

Estimating 3D Facial Pose in Video with Just Three Points

Gines Garcia-Mateos, Alberto Ruiz
Dept. Informatica y Sistemas
University of Murcia, 30100 Murcia, Spain
{ginesgm, aruiz}@um.es

Pedro E. Lopez-de-Teruel, Antonio L. Rodriguez, Lorenzo Fernandez
Dept. Ingenieria y Tecnologia de Computadores
University of Murcia, 30100 Murcia, Spain
{pedroe, lfmaimo}@ditec.um.es, alr11@alu.um.es

Abstract

Estimating facial pose along a video sequence is an important problem in many applications such as perceptual user interfaces, people monitoring, and aid for the disabled. Currently, most existing methods to compute facial pose are designed to obtain high accuracy under controlled conditions. However, efficiency and robustness are preferable in most practical situations. In this paper, we deal with the problem of 3D pose estimation with the aim to design a fast and approximate solution that requires a minimum amount of information: our method is able to work with just the location of three points –the eyes and the mouth– in each frame. Using that information, and introducing several assumptions –that could hold in many cases–, we show how to estimate the six degrees of freedom of facial position and orientation. Moreover, one single point could be enough after initializing the system. The obtained accuracy greatly depends on the precision of the underlying 2D face tracker. The system is generic and does not require to be trained for each new user.

1. Introduction

Three-dimensional head pose –in particular 3D orientation– is an important element in non-verbal human communication. With certain movements of the head, like nodding or shaking, a person may express agreement or disagreement. A facial pose can also indicate a point of interest to the human, since it signals where he/she is looking to. For example, in human/computer interaction, it could indicate which part of the screen the user is looking to.

In consequence, 3D facial pose estimation has attracted much interest in the face processing community. The prob-

lem can be defined as follows [12]: recover the six-degree-of-freedom pose of a target structure that has a rigid geometric relationship to the skull of a particular individual, given a stream of images. These 6 parameters correspond to 3D position, (p_x, p_y, p_z) , and orientation, $(roll, pitch, yaw)$, of the user's head.

By 1998, in an early review by Toyama [12], up to eighteen different approaches to 3D face tracking and pose computation were described. Obviously, new methods and approaches have been proposed since then. Most of them are based on techniques such as active appearance models [2], cylindrical face models [11], 3D morphable models [6, 9], texture models [14], and others [7]. All of these systems work with a great amount of information, which involves complex implementations, high computational load, and user specific training. However, in theory, just 3 points should be enough to estimate the 3D pose of a rigid object.

Our purpose is to develop a new method working with the minimum necessary information, with application on a perceptual user interface. Therefore, high efficiency and robustness are preferable to accuracy. We introduce some assumptions that lead to a simpler although approximate process. In essence, we perform a model-based object localization in 3D, where the model is a simple triangle containing eyes and mouth. The classical problem of parameterized 3D model location was studied by Lowe [8], where pose is found by iteratively solving coupled, non-linear equations.

The rest of the paper is structured as follows. In section 2 we briefly describe our solution to the problems of face detection, location and tracking, that are previous to pose estimation. Using the obtained locations of eyes and mouth, section 3 presents our method to estimate 3D pose of the face. The experiments and sample application are explained in sections 4 and 5. Finally, in section 6 we extract some relevant conclusions.

2. Face detection, location and tracking

Face detection and tracking are preliminary problems in computer vision systems that work with human faces [15]. For example, before 3D pose can be estimated, at least one face has to be localized in the images. For completeness, in this section we describe the face detector, locator and tracker used in the experiments of section 4. However, it is important to note that our pose estimator is independent from these initial steps. That is, in our approach 2D face tracking and 3D pose computation are separate problems.

2.1. Combined face detection

Many facial detection techniques have been proposed in the literature. For example, in a survey by Yang *et al.* [15] more than 150 different approaches are exposed. In our case, we have used a combination of methods in order to improve detection reliability.

Firstly, the face detector proposed by Viola and Jones [13] is applied on the first frame of the sequence. This method constructs an optimum face/non-face classifier through the composition of *weak* classifiers, using the well-known AdaBoost algorithm. Each weak classifier is based on the response to a Haar-like wavelet filter. A sample detection result is shown on the left of Figure 1.

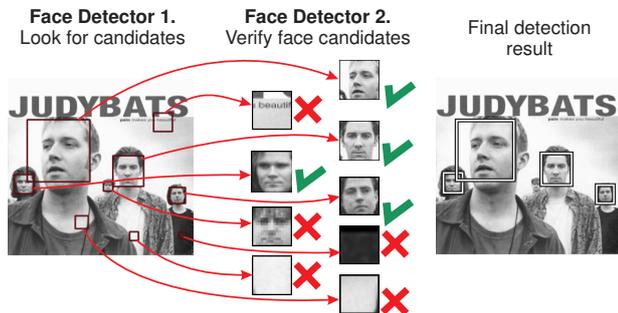


Figure 1. Human face detection using a combination of detectors. First, we apply Viola and Jones’ detector [13] on the input images. The resulting face candidates are passed to a second face detector [5]; only the faces that pass both tests are considered. Most false positives are rejected, although some real faces may be lost. The image was taken from the CMU/MIT face database [10].

Thanks to its public and free implementation in OpenCV libraries [1], Viola and Jones’ detector is probably one of the most popular techniques. In our own tests, it achieves a detection rate about 90% of the existing faces, with 10% false-positives per image (73 false alarms in 737 images). However, such false-positive rate could be inadmissible in some applications. In order to reduce it, we apply a second face detector on the results of the first one. In this way, we can say that the first detector looks for candidates, and the second detector verifies them.

Concretely, candidates are verified with the face detector proposed by Garcia *et al.* [5]. This method is based on vertical and horizontal integral projections –i.e., the sum of pixels along a row or column of the image, respectively–. In particular, the vertical projection of each candidate, PV_{face} , is computed and compared with a trained model of the face, MV_{face} , using a normalized Euclidean distance¹. If the distance is low enough, the horizontal projection of the eyes’ region, PH_{eyes} , is obtained and compared with a model MH_{eyes} . When both distances are low, the face candidate is accepted. Otherwise, it is rejected. A sample of the process is depicted in Figure 2.

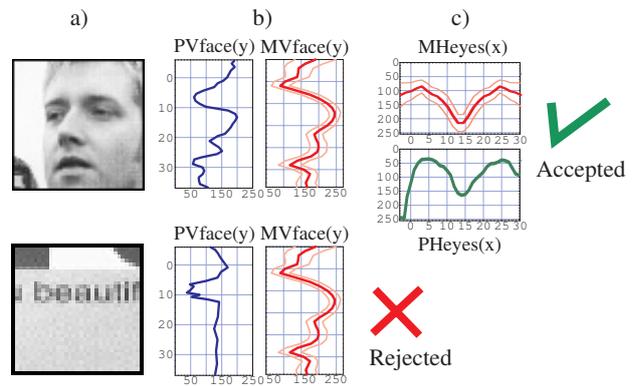


Figure 2. Face candidate verification using integral projections. a) Two candidates resulting from Viola and Jones’ detector, taken from Figure 1. b) For each candidate, the vertical integral projection, PV_{face} , is compared to the model, MV_{face} . c) The horizontal projection, PH_{eyes} , is compared to the model, MH_{eyes} . The upper sample is accepted; the lower is rejected after the first test.

In our experiments, this combination of methods is able to reduce 4 times the false-positive rate, obtaining 2.5% false alarms per image with a detection rate of 90% of the existing faces. Moreover, while Viola and Jones’ method takes an average of 292.5 ms per image in a Pentium IV at 2.6GHz, our combined detector just requires 295.6 ms. Clearly, the increment in computational time is negligible.

2.2. Facial features location

The detection process presented in subsection 2.1 is applied on the first frame of the sequence, or whenever the face is lost. In the following frames, a tracker is used to update the location of the faces through the sequence. In our case, we have used a 2D facial tracker that is able to locate the eyes and the mouth of each face. Later on, in section 3, we will describe how to translate this 2D information into a 3D facial pose.

¹The projection model, MV_{face} , consists of the mean and variance of $PV_{face}(y)$ for a training set of faces, for each point y of the 1D signals. Similarly for the projection model of the eyes’ region, MH_{eyes} .

Once the face has been detected, the next step is to find the precise location of the main facial features, such as eyes and mouth. Like most face detectors, Viola and Jones' method [13] simply outputs a rough location of the found faces, described by bounding rectangles. Thus, we apply the facial location technique presented in [5] on the given rectangles, obtaining the position of right eye, $(e1_x, e1_y)$, left eye, $(e2_x, e2_y)$, and mouth, (m_x, m_y) .

Again, the facial features locator is based on integral projections of the bounding rectangle of the face. The key element of this approach is an iterative algorithm for the alignment of 1D signals [5], the projections. The location of eyes and mouth are obtained by aligning vertical and horizontal projections of the face, as shown in Figure 3.

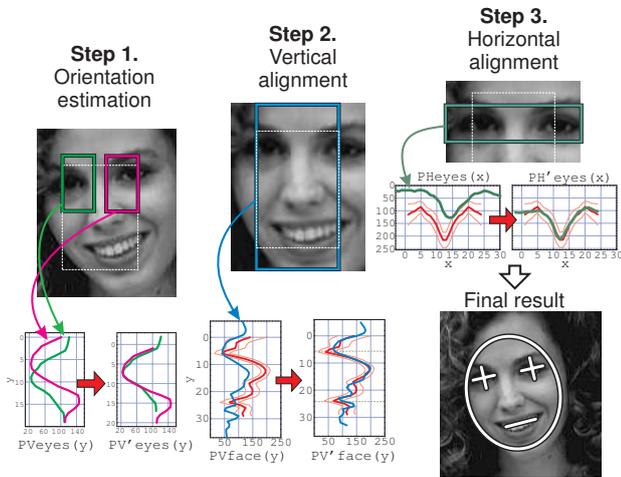


Figure 3. Structure of the facial features location algorithm [5]. In step 1, the orientation of the face is estimated using the vertical projections of both eyes. Then, in step 2, the vertical projection of the whole region is used to find the vertical position of the face. Similarly, the horizontal projection of eyes is computed and aligned in step 3.

First, the vertical integral projections of the expected left and right eye's regions are used to estimate 2D orientation of the face. Then, the vertical projection of the face is aligned with respect to a model, MV_{face} , to find the vertical position of eyes and mouth. Finally, the horizontal projection of the eyes is aligned to MH_{eyes} in order to determine horizontal position of the features.

2.3. Face tracking

The problem of 2D facial tracking is closely related to that of facial features location. In this sense, we can consider tracking as a continuous relocation of the features of interest for each new frame of the sequence. That way, our face tracker is very similar to the technique just described in subsection 2.2. There are only two main differences:

1. The projection models for MV_{face} and MH_{eyes} are

computed from the sequence itself. Observe that, in the case of facial features location, generic models were used for these integral projections.

2. The orientation estimation step is performed *after* vertical and horizontal alignments, instead of *before*. While in facial features location the observed inclination can be relatively high, in tracking only a presumably slight variation needs to be considered.

Besides, the sum of alignment distances is used to detect the end of tracking: when the distance of aligned projections to the models is above a given threshold, the face is considered lost. A more detailed description of the face tracker used in the experiments can be found in [3].

3. 3D facial pose estimation

As we have seen, the main novelty of our approach consists of separating the problems of tracking and pose estimation. That separation makes it possible to construct a 3D pose estimator which works with a reduced amount of data: the 2D location of three facial features. This way, a complex 3D tracker is not required, but almost any of the existing 2D facial trackers could be applied in our system.

Henceforth, we suppose that a face has been detected, located and tracked in a video sequence. The right eye is initially located on $(e1_x, e1_y)$, the left eye on $(e2_x, e2_y)$, and the mouth on (m_x, m_y) .

3.1. Estimation of 3D location

Suppose we have a three-dimensional coordinate system centered on the camera, as in Figure 4, with the Z-axis along the optical axis. The center of the face –defined as the mean position of eyes and mouth– is named $p = (p_x, p_y, p_z)$; f is the focal length of the camera; and the center of the face in the image is $c = (c_x, c_y)$.

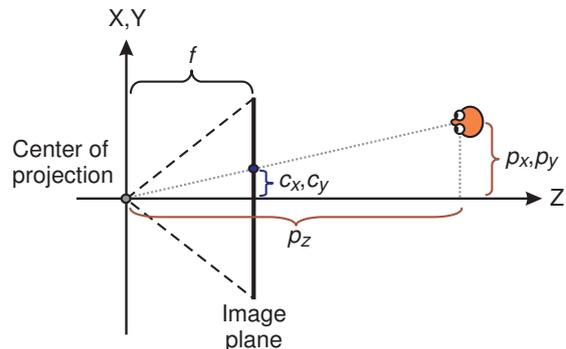


Figure 4. 3D model of the face and the camera. The face is situated in $p = (p_x, p_y, p_z)$ and the center of projection in $(0, 0, 0)$.

Under a perfect perspective projection, the relationship between p and c is given by:

$$\frac{c_x}{f} = \frac{p_x}{p_z}; \frac{c_y}{f} = \frac{p_y}{p_z} \quad (1)$$

Where:

$$c_x = \frac{e1_x + e2_x + m_x}{3}; c_y = \frac{e1_y + e2_y + m_y}{3} \quad (2)$$

So the position of the face in X and Y is:

$$p_x = \frac{e1_x + e2_x + m_x}{3} \frac{p_z}{f}; p_y = \frac{e1_y + e2_y + m_y}{3} \frac{p_z}{f} \quad (3)$$

Observe that both are given with respect to the distance to the camera, p_z , and the focal length, f . The first is inversely proportional to the size of the face in the image, r . Let t be the *real* size of the face in the 3D world. We have:

$$p_z = f \frac{t}{r} \quad (4)$$

Some previous works have associated r with the distance between the eyes. However, such heuristic *size* is falsely reduced with pitch and yaw, as shown in Figure 5. For example, when the person looks to the left or to the right the interocular distance is reduced.

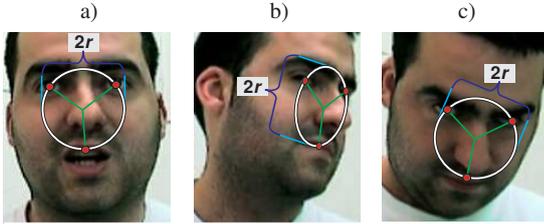


Figure 5. Estimation of the apparent facial size, r . Although the distance to the camera, p_z is the same in all cases, the eye-to-eye and eye-to-mouth distances are not constant. However, the major axis of the ellipse that contains eyes and mouth is always the same.

We propose a more robust method. Consider a hypothetical circumference that passes through the eyes and the mouth of a human face, as in Figure 5a). The perspective projection of a circumference is always an ellipse, regardless of its pose. Moreover, under weak perspective conditions², the major axis of the ellipse remains constant for all the possible poses. The demonstration is straightforward. Some samples of this property can be seen in Figure 5.

In the 3D world, the eyes and the mouth approximately form an equilateral triangle. Thus, those three points can be seen as a uniform sampling of the hypohetic circumference. In consequence, the covariance matrix of the points is related with the ellipse. This matrix takes the form:

²This is not contradictory with previous considerations. Simply, the effect of perspective *inside* the face is negligible.

$$\Sigma = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix} = \frac{1}{3} P^T \cdot P; P = \begin{bmatrix} e1_x - c_x & e1_y - c_y \\ e2_x - c_x & e2_y - c_y \\ m_x - c_x & m_y - c_y \end{bmatrix} \quad (5)$$

The first eigenvalue of Σ corresponds to the major radius of the ellipse, that is the projection of the hypothetical circumference. It can be computed in closed form with:

$$r = \sigma_x^2 + \sigma_y^2 + \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4\sigma_{xy}^2} \quad (6)$$

Replacing the value of r , calculated in equation 6, within equation 4 and then in 3, we obtain the 3D location of the face (p_x, p_y, p_z) .

Parameters f and t in equations 3 and 4, can be estimated with prior calibration. For example, the size t is normally around 4 cm for an average human head. In fact, both values just affect the scale factor.

3.2. Estimation of roll angle

When pitch and yaw angles are small (i.e., near frontal look to the camera), roll angle can be approximately associated with the 2D rotation of the face in the image, that is, the angle of the line passing through both eyes. In consequence, roll estimation is straightforward:

$$roll = \arctan \frac{e2_y - e1_y}{e2_x - e1_x} \quad (7)$$

Some sample estimations of roll can be seen in Figure 6.

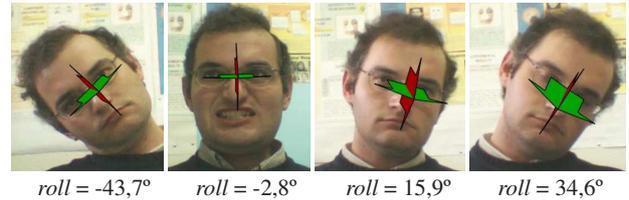


Figure 6. Estimation of roll angle of the face using the location of the eyes. Some faces and the resulting *roll* are shown. Observe that faces can have different expressions.

Although equation 7 is valid in most practical situations, we have to observe that it is not precise in all cases. For example, if the face and the camera are at different heights, a horizontal displacement can result in an erroneous roll not equal to 0.

3.3. Fixed body assumption

Pitch and yaw are normally the most difficult parameters to be estimated of facial pose. However, the problem can be greatly simplified by introducing two assumptions: (1) the location of the person does not change along the sequence

–e.g. the user is sitting on a chair, so his/her body remains still–; and (2) the user is initially looking frontally to the camera. If the first assumption does not hold, an additional step of body detection and tracking could be used to compensate body movement. The second assumption can be accomplished, for example, by requesting the user to look directly to the camera.

Taking into account human anatomy, the pair head/neck can be modeled as a *robotic arm* with three rotational degrees of freedom, that we have called *roll*, *pitch* and *yaw*. This simplified anatomical model is shown in Figure 7.

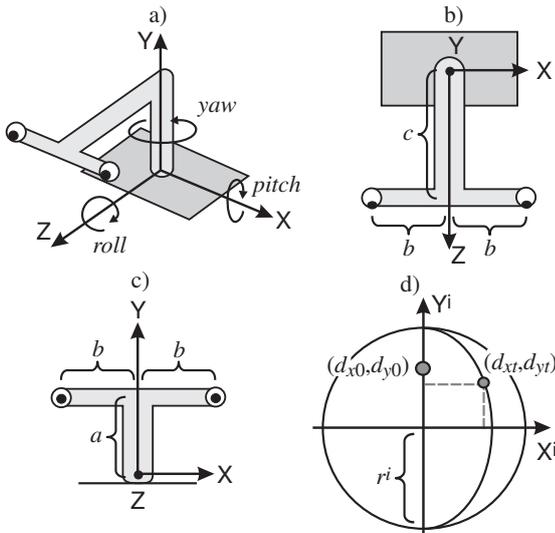


Figure 7. Simplified face model for 3D pose estimation. a) The human face as a robotic arm. b) Top view. c) Front view. a - height of the eyes with respect to the center of rotation; b - half of the eye-to-eye distance; c - horizontal distance from the center of the eyes to the neck. d) The mean position between the eyes (d_x, d_y) , and the circle generated for different poses, in image coordinates $(X^i$ and $Y^i)$.

Roll angle has been estimated independently, so we have to compute pitch and yaw. Let $p_0 = (p_{x0}, p_{y0}, p_{z0})$ be the initial location of the base of the neck in the 3D world. The left eye would be situated in $(p_{x0} - b, p_{y0} + a, p_{z0} - c)$, and the right one in $(p_{x0} + b, p_{y0} + a, p_{z0} - c)$. a , b and c are anatomic parameters defined in Figure 7b,c). The mean point of the eyes is situated in $(p_{x0}, p_{y0} + a, p_{z0} - c)$.

Clearly, in this *ideal* robotic model any point of the head lies in a sphere when pitch and yaw change, as shown in Figure 7d). So the problem can be reduced to a classical computation of latitude/longitude in a spherical orbit. Obviously this result is a mere approximation –because of the different simplifications imposed–. But neck anatomy greatly reduces the possible angles of rotation of the head, and our approximation is valid and very accurate for small angles.

3.4. Estimation of pitch and yaw angles

Consider the mean point of the eyes in the 3D world. The radius of the sphere where this point lies for different pitch and yaw angles (in units of the real world) is:

$$r^w = \sqrt{a^2 + c^2} \quad (8)$$

In coordinates of the image, this hypothetic 3D sphere is projected into a circle of radius:

$$r^i = r^w \frac{f}{p_z} \quad (9)$$

Making use of assumption number (2), and being $d_0 = (d_{x0}, d_{y0})$ the initial mean location of the eyes –i.e. the mean of $(e1_x, e1_y)$ and $(e2_x, e2_y)$ in the first frame of the sequence–, the center of the circle is located in:

$$(d_{x0}, d_{y0} - a \frac{f}{p_z}) \quad (10)$$

Now, we can compute pitch and yaw for each frame t where the face has been tracked. Let $d_t = (d_{xt}, d_{yt})$ be the mean location of the eyes in frame t ; the estimation of pitch would be:

$$pitch = \arcsin \frac{d_{yt} - (d_{y0} - a \cdot f/p_z)}{r^i} - \arcsin \frac{a}{c} \quad (11)$$

Observe that $\arcsin(a/c)$ is always subtracted because the pose of “looking to the front”, $pitch = 0$, does not necessarily correspond to the center of the circle (see Figure 7d). Some sample results of pitch estimation can be seen in Figure 8.



Figure 8. Sample estimations of pitch angle of the face. An arrow is drawn in the images according to the resulting 3D pose parameters.

On the other hand, yaw can be easily computed with:

$$yaw = \arcsin \frac{d_{xt} - d_{x0}}{r^i \cdot \cos(pitch + \arcsin(a/c))} \quad (12)$$

Figure 9 contains some examples of yaw estimation.



Figure 9. Sample estimations of yaw angle. An arrow is drawn in the images according to the resulting 3D pose parameters.

4. Experimental results

We have compared the proposed technique with a heuristic pose estimation system described in [4]. This method is based on integral projections, and is centered on the calculation of pitch and yaw. Basically, pitch is computed using vertical projections, and yaw with horizontal projections. In many aspects, this method is comparable with the technique presented in this paper: it is approximate, fast, intended for pose estimation in video, and requires no training.

A video sequence with facial rotations has been used in the experiments. This sequence contains a great variety of rotations: nodding and shaking while looking to the camera; nodding while looking to the right and to the left; and shaking while looking up and down. The user is sitting on a chair, so the fixed body assumption is met. The acquisition system was an off-the-shelf webcam at a 640×480 resolution, and the sequence contains a total of 490 frames. Some extracts of the sequence are presented in Figure 10.

The 3D position of the face in the first frame, with respect to the camera, was estimated in $p_{x0} = -5.6$ cm, $p_{y0} = -1.2$ cm, $p_{z0} = 73.1$ cm. The initial roll was 2.3° , and by definition pitch and yaw were set to 0. As the movement in the video consists of rotations in pitch and yaw, we center our attention in the estimation of both parameters. Figure 11 shows the results obtained with the method proposed in this paper and the one described in [4].

We can extract some relevant conclusions from this experiment:

- **Global assessment.** In general, the proposed method offers good estimations of the 3D facial pose, in particular rotation angles. It greatly improves the heuristic estimation, which presents bigger instabilities and erroneous values for higher angles. A ground truth 3D pose is not available to compare with.
- **Combined rotations.** The gestures that present more difficulties are those with high angles in different axis, for example, high roll and pitch. An extreme case of

this problem happens between frames 270 and 300, where face tracking is partially lost due to a great pitch and yaw; face location falls outside the expected sphere of possible rotations, so the computed angles are not correct. In fact, this is not a problem of pose estimation, but of the 2D facial tracker we have used.

- **Range of valid angles.** With respect to the previous point, the facial tracker used in the experiments is able to work correctly in a range of $\pm 20^\circ$ in pitch, and $\pm 40^\circ$ in yaw; roll angle is not limited. Within this range of rotations, the proposed 3D pose estimator is able to produce very accurate results.
- **Computational efficiency.** It is clear that our method is extremely fast, since it consists of a small and fixed number of calculations. In a Pentium IV at 2.60GHz, it just takes 0.006 ms per frame. The heuristic method took an average 0.18 ms per frame. In comparison, face tracking required about 8 ms per frame, including file reading and decompression.



Figure 10. Sample frames from the video used in the experiments. Only an extract of the faces is shown.

5. Application in a perceptual interface

Our pose estimation method is intended for applications where a rapid response is preferable to an accurate but slow computation. Its low computational load makes it very adequate in cases where this factor is important, for example in mobile devices. One such application is perceptual interfaces. The user interacts with the system through the movements of his/her face; accuracy is not critical, but the com-

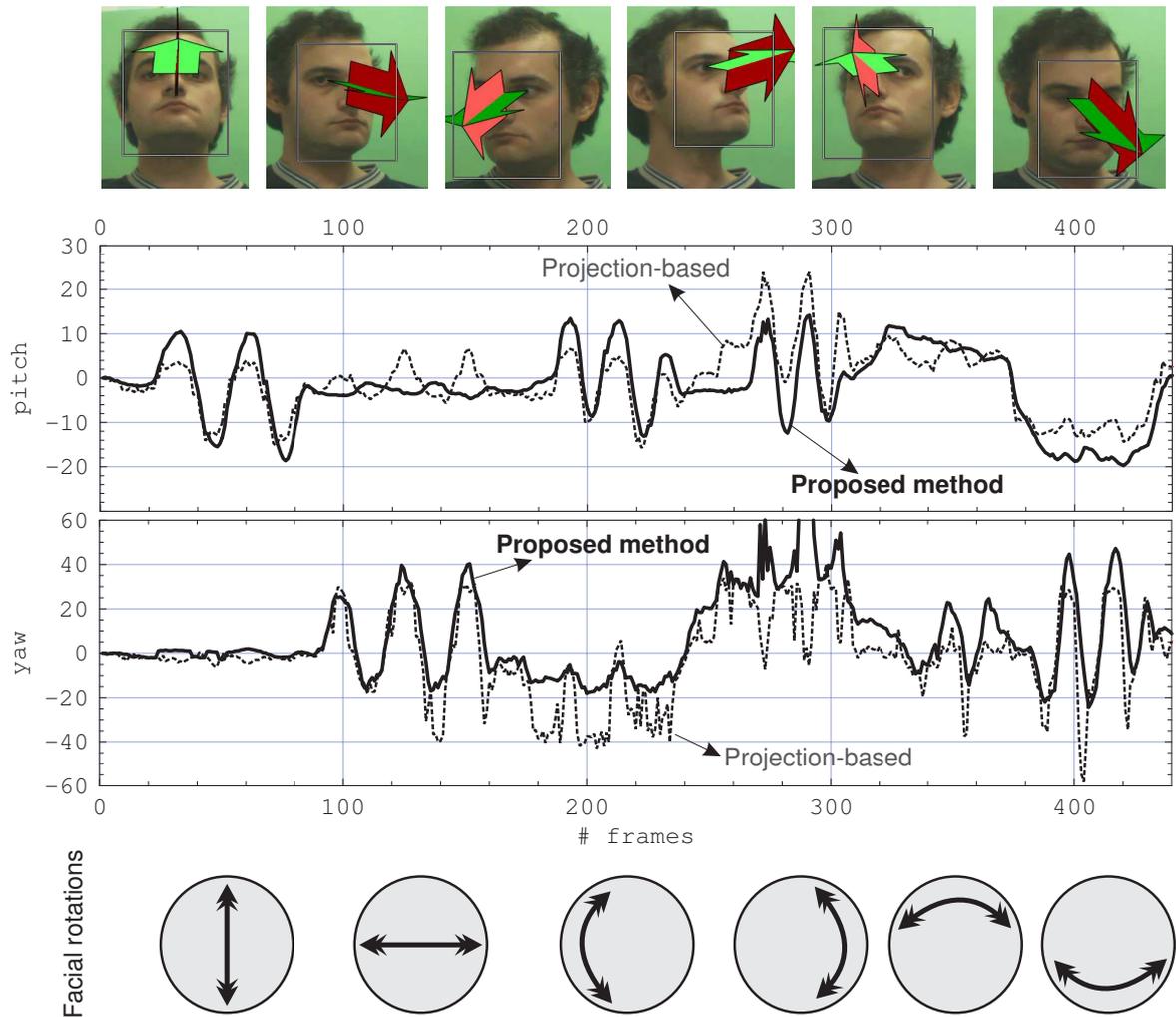


Figure 11. Facial pose estimation in a video sequence. The resulting vertical rotation (*pitch*) and horizontal rotation (*yaw*) are shown along time, for the projection-based method (dashed line) and the proposed method (thick line). Upper line: some sample faces (in the approximate positions where they happen) with the tracked position and a representation of the estimated 3D pose. Lower line: indication of the rotations done along time. The resulting file is available in: <http://dis.um.es/profesores/ginesgm/fip>

puter vision tasks –which are only one part of the system– have to be very efficient.

In [4], a perceptual interface is used to control the navigation parameters in a virtual 3D world. The action in the program is carried out in a first-person view, like a 3D video-game. Some sample images of this system are presented in Figure 12. Basically, p_x parameter is transformed into a lateral movement; p_y controls the height of the virtual observer; p_z is used to move forward or backwards; both *roll* and *yaw* involve a rotation in the world to the left or to the right; and *pitch* controls the look up/down state.

The perceptual interface presented in [4] uses the integral projection-based technique described in section 4. We are working on the integration of the method proposed here

into this virtual world. Currently, the system is able to work correctly with roll, pitch and yaw angles. To allow a control by 3D location, (p_x, p_y, p_z) , the fixed body assumption has to be removed. We have devised two ways of doing it. The first one is to perform a 2D tracking of the user’s body –for example, of the shoulders–. This way, simple movements like going to the left, right, forward, etc., could be detected and separated from the head movement; the pose of the head would be measured relatively to the body. The second possibility is to initialize the system whenever a frontal pose is detected. For example, a face detector could be applied periodically; as the detector is designed to find upright frontal faces, when a face is detected with high probability, the current position could be recomputed.

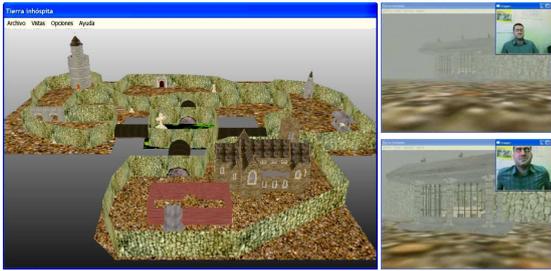


Figure 12. A view of the virtual world perceptually controlled by the movements of the user's head. Some sample uses of the program are shown on the right. This program is available in: <http://dis.um.es/profesores/ginesgm/fip>

6. Conclusions

Three-dimensional pose estimation has been, so far, one of the main problems in 3D human face processing. It remains an open problem, since it is difficult to find a method that is, at the same time, accurate, efficient, robust and generic (in the sense of no need to be trained for each user). We have presented a new method to estimate the 6 parameters of 3D pose, centering our attention in efficiency and robustness. These properties could be preferable in some applications where precise estimations are not necessary.

The main contribution of our work consists of developing a method that only requires the location of the eyes and the mouth. This is done by separating the preliminary steps of face detection, location and 2D tracking, from the computation of 3D facial pose. As we have discussed, the problem can be greatly simplified by introducing some assumptions. In particular, when we consider only the possible movements of the head (not of the user's body), human anatomy allows us to define a simple model of the head/neck system; this makes the problem of 3D orientation estimation straightforward. Curiously, this important property has not been exploited so far by previous research.

Our experiments prove that, compared to other fast pose estimation methods, the achieved accuracy of our system is very high, and suitable for applications such as perceptual user interfaces and aid for the disabled. Nevertheless, our main goal was not precision, and some of the assumptions may produce noise in the obtained values. We believe that future research in the field of 3D facial pose should make use of the anatomical limitations of the head/neck system; in particular, it would be very interesting to devise new kinetic models to explain the admissible positions and rotations of the head with respect to the rest of the human body.

Acknowledgements

This work has been jointly supported by the Spanish MEC and European Commission FEDER funds un-

der grants "Consolider Ingenio-2010 CSD2006-00046" and "TIN2006-15516-C04-03".

References

- [1] I. Corporation. The open source computer vision (OpenCV) library homepage. URL: <http://www.intel.com/research/mrl/research/opencv/>. 2
- [2] F. Dornaika and J. Orozco. Real time 3D face and facial feature tracking. *Journal of Real-Time Image Processing*, 2(1):35–44, 2007. 1
- [3] G. Garcia-Mateos. Refining face tracking with integral projections. In *4th Intl. Conf. on Audio- and Video-based Biometric Person Authentication (AVBPA)*, volume LNCS 2688, pages 360–368, Guildford, UK, 2003. 3
- [4] G. Garcia-Mateos and S. Fructuoso-Munoz. A fast and stable facial interface using integral projections. *Pattern Recognition and Image Analysis*, 17(4):462–469, 2007. 6, 7
- [5] G. Garcia-Mateos, A. Ruiz, and P.E. Lopez-de-Teruel. Human face processing with 1.5D models. In *3rd Int. Workshop on Analysis and Modeling of Faces and Gestures (AMFG 2007)*, volume LNCS 4778, pages 220–234, 2007. 2, 3
- [6] H. Kawanaka, H. Fujiyoshi, and Y. Iwahori. Human head tracking in three dimensional voxel space. In *ICPR06*, pages III: 826–829, 2006. 1
- [7] W. Liao, D. Fidaleo, and G. Medioni. Integrating multiple visual cues for robust real-time 3D face tracking. In *AMFG07*, pages 109–123, 2007. 1
- [8] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987. 1
- [9] B. Moghaddam, J. Lee, H. Pfister, and R. Machiraju. Model-based 3D face capture with shape-from-silhouettes. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG 2003)*, pages 20–27, 2003. 1
- [10] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–28, 1998. 2
- [11] W. Ryu and D. Kim. Real-time 3D head tracking under rapidly changing pose, head movement and illumination. In *4th Int. Conference on Image Analysis and Recognition (ICIAR 2007)*, volume LNCS 4633, pages 569–580, 2007. 1
- [12] K. Toyama. Prolegomena for robust face tracking. In *Workshop on Automatic Facial Image Analysis and Recognition Technology (ECCV '98)*, 1998. 1
- [13] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Intl. Conf. on Computer Vision and Pattern Recognition, CVPR 2001*, pages 12–14, Kauai, Hawaii, 2001. 2, 3
- [14] J. Xiao, T. Moriyama, T. Kanade, and J. Cohn. Robust full-motion recovery of head by dynamic templates and re-registration techniques. *International Journal of Imaging Systems and Technology*, 13:85–94, 2003. 1
- [15] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002. 2