Int J Comput Vis
DOI 10.1007/s11263-013-0691-3

Visual Focus of Attention in Non-calibrated Environments using Gaze Estimation

Stylianos Asteriadis · Kostas Karpouzis · Stefanos Kollias

Received: 24 May 2012 / Accepted: 2 December 2013

Abstract Estimating the focus of attention of a person highly depends on her/his gaze directionality. Here, we propose a new method for estimating visual focus of attention using head rotation, as well as fuzzy fusion of head rotation and eye gaze estimates, in a fully automatic manner, without the need for any special hardware or a priori knowledge regarding the user, the environment or the setup. Instead, we propose a system aimed at functioning under unpretending conditions, only with the usage of simple hardware, like a normal web-camera. Our system is aimed at functioning in a human-computer interaction environment, considering a person is facing a monitor with a camera adjusted on top. To this aim, we propose in this paper two novel techniques, based on local and appearance information, estimating head rotation, and we adaptively fuse them in a common framework. The system is able to recognize head rotational movement, under translational movements of the user towards any direction, without any knowledge or a-priori estimate of the user's distance from the camera or camera intrinsic parameters.

Keywords Head pose estimation · User attention estimation · Facial feature tracking · Facial feature detection · Face tracking · Eye Gaze estimation · Human computer interaction

S. Asteriadis (✉) · K. Karpouzis · S. Kollias
Image, Video and Multimedia Systems Lab, National Technical University of Athens, 9 Iroon Polytechniou str, 157 80 Athens, Greece
e-mail: stiast@image.ntua.gr

K. Karpouzis
e-mail: kkarpou@image.ntua.gr

S. Kollias
e-mail: stefanos@cs.ntua.gr

1 Introduction

Estimating focus of attention in an human-computer interaction (HCI) environment has received a large amount of attention in bibliography during the last years. There is an abundance of methods, addressing the problem from various points of views: estimation of the pose of the whole body (Fathi and Mori 2007), eye gaze estimation (Wang 2003), estimation of the head orientation (Murphy-Chutorian et al. 2007) or fusion of different methods (Weidenbacher et al. 2006), where head pose and eye gaze are combined. We address here the problem of estimating the focus of attention of a person. Especially, we examine the problem in conditions where only the head is visible or necessary to infer his/her focus of attention. To this aim, a fully automatic method for estimating head rotations is considered, attempting to overcome certain conventions and constraints usually imposed on such systems (especially dedicated hardware, headmounted or calibrated systems). Additionally, eye gaze cues are also considered for the overall estimate of gaze directionality. The system is intended to function with the usage of non-calibrated remote cameras, with the aim of maximizing accessibility and easiness of use in HCI environments.

Numerous applications can take advantage of such a system: estimating the orientation of the head, it is easy to estimate the levels of attention of a driver (Murphy-Chutorian et al. 2007) or enrich e-learning systems (Asteriadis et al. 2009c). Also, HCI systems with personalized virtual agents are among near future applications of the field (Peters et al. 2009). Typical works on visual attention estimation, are the methodologies presented in Voit and Stiefelhagen (2010); Ba and Odobez (2011), where the authors estimate focus of attention of participants in dynamic meeting scenarios, taking advantage of information coming from speech and motion activity of other participants, utilizing, in this sense, context-

Published online: 10 December 2013

related information regarding social dynamics. Head pose in Voit and Stiefelhagen (2010) is monitored by four far-field views from the room's upper corners, while (Ba and Odobez 2011) jointly track head position and pose, in a Bayesian probabilistic framework solved with particle filtering techniques. In Murphy-Chutorian et al. (2007), the authors propose a framework for drivers attention estimation. To this aim, they use a camera sensitive to near-infrared wavelengths. Authors in Asteriadis et al. (2009c) estimate attention levels of children with learning difficulties during their interaction with learning materials. In Peters et al. (2009), the authors employ head pose to infer user's attention towards objects presented by a virtual seller and, through the above estimated and different interaction scenarios, they analyze people's reactions to a variety of conditions in interaction. Apart from estimating visual focus of attention, gaze estimation can a play critical role at estimating cognitive states; recognizing such cues can be very important at estimating intentions (e.g. head nods), interest (fixation), and facial expressions. A typical example is the work in Shaker et al. (2011), where head expressivity parameters and game interaction events are correlated, with a view to game content adaptation and personalization.

2 Related Work

The estimate of head pose is usually related to calculating the three degrees of freedom of the head rotation: *yaw*, *pitch* and *roll*. Using these values (or, in cases, subsets of them), one can get important clues regarding the gaze directionality of a person, especially in conditions where eyes are not visible, or the resolution of the image/video is too low to extract information regarding gaze, only from the eyes. We consider here, as head pose, the ability to infer head rotation parameters relative to the view of a camera (Murphy-Chutorian and Trivedi 2009). More specifically, head rotation is considered to be null when the face appears to be symmetrical around the y axis, and the triplet formed by the eyes and mouth forms a plane parallel to the image plane (Gee and Cipolla 1994a). In practice, here, we employ the frontal face detector of Viola–Jones (Viola and Jones 2001), followed by trained models in order to infer frontal view. Horizontal and vertical rotation (yaw and pitch rotation angles) refer to the angles formed by the face plane with that of the camera and, in this paper, the proposed algorithm employs 2-D techniques for extracting the corresponding parameters, based on facial features spatial positions, as well as face area information. Roll angle is defined as the angle formed by the inter-ocular line segment and the x-axis (Horprasert et al. 1996; Ma et al. 2008).

The methodologies presented in recent bibliography, regarding the issue of head pose estimation, use different approaches, varying in terms of hardware necessities or algorithmic restrictions. Should someone want to make a categorization of the existing techniques, a major separation would have to do with methods requiring specific hardware, like helmets and magnetic sensors (intrusive methods) and methods that do not require the user to attach any special equipment on him or her (un-intrusive methods). In this description, we will mainly concentrate on the second group of approaches. Within the group of unintrusive methods, the proposed methodologies—based on the principles they use—may depend on different restrictions, such as specialized hardware or camera/lighting set-ups, alignment between training and test data, exact and robust feature tracking, etc. Furthermore, many systems require a priori knowledge of camera parameters or/and multi-camera systems in order to use stereo vision techniques for inferring head pose parameters. In conditions where it is not possible to have knowledge regarding camera parameters, these methods usually fail to give accurate results when there is movement that goes beyond the system assumptions (e.g. along the z-axis).

In terms of the algorithms used for head pose estimation, mainly in monocular systems, the techniques can be further classified in terms of the principles they use, each presenting different advantages and disadvantages. Although it is not easy to classify methods in a strict manner, as each of them has its own characteristics and specificities, a coarse classification would be the following:

Holistic techniques According to these techniques, the face is compared against certain models, each representing a different pose. The most similar model, or a weighted fusion of different models, is the final estimate of the pose. One of the major disadvantages of the majority of these systems is that they require that the detected facial area is aligned to the training images facial contours (i.e. the amount of background allowed, or the amount of facial area not included in the image should be standardized). Thus, although these methods, theoretically, are alleviated from the problem of finding exact facial feature positions, aligning the detected facial regions with conditions met during training, does in fact require detecting specific features. A typical work is the one described in Stiefelhagen (2004), where neural networks are used for head pose estimation. In Morency et al. (2010), the authors use the generalized adaptive view-based appearance model (GAVAM), which is an extension of the AVAM (Morency et al. 2003). After segmenting the face region at the current frame, according to the model, its pose is estimated using the following modules: a static pose estimator for current frame, a differential tracker between the current frame and the previous one, and a set of keyframes of similar view to the current frame are used, as well as a reference keyframe. In the work in Osadchy et al. (2007), the authors employ convolutional neural networks (CNNs), as a means to overcome errors stemming from unprecise alignment between training and testing data. Using the trained models, input images are mapped to low-dimensional manifolds, parameterized accordingly, to account for different head poses.

Local techniques This group of techniques uses specific facial features positions in order to employ geometric properties for determining head pose. Although using geometrical properties of landmark points is quite straightforward, efficient and accurate facial feature detection and tracking over a large series of images is crucial, and even small errors may introduce large deviations from correct head pose estimates. Another issue that arises, is that of occlusions, when positions of facial points become, either un-available or arbitrary. An example is the work presented in Gee and Cipolla (1994b), where the authors use the ratio between the outer eye corners distance and the distance of the mouth from the eyes middle point as a face model, in order for the face plane orientation to be calculated. Another typical work is the one reported in Gourier et al. (2004), where the relative position of facial characteristics with regards to the rest of the face is used. In Nguyen et al. (2008), support vector machines (SVMs) are employed for finding the location of the iris centers in approximately detected eye regions. The authors report results on the CMU (Sim et al. 2003) Face Dataset to distinguish between frontal and looking-up head poses.

Facial motion recovery This set of methods relies on tracking the face area and estimating the movement between successive frames of a sequence. The results yielded with these methods are usually very accurate but, unless combined with other techniques, either require knowledge of the camera parameters or/and estimates of the distance of the user from the camera, or pre-assume frontal pose at start-up. Typical work is the one reported in Cascia et al. (2000), where the authors model the head as a cylinder to recover its motion parameters, considering the camera model to be known. Due to perspective projection, not all pixels have the same confidence value during registration. In the work described in Dornaika and Davoine (2008), the authors present a method for solving the challenging problem of facial actions and expression recognition under head rotation. For head pose, they employ a deterministic registration technique. This method adopts a weak perspective approach and, thus, does not depend on prior assumptions related to camera or environment parameters. The authors in Lefevre and Odobez (2009), using the Candide model (Ahlberg 2001), jointly estimate head pose and facial actions, even under challenging lighting and motion conditions. Here, also weak projection model is hypothesized and no camera parameters are known. The authors achieve very good results on the BU dataset, and also cater for occlusion by assigning different weights according to points orientation, but fixed patches around trained features are considered.

Non-rigid model fitting Using a trained non-rigid model for mapping on a face region allows multiple transformations of its nodes, so that to match the texture of an input face. Such methods have drawn much attention in the recent years. A major factor to be taken into account is that such models require good initialization of their position and size, as they are prone to falling into local minima when compared to a face image. A typical example is the work proposed in Cootes et al. (2000), where the authors use Active Appearance Models (Cootes et al. 2001) for estimating the rotation of a face around the vertical axis.

Fusion of methods There are also many hybrid techniques that try to avoid the disadvantages of one method, using advantages offered by the other. For example, in Sung et al. (2008), the authors combine active appearance models (AAMs) with cylinder head models (CHMs) (Xiao and Cohn 2003), in an attempt to combine the local character of AAMs with the global motion properties of CHMs. Correct tracking rates improve in comparison to AAMs. However, the "pose coverage" (the spectrum of pose angles that can be detected) does not outperform 45° for yaw angles.

Eye gaze estimation with the usage of simple, ordinary cameras is a less studied issue. Authors in Kourkoutis et al. (2007) compare the position of the face center with the pupil centers, in order to distinguish among different eye gaze directionalities. Holistic methods are utilized in Tan et al. (2002), where manifolds, based on the eye area, are created and different gaze directionalities are mapped to different points on the manifold. In the work described in Ji and Yang (2002), the authors use an infrared light sensitive camera in order to make use of the bright pupil effect. In Magee et al. (2008), a template matching technique is used for localizing the eyes and, subsequently, eye regions are compared to each other in order to infer eye gaze directionality.

The above methods on eye gaze directionality estimation, either state or imply that head needs to be frontally rotated towards the camera, in order for eye gaze to be effectively reconstructed. Should this not be the case, dedicated hardware (like infrared-sensitive cameras) are necessary, in order to estimate gaze directionality. As a result, not a lot of works exist in bibliography regarding the issue of combining head pose and eye gaze, only with the usage of one ordinary camera, due, mainly, to the challenging nature of the problem. In the system described in Weidenbacher et al. (2006), the authors use elastic graphs for the estimate of the horizontal head rotation. For eye gaze estimation, they employ gabor filters. Based on training data, they build lookup tables that match the focus of attention with eye gaze and head pose calculations. Typical work on eye gaze and head pose estimation is the one described in Valenti et al. (2012), where the authors model heads with cylindrical shapes and, using the cylinder parameters, estimate the location of the eyes. These positions are projected on a normalized model view and are compared to reference positions in order to acquire eye gaze directionality.

3 Method Overview

Here, a system that functions in a monocular environment, without any special needs in terms of hardware, or knowledge of internal camera or set-up parameters is presented. These properties constitute this work independent of intrusive mechanisms or environment-related parameters. The system is designed to work in a HCI scenario, with a person sitting in front of a computer monitor, with a common [1] camera adjusted on top of it. The proposed system can handle large head translations, both parallel and perpendicular to the camera plane: by employing holistic information, it uses the appearance of the skin region, which is independent of its size and, by re-starting at frequent intervals, the system can re-initialize information related to local features in order to avoid error accumulation. Furthermore, it uses a combination of appearance information and local features, in order to use properties of each of the two families of methods, that alleviate the drawbacks of each other by fusing them in a common framework. Distance vector fields (DVFs) have been used in the past (Asteriadis et al. 2009a) for the purpose of detection, showing promising results in normal lighting and non-pretending conditions. Here, we propose their use as a local information flow (i.e. feature tracker), due to their property to encode efficiently local structures in conditions of poor lighting conditions. Furthermore, we propose a methodology for mapping facial feature positions to actual head rotation angles. More specifically, although our method introduces a facial feature tracker that uses DVFs, it does not utilize strict relationships among the positions of the features, but their relative movements within the co-ordinate system extracted from the face region of the user, at each frame. This makes our system robust to small tracking errors of the positions of features during tracking. Moreover, the system is able to re-initialize when certain conditions are met, not allowing, in this way, error accumulation due to tracking failure. For holistic appearance information, here, we introduce the use of a CNN architecture for head pose estimation, due to a series of properties that CNNs have. CNNs have been extensively used for digits recognition (LeCun et al. 1998a) and have similar topology with the standard multi-layer perceptrons (MLPs), but act as filters on the input image and subsequent transformations. More specifically, they build the optimal filters in order to extract the appropriate features of an image, necessary for the task of recognition in hand. Furthermore, thanks to frequent sub-samplings of the image, CNNs are less prone to image distortions and small shifts. This last property is very useful for the problem we want to tackle: Since it is our desire to create a system that would work under real conditions, it is expected that not all input images can be aligned in the same manner during testing. Moreover, using CNN's as static pose estimators, they are employed here as a refining step to infer frontal pose, before local information is employed, and a reference frontal pose frame is labelled. Fusion of the above two techniques has been shown to give very good results at estimating *yaw* and *pitch* angle of the head pose. *Roll* angle is more straightforward to detect. Figure 1 gives a brief overview of the methodology on head pose estimation. The system overcomes as many as possible of the common constraints met in bibliography, as described above, although there do exist methodologies overcoming a large amount of such limitations and, as will be seen in the experimental results section, its ability to achieve accurate results on non-posed datasets is grounded.

The perspective of combining estimates of Head Pose with Eye Gaze is also considered in this paper, building on the work presented in Asteriadis et al. (2011), where the core components of a system for validating prior hypotheses regarding user attention, were described. To this aim, a dataset combining Head Pose and Eye Gaze, specifically designed for the current research has been developed, in order to test the system's ability to infer gaze under large head translations. Furthermore, a commonly used dataset (Cascia et al. 2000) has been annotated regarding its participants' overall gaze towards the camera. Within this framework, we did not focus on inferring exact gaze estimation, but rather, we were interested in detecting degrees of confidence, through fuzzy logic, regarding hypotheses that a person is looking towards a specific point. A frame sequence dataset with clear annotations referring to focus of attention values, coming from both head rotation and eye gaze, to the authors' knowledge, was not publicly available at the time of conducting our experiments. However, a series of participants were asked to gaze towards certain points within a limited field of view, in order to test the ability of the overall system to generalize for unknown users, requiring no specific calibration. Moreover, the problem of combining the two cues in a common framework, using non-calibrated remote and monocamera mechanisms is not a thoroughly studied issue, due to the challenging nature of the problem and high variability among different subjects.

The structure of the rest of the paper is the following: In Sect. 4, we briefly discuss the ideas behind DVFs for facial feature detection, and we explain how we obtain personalized facial chrominance models for successful face tracking. Next, we give details regarding facial feature tracking and propose a model for estimating head rotation based on feature positions with regards to face boundaries. Section 5 presents Convolutional Neural Networks (CNNs) and we state the reasons why they were chosen for utilizing holistic information. The proposed architecture is presented, as well as the training and reasoning scheme. In Sect. 6, we explain how we fuse the two types of information. Section 7 extends

[1] Here, the word 'common' is used to distinguish from different types of web-cameras, such as wide or narrow angle, or infrared.

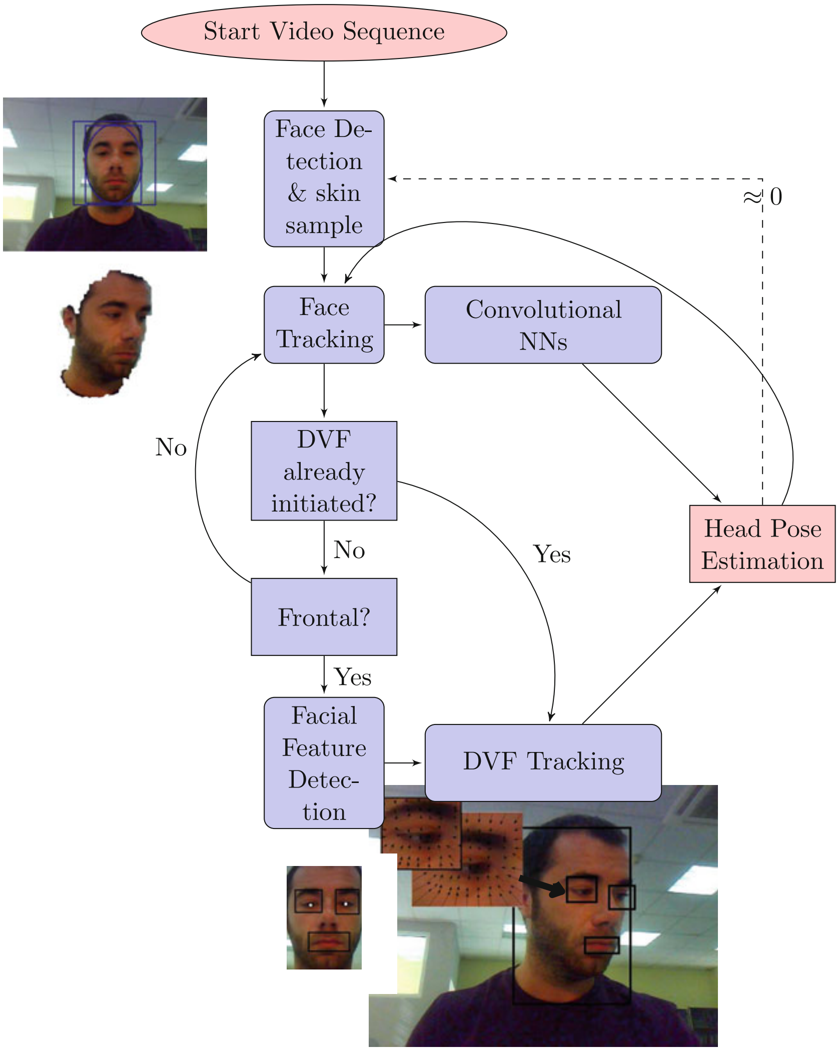


Fig. 1 Overview of the method: face detection and samples of skin are extracted at start-up. Subsequently, the face area is tracked and provides input to a series of CNN. If head pose vector is almost zero, facial feature detection and tracking are launched and provide local information. The algorithm can re-initialize at frequent intervals

the idea of visual focus of attention estimation by including eye gaze directionality, while Sect. 8 presents experimental results. A discussion and conclusions follow in Sects. 9 and 10, respectively.

4 Head Pose Estimation Using Distance Vector Fields

For the estimate of Head Pose using local information, we present a novel technique for Facial Feature Tracking. Based on tracking the face and facial areas (eyes and mouth), we extract information regarding the head's position, as well as its size on the image. The positions of the tracked features will be used for inferring head pose, as will be discussed in Sect. 4.3.

4.1 Distance Vector Fields

Distance vector fields for facial feature localization in frontal face images have been introduced in Asteriadis et al. (2007). Their applicability as detectors has been evaluated on two widely used datasets, the XM2VTS (Messer et al. 2003) and the BioID (Jesorsky et al. 2001), showing promising results on facial analysis. Specifically, the latter case consists of images taken under normal lighting conditions, depicting people posing various spontaneous expressions and, as the authors report, facial feature detection was very successful. To give an overview of the idea behind DVFs, they are image representations that encode shape geometry. This is done by attributing, to every image pixel, a vector pointing to the closest edge pixel, rather than just a scalar value (its distance), as is the case with distance maps. Considering a binary edge map of an image containing edge and non-edge pixels, its DVF is a vector field that is created by assigning to each pixel (i, j) the vector $\mathbf{v}$ pointing to the closest edge pixel (k, l) (Eqs. 1, 2).

$$\mathbf{v}_{ij} = [k - i, l - j]^T, (i, j) \in \underline{B} \tag{1}$$

$$(k, l) = \arg \min_{m,n \in \underline{E}} D((i, j), (m, n)) \tag{2}$$

where D is some metric (here, we use the euclidean distance) and $\underline{E}$, $\underline{B}$ are the sets of edge and non-edge pixels respectively. From the above equations, it can be noticed that every shape can be reconstructed by its corresponding DVF and, in pattern recognition problems, every pixel is given a 2-D vector that characterizes its position with regards to the closest edge pixels. The above has been utilized in Asteriadis et al. (2009a) for facial feature detection. The above properties make DVFs insensitive to luminance variations and, thus, appropriate for detection and tracking: what is encoded is the feature's shape geometry, which is not dramatically altered due to lighting differences between training and test data or throughout a frame sequence; eyes, eyebrows and mouth provide geometrical structures where basic objects (eye, eyebrow, mouth) are easily separable from the background (skin), at least to a certain degree, under variable lighting conditions. Consequently, resulting DVF structures have similarities even under different lighting circumstances, providing high robustness to the problem of detection. The situation during tracking is similar and can also cope with changes in lighting conditions. Consequently, tracking will not fail even if DVF structure changes due to lighting conditions, as long as these changes do not occur abruptly.

4.2 Tracking Facial Features Using DVFs

4.2.1 Facial Feature Detection

Prior to tracking, the face is initially detected using the Boosted cascade method (Viola and Jones 2001), followed by ellipse fitting, as was done in Asteriadis et al. (2007), for a more precise estimate of face boundaries. Subsequently, the face bounding box is brought to certain dimensions (we used $H_f = 130$ for height and $W_f = 105$ for width) and eye and mouth areas are detected using DVFs.

After detection of the position of the eyes and the mouth, the detected facial areas are brought back to their original dimensions, that agree with those of the face in the current image; the above is of high importance, since it constitutes the basic element that renders the proposed method invariant to face size (i.e. position of the user with regards to his/her distance from the camera). More details regarding the detection step can be found in Asteriadis et al. (2009a).

4.2.2 Face Tracking

The ideas discussed above will be used for tracking the detected facial features. To impose constraints on facial features positions, the bounding box of the face area is used at every frame, and the desired features are limited within this area, at a minimum of 50 % of their size. To this aim, each time a face is detected, a region of interest C_{skin} in the centre of the detected face area is used as the skin color predicate of the face, and the saturation[2] values of its pixels are calculated. Candidate facial pixels saturation values, for the subsequent frames, are expected to be within certain limits with regards to the mean s_M of the saturation values of C_{skin}. In this way, a binary image C_{fp} is created, marking with 1 those pixels x that satisfy the below hypothesis (face candidate pixels):

$$C_{fp} = \{x \in \Omega : \|s_M - s_x\| < T\} \tag{3}$$

where Ω is the set of all pixels belonging to the frame, x are candidate facial pixels, s_x their corresponding saturation values and T a threshold. Binary opening is subsequently applied to remove small areas, falsely attributed to skin regions.

The threshold T is automatically selected for each user separately, at the detection step (see Sect. 4.2.1), according to Eqs. (4) and (5):

$$T = \arg \min_{0.05<T<0.35} \left(\sum_{x \in \Omega} \delta(k_x) - Face_{size} \right) \tag{4}$$

with

$$k_x = \begin{cases} 0, & \|s_M - s_x\| \leq T \\ 1, & \|s_M - s_x\| > T \end{cases} \tag{5}$$

with δ being the Kronecker delta function and $Face_{size}$ the size of the face as defined by the ellipse containing the face at the detection step (see Sect. 4.2.1). The above procedure resulted in selecting a threshold automatically for each user, illumination conditions and face size with regards to the camera, thus, helping the system to adapt to any conditions in terms of lighting and user position. According to Eqs. (4) and (5), T is chosen based on the hypothesis below: it was expected that, at the first frame, the amount of pixels with saturation values close to the mean of C_{skin} is close to the amount of pixels that account for the real face region. The above procedure is summarized in Fig. 2, where the optimum threshold T to be used in Eq. (3) is based on the size of the face at the face detection step. To reduce the number of candidate facial pixels, the rules defined in Kovac et al. (2003) are used, in order for a map C_{sp}, extracted based on these rules, of candidate skin pixels to be built. According to these rules, a model for segmenting face regions is proposed. However, they are are quite relaxed, resulting to the inclusion of a lot of background regions.

C_{fp} and C_{sp} are combined using the logical AND operation, and binary closing (using a 10×10 structuring element, accounting for a 0.13 % of the frame size of the images where we conducted our experiments) is applied. This removes small holes like the eyes. Finally, the proposed method uses connected component labelling (Haralick and Shapiro 1992) and chooses the largest component as the final face region,

[2] Here, saturation is used, although different color channels (or combinations) can be used

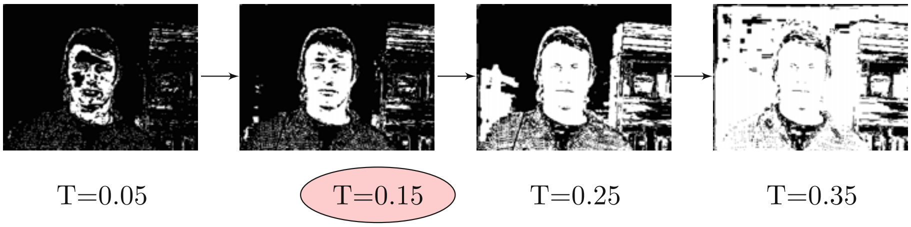


Fig. 2 Overview of selection of threshold T for segmenting face regions based on Eq. 3: threshold $T = 0.15$ was decided in this sequence, as the total number of pixels whose values are close to that of the initially selected skin region is close to the number of pixels belonging to the face region

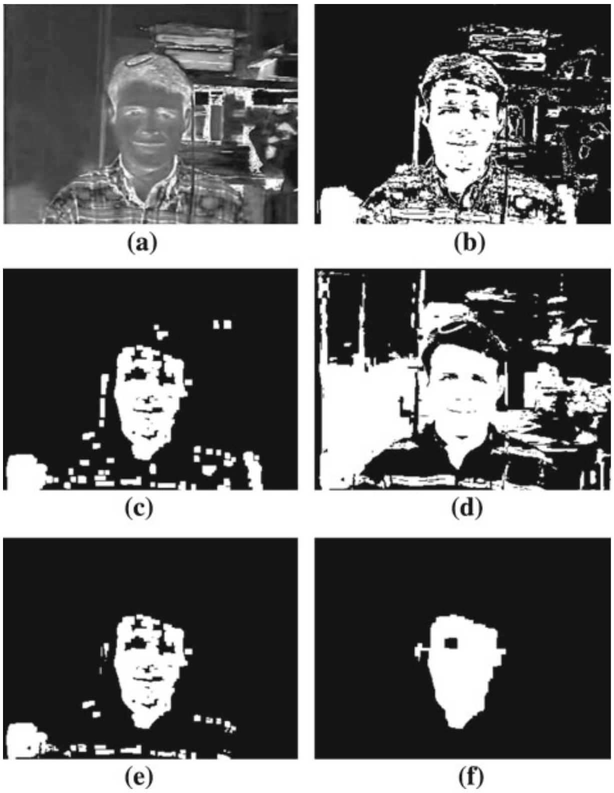


Fig. 3 **(a)** Original image saturation values; **(b)** thresholded saturation values; **(c)** face candidate pixels Cfp, extracted after morphological opening; **(d)** skin candidate pixels Csp; **(e)** face candidate pixels after logical AND between Cfp and Csp; **(f)** final face mask after morphological operations

while objects suddenly appearing at a distance not close to the position of the skin region of the previous frame, are also discarded. This last step ensures that background objects, falsely attributed to belonging to facial areas, are removed and, thus, only the face skin region remains. An overview of the steps of face tracking can be viewed in Figs. 2 and 3.

4.2.3 Facial Feature Tracking

Having defined the face region and having detected eye and mouth areas at start-up, tracking of facial features follows. In this paper, for local information, we introduce the use of Distance Vector Fields for facial feature positions tracking, thanks to the properties and analysis described in previous sessions. For mouth tracking, further image transformations have been employed.

Eye tracking For every eye, the DVF $f_{k,\mathbf{i}}$ of its region R_k is extracted in frame k and position $\mathbf{i}$, and the new position $\mathbf{i{+}p}$ of the feature in frame $k{+}1$, that minimizes the L_2 norm is extracted. The search region in frame $k+1$ is an extended area around the previous region in frame k. Here, we extended the area by ±33.3 % of the eye region, both along the horizontal and vertical axes. The motion vector $\mathbf{p}$ that gives the new position of the eye with regards to the previous frame is expressed by the following formula:

$$\mathbf{p} = \arg\min_{\mathbf{x}} \sum_{i \in R_k} \| f_{k,\mathbf{i}} - f_{k+1,\mathbf{i+x}} \|_2 \tag{6}$$

Every time an eye region is localized in a frame, the eye center (Asteriadis et al. 2009a) is found and the region position is updated so that it is centered around the eye center. Through experiments, it was shown that employing this update step of centering the eye area around the eye center helps to avoid erroneous tracking as, even if the DVF shows a tendency of slipping away from its correct trajectory, causing it to get to a position around the eye center, brings it back to the desired position. The advantage of using DVFs for eye tracking is that, as DVFs function on a frame-by-frame basis, eyes are tracked based on the geometrical similarity between two consecutive frames. Consequently, even if there are lighting variations throughout the sequence, as long as two consecutive frames are not dramatically different, in terms of the eye area shape, tracking would not fail.

Mouth tracking For mouth tracking, rapid lip movements, especially in the case where skin color cannot be easily distinguished from lips, cause DVFs to change very quickly. To tackle mouth tracking, a search area around the perpendicular bisector of the inter-ocular line segment is used to search for regions with high hue values and high horizontal edges concatenation. The combination of the two features maps is achieved by multiplying the binary edge values with the hue component values of the search area (see Fig. 4). The region with the highest mean of the resulting map is denoted as the new position of the mouth area. The mask used for checking

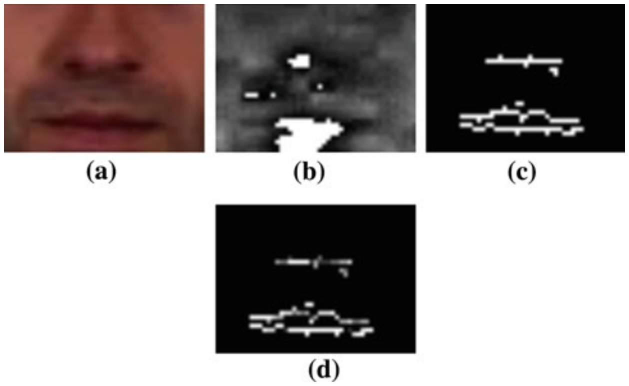


Fig. 4 **(a)** Mouth search area; **(b)** Hue component of mouth search area; **(c)** horizontal edge map of mouth search area; **(d)** Hue multiplied with horizontal edge map;

the means is of equal dimensions to that of the mouth at the detection step (brought back to real image dimensions).

Optimization of the tracker To further restrict the search areas for eye tracking and push results to obey to anthropometric measurements, it was assumed that the fraction between the inter-ocular distance and the vertical distance between the eyes and the mouth follows a normal distribution $f(\mu, \sigma^2)$ with μ and σ being the mean and standard deviation respectively. To accommodate each user's characteristics, μ was considered as the inter-ocular distance to eye–mouth distance at start-up, when the user is facing the camera frontally, while σ was extracted from training data of faces posing various head rotations [we used the dataset in Gourier et al. (2004)]. Thus, the extra factor corresponding to this distribution changes Eq. (6) as follows:

$$\begin{aligned} \mathbf{p} &= \arg\min_{\mathbf{x}} \left(\sum_{i \in R_k} \| f_{k,\mathbf{i}} - f_{k+1,\mathbf{i+x}} \|_2 f(d_{k+1,\mathbf{x}}; \mu, \sigma^2)^{-1} \right) \\ &= \arg\min_{\mathbf{x}} \left(\sum_{\mathbf{i} \in R_k} \| f_{k,\mathbf{i}} - f_{k+1,\mathbf{i+x}} \|_2 e^{\frac{(d_{k+1,\mathbf{x}} - \mu)^2}{2\sigma^2}} \right) \qquad (7) \end{aligned}$$

with $d_{k+1,x}$ standing for the fraction between the inter-ocular distance and the distance between the eyes midpoint and the mouth, at frame $k+1$, and translation x of the tracked eye with regards to its position at frame k. The above equation is used when tracking each eye separately and uses the coordinates of the other two features in frame k for estimating $d_{k+1,x}$.

4.3 Estimation of Yaw, Pitch, Roll Angles Based on DVF Tracking

Facial features' positions with regards to head pose contour play a key role for human perception of head pose (Gourier et al. 2004). Knowing the skin contour boundaries, as extracted from the face tracking process, the eye midpoint's $E_k = (E_{x,k}, E_{y,k})$ and mouth centre's $M_k = (M_{x,k}, M_{y,k})$ positions at each frame k, we calculate the yaw (pitch) angle as follows: it is modelled as the relative changes of the distance of E and M from the skin region rightmost and leftmost $C_{r,x}$, $C_{l,x}$ (uppermost and lowermost $C_{up,y}$, $C_{lo,y}$) boundaries' midpoint, with regards to a frame where the subject is facing the camera frontally. Linear regression has been shown to give a sufficient modelling mechanism for mapping the above measures to actual head rotations.

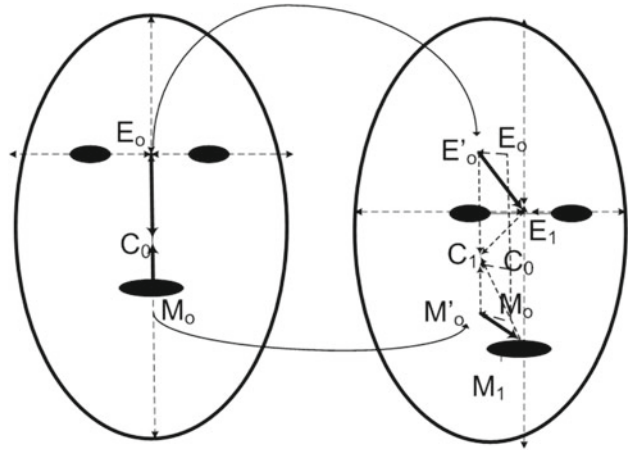


Fig. 5 Head pose vector estimation explaining the process described in Eqs. 8 and 9. *Dark arrows on the right image* are head rotation, as defined by eyes and mouth, separately.

The above are illustrated in Fig. 5 [3] and Eqs. 8 and 9, with $y_{DVF,k}$ and $p_{DVF,k}$ being the values of *yaw* and *pitch* at frame k.

$$\begin{aligned} y_{DVF,k} &= b_{1y} \times \left[\frac{(E_{x,k} - C_{x,k}) - (E_{x,0} - C_{x,0})}{d_{eyes,0}} \right] \\ &\quad + b_{2y} \times \left[\frac{(M_{x,k} - C_{x,k}) - (M_{x,0} - C_{x,0})}{d_{eyes,0}} \right] \qquad (8) \end{aligned}$$

$$\begin{aligned} p_{DVF,k} &= b_{1p} \times \left[\frac{(E_{y,k} - C_{y,k}) - (E_{y,0} - C_{y,0})}{d_{eyes,0}} \right] \\ &\quad + b_{2p} \times \left[\frac{(M_{y,k} - C_{y,k}) - (M_{y,0} - C_{y,0})}{d_{eyes,0}} \right] \qquad (9) \end{aligned}$$

with $C_{x,k}$ and $C_{y,k}$ being the vertical and horizontal coordinates of the face midpoint C_k at frame k, respectively, and b_{1y}, b_{2y} and b_{1p}, b_{2p} the regression weights used for fusing the information coming from the eye midpoint and mouth centre for yaw and pitch angles, respectively. Normalization with inter-ocular distance (as calculated at frame 0 where the user was looking frontally) $d_{eyes,0}$ is done to cater for scale variations between different subjects, while the inter-ocular

[3] E'_0 and M'_0 are the coordinates of E_0 and M_0, translated on the second frame so that C_0 coincides with C_1. This has been done, in order for a visual explanation of Eqs. 8 and 9 to be given.

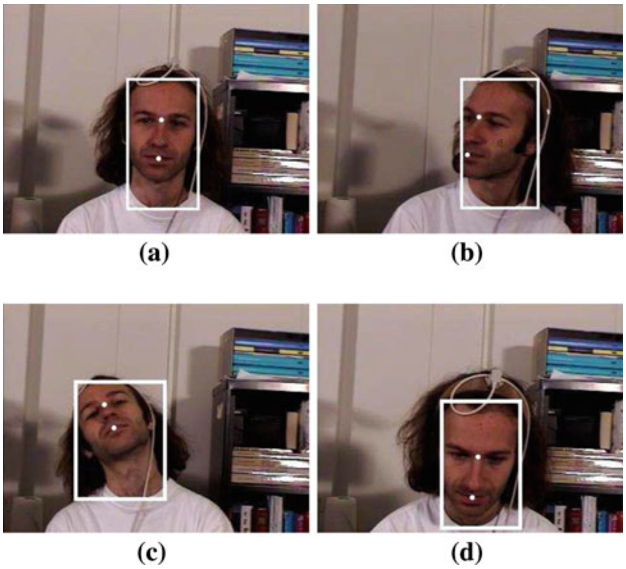


Fig. 6 Examples of skin regions boundaries and relation with the used features' positions

distance is re-calculated when the face is facing the camera frontally. By doing so, head rotation estimation is invariant to head movements along the z axis. To suppress noisy data, y_{DVF} and p_{DVF} can be convolved with a Nth order FIR filter (here, $N = 12$). In Eq. (8), the terms corresponding to frame 0 on the numerators are introduced as, in practice, the eyes midpoint does not always coincide precisely with the face midpoint when the user is facing the camera frontally. The above methodology can be intuitively explained as shown in Fig. 6.

Roll angle derives from the angle defined by the eye centres line segment and the horizontal axis (Horprasert et al. 1996; Ma et al. 2008). Here, weak perspective projection model is adopted (ignoring, in this way, the perspective effect). The values are again filtered as is done for the *yaw* and *pitch* angles. For maximizing robustness, the algorithm restarts, at frequent intervals, if certain conditions are met (here, when both horizontal and vertical rotations are below 3°).

4.4 Comparison Between DVF Tracking and Optical Flow

For evaluating the tracking performance described above, we conducted an experiment on the first session of the HPEG (Asteriadis et al. 2009b) dataset. The sequences were recorded using a simple webcamera, while the background is uncontrolled, with intense human action taking place in many cases, and the length of each sequence is 200 frames. In this session, participants were asked to move freely towards any direction and speed they wanted. The lighting conditions were those of an office environment. Examples from the dataset can be seen in Fig. 7. To assess the effectiveness of using DVFs for facial feature tracking, we compared the tracker described in Sect. 4.2.3 with the optical flow method.

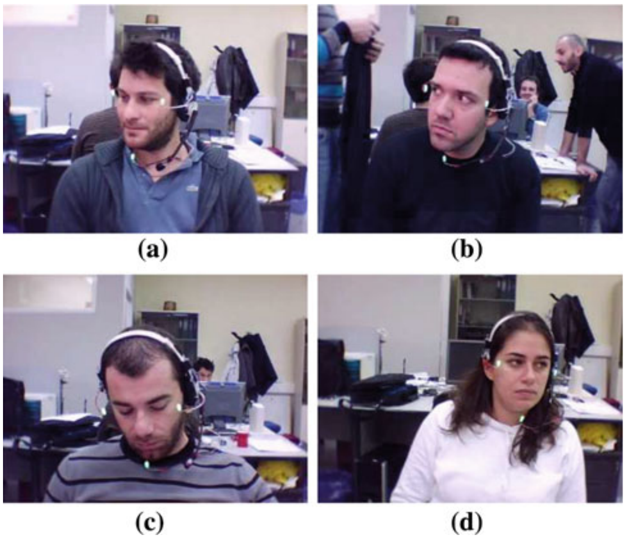


Fig. 7 Examples from the HPEG dataset

After detecting the facial features, as described in Sect. 4.2.1, we used the standard Lucas–Kanade algorithm to track the eyes' positions, while, for mouth tracking, the same procedure was followed as the one described in Sect. 4.2.3. The windows used for tracking were the same as the ones calculated at the detection step (after scaling to match their original dimensions) and, similar to DVF tracking, search regions were limited only in the face skin area bounding box. Furthermore, here as well, we introduced the gaussian term used in Eq. (7) [see Eq. (10)] in order to reinforce tracking of the most possible positions of the eyes and, after each frame, the tracked area was centered around the eye centre.

$$\mathbf{p} = \arg\min_{(u,\,v)} \left(\sum_{i \in R_k} \|(I_x \cdot u + I_y \cdot v + I_k)\|_2 e^{\frac{(d_{k+1,\mathbf{v}}-\mu)^2}{2\sigma^2}} \right) \qquad (10)$$

with $\mathbf{v} = (u, v)$ being the feature's translation between two consecutive frames, I_x, I_y, I_k the image derivatives at coordinates (x, y) and frame k. As before, $d_{k+1,\mathbf{v}}$ stands for the fraction between the inter-ocular distance and the distance between the eyes midpoint and the mouth, at frame $k + 1$, and translation (u, v) of the tracked feature.

Table 1 shows the RMS[4] and MAE errors when using Optical Flow tracking and DVF tracking on the HPEG dataset. It can be seen that DVF—under the conditions considered in our experiments—is more appropriate for head pose estimation using facial feature tracking, as optical flow searches for similarities of chrominance between consecutive frames (which might have more than one solutions), while

[4] RMS is also calculated, here, as a stricter criterion than the mean absolute error (MAE), since it 'punishes' large errors

Table 1 Head pose estimation using feature tracking with optical flow and distance vector fields

	Optical flow			DVF		
	RMS	MAE	STD	RMS	MAE	STD
Yaw	8.41°	6.65°	7.93°	6.65°	5.18°	5.41°
Pitch	5.68°	4.42°	4.09°	5.59°	4.30°	4.04°
Average	7.05°	5.53°		6.12°	4.74°	

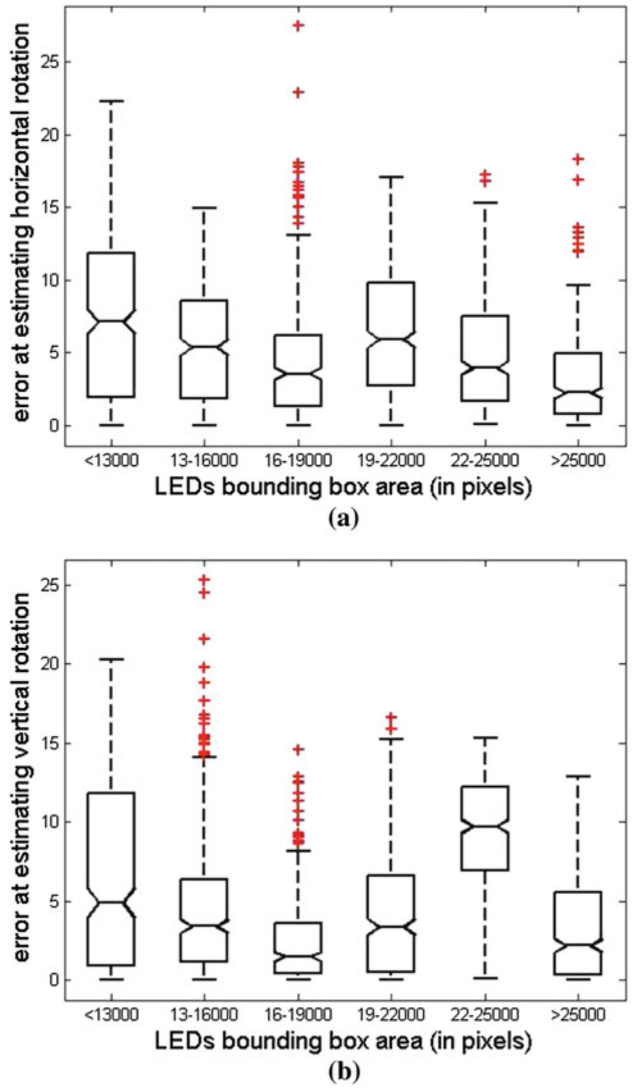


Fig. 8 MAE error for horizontal (**a**) and vertical (**b**) rotations, with regards to bounding box area, defined by LEDs positions

DVFs search for similar shapes, as the position of each pixel and its relation with their neighboring shapes is encoded.

4.5 DVF Tracking and Scale Invarianvce

The HPEG dataset consists of a lot of variations in terms of participants' distances from the camera. Figure 8 shows boxplots of medians depicting MAE at estimating horizontal and vertical rotation for different sizes of facial areas[5]. It can be seen that there is no evident relation between detected face size and resulted error at estimating head rotations.

5 Head Pose Estimation Using Convolutional Neural Networks

For utilizing holistic information for head horizontal and vertical rotation estimation, in this paper, the use of a certain type of Neural Networks, the so called CNNs, is proposed. The main reason for this choice is that CNNs require a relative small number of parameters to be learnt, thanks to the property of weight sharing. As it will be seen below, a large number of classifiers was built, taking as input certain poses. This limitation reduced the number of training images and, thus, building neural classifiers with a limited number of free parameters would help avoid the problem of overfitting. Second, CNN are known for their ability to be trained on non-aligned datasets (LeCun 1989); this property is ideal for not posed environments, with complex background and intra-person variability, where testing and training data are not expected to be perfectly aligned. Third, CNNs take advantage of spatial relations among features, by extracting local features, restricting the receptive fields of hidden units to be local (LeCun 1989; LeCun et al. 1998a).

Convolutional Neural Networks are bio-inspired networks whose applicability is known for character recognition (LeCun et al. 1990). Research on the potentiality of Convolutional Neural Networks as face pose estimators is presented in Osadchy et al. (2007), where the authors train CNNs that map input images X to low-dimensional spaces. The "face region" on these spaces is denoted with a manifold, parameterized accordingly to account for different poses. In our work, following a different architecture and reasoning scheme, we create models that calculate the exact head rotation angles by fusing a series of pre-trained CNNs. For training, a smaller amount of data is required, captured under a finite number of head rotations. Moreover, the total number of free parameters in our networks is less than 26,000, while the architecture described in Osadchy et al. (2007) necessitates training of 63,493 weights and kernel coefficients. A comparison (on different datasets) of the two systems has shown that the proposed work can approximate the head rotation angle with better accuracy. In Osadchy et al. (2007) in 89 % of cases horizontal rotation estimates is within 15° from the corresponding ground truth value, while, the proposed scheme achieves an overall 93.1 % for the same criterion. In Osadchy

[5] As, in HPEG dataset, no depth information is given, here, we approximated distance from camera through the area formed by LEDs positions when the subject is facing frontally

Table 2 Interconnection table of subsampling layer $S2$ (rows) with convolutional layer $C3$ (columns)

	0	1	2	3	4	5
0	X				X	X
1	X	X				X
2	X	X	X			X
3		X	X	X		X
4			X	X	X	X
5				X	X	X

Each X stands for a connection

et al. (2007), no results are reported regarding vertical head rotations.

In a generic image recognition problem, an image is used as input in the first layer of a CNN and filters' parameters are learnt for alternating layers of convolutions and subsamplings. The above guarantees the following: first, the ability of the network to learn robustly from a small amount of training data, at reasonable time, is feasible, as the number of the free parameters is significantly reduced, since feature maps' units share the same weights. Second, subsampling renders the network more resisting to small distortions or translations of the input image. Also, alternating convolutional and subsampling layers, makes it easy to form layers that start with detecting simple features (e.g edges, corners) and end up to combining features with each other in subsequent layers, to achieve information coming from spatial combinations of them. A typical CNN is the one described in LeCun et al. (1998a), where the authors have built an architecture with 60,000 free parameters for the purpose of digit recognition.

5.1 Proposed CNN Architecture

The architecture employed in the proposed scheme, is a 6-layer CNN, with the layers being in the order $C1$, $S2$, $C3$, $C4$, $F5$, $F6$ (output), which is a 2-element vector. The convolutional layers use 7×7 kernels, while the sub-sampling layer uses a downscale factor of 2. Layers $F5$ and $F6$ consist of 10 and 2 (output) fully connected typical neurons, respectively. We used 6 feature maps for layers $C1$ and $C3$, and 80 for layer $C4$. The main difference from a typical CNN, here, is that, between the third and forth layers, we did not include a subsampling layer. This gave better results, since the input images' resolution is 32×32 and, subsequent sub-samplings has been shown to cause significant loss of information at discriminating between similar poses, as will be explained later. Furthermore, similar to (LeCun et al. 1998a), we used a non-fully connected interconnection between layers $S2$ and $C3$ (Table 2). A schematic representation of the network can be seen in Fig. 9. The total number of free parameters of the network is 25,779. More specifically, layer $C1$ consists of 300 free parameters (49 kernel weights per feature map and a bias), layer $S2$ has 12 trainable weights (one multiplicative and one additive bias per map) and, similar, $C3$ has 1035, $C4$ has 23600, and layers $F5$ and $F6$ 810 and 22, respectively.

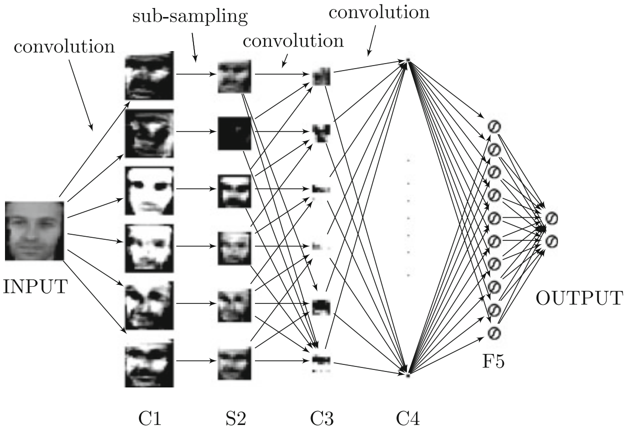


Fig. 9 Convolutional neural network architecture for head pose estimation

5.2 Training Procedure

Training was done using stochastic Levenberg–Marquardt (LeCun et al. 1998b), and the hyperbolic tangent sigmoid function was used as activation function throughout the network. For training, the face dataset in Gourier et al. (2004) was used. The dataset consists of static images of people posing various head rotations in the *yaw* and *pitch* space, at certain angles. In order to include more variability in the trained models, we shifted the training images by one until three pixels towards all directions, and included the new images in the training dataset. To further increase training data, all images have been mirrored around the vertical axis. For training, we created a pose space consisting of classes centered at pitch angles $\{-60°, 0, 60°\}$ and yaw angles $\{-90°, -45°, 0, 45°, 90°\}$, thus, a total of 15 combinations of yaw and pitch (Fig. 10) This coarse discretization was preferred to a more thorough one, as, much different poses are much easier to distinguish from each other. Furthermore, as will be discussed in the following subsection, using a finer quantization in the classes space, would eliminate the risk of jumping to erroneous classes. Each class contained images with ±15° (or ±30° in the case of class centres with ±60° pitch angles) deviation from the corresponding class centre. Thus, the total of training data, depending on the subclassifier and the position of the training classes on the pose space, varied from 29,400 to 52,920 images, and we applied early stopping at the training procedure, to avoid overtraining (Sarle 1995). We trained one CNN for each combination

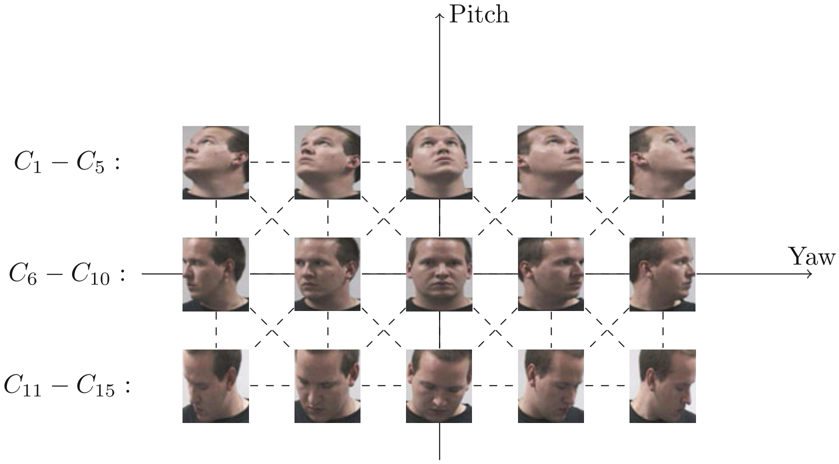


Fig. 10 Head Pose classes used for training the convolutional neural networks (CNNs). Each trained CNN is denoted with a *dashed line*

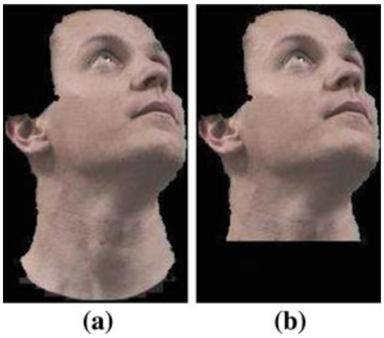


Fig. 11 **(a)** Extracted skin region, **(b)** cropped skin region, as used for training

of neighboring classes, resulting in a total of 38 classifiers. Consequently, at this stage, training was done using pairs of images belonging to different, but adjacent points, in the pose space, and the target values of the output were $\{-1, 1\}$ or $\{1, -1\}$, depending on the training pair. Prior to training, all images were normalized to have zero means and standard deviation equal to 1. Furthermore, to reinforce uniformity in our data, all input face images were cropped so that the skin region's length is 1.3 times larger than width, before setting them to 32×32 pixels (Fig. 11).

5.3 Estimation of Yaw and Pitch Rotation Using Trained CNNs

For estimating head pose, solely based on the CNN classifiers, the skin bounding boxes are cropped according to Fig. 11 and, after being downscaled to 32×32 pixels, they are normalized in order to have zero means and standard deviation equal to 1, handling, in this way, lighting variations throughout sequences. Using *yaw* and *pitch* information from the previous frame, all n networks considered include the class C_c whose center is closer to the *yaw* and *pitch* values of the previous frame. In this way, only a subset of CNNs is used at each frame, constituting the system faster and more reliable, as the possibility of erroneous classification is reduced. For example, if *yaw* and *pitch* values of the previous frame are closer to the class C_8 centre, the CNNs employed are the C_8–C_3, C_8–C_2, C_8–C_4, C_8–C_7, C_8–C_9, C_8–C_{12}, C_8–C_{13}, C_8–C_{14}. In the general case, let's consider the first output of each network as out_1, and the second one, as out_2. Here, we have used the difference between the two outputs to compare class C_c against neighboring classes and, thus, after each frame is processed, the overall output is a vector consisting of 8 elements (out_{up}, out_{down}, out_{left}, out_{right}, $out_{up,left}$, $out_{up,right}$, $out_{down,left}$, $out_{down,right}$, depending on the topological relation of class C_c against each class it is compared against). If class C_c is at the boundaries of the pose space, dummy classifiers giving output equal to 2 are hypothesized (setting $out_1 = 1$ and $out_2 = -1$ for the existing C_c and non-existing class, respectively) for the missing hypothesized classes. The final estimate of the yaw (or pitch) angle is done by employing regression models including elements of the vector, as well as the centre of class C_c. At estimating the yaw angle, out_{up} and out_{down} were excluded from the model. Similar, out_{left} and out_{right} were omitted at estimating the pitch angle, thus, giving 6-element vectors (as well as the centre of class C_c) at the regression model. It should be noted here that, as faces rotated parallel to the image plane have not been used during the training procedure, all input faces' bounding boxes have been rotated so that the inter-ocular line is parallel to the horizontal axis.

6 Fusion of Holistic and Local Information for Head Pose Estimation

Based on the observation that our local and holistic techniques have different levels of reliability, depending on the context of the interaction, in this paper, we take this parameter into account during fusion. For this reason, we used Bayesian modality fusion, so that reliability of each cue is

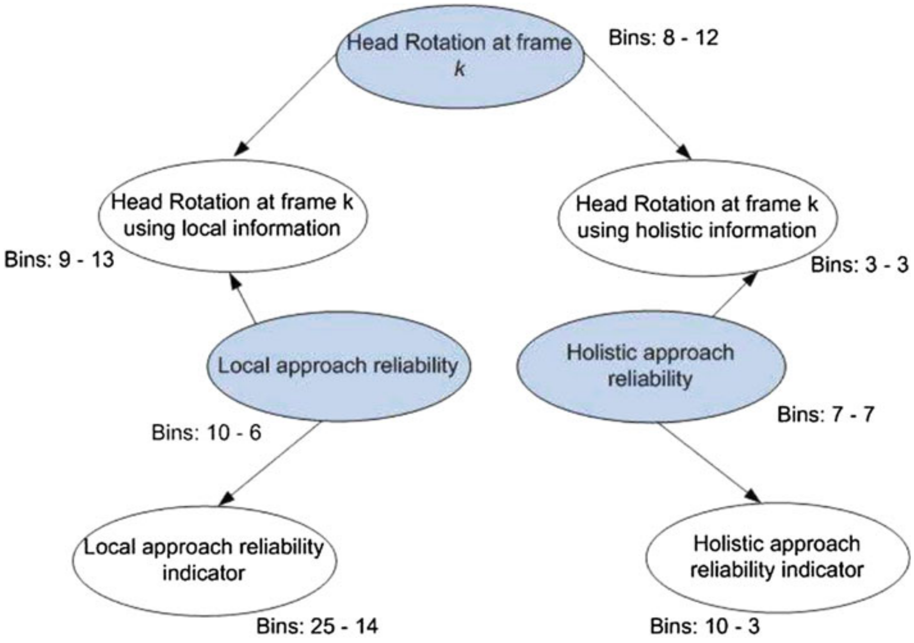

Head Rotation at frame k
Bins: 8 - 12
Head Rotation at frame k using local information
Bins: 9 - 13
Head Rotation at frame k using holistic information
Bins: 3 - 3
Local approach reliability
Bins: 10 - 6
Holistic approach reliability
Bins: 7 - 7
Local approach reliability indicator
Bins: 25 - 14
Holistic approach reliability indicator
Bins: 10 - 3

Fig. 12 Bayesian network architecture used for fusing local with holistic technique (*number of bins* for horizontal–vertical rotation)

modelled, according to the phase of the interaction. The proposed architecture is explained in the next subsection.

6.1 Bayesian Networks for Head Rotation Estimation

In literature, the term *Bayesian Network* refers to a directional acyclic graph that represents the joint probability distribution for a set of random variables (Horvitz et al. 1988; Jensen 1996). In such networks, nodes are random variables and arcs stand for the statistical dependencies among pairs of nodes. Such dependencies, in a bayesian network model deterministic influences among the variables.

In this paper, estimated head rotation (horizontal or vertical) has been considered to be a random, observable variable. On a second level, true head rotation affects visual systems' outputs (observable variables), which are also affected by each modality's reliability (hidden node). Reliability varies depending on the context of the interaction. As modality reliability cannot be observed during the sequence, an indirect way to infer it, is through measurable variables, correlated with it, namely modality reliability indicators. Figure 12 shows a schematic representation of the employed network, which is an adaptation of the scheme proposed in Toyama and Horvitz (2000). Graph nodes represent variables of interest, with the white ones corresponding to observable quantities and those with grey color corresponding to hidden variables. Node *Head rotation at frame k* is the final output variable (target node). The mean of the integral of the probability distribution of the target node gives the final estimate of head rotation.

6.2 Local Information Reliability Indicator

As reliability indicator for local information, here is considered the fraction between vertical distance between mouth and eyes with eye distance:

$$rel_{DVF,k} = \frac{\| Eyes_{middle,k} - Mouth_{middle,k} \|}{\| Eyes_{right,k} - Eyes_{left,k} \|} \tag{11}$$

The values taken by this parameter are correlated with expected rotations. For example, arbitrary values would be linked with a low degree of reliability.

6.3 Holistic Information Reliability Indicator

For each instance of the estimate of horizontal and vertical rotation with CNNs, at frame k, the confidence value modelled as reliability indicator derives from Eqs. (12) and (13) for horizontal and vertical rotations, respectively:

$$rel_{CNN,y,k} = 1 - \frac{|y_{CNN,k} - m_{y,k:k-n+1}|}{std_{y,k:k-n+1}} \tag{12}$$

$$rel_{CNN,p,k} = 1 - \frac{|p_{CNN,k} - m_{p,k:k-n+1}|}{std_{p,k:k-n+1}} \tag{13}$$

with $m_{y,k:k-n}$, $std_{y,k:k-n}$ and $m_{p,k:k-n}$, $std_{p,k:k-n}$ being average values and standard deviation for horizontal and vertical rotation, respectively, for temporal windows of the n previous frames (here, we used $n = 5$). The values of reliability indicators, under normal circumstances, are within specific values but, when the corresponding modality reliability is low, they can take arbitrary values (Liu et al. 2003), due to sudden peaks or valleys, usually attributed to face tracking failure, resulting in arbitrary facial area and subsequent estimates of rotation.

6.4 Network Parameters

Network training was based on learning conditional probability tables for the nodes which were learnt by quantizing variables into bins. The discretization that gave the optimum trade-off between variance and bias can be seen in Fig. 12. Tables parameters are learnt by simply counting (and normalizing) those frames where two events co-occur.

For maximizing the possibility of accurate facial feature localization (Asteriadis et al. 2009a), during the first frames of a video sequence, only CNNs are applied, and facial feature detection is applied only when CNN output for yaw and $pitch$ is below a certain threshold (thus, frontal face and eye/mouth detection is more accurate—see Fig. 1).

7 Eye Gaze Estimation

For estimating eye gaze, we propose a technique that models the face area around the eyes (Fig. 13) by a cylindrical shape with pose parameters equal to $p = [\omega_\chi, \omega_y, \omega_z, t_\chi, t_y, t_z]$, where $\omega_\chi, \omega_y, \omega_z$ the cylinder rotation angles and t_χ, t_y, t_z the translation parameters. As the input image is solely the area around the eyes, we considered t_χ and t_y to be equal to zero, while t_z only needs to be approximated. Similar, ω_χ (pitch angle) is considered zero here, and ω_z (roll angle) is also considered to be null, since it can be eliminated by rotating the image, as the eye positions are known (the image is rotated so that both eyes lay on the same level). ω_y is the horizontal angle (yaw), as calculated from the head rotation estimation methodology. In our experiments, we considered that the camera focal length is $f = 500$ (in pixels), while different considerations have also been made, as shall be seen in the experimental results section, giving us similar results.

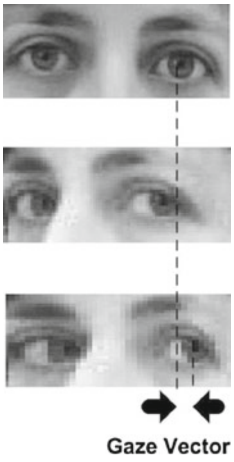


Fig. 13 Extraction of Eye Gaze Vector. The eye position in the warped image (*bottom*) is compared to that of the frontal position, after yaw angle has been removed

Subsequently, the input image is warped so that ω_y is zero (Fig. 13), similar to (Begley et al. 2008; Valenti et al. 2012). From the two new positions of the eye centers, the one used is that of the eye that is closer to the camera, as the error caused by perspective projection is smaller. Its position on the horizontal axis is then compared to that of a frame when the person is looking frontally, in order to estimate the gaze vector. The resulting value is normalized with the inter-ocular distance, as calculated at a frame when the person faces the camera frontally, in order to handle scale variations.

8 Experimental Results

8.1 Estimating Head Pose: Results on the Boston University Dataset

In order for the proposed scheme on head pose estimation to be comparable with existing methodologies in bibliography, experiments were carried out on the BU (Boston University) dataset. It consists of 45 image sequences of 200 320 × 240 frames each, and contains 5 people, each of them appearing in 9 videos. As they appear in the database, the participants were allowed to move freely, along any direction. Typical example frames of the database can be seen in Fig. 14. Three types of experiments were conducted: head pose estimation using DVFs and CNNs separately, as well as using fusion of both modalities. Table 3 shows the RMS errors and standard deviation when using DVF, CNNs and fusion of both. We also compare results with other methods in literature, which use the RMS error on the same dataset. Similar, Table 4 shows the same errors using the MAE, as well as comparisons with works considering the same criterion. Training of regression

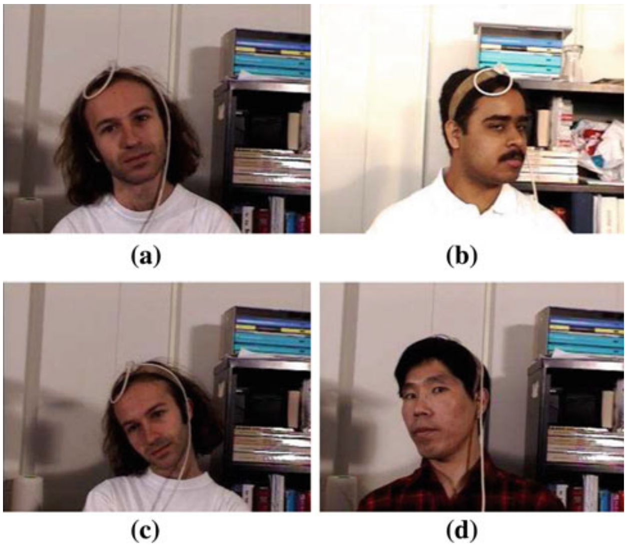


Fig. 14 Example frames from the Boston University dataset

Table 3 RMS error results on the BU

	DVF Error STD	CNN Error STD	DVF+CNN Error STD	Valenti et al. (2012) Error STD	Sung et al. (2008) Error
Yaw	5.72° 4.90°	7.07° 6.45°	5.66° 5.02°	6.10° 5.79°	5.40°
Pitch	4.89° 4.04°	5.81° 5.13°	4.80° 4.2°	5.26° 4.67°	5.60°
Roll	3.56° 3.05°	–	3.56° 3.05°	3.00° 2.82°	3.10°

Table 4 MAE error results on the BU

	DVF	CNN	DVF+CNN	Morency et al. (2010)	Lefevre and Odobez (2009)
Yaw	4.39°	5.63°	4.29°	4.97°	4.40°
Pitch	3.87°	4.74°	3.74°	3.67°	3.30°
Roll	2.61°	–	2.61°	2.91°	2.00°

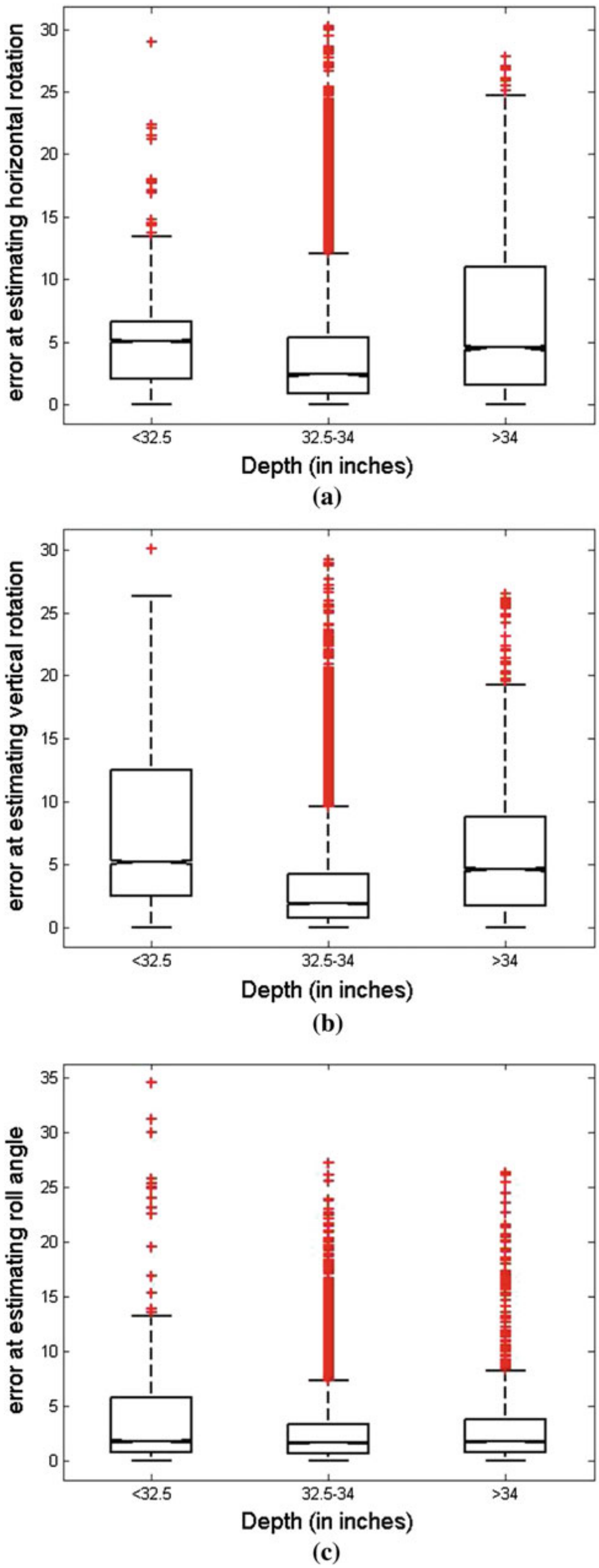
models (see Sect. 4.3) and the bayesian network was done by following a leave-one-subject-out cross-validation method for each user.

It can be seen from the results, that our method on head pose estimation is comparable and, in cases, performs even better than published results. However, the advantage of the proposed scheme is that no a priori knowledge regarding camera parameters or distance of the user from the camera is needed, or limitation that the user's face bounding box size is stable. Figure 15 shows boxplots of medians corresponding to errors for horizontal, vertical and around z-axis rotation estimation with regards to person's distance from the camera. The data were extracted using all image frames in the database, along with depth information for each of them. Although the vast majority of frames correspond to people having distance from the camera between 32.5 and 34 in., there were cases when people would come closer or move further from the camera, thus, resulting to movements along the z-axis. From this figure, it can be concluded that the impact of position of the subject on the z-axis does not have a significant impact on the error at estimating yaw, pitch and roll head rotations.

Methods in Valenti et al. (2012) and Sung et al. (2008) use head models, making assumptions regarding, either camera focal length or user-camera distance fixation (Aggarwal et al. 2005), while, in the method in Lefevre and Odobez (2009), the authors use fixed windows around trained features to estimate head pose (thus, scale variations due to translations perpendicular to the image plane, are not handled), and the head has to face the camera frontally at start-up. Here, using CNNs at start-up, we do not limit the approach to functioning only after a frontal pose is detected but we can have reliable estimates of arbitrary initial head rotation angles. When CNN method declares very small head rotation, facial feature detection and DVF tracking are launched and, only then is frontal pose inferred. Also, as the proposed approach takes, as input, face areas, and can re-initialize when certain conditions are met, it can track with a high degree of reliability head movements that employ translations perpendicular to the image plane, leaving a lot of freedom for variability at movements. Figure 16 shows typical examples of estimated rotation angles, together with the ground truth data, on three different video sequences.

Furthermore, the proposed system is designed to handle effectively near profile poses: the tracked eye and mouth areas are always forced to stay within the limits of the face region, at least at a minimum of 50 % of their size. Consequently, even if the eye centre is not visible and the method falsely considers the eye centre outside the face area, the algorithm uses its position in the previous frame, so that eye areas are kept within the face area. Practically, this means that, when the eye is occluded due to large rotational head movements, its position will be considered to be the one corresponding to the last frame, when it was not occluded. This should introduce some error in the case of the local technique alone but, it would only be evident in extreme poses. Using CNN head pose estimation, trained classifiers did actually contain cases of 90° of horizontal rotation.

The system relies a lot on re-initialization when certain conditions are met. Thanks to this, face tracking uses new face chrominance samples each time the system is re-initialized. In this way, the algorithm adjusts to possible changes in lighting conditions. Furthermore, as demonstrated in Sect. 4.2.2, expected face size is also used, catering for changes in scale. This method of system recovery increases robustness and contributes to eliminating error accumulation. Also, all objects classified as skin regions, but at a distance not close to the expected position of the face, are automatically discarded to cater for background color noise (see Sect. 4.2.2). The above limitations and rules have been very essential for the robustness of the component of face tracking. One issue that may arise during face tracking is that face boundaries are not very strict from frame to frame. This is one of the major reasons why CNN have been employed for utilizing holistic information. System automatic recovery is also essential for facial feature tracking, since every time face detection occurs, facial features are re-detected and tracking begins from the new positions. Furthermore, the Gaussian model that describes the geometrical constraints among the positions of the eyes and the mouth favours

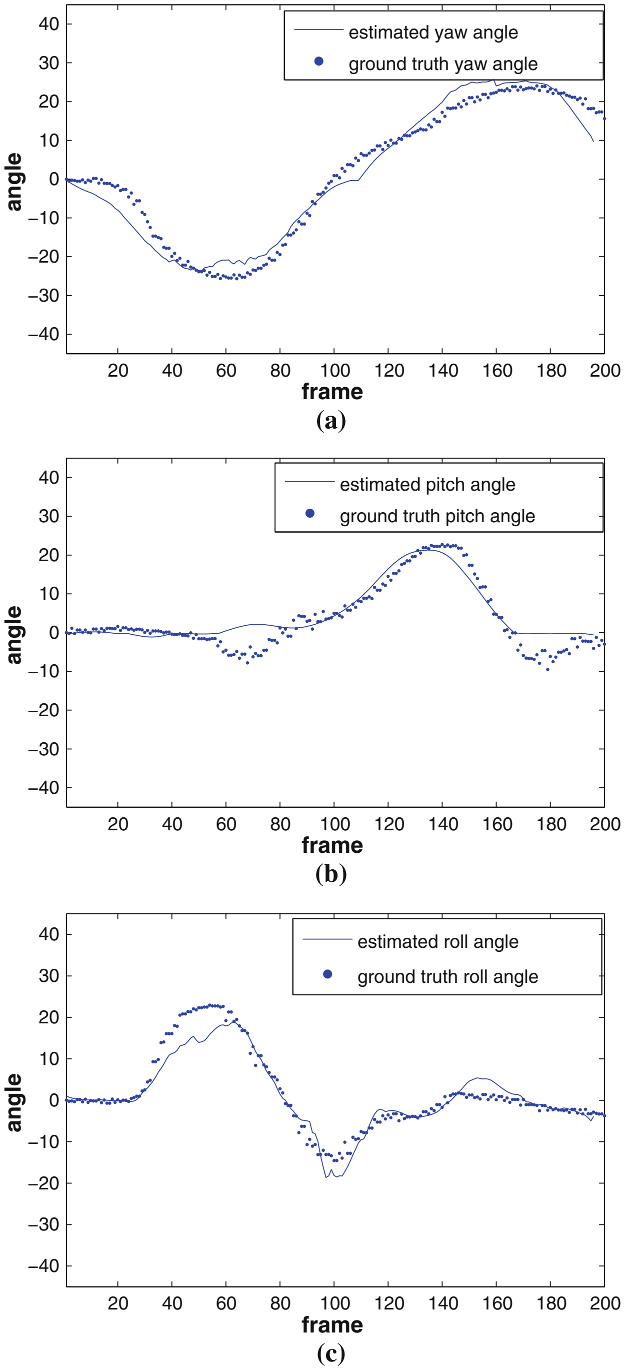


(a)

(b)

(c)

Fig. 15 MAE error for horizontal **(a)**, vertical **(b)** and roll **(c)** rotations, with regards to depth information on the Boston University dataset



(a)

(b)

(c)

Fig. 16 Estimated head pose angles and corresponding ground truth

expected positions of the features. The impact of roll angle is also explored in Fig. 17 where local, holistic and combined information errors for horizontal and vertical rotations against true roll angles of the face are shown. It can be seen that, for almost the whole range of roll angles, estimation errors are similar. It only fails for a given series of frames, where face tracking was unsuccessful, leading to unsuccessful estimates.

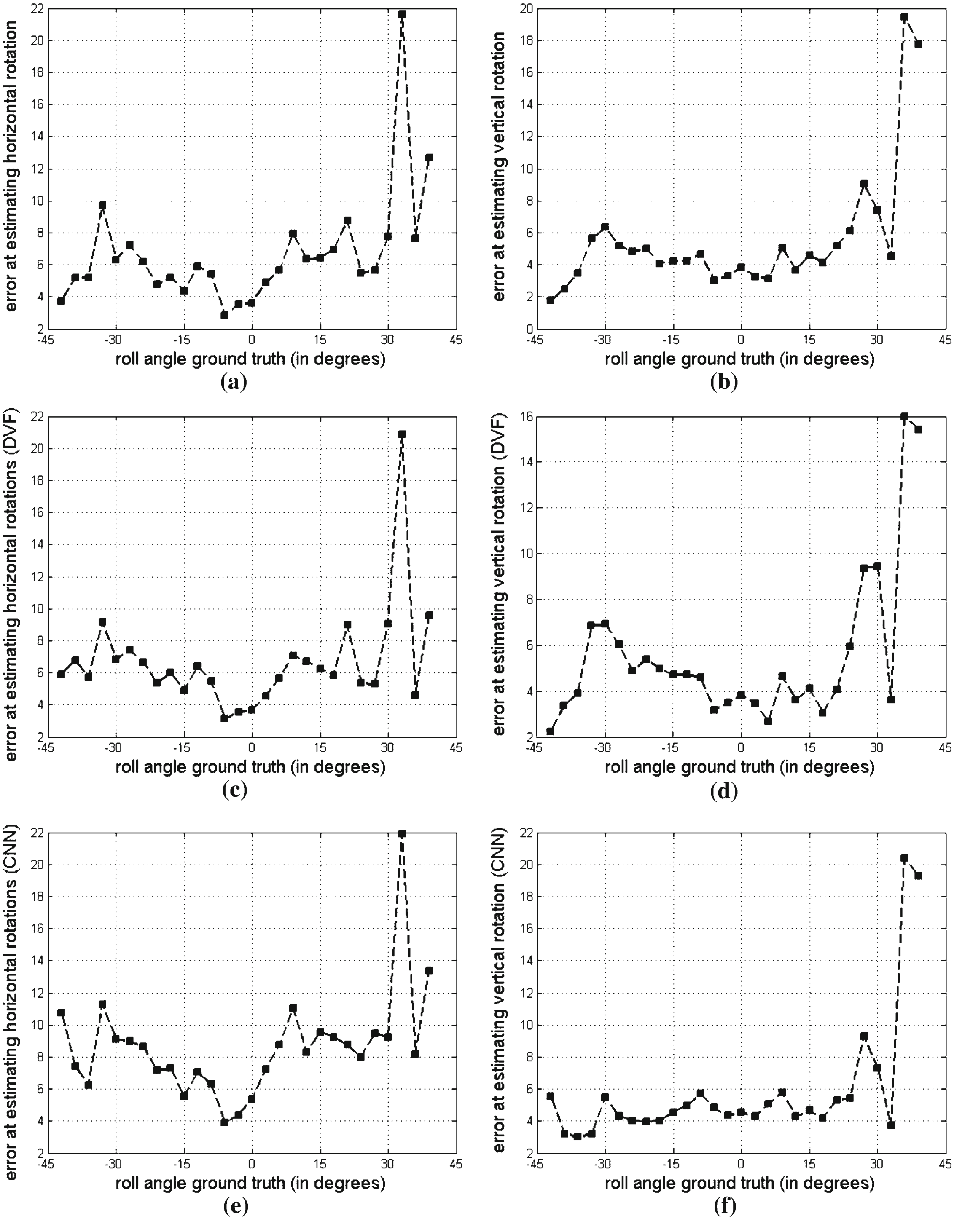


Fig. 17 Mean absolute error at estimating horizontal and vertical rotation using fused (**a**, **b**), local (**c**, **d**) and holistic (**e**, **f**) information, against true roll angles

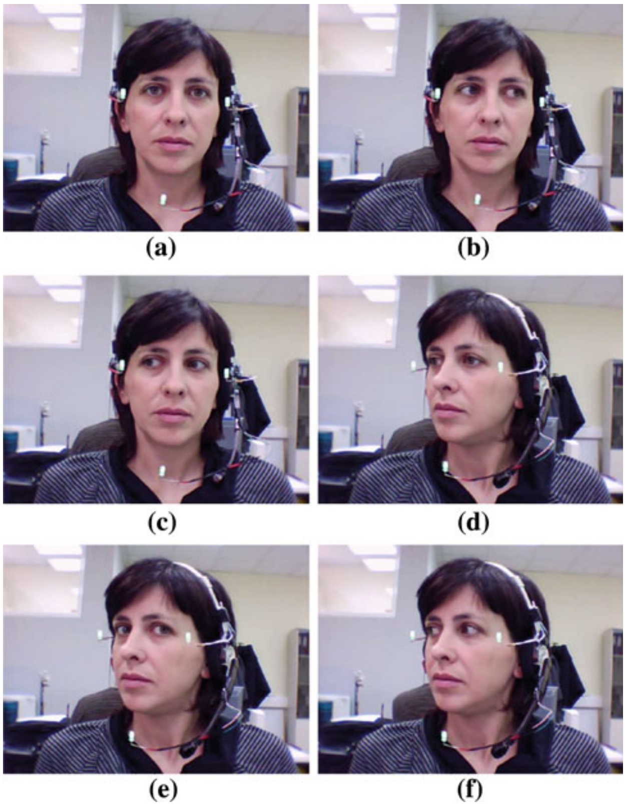


Fig. 18 The six gaze classes of the second session of the HPEG dataset

Table 5 Eye Gaze directionality on HPEG session B

	Frontal head rotation			Right head rotation		
	PH	R	L	PH	R	L
PH	10	0	0	6	0	4
R	0	10	0	1	8	1
L	0	0	10	0	0	10

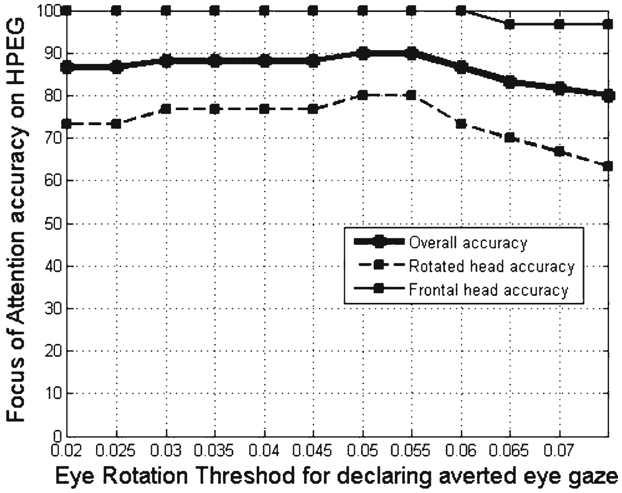


Fig. 19 Overall focus of attention estimates for different eye gaze thresholds T

8.2 Estimating Visual Focus of Attention using Head Pose and Eye Gaze

8.2.1 Estimating Visual Focus of Attention: Results on the Head Pose and Eye Gaze dataset

Session B of the HPEG[6] dataset consists of 10 frame sequences where the participants pose pre-determined head rotations combined with eye gaze directionality variations. More specifically, volunteers were first asked to look directly at the camera, then turn their eyes left and right, keeping their head at a frontal position. Subsequently, the volunteers would turn their heads to the right, with eye gaze directionality parallel to head rotation. Keeping the head stable, eyes were rotated towards the camera and, finally, the eyes were turned towards the other direction. The above instances resulted in 6 different classes. Figure 18 shows typical instances of the above classes. To the authors' knowledge, no publicly available dataset, including ground truth for both head rotation and gaze directionality combined, existed during writing this paper, while, the development of a dataset with spontaneous movement of the two cues falls within the scope of near future research.

For each sequence, the algorithms of Head Pose and Eye Gaze estimation were applied and the ability of the system to separate among the three different eye gaze directionalities, with head turned frontally and head rotated was evaluated. The threshold set for declaring whether eyes were rotated right or left was chosen $T_r = 0.05$ and T_l -0.05, respectively (Sect. 7). Results can be seen in Table 5, as the confusion matrix of classes eye gaze Parallel to Head pose (PH), Right eye gaze (R) and Left eye gaze (L) with overall accuracy equal to 90 %. Figure 19 shows accuracies obtained for different thresholds $T_{r,l}$. It can be seen that, for absolute values between 0.03 and 0.055 (or 3–5.5 % of the inter-ocular distance), results are almost stable, while, considering that eye gaze is averted for larger thresholds mainly affects accuracy of overall focus of attention estimation for large head rotations.

8.2.2 Estimating Visual Focus of Attention: Results on the Boston University Dataset

The Boston University dataset has also been used, and was annotated regarding the degree at which its participants are focused on the camera. For each sequence, we used 14

[6] The HPEG dataset is freely available at http://www.image.ece.ntua.gr/~stiast/

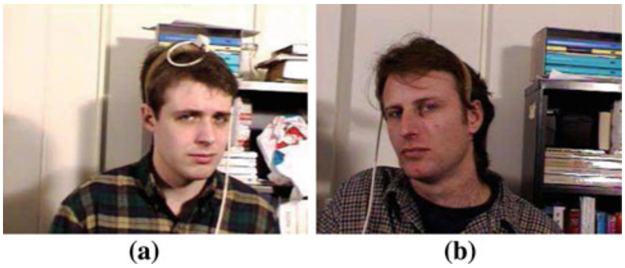


Fig. 20 Examples of annotated images with annotations equal to 0.53 and 0.6, respectively

images, taken at intervals of 15 frames resulting to a total of 630 images (the dataset contains 45 sequences of 200 frames each). The extracted images were uploaded on a server and 102 people were asked to annotate up to 60 randomly selected images, each, regarding the degree of attention towards the camera they think the person in each image has (at a scale of 0–1, with 0 standing for complete distraction and 1 for gaze in the camera). In this way, each image has been annotated 8.75 times on average, and has been assigned the average of its annotations. Examples of images can be seen in Fig. 20. The use of the Boston University dataset, here, as our workbench, is due to the dataset's nature: the lighting conditions are normal and participants move freely, with high degree of spontaneity, changing both head rotations and their eye directionality. Thus, although the dataset offers ground truth regarding head pose only, here, during the annotation set up for the current work, volunteers were expected to take into account eye gaze directionality as well for declaring their degree of confidence that someone is facing the camera or not. Due to the nature of the annotation data, at this stage, experiments were confined to extracting a degree of confidence regarding participants' attention towards certain areas (in this case, the camera).

Head pose and eye gaze are used as inputs to a Sugeno-type (Takagi and Sugeno 1985) fuzzy inference system to infer confidence values regarding focus of attention towards the camera, utilizing the annotation described above, as ground truth data. Prior to training, our data were clustered using the sub-cluster algorithm described in Chiu (1994). This algorithm, instead of using a grid partition of the data, clusters them and, thus, leads to fuzzy systems deprived of the curse of dimensionality. For clustering, many radius values for the clusters were tried and the ones that gave the best trade-off between complexity and accuracy were 25° for head horizontal rotation, 0.15 for gaze vectors, and 0.40 for the output variable. The number of clusters created by the algorithm determines the optimum number of the fuzzy rules. After defining the fuzzy inference system architecture, its parameters (membership function centers and widths), are acquired by applying a least squares and back-propagation gradient descent method (Jang 1993).

Training of regression models (Sect. 4.3), Bayesian network, as well as the Fuzzy Inference System was done by following a leave-one-subject-out cross-validation method for each user, exempting all video sequences corresponding to her/him and using only those belonging to the rest of the participants. In this way, our system's aim is to be able to generalize and be used in applications where a user-specific calibration phase is supposed to be avoided. Taking into account that the overall settings of the dataset are not posed (every user moves in a personalized manner and lighting is normal), experimental performance shows that the system's ability to generalize to unknown users is promising. Testing for each user (with $t_z = 80$ cm and $f = 500$) showed that the overall system was capable to estimate ground truth, as it was annotated by the raters, with an absolute error $E_{att} = 0.162$. Trying different combinations of the above parameters would yield similar results [e.g. for $\{t_z = 70$ cm, $f = 500\}$, $\{t_z = 90$ cm, $f = 500\}$, $\{t_z = 80$ cm, $f = 700\}$, $\{t_z = 120$ cm, $f = 300\}$, $E_{att} = 0.159, 0.163, 0.158$ and 0.166, respectively].

To get a more precise picture of the system's ability to estimate those moments when the user is looking at specific points, raters' annotation, when larger than a certain threshold was considered to correspond to gaze patterns on the camera. When annotations were smaller than this threshold, it was considered that users were looking away from the camera. Visual inspection of the annotations and the corresponding images revealed that there was high variance when head would pose a high rotation with regards to the camera plane, but the eyes were actually looking at it. In such images, qualitative assessment of the annotation showed that raters would consider users looking at the camera at a degree around ~0.5 out of 1 (Fig. 20). Thus, setting a threshold at the fuzzy system's output, equal to $T = 0.5$ for declaring a user as *looking at the camera*, overall recall and precision were 89 and 75 %, respectively (f-measure = 0.79).

In these experiments, enhancing head pose estimation with eye gaze cues, the ability of the system to infer focus of attention towards a task has been addressed. It was shown that the estimates are reliable, indicating that the proposed methodology on head pose and eye gaze combination for a cumulative estimate of user focus of attention estimation is promising, especially taking into account that not a lot of work has been done towards fusing these two cues in a non-calibrated, mono-camera environment. One of the main difficulties of such a system consists mainly in discriminating between eye gaze directionalities in case of large head rotations and, more in particular, in those cases when eyes are totally averted of the camera. However, results show that a combination of the two streams of information, using remotedly positioned systems is feasible, giving promising outcome.

8.2.3 Estimating Visual Focus of Attention Within Limited Target Spaces

In Sect. 8.2.1, a dataset for estimating focus of attention under large head rotations and eye gaze directionalities was tested. For testing the validity of combining head pose and eye gaze, in an overall framework, for estimating the focus of attention in more detailed spaces, a dataset of 8 persons was developed. More in particular, volunteers were asked to sit in front of a web-camera, positioned above a computer monitor. Subsequently, they were asked to look at three specific points: first, one on the camera and then, one on the right and one on the left of it. The line connecting the head and the camera and the lines connecting the head and the lateral points, formed ±19° angle, on average. This figure varied depending on the distance of the user from the camera (on average about 50cm). Participants were asked to look freely, as they would do if they were not recorded (head rotation or eye gaze were not mentioned at all, in order to increase spontaneity - people were simply asked to gaze at the points of interest). Moreover, they were expected to fixate at each point for as long as they desired, something that resulted in a few seconds (or hundreds of frames) for each user, which were enough for creating mappings among head rotation—eye gaze and fixation points. Figure 21 shows typical instances of the data. It was typical in the dataset, that participants, within only a few seconds, would pose different combinations of the two cues for looking at the pre-defined points (Fig. 21a–d). Thus, within the dataset, people would pose different head rotational positions, small translations and eye movements, while looking at the same point. Annotation was done separately for each user, as they were allowed to fixate for as long as they desired at each point.

Results have shown that, fusing the two cues, following a linear regression scheme, high accuracy at estimating focus of attention was achieved. The fusion scheme followed was the same for all participants, showing the ability of the proposed scheme to generalize for more persons, without needing extra calibration. The overall accuracy at estimating gaze directionality using only eye gaze was 8.16°, using only head rotation, it was 8.42°, while the combination of the two gave an average error of 6.72°. In fact, out of the 8 participants, only for 2 would fusion give lower accuracy than at least one of the two cues separately.

Following a person-specific fusion scheme (learning different regression schemes per participant) results can be further improved, with a mean error smaller than 3° for most of the subjects, approaching the visual field of the human fovea, which is about 2°. These results are notable, if one takes into account the fact that the proposed scheme does not depend on exact hypotheses regarding camera internal parameters, or exact knowledge of the head's position in the 3D space.

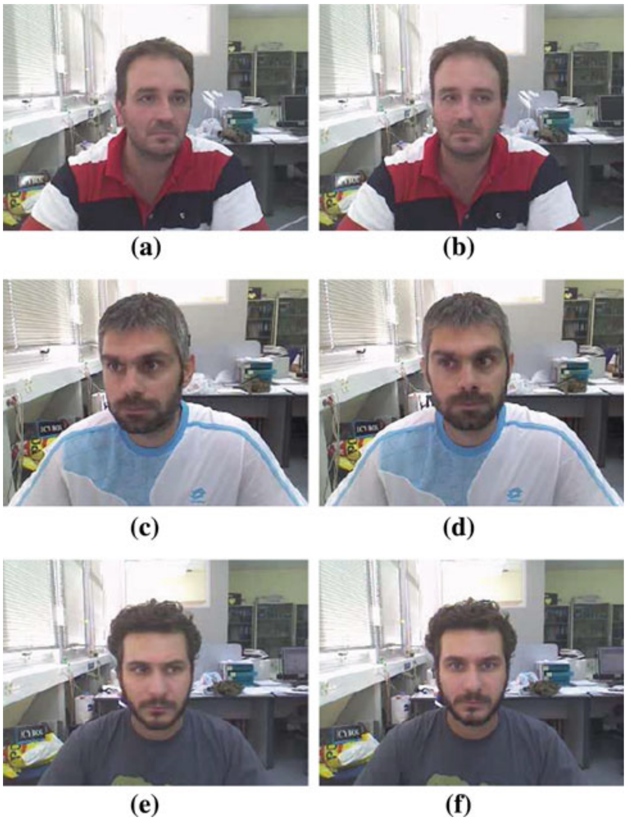


Fig. 21 Participants moving freely both head and eyes for spotting certain points in front of them

The above are indicative of the system's capability of estimating the overall focus of attention without forcing participants to keep their head stable, as most modern non-obtrusive eye trackers require. Table 6 shows results, for each participant, with mapping schemes trained following a leave-one-out protocol and following three different gaze estimation approaches:

– *Using head rotation, only* Eye gaze directionality was ignored, here, considering that head rotations are the only cue from which visual focus of attention should be extracted.
– *Using eye gaze, only* Contrary to the above scheme, an exclusively eye gaze dependent scheme was followed, considering that head would remain stable as participants would stare at different points.
– *Combining both head rotation and eye gaze* Here, the overall focus of attention (from the point of view of the user) was considered, as fusion of both cues.

The corresponding results, with head pose and eye gaze fused following a person-specific protocol are shown in Table 7. Each frame was used as a test instance with regressors trained using the rest in the user's sequence.

Table 6 Mean error for the limited target space experiment following a leave-one-out protocol

Participant	Head and eyes	Head only	Eyes only
1	6.46°(4.51)	5.52°(7.71)	8.42°(5.19)
2	7.51°(5.35)	10.76°(6.59)	8.30°(4.86)
3	3.80°(3.87)	4.95°(6.84)	4.49°(3.42)
4	10.11°(8.82)	11.42°(10.81)	14.32°(14.30)
5	5.03°(5.46)	8.90°(10.67)	5.45°(5.49)
6	12.10°(5.91)	13.85°(8.24)	11.90°(6.21)
7	5.20°(6.69)	6.34°(9.51)	7.69°(5.58)
8	3.58°(5.10)	5.66°(7.70)	4.72°(4.34)
Average	6.72°(5.72)	8.43°(8.51)	8.16°(6.18)

Table 7 Mean error and standard deviation for the limited target space experiment for personalized models

Participant	Head and eyes	Head only	Eyes only
1	2.99°(3.29)	3.77°(5.80)	7.38°(6.07)
2	4.60°(7.00)	6.47°(10.01)	7.16°(6.26)
3	2.93°(3.94)	3.38°(4.45)	5.39°(5.74)
4	6.85°(6.53)	11.42°(10.81)	12.08°(8.53)
5	2.85°(5.26)	6.42°(8.45)	4.85°(4.86)
6	7.02°(7.01)	11.53°(9.76)	8.68°(8.02)
7	2.66°(3.18)	3.75°(5.57)	5.70°(5.28)
8	2.85°(5.72)	4.49°(8.70)	4.43°(4.39)
Average	4.09°(5.24)	6.40°(7.94)	6.96°(6.15)

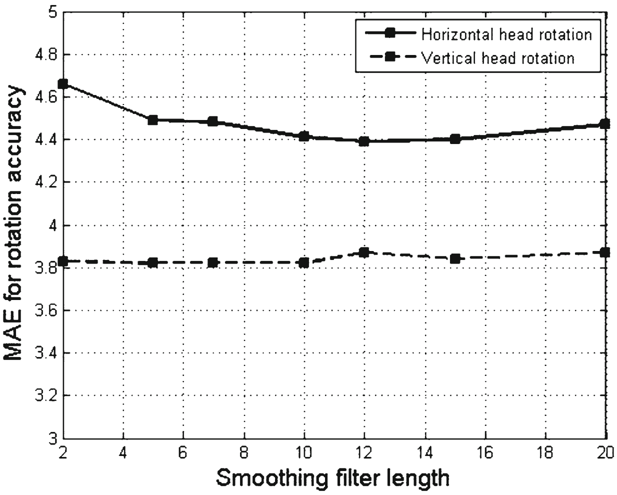


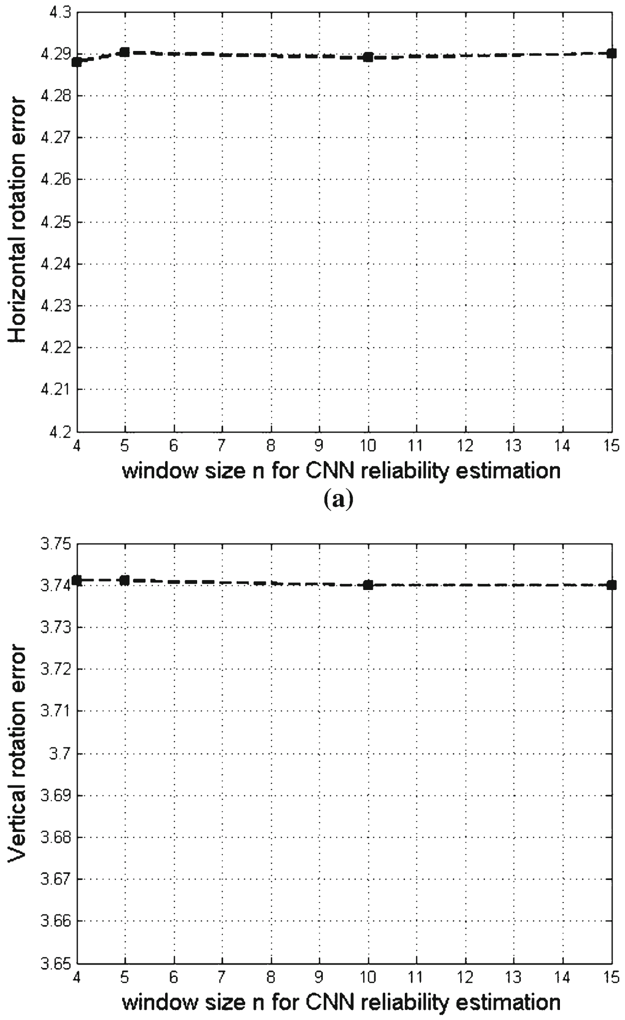
Fig. 22 Local horizontal and vertical head rotation estimation for various lengths of smoothing filters

9 Discussion

The proposed methodology is a top-to-down approach for estimating combinations of head and eye gaze directionality for inferring user focus of attention. Automatic extraction of face size and adaptation to different skin colors guarantee that tracking of facial area is robust, while DVF tracking contributes to the overall system's invariance to different lighting conditions. Further filtering of extracted measurements (Sect. 4.3) with a simple filter, smoothes out erroneous estimates. Figure 22 shows resulted MAEs of DVF tracking for different lengths of FIR filters. It can be seen that, practically, results do not differ significantly for various different lengths, from 5 to 20. The contribution of exploiting holistic information based on the proposed CNN architecture is also important for the robustness of the system. Based on normalized (and, thus, different lighting conditions invariant) face inputs, CNNs can handle face misalignments, which would be more than expected in a real scenario. Furthermore, using CNNs contributes to the overall capability of the system to be launched from arbitrary angles, not depending, in this sense, on initially frontally detected faces. The outcome of the fusion between the two modalities is dynamically updated, based on reliability indicators, that take into account expected facial geometry (local cue), as well as history of rotations (holistic cue) in the n previous frames (Eqs. 12 and 13). Figure 23 depicts the impact of different values for n. It can be seen, from this figure, that resulting error is practically not influenced by using a large range of values for n.

Frequent re-initializations of the system, when head rotation is small enough, update metrics used for the estimates of rotation using local information, rendering the system robust to translations perpendicular to the camera plane. Facial feature detection is independent of detected face size, while tracking, depending on frame-by-frame similarities, can handle gradual variations in feature sizes, as projected on the image plane. Similar, holistic information is scale independent as well, since skin regions are brought to dimensions that agree with those of the trained classifiers' inputs.

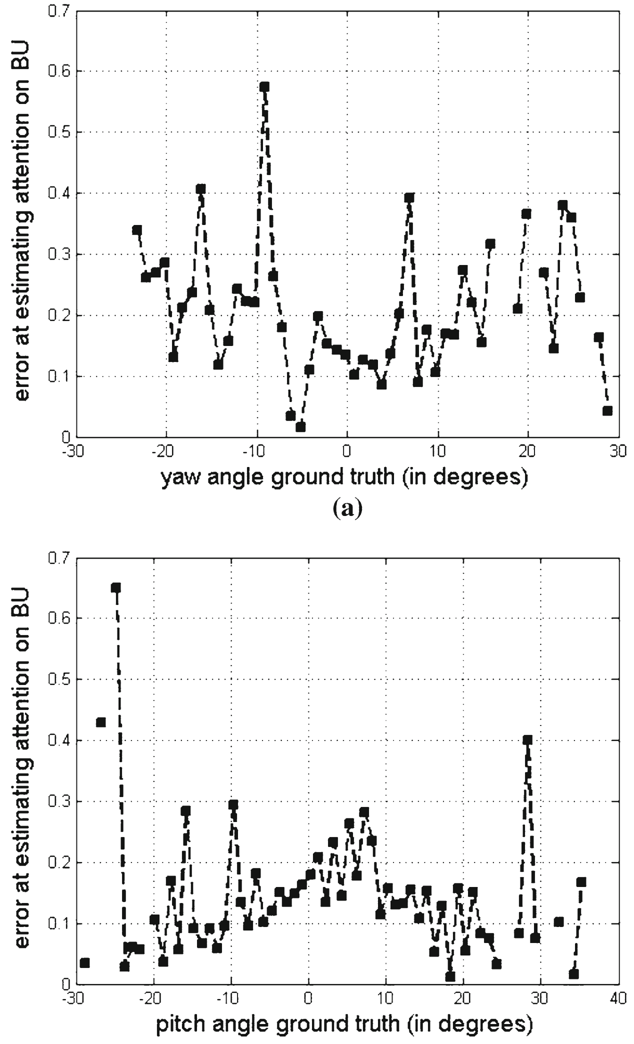
One of the main reasons for failure of the system is cases of fast movements and low video quality in terms of compression, as certain frames appear blurred. In this case, locally tracking would not function but, as the main structure of the face would be visible, the holistic nature of CNNs can give better approximations of head pose. Combinations of eye gaze and head pose would usually not give accurate results at estimating overall focus of attention, in those cases when head is highly rotated and eyes directionality would be towards the same direction with it, but highly averted from being parallel with head direction. However, as seen in Fig. 24, overall error at estimating raters' annotations regarding people's focus of attention is not biased with regards to facial angle, in general; this is a very promising element for creating a system able to imitate people's capability to infer others' focus of attention, in a non-calibrated, user agnostic way.



(a)

(b)

Fig. 23 Fused horizontal **(a)** and vertical **(b)** head rotation estimation for various lengths of windows in CNN's confidence values



(a)

(b)

Fig. 24 System's ability of estimating overall attention, with regards to true horizontal and vertical head rotations

10 Conclusions

We have proposed a novel method for estimating head pose rotation angles, using an adaptive combination of local and holistic information. As local method, in this paper, we propose a facial feature tracker based on DVFs, due to their robustness to various lighting conditions. The use of CNN for estimating horizontal and vertical rotations is explored in this paper, as an appearance-based source of information, and a novel architecture is explored. The two methodologies are fused following a dynamic and reliability-aware Bayesian scheme. Our technique uses face tracking as preprocessing step, in order for the exact boundaries of the face region to be known at every frame. Accurate and stable face tracking is achieved by using personalized skin-color models, extracted at frequent intervals, when the system detects a face. The method was built in order to handle in and out-of-plane face translational movements, and does not make any prior assumptions regarding the distance of the user from the camera, or camera internal parameters, although it is considered that the shape of the face does not look unnatural due to perspective distortion (we assume we do not use wide or narrow angle of view cameras). The proposed scheme has achieved very promising and competitive results on a widely used dataset, taking into account the aforementioned challenging factors. The method was tested on the Boston University Dataset and tracking was successful during all videos. Our technique achieves high success rates as a result of efficient facial boundaries tracking, adoption of a gaussian model describing the triangle formed by the eyes and the mouth, and continuous tracking of certain points, to ensure that facial feature regions are correctly localized during large sequences. Furthermore, the introduction of CNN, as a holistic technique, insensitive to small distortions and shifts has improved the results and alleviated the demand that the user is posed frontally to the camera at start-up.

The above constraints have also been imposed on the challenging task of inferring focus of attention combining head pose and eye gaze directionality. Fusion of these two flows of information is a challenging and not thoroughly studied problem, especially under user independent and calibration-free conditions. The proposed scheme has been tested on two datasets: one involving large rotational movements, for inferring focus of attention in and out of the camera area, and one involving more detailed fields of view. The obtained results are promising for further enrichment of the system. In particular, extracted information regarding gaze directionality will be mapped to certain points of interest. To this aim, the position of the user with regards to a target area (e.g. a computer screen) should be known. Following a calibration and user independent scheme, this is a challenging issue. Future research will deal with this problem, taking advantage of the herein presented results and conclusions, and will tackle the issue of finding appropriate mappings between 2-D projections and head/eye gaze analysis to certain points on a target plane.

The core idea of the presented work is encouraging users in HCI installations to adopt spontaneous behaviours, while modelling their degree of attentiveness towards the area of interaction. Current systems are either obtrusive, or require that the user keeps their head stable. Towards this direction, the possibilities of fuzzy logic for modelling human attention, engagement and cognitive state will be further exploited. Our work is expected to support human-robot interaction environments, where the notions of shared attention and imitation are vital for natural dialogues, and adaptation to human preferences.

Acknowledgments This research was supported by the FP7 ICT European project SIREN (project no: 258453). We would also like to thank all participants of the HPEG dataset for their helpful participation.

References

Aggarwal, G., Veeraraghavan, A., & Chellappa, R. (2005). 3D facial pose tracking in uncalibrated videos. In textitProceedings of the International Conference on Pattern Recognition and Machine Intelligence (PReMI) (pp. 515–520).

Ahlberg, J. (2001). An active model for facial feature tracking. *EURASIP Journal on Applied Signal processing*, *2002*, 566–571.

Asteriadis, S., Nikolaidis, N., Pitas, I., & Pardàs, M. (2007). Detection of facial characteristics based on edge information. In textitProceedings of the Second International Conference on Computer Vision Theory and Applications (VISAPP) (vol 2, pp. 247–252). Barcelona, Spain.

Asteriadis, S., Nikolaidis, N., & Pitas, I. (2009a). Facial feature detection using distance vector fields. *Pattern Recognition*, *42*(7), 1388–1398.

Asteriadis, S., Soufleros, D., Karpouzis, K., & Kollias, S. (2009b). A natural head pose and eye gaze dataset. In *Proceedings of the International Workshop on Affective-Aware Virtual Agents and Social Robots*, November 2–6, Boston, MA.

Asteriadis, S., Tzouveli, P., Karpouzis, K., & Kollias, S. (2009c). Estimation of behavioral user state based on eye gaze and head pose: Application in an e-learning environment. *Multimedia Tools and Applications*, *41*(3), 469–493.

Asteriadis, S., Karpouzis, K., & Kollias, S. D. (2011). Robust validation of visual focus of attention using adaptive fusion of head and eye gaze patterns. In: *ICCV Workshops* (pp. 414–421).

Ba, S. O., & Odobez, J. M. (2011). Multiperson visual focus of attention from head pose and meeting contextual cues. *IEEE Transactions Pattern Analysis Machine Intelligence*, *33*(1), 101–116.

Begley, S., Mallon, J., & Whelan, P. F. (2008). Removing pose from face images. In: *International Symposium on Visual Computing* (pp. 692–702).

Cascia, M. L., Sclaroff, S., & Athitsos, V. (2000). Fast, reliable head tracking under varying illumination: An approach based on robust registration of texture-mapped 3d models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 322–336.

Chiu, S. L. (1994). Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, *2*(3), 267–278.

Cootes, T., Walker, K., & Taylor, C. (2000). View-based active appearance models. In: *Fourth IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 227–232).

Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(6), 681–685.

Dornaika, F., & Davoine, F. (2008). Simultaneous facial action tracking and expression recognition in the presence of head motion. *International Journal of Computer Vision*, *76*, 257–281.

Fathi, A., & Mori, G. (2007). Human pose estimation using motion exemplars. In: *Proceedings of IEEE International Conference on Computer Vision* (pp. 1–8).

Gee, A., & Cipolla, R. (1994a). Determining the gaze of faces in images. *Image and Vision Computing*, *12*, 639–647.

Gee, A., & Cipolla, R. (1994b). Non-intrusive gaze tracking for human-computer interaction. *Proceedings of the International Conference on Mechatronics and Machine Vision in Practice* (pp. 112–117). Australia: Toowoomba.

Gourier, N., Hall, D., & Crowley, J. (2004). Estimating face orientation from robust detection of salient facial features. In *International Workshop on Visual Observation of Deictic Gestures* (ICPR). Cambridge.

Haralick, R. M., & Shapiro, L. G. (1992). *Computer and robot vision*. Reading, MA: Addison-Wesley.

Horprasert, T., Yacoob, Y., & Davis, L. S. (1996). Computing 3-d head orientation from a monocular image sequence. pp. 242–247.

Horvitz, E., Breese, J. S., & Henrion, M. (1988). Decision theory in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, *2*(3), 247–302.

Jang, J. S. R. (1993). ANFIS adaptive-network-based Fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics*, *23*, 665–684.

Jensen, F. V. (1996). *An introduction to Bayesian networks*. New York: Springer.

Jesorsky, O., Kirchberg, K., & Frischholz, R. (2001). Robust face detection using the Hausdorff distance. *Lecture Notes in Computer Science* (pp. 90–95).

Ji, Q., & Yang, X. (2002). Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. *Real-Time Imaging*, *8*(5), 357–377.

Kourkoutis, L., Panoulas, K., & Hadjileontiadis, L. (2007). Automated iris and Gaze detection using chrominance: Application to human-computer interaction using a low resolution webcam. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence. IEEE Computer Society* (vol 01, pp. 536–539).

Kovac, J., Peer, P., & Solina, F. (2003). Human skin colour clustering for face detection. In *IEEE International Conference on Computer as a Tool* (vol 2).

LeCun, Y. (1989). Generalization and network design strategies. Conectionism in perspective.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1990). Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems* (pp. 396–404). San Mateo, CA: Morgan Kaufmann.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE* (vol 86, pp. 2278–2324).

LeCun, Y., Bottou, L., Orr, G., & Muller, K. (1998b). Neural networks: Tricks of the trade. In K. Muller (Ed.), *Efficient backprop*. New York: Springer.

Lefevre, S., & Odobez, J. M. (2009). Structure and appearance features for robust 3d facial actions tracking. In *International Conference on Multimedia Computing and Systems/International Conference on Multimedia and Expo* (pp. 298–301).

Liu, F., Lin, X., Li, S. Z., & Shi, Y. (2003). Multi-modal face tracking using bayesian network. In *Proceedings of IEEE International Workshop on Analysis and Modeling of Faces and Gestures*.

Ma, B., Shan, S., Chen, X., & Gao, W. (2008). Head yaw estimation from asymmetry of facial appearance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, *38*(6), 1501–1512.

Magee, J. J., Betke, M., Gips, J., Scott, M. R., & Waber, B. N. (2008). A human-computer interface using symmetry between eyes to detect gaze direction. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, *38*(6), 1–1261.

Messer, K., Kittler, J., Sadeghi, M., Marcel, S., Marcel, C., Bengio, S., Cardinaux, F., Czyz, J., Srisuk, S., Petrou, M., Kurutach, W., Kadyrov, E., Kepenekci, B., Tek, F. B., Akar, G. B., & Deravi, F. (2003). Face verification competition on the xm2vts database. In *Proceedings of the International Conference on Audio and Video Based Biometric Person Authentication* (pp. 964–974).

Morency, L. P., Rahimi, A., & Darrell, T. (2003). Adaptive view-based appearance model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 803–810).

Morency, L. P., Whitehill, J., & Movellan, J. R. (2010). Monocular head pose estimation using generalized adaptive view-based appearance model. *Image Vision Computing*, *28*(5), 754–761.

Murphy-Chutorian, E., & Trivedi, M. M. (2009). Head pose estimation in computer vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(4), 607–626.

Murphy-Chutorian, E., Doshi, A., & Trivedi, M. M. (2007). Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation. In *Proceedings of the IEEE Conference on Intelligent Transportation Systems* (pp. 709–714).

Nguyen, M. H., Modrego Pardo, P. J., & la Torre, F. D. (2008). Facial feature detection with optimal pixel reduction SVM. In *Proceedings of the eighth IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 1–6).

Osadchy, M., LeCun, Y., & Miller, M. (2007). Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research*, *8*, 1197–1215.

Peters, C., Asteriadis, S., & Karpouzis, K. (2009). Investigating shared attention with a virtual agent using a gaze-based interface. *Journal on Multimodal User Interfaces*, *3*(1–2), 119–130. doi:101007/s12193-009-0029-1.

Sarle, W. S. (1995). Stopped training and other remedies for overfitting. In *Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics* (pp. 352–360).

Shaker, N., Asteriadis, S., Yannakakis, Y., & Karpouzis, K. (2011). A game-based corpus for analysing the interplay between game context and player experience. *EmoGames workshop, International Conference on Affective Computing and Intelligent Interaction* (ACII2011) (pp. 547–556), October 9, Memphis, TN.

Sim, T., Baker, S., & Bsat, M. (2003). The cmu pose, illumination, and expression database. *IEEE Transactions Pattern Analysis Machine Intelligence*, *25*(12), 1615–1618.

Stiefelhagen, R. (2004). Estimating head pose with neural networks: Results on the pointing04 ICPR workshop evaluation data. In *Pointing 04 Workshop* (ICPR), Cambridge.

Sung, J., Kanade, T., & Kim, D. (2008). Pose robust face tracking by combining active appearance models and cylinder head models. *International Journal of Computer Vision*, *80*(2), 260–274.

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, *15*(1), 116–132.

Tan, K., Kriegman, D., & Ahuja, N. (2002). Appearance-based eye gaze estimation. In *IEEE Workshop on Applications of Computer Vision* (pp. 191–195).

Toyama, K., & Horvitz, E. (2000). Bayesian modality fusion: Probabilistic integration of multiple vision algorithms for head tracking. In *Proceedings of 4th Asian Conference on Computer Vision* (ACCV).

Valenti, R., Sebe, N., & Gevers, T. (2012). Combining head pose and eye location information for gaze estimation. *IEEE Transactions on Image Processing*, *21*(2), 802–815.

Viola, P. A., & Jones, M. J. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition* (vol 1, pp. 511–518).

Voit, M., & Stiefelhagen, R. (2010). 3D user-perspective, voxel-based estimation of visual focus of attention in dynamic meeting scenarios. In *Proceedings of ICMI-MLMI*.

Wang, J. G. (2003). Eye Gaze estimation from a single image of one eye. In *IEEE International conference on Computer Vision* (pp. 136–143).

Weidenbacher, U., Layher, G., Bayerl, P., & Neumann, H. (2006). Detection of head pose and Gaze direction for human-computer interaction. *Perception and Interactive Technologies*, *4021*, 9–19.

Xiao, J., & Cohn, J. F. (2003). Robust full-motion recovery of head by dynamic templates and re-registration techniques. *International Journal of Imaging Systems and Technology*, *13*, 85–94.