Head Pose Estimation with One Camera, in Uncalibrated Environments

Stylianos Asteriadis
Image, Video and Multimedia Systems Lab
National Technical University of Athens
9, Iroon Polytechniou str, Athens, Greece
stiast@image.ntua.gr

Kostas Karpouzis
Image, Video and Multimedia Systems Lab
National Technical University of Athens
9, Iroon Polytechniou str, Athens, Greece
kkarpou@cs.ntua.gr

Stefanos Kollias
Image, Video and Multimedia Systems Lab
National Technical University of Athens
9, Iroon Polytechniou str, Athens, Greece
stefanos@cs.ntua.gr

ABSTRACT

Head pose together with eye gaze are a reliable indication regarding the estimate of the focus of attention of a person standing in front of a camera, with applications ranging from driver's attention estimation to meeting environments. As gaze indication, eye gaze in non-intrusive or non highly specialized environments is, most times, difficult to detect and, when possible, combination with head pose is necessary. Also, in order to successfully track the rotation angles of the head, a priori knowledge regarding the equipment set-up parameters is needed, or specialized hardware, that can be intrusive is required. Here, we propose a novel facial feature tracker that uses Distance Vector Fields (DVFs) and, combined with a new technique for face tracking, successfully detects facial feature positions during an image sequence and estimates head pose parameters. No a priori knowledge regarding camera or environmental parameters is needed for our technique.

Author Keywords

Head Pose, Facial Feature Tracking, User Attention Estimation

ACM Classification Keywords

H.5.2 Information Interfaces and Representation: User Interfaces

INTRODUCTION

For estimating the visual focus of view, it is necessary, apart from the eye gaze, to incorporate information coming from head pose [10]. This means that, in order to know the exact orientation of visual attention, both modalities should be added. However, there are cases where eye gaze information is not easy to retrieve, due to limitations coming from necessary equipment or the intrusive nature of the relevant devices. On the contrary, head pose estimation has been studied more thoroughly in less intrusive environments, with off the shelf equipment, like simple web-cameras [12]. Besides, it has been experimentally shown that head pose alone, in certain environments, is a very reliable criterion for determining the directionality of gaze. Typical research is the one presented in [14], where the authors have conducted experiments in a setting of four people participating in a meeting: It was proven that head pose directionality alone is strongly correlated with the total of gaze directionality, and can be useful at 88.7% of the cases for inferring the focus of attention of a participant. For the above reasons, in this paper we are focusing on the issue of automatic Head Pose Estimation in environments where intrusive mechanisms are not a prerequisite. The described work is aimed at environments where gaze directionality can be approximated efficiently by head pose alone.

Recent bibliography consists of a variety of methodologies regarding the issue of estimating the rotation of a head (*yaw, pitch, roll* angles). The various systems that have been presented are various in terms of algorithms or hardware / equipment employed. In an un-intrusive environment (not necessitating dedicated equipment like helmets, or infrared light cameras) there might exist restrictions regarding knowledge of environmental or camera parameters. That is, many methods require to know, e.g the approximate distance of the user from the camera, or they need to know the camera intrinsic parameters. Lacking such knowledge may lead these systems to erroneous estimation of head pose parameters, when particular movements are taking place (e.g. when the user is moving along the *z*-axis). Here, we will mainly deal with methods that do not depend on specific set-ups in terms of hardware, do not rely on any dedicated equipment and are, as parameter-independent as possible.

A coarse classification of methodologies has proved that no ideal group of one-camera systems exists: Each class of methods have their advantages and disadvantages, while there is a tendency for better results when it comes to hybridic techniques, where the authors attempt to use the advantages of one method to alleviate the disadvantages of the other. Although it is not easy, and sometimes not straightforward to categorize methods, a coarse classification is the one pre-

Presented at International IUI 2010 Workshop on Eye Gaze in Intelligent Human Machine Interaction.

sented below:

Holistic techniques: The face is aligned and compared with trained models and, using, either regression or classification, a final outcome regarding the pose of the test face is acquired. These methods are usually the most accurate but the major disadvantage of this group of approaches is the fact that, usually, the face needs to be exactly aligned with the models against which it will be compared, and this necessitates very good detection of the face boundaries. Typical methods are reported in [15] and [11], where the authors use neural networks and keyframes belonging to past appearances of the face during the video, respectively.

Local techniques: These approaches depend on accurate facial feature localization and tracking, as they use geometrical relations among them to infer head pose. They make use of the saliency of some facial features, which makes it easy to follow them. The drawback of these approaches is that they tend to be quite sensitive to erroneous tracking and it is not easy to recover when some features become occluded. Typical work is the one presented in [6], where the authors use the ration of the inter-ocular distance and the distance of the mouth from the eyes midpoint to model the face, while [7] has proposed models relating facial landmarks to the rest of the face. In [13], the authors, based on expected eye positions on faces, and SVMs, detect eye features and discriminate between poses around the horizontal axis.

Facial motion recovery: Facial motion recovery usually calculates motion flow between successive frames and, based on this, estimates head pose parameters, using, e.g, cylindrical models [3]. This family of methods is usually the most robust but, most of the times, knowledge of camera intrinsic parameters or a priori knowledge of the distance between camera and user is needed.

Non-rigid model fitting: These methods use non-rigid trained models, which encode information regarding shape and texture of a face [4] [5], and have drawn much attention in the recent years. A major factor to be taken into account, though, is the requirement for good initialization as, such models, can easily fall into local minima.

Hybrid techniques: Many methods have also been proposed, as attempts to overcome problems imposed by one technique, by fusing it with other techniques. Typical example is the one reported in [16], where the authors fuse AAMs with cylindrical models, and the work of [11], where a static pose estimator for the 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.

For a more extensive and analytical description of the literature on the Head Pose estimation issue, the reader is referred to the work of Murphy-Chutorian et al [12].

Here, we propose a method for inferring head pose rotation parameters (*yaw*, *pitch* and *roll* angles), that does not rely on dedicated hardware or a priori knowledge of any parameters

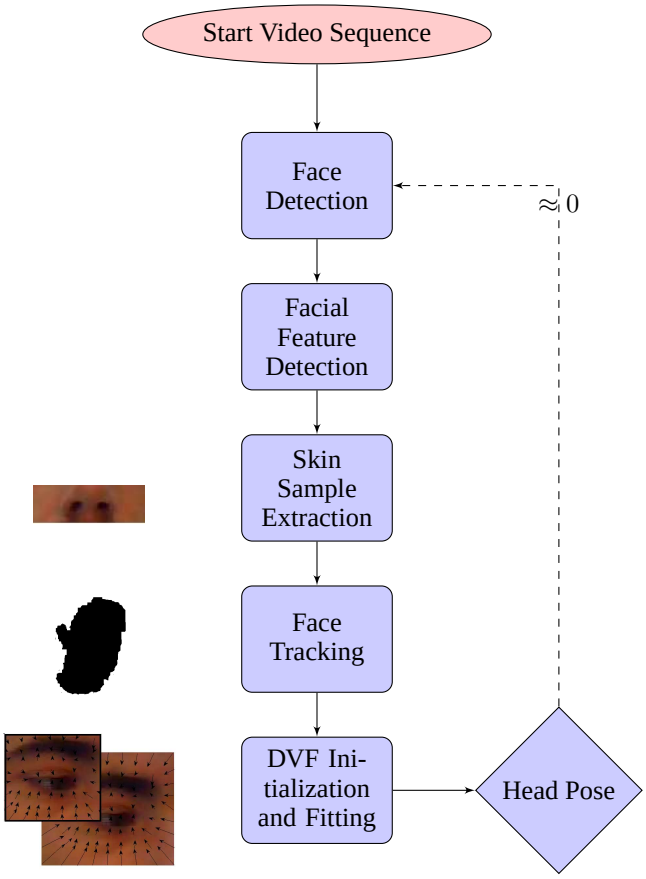


Figure 1. Overview of the method: Face/Facial feature detection and samples of skin are extracted at start-up. Subsequently, the face area is tracked and DVF tracking takes place. DVFs positions and face area are used to infer Head Pose Estimation. If Head Pose Vector is almost zero, the algorithm can re-initialize.

of the set-up. Instead, we use a monocular system, and do not exploit any specific a-priory knowledge regarding the environment or the user. We track facial features using a novel tracker, that employs Distance Vector Fields (DVFs) [1] and relate features' location to face boundaries. The tracker is compared to the Optical Flow algorithm on raw pixel data, to show the applicability of the method. First, the face is detected and a skin area is used as sample for face tracking in subsequent frames. Using face detection as a preliminary step, allows the system to be scale-independent, as the face is searched for at various scales, while face tracking is adapted to each user's skin chrominance. In this way, we avoid using generic thresholds but personalize skin color segmentation, in order to improve robustness of our face tracker. Within the segmented face region, DVFs of certain facial features are only searched for within the face area and, based on features' location, head pose vector is extracted. An overview of the method can be seen in Figure 1.

DISTANCE VECTOR FIELDS

Distance Vector Fields [1] are image representations encoding the shape of a deformable area. More specifically, each image pixel (i, j) is given a vector $\underline{\mathbf{v}}$ pointing to the clos-

est edge pixel (k, l), thus, forming the Distance Vector Field (DVF)(Equations 1,2).

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

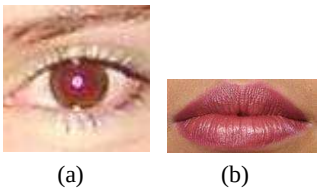
$$(k,l) = \arg \min_{m,n \in \underline{E}} D((i,j),(m,n)) \quad (2)$$

where D is (here) the euclidean distance and $\underline{\text{E}}$, $\underline{\text{B}}$ are the sets of edge and image pixels respectively.

With DVFs, every shape can be reconstructed and, furthermore, every pixel in an image can be used to inform regarding the shape of the object it belongs to. Furthermore, using edge maps to extract DVFs, usual variations in terms of lighting do not introduce large differences of the appearance of the corresponding DVFs. These ideas led us to using DVFs for tracking facial feature areas.

FACIAL FEATURE DETECTION

Before Head Pose Estimation, facial features are detected. As a pre-processing step, the face is initially detected [1]. The method described in [1] for face detection, employs ellipse fitting for finding the exact boundaries of the face. For the extraction of prototype Distance Vector Fields corresponding to eye and mouth areas, random face images of good resolution were collected from the web. The eye areas have been cropped in such a way that the upper edge of the eyebrows defines the upper boundary of the eye patch and the lip upper/lower/right and leftmost points define the boundaries of the mouth patches. The eye centers were aligned so that they were located at the center of the eye patches. Example training images can be seen in fig. 2. To detect facial features, we used predefined areas of the face and compared the mean DVF of the prototype eye and mouth patches with the corresponding DVFs of candidate facial areas. More precisely, the right eye was first detected on the upper right part of the face and, based on its position, the left eye was searched for at an even more restricted area, on the left of the detected right eye. Subsequently, the mouth is searched for at a region under the two detected eye centers. In order to compare features' DVFs with facial areas' DVFs, the face region is brought to certain dimensions, that agree with the scale at which prototype DVFs were extracted. After detection, the face and corresponding DVFs are brought back to the dimensions that agree with the real face region. Further details of the detection algorithm can be found in [1].



(a) (b)

Figure 2. a) example of eye training image; b) example of mouth training image;

FACE TRACKING

As head pose estimation will be based on tracking with DVFs and, in order to eliminate erroneous tracking at the maximum extent, it is required to specify the boundaries beyond which tracking will not be permitted. To this aim, the exact contour of the face is searched for at each frame. Face contour is based on tracking face areas that have skin-similar color.

As skin color varies among people, after facial feature detection, a sample area C_{skin} of face is used for each person; we used the saturation values of this area, and the saturation values of face pixels C_{fp} in subsequent frames are expected to be within certain limits with regards to the mean saturation value s_M of C_{skin} (Equation 3):

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

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, at the detection step, according to equations 4-5:

$$T = \arg \min_{0<T<0.35} (\sum_{x \in \Omega} \delta(k_x) - Face_{size}) \quad (4)$$

with

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

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. 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 equations 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 Figure 3, where the optimum threshold T to be used in equation 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 [9] are used, in order for skin clusters in RGB colorspace to be built. According to the authors in [9], a map C_{sp} of candidate skin pixels is built according to a series of chrominance rules.

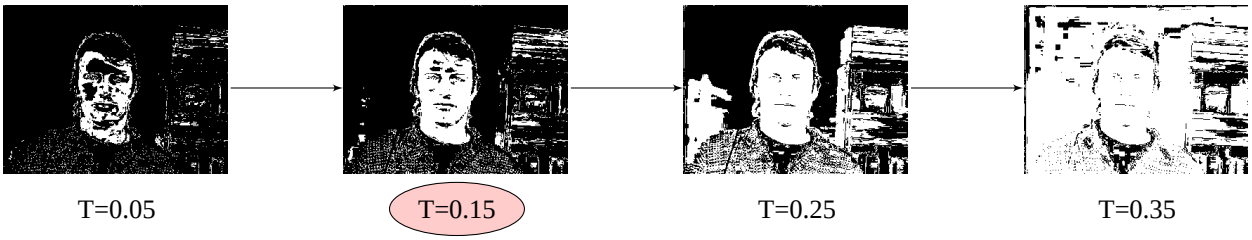


Figure 3. Overview of selection of threshold T for segmenting face regions based on equation 3: Threshold $T = 0.15$ was decided in this sequence, as the total number of pixels whose values are close to that of the initially selected skin region is close to the number of pixels belonging to the face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

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 [8] and chooses the largest component as the final face region. Additionally, in order to avoid false alarms when estimating position and size of the face, information from previous frames is taken into account. More specifically, when a component is marked as face region but does not overlap with the face at the previous frame,the second largest component is chosen and checked, while, if there are overlaps between the current and the previous detected face regions but they differ significantly in shape, the position of the face at the previous frame is considered and the skin area (used as color predicate) is expanded and a new threshold T is calculated. The above, multi-step procedure gave very good results at tracking efficiently the face region. An overview of the steps of face tracking can be viewed in figures 3 and 4.

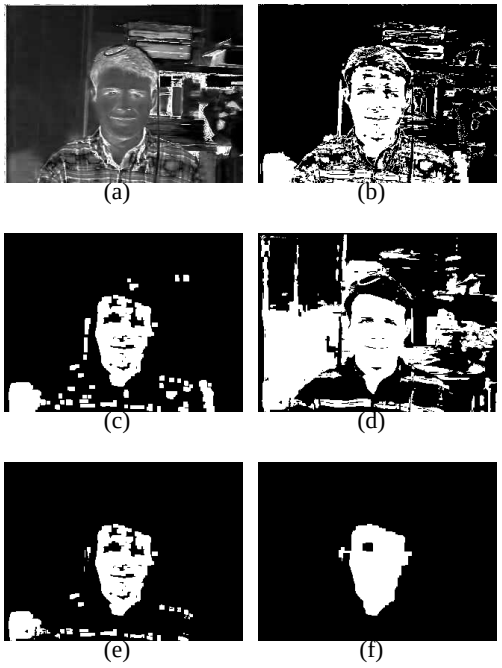


Figure 4. a) Original image saturation values; b)thresholded saturation values; c)Face candidate pixels C_fp, extracted after morphological opening; d) Skin candidate pixels C_sp; e) Face candidate pixels after logical AND between Cfp and Csp; f) Final face mask after morphological operations

FACIAL FEATURE TRACKING

As mentioned earlier, head pose angles will be inferred based on the position of the eye and mouth centers, and their relative distances from facial boundaries. For this reason, DVFs will be used to detect facial areas at each frame, by comparing a feature's DVF $f_{k,i}$ at frame k and position i with the DVF $f_{k+1,i+x}$ at frame $k+1$ and candidate areas $i+x$ of an extended area around its position i at the previous frame. Experiments showed that, at this stage, DVFs are more robust when used for eye tracking, while for mouth tracking, we used eye positions as initialization and utilized further features as will be discussed later on in this section.

Eye tracking

The position of the new eye area in frame $k+1$ is the one that minimizes the L_2 norm, and the motion vector $\mathbf{p}$ for each eye is the one described in (6):

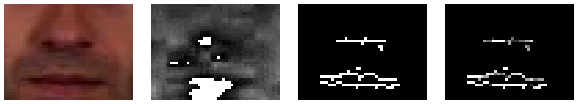
$$\mathbf{p} = \arg\min_{\mathbf{x}} \sum_{i \in R_k} \|f_{k,i} - f_{k+1,i+x}\|_2 \qquad (6)$$

To reinforce correct tracking, after a new eye area is defined at a frame, its position is updated in order to be centered around the eye center. For this reason, when a new eye area

is defined based on equation 6, the iris center is searched for based on the derivative images and their projections on the horizontal and vertical axis, as well as luminance information [1]. More precisely, the authors in [1] search for eye centers in eye areas by using the derivative images of the eye areas both row-wise and column-wise and use a set of their maximum projections on the horizontal and vertical axis respectively for an initial estimate of the eye center. Subsequently, a small window is used to search for the darkest region in a neighborhood close to this initial estimate. Through experiments, it was proven 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.

Mouth tracking

For mouth tracking, rapid lip movements, especially in the case where skin color cannot be easily distinguished from lips, cause DVF's to change very rapidly. As a result, for some cases, the mouth area is localized at the region between the mouth and the nose, or a region under the chin. 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 is achieved by multiplying the binary edge values with the hue component values of the search area (see fig. 5). The mouth is then tracked as a mask of predefined size and is localized at the positions of maxima of the map that combines hue and horizontal edges. It was proved that this technique alleviated the problems caused by fast lip movements and the mouth area was efficiently tracked.



(a) (b) (c) (d)

Figure 5. 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;

Optimization of 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 face'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 [7]). Thus, the extra factor corresponding to this distribution changes eq. 6 as follows:

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

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 $d_{k+1,x}$. μ is acquired automatically at the face/facial features detection step and σ is extracted offline from training data, and can be thus used at every set-up.

Estimation of Yaw, Pitch, Roll angles

Wilson's experiments [18] demonstrated that facial features' positions with regards to head pose contour play a key role for human perception of head pose [7]. Knowing the skin contour boundaries, the eye midpoint's $E_i = (E_{x,i}, E_{y,i})$ and mouth centre's $M_i = (M_{x,i}, M_{y,i})$ positions at each frame i, we calculate the yaw (pitch) angle as follows: it is modeled as the average relative changes of the distance of E and M from the skin region vertical C_{1x}, C_{2x} (horizontal C_{1y}, C_{2y}) boundaries, with regards to a frame where the subject is facing the camera frontally. The resulting values coming from the eye midpoint's and mouth's positions were fused using linear regression.

The above are illustrated in equations 8 and 9, with Y_i and P_i being the values of yaw and pitch at frame i.

$$\begin{aligned} Y_i &= b_{1y} \times \Big[\frac{(E_{x,i} - C_{1x,i}) + (E_{x,i} - C_{2x,i})}{2 \times d_{eyes,0}} \\ &- \frac{(E_{x,0} - C_{1x,0}) + (E_{x,0} - C_{2x,0})}{2 \times d_{eyes,0}} \Big] \\ &+ b_{2y} \times \Big[\frac{(M_{x,i} - C_{1x,i}) + (M_{x,i} - C_{2x,i})}{2 \times d_{eyes,0}} \\ &- \frac{(M_{x,0} - C_{1x,0}) + (M_{x,0} - C_{2x,0})}{2 \times d_{eyes,0}} \Big] \end{aligned} \quad (8)$$

$$\begin{aligned} P_i &= b_{1p} \times \Big[\frac{(E_{y,i} - C_{1y,i}) + (E_{y,i} - C_{2y,i})}{2 \times d_{eyes,0}} \\ &- \frac{(E_{y,0} - C_{1y,0}) + (E_{y,0} - C_{2y,0})}{2 \times d_{eyes,0}} \Big] \\ &+ b_{2p} \times \Big[\frac{(M_{y,i} - C_{1y,i}) + (M_{y,i} - C_{2y,i})}{2 \times d_{eyes,0}} \\ &- \frac{(M_{y,0} - C_{1y,0}) + (M_{y,0} - C_{2y,0})}{2 \times d_{eyes,0}} \Big] \end{aligned} \quad (9)$$

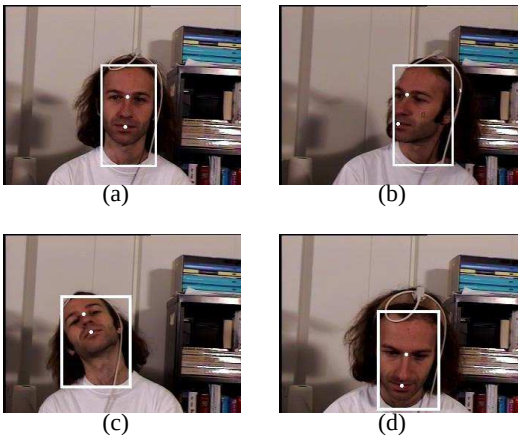


Figure 6. Examples of skin regions boundaries and relation with the used features' positions

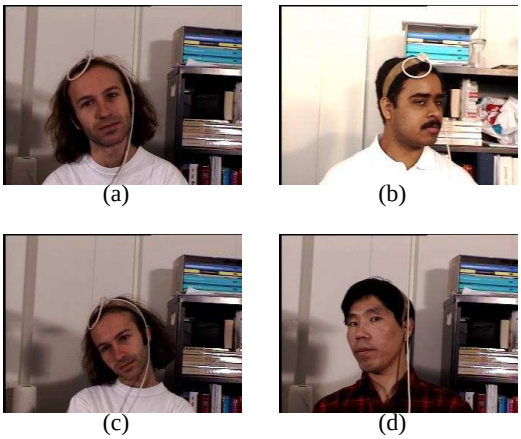


Figure 8. Example frames from the Boston University Dataset

with Y_i and P_i the estimated yaw and pitch angle at the current frame, 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 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 and P can be convolved with a N^{th} order FIR filter (here, N=12). The above methodology can be intuitively explained as shown in figure 6.

Estimation of Roll angle

Calculating the roll angle is straightforward: It derives from the angle defined by the eye centers line segment and the x axis (Fig. 7). The values are again filtered as is done for the yaw and pitch angles.

Figure 7. Example of roll angle estimation on the BU dataset;

EXPERIMENTAL RESULTS

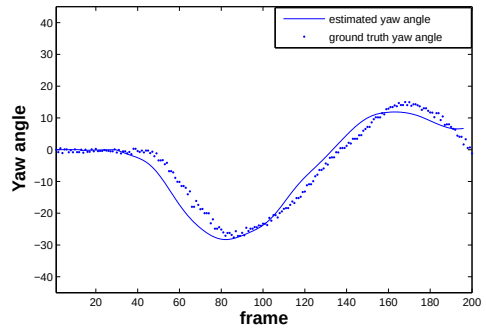
Results on the Boston University Dataset

To evaluate the effectiveness of the described method, we used a widely known database, namely the Boston University database [3]. 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, while ground truth is offered regarding their head rotation at any moment, using a "Flock of Birds" magnetic tracker. Also, the sequences were digitized at a 30 fps rate and the participants appear to be sitting at a distance of about one meter from the camera. Typical example frames of the database can be seen in Figure 8.

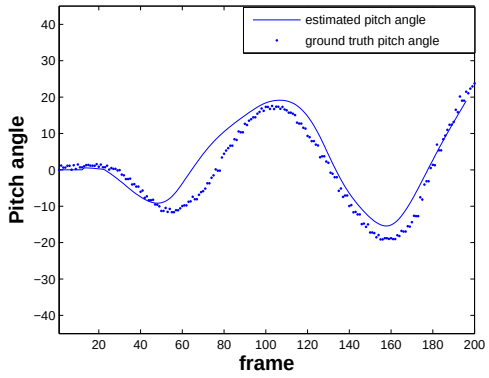
When testing a person's head rotation, weights b_{1y},b_{2y},b_{1p} and b_{2p} were learnt using the rest of the subjects' videos. Although, here, weights were learnt from videos that used the same camera for recording as the testing videos, in our work we consider that the cameras do not distort the face shape (they are neither of wide nor narrow angle) and, thus, the weights would be expected to have the same values for any type of camera within the scope of our research. Table 1 shows the mean errors at estimating roll, pitch and yaw angles on the Boston University dataset, using the RMS error. We state results of other methods in literature, stating that they are using the RMS error. It can be seen from the results, that our method is comparable and, in cases, performs even better. 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 9 shows typical examples of angles as extracted by our method, against ground truth data.

Results on the HPEG dataset

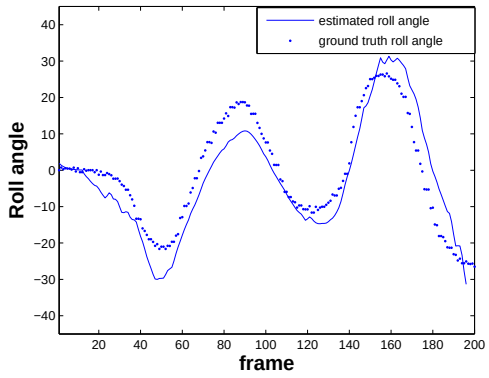
To test the validity of our algorithm, we used the HPEG dataset's [2] first session. This dataset was developed in our laboratory, and it is challenging, in the sense that it consists of recordings of ten people, sitting in front of a computer monitor, and free to rotate their heads (some of them also



(a)



(b)



(c)

Figure 9. Estimated head pose angles and corresponding ground truth.

Table 1. RMS error results on the BU

	DVF tracking	method in [17]	method in [16]
Yaw	5.72°	6.10°	5.40°
Pitch	4.89°	5.26°	5.60°
Roll	3.56°	3.00°	3.10°

(a) (b) (c) (d)

Figure 10. Examples from the HPEG dataset

Table 2. Head Pose Estimation RMS error, using feature tracking with Optical Flow and Distance Vector Fields

	Error: Optical flow	Error: DVF
Yaw	8.39°	6.65°
Pitch	5.51°	5.59°
Average	6.95°	6.12°

their torso) towards any direction they want. Furthermore, the dataset was acquired using a standard web-camera, while the lighting conditions were those of a usual indoors, office environment, with complex background and some human action taking place in the background. The frame resolution is 640×480, while the frame rate of the recorded videos is 30fps. Some examples of the dataset can be seen in figure 10.

To illustrate the validity of using DVFs for tracking facial feature areas, we compared the Head Pose results with those obtained by using the standard Lucas-Kanade algorithm. The same restrictions as before were imposed (features limited within the skin area, gaussian model limiting the eye-mouth geometrical relations, search areas the same as DVF tracking). Table 2 shows the results of using Optical Flow tracking and DVF tracking on the HPEG dataset. It can be seen that, in general, DVF 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 on the face), while DVFs search for similar shapes, as the position of each pixel and its relation with their neighboring shapes is encoded. These results have shown that, for the purpose of Head Pose Estimation, and the search areas that we use, Distance Vector Fields are more appropriate for tracking.

CONCLUSIONS AND FUTURE WORK

A new method for facial feature tracking for head pose estimation has been proposed in this paper, as a necessary part of inferring gaze of a person. As head pose is a vital component for inferring people's focus of attention, and sometimes it can be a reliable stand-alone indicator, we focused on un-

intrusive and uncontrolled environments. Our technique introduces Distance Vector Fields for tracking facial features and adopts a new methodology for efficient face tracking. This work was inspired by the need to conduct research towards un-intrusive methods for facial attention estimation, without any prior knowledge regarding camera parameters, environment, user, etc. The results obtained by our method are extremely promising and tracking was successful during all image sequences. This is largely due to the fact that we incorporated a face tracking scheme, automatically adapted to the user, as well as a gaussian geometrical model, which is also acquired online. Gaze directionality estimation using the present technique and eye gaze estimation in a common system is within the scope of our future work and an overall framework for user attention recognition, taking into account user profile/character/emotional state is going to be our main focus of attention in the near future.

Acknowledgments

This work was funded by IST Project 'FEELIX',(under contract FP6 IST-045169).

REFERENCES

1. S. Asteriadis, N. Nikolaidis, I. Pitas, and M. Pardàs. Detection of facial characteristics based on edge information. In *Proc. VISAPP*, volume 2, pages 247–252, 2007.
2. S. Asteriadis, D. Soufleros, K. Karpouzis, and S. Kollias. A natural head pose and eye gaze dataset. In *Proc. of the AFFINE Workshop*, 2009.
3. M. L. Cascia, S. Sclaroff, and V. Athitsos. Fast, reliable head tracking under varying illumination: An approach based on robust registration of texture-mapped 3d models. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:322–336, 2000.
4. T. Cootes, K. Walker, and C. Taylor. View-based active appearance models. In *Proc. Fourth FG IEEE International Conference*, pages 227–232, 2000.
5. T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.
6. A. Gee and R. Cipolla. Non-intrusive gaze tracking for human-computer interaction. In *Proc. International Conference on Mechatronics and Machine Vision in Practice.*, pages 112–117, 1994.
7. N. Gourier, D. Hall, and J. Crowley. Estimating face orientation from robust detection of salient facial features. In *International Workshop on Visual Observation of Deictic Gestures (ICPR)*, 2004.
8. R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley, 1992.
9. P. P. Jure Kovac and F. Solina. Human skin colour clustering for face detection. In *Proc. IEEE International Conference on Computer as a Tool*, volume 2, 2003.
10. C. L. Kleinke. Gaze and eye contact: a research review. *Psychological Bulletin*, 100(1):78–100, 1986.
11. L.-P. Morency, J. Whitehill, and J. Movellan. Generalized adaptive view-based appearance model: Integrated framework for monocular head pose estimation. In *Proc. IEEE FG International*, 2008.
12. E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(4):607–626, 2009.
13. M. H. Nguyen, J. Perez, and F. D. la Torre. Facial feature detection with optimal pixel reduction svm. In *Proc. 8th IEEE FG International Conference*, 2008.
14. R. Stiefelhagen. Tracking focus of attention in meetings. *Proc. IEEE ICMI*, 0, 2002.
15. R. Stiefelhagen. Estimating Head Pose with Neural Networks - Results on the Pointing04 ICPR Workshop Evaluation Data. In *Pointing 04 Workshop (ICPR)*, 2004.
16. J. Sung, T. Kanade, and D. Kim. Pose robust face tracking by combining active appearance models and cylinder head models. *International Journal of Computer Vision*, 80(2):260–274, 2008.
17. R. Valenti, Z. Yucel, and T. Gevers. Robustifying eye center localization by head pose cues. In *Proc. IEEE CVPR Conference*, 2009.
18. H. Wilson, F. Wilkinson, L. M. Lin, and M. Castillo. Perception of head orientation. *Vision research*, 40(5):459–472, 2000.