robot interaction," in *Proc. Humanoids*, 2003, pp. 140–146.

[19] K. V. Price, J. A. Lampinen, and R. M. Storn, *Differential Evolution: A Practical Approach to Global Optimization*. New York: Springer-Verlag, 2005.

[20] E. Seemann, K. Nickel, and R. Stiefelhagen, "Head pose estimation using stereo vision for human–robot interaction," in *Proc. Int. Conf. Autom. Face Gesture Recog.*, 2004, pp. 626–631.

[21] K. Smith, S. Ba, D. Gatica-Perez, and J. M. Odobez, "Tracking the multi-person wandering visual focus of attention," in *Proc. ICMI*, 2006, pp. 265–272.

[22] R. Stiefelhagen, J. Yang, and A. Waibel, "Modeling focus of attention for meeting indexing based on multiple cues," *IEEE Trans. Neural Netw.*, vol. 13, no. 4, pp. 928–938, Jul. 2002.

[23] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[24] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

[25] J. G. Wang and E. Sung, "Study on eye gaze estimation," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 3, pp. 332–350, Jun. 2002.

[26] J. G. Wang, E. Sung, and R. Venkateswariu, "EM enhancement of 3D head pose estimates by perspective invariance," in *Proc. ECCV Workshop Human Comput. Interaction*, N. Sebe *et al.*, Ed., 2004, vol. 3058, pp. 187–199.

[27] M. Yeasin and S. Chaudhuri, "Toward automatic robot programming: Learning human skill from visual data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 1, pp. 180–185, Feb. 2000.

[28] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic recognition of human faces from video," *Comput. Vis. Image Underst.*, vol. 91, no. 1/2, pp. 214–245, Jul./Aug. 2003.

Fadi Dornaika received the B.S. degree in electrical engineering from the Lebanese University, Tripoli, Lebanon, in 1990 and the M.S. and Ph.D. degrees in signal, image, and speech processing from the Institut National Polytechnique de Grenoble, Grenoble, France, in 1992 and 1995, respectively.

He has worked with several research institutes including Institut National de Recherche en Informatique et en Automatique Grenoble-Rhône Alpes, Saint Ismier, France; The Chinese University of Hong Kong, Shatin, Hong Kong; Linkoping University, Linkoping, Sweden; and the Computer Vision Center, Barcelona, Spain. He is currently with the Institut Géographique National, Saint-Mandé, France. He has published more than 90 papers in the field of computer vision. His research concerns geometrical and statistical modeling with focus on 3-D object pose, real-time visual servoing, calibration of visual sensors, cooperative stereomotion, image registration, facial gesture tracking, and facial expression recognition.

Bogdan Raducanu received the B.Sc. degree in computer science from the University "Politehnica" of Bucharest, Bucharest, Romania, in 1995 and the Ph.D. degree "*cum laude*" from the University of the Basque Country, Bilbao, Spain, in 2001.

He is currently a Postdoctoral Researcher with the Computer Vision Center, Barcelona, Spain. His research interests include computer vision, pattern recognition, artificial intelligence, and social robotics.