Improved strategies for HPE employing learning-by-synthesis approaches

Andoni Larumbe, Mikel Ariz, José J. Bengoechea, Rubén Segura, Rafael Cabeza, Arantxa Villanueva
Department of Electrical and Electronic Engineering, Public University of Navarre
Navarre, Spain
{andoni.larumbe,mikel.ariz,rcabeza,avilla}@unavarra.es

Abstract

The first contribution of this paper is the presentation of a synthetic video database where the groundtruth of 2D facial landmarks and 3D head poses is available to be used for training and evaluating Head Pose Estimation (HPE) methods. The database is publicly available and contains videos of users performing guided and natural movements. The second and main contribution is the submission of a hybrid method for HPE based on Pose from Ortography and Scaling by Iterations (POSIT). The 2D landmark detection is performed using Random Cascaded-Regression Copse (R-CR-C). For the training stage we use, state of the art labeled databases. Learning-by-synthesis approach has been also used to augment the size of the database employing the synthetic database. HPE accuracy is tested by using two literature 3D head models. The tracking method proposed has been compared with state of the art methods using Supervised Descent Regressors (SDR) in terms of accuracy, achieving an improvement of 60%.

1. Introduction

In the computer vision field, Head Pose Estimation (HPE) is understood as the computation of the head position and orientation of a subject with respect to a given coordinate system, usually the camera one, *i.e.* the camera is considered to be the origin of the world coordinate system (WCS). Head pose information can be employed in alternative fields such as: human behavior analysis [3, 25], driver assistance [26] or gaze estimation systems [27]. It is a rich communication tool and it can be considered as a bridge for the communication between subjects and computers in applications belonging to the field of Human Computer Interaction (HCI) [16]. HCI has experienced an important rise in the past decade due to its multidisciplinary nature and its application in a vast number of fields, such as artificial intelligence or the control of mobile devices. Lately, research on HCI has focused on developing control methods without the need of touch, such as hand gesture recognition [30, 13, 28], head tracking [20, 33, 32] or gaze estimation [17] among others. For applications based on gaze estimation using off the shelf cameras, in which the eye area is not represented in great detail, it becomes of critical importance to have an accurate HPE system. The automatic recognition of the orientation of the head, eventually strengthened by the more accurate estimation of the gaze direction, is crucial in a number of assistive technologies applications as autism diagnosis, monitoring of social development, depression detection and human behavior analysis [14].

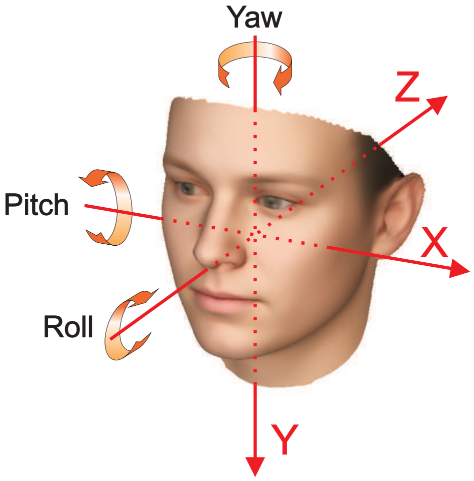


Figure 1. Definition of the spatial translations and rotations that determine the Head Pose employing the camera coordinate system. The head 3D model has been placed at the origin for simplicity.

Head Pose is generally defined as a six degrees of freedom (DOF) variable, *i.e.* three translation parameters (t_x, t_y, t_z) and three rotation angles (*roll, yaw, pitch*). These parameters are shown in Figure 1.

The methods for HPE can be divided into different types of approaches according to the work presented by Murphy-Chutorian and Trivedi [16]. Methods using appearance templates or detector arrays provide a coarser estimate of head position. Finer HPE values can be obtained if non-linear regression methods of manifold embedding methods are employed. The accuracy of HPE can be improved if other methods are utilized, such as, flexible models, geometric methods, tracking methods or hybrid methods.

Hybrid methods are ones of the most employed due to their flexibility and high accurate results obtained as a combination of different approaches. Pose from Ortography and Scaling by Iterations (POSIT) [6] retrieves HPE from the image assuming a correspondence between image features (2D landmarks) and head 3D model (3D landmarks) knowing camera intrinsic parameters. One of the most critical parts of tracking methods is the detection of 2D key features in the image to be tracked through a sequence of images. Temporal continuity and smooth motion are usually assumed. Tracking of face features in the wild has been largely studied in order to overcome problems such as occlusions and light variations among others [15]. Face detection has been studied with special emphasis paid to the fact of working in challenging conditions and with the aim to retrieve higher accuracies in terms of HPE.

One of the combinations retrieving a satisfying outcome results from the combination of a geometrical model such as POSIT and Supervised Descent Method (SDM) [31] as a method for 2D tracking. The SDM is an algorithm designed to minimize the non-linear minimum quadratic error avoiding the use of Jacobian and Hessian matrices. The solution provided consists in a previous training step in which the optimization procedure is learned for a given problem. A series of points are sampled around the global minimum, thus, the SDM regressor learns the optimal trajectories starting from the alternative sampled points to reach that minimum. Once the regressor has been trained for a given database new samples can be tested.

IntraFace [5] is a commercial software employing SDM in which face tracking is provided together with HPE and gaze direction among others. The authors do not provide detailed information about the implementation of the training procedure. However, it is known that a proprietary version of Scalar Invariant Feature Transform (SIFT) is employed.

Feng *et al*. have devised a publicly available alternative method for image tracking using SDM [7]. A variation of the Histogram of Oriented Gradient (HoG) is used as feature and the structure of the regressor provided is a Random Cascaded-Regression Copse (R-CR-C). The main idea behind the R-CR-C is to design multiple cascaded regressors (CR) and fuse their estimates instead of employing a single CR as in the original implementation. Each CR thread have a series configuration. The results obtained provide a better balance between the loss of precision and the risk of overfitting. Moreover, the R-CR-C is independent from the scale, *i.e*. the size of the face does not interfere in the result.

Evaluating the accuracy of the alternative tracking and HPE methods is not trivial. In most cases the assessing of the different methods is based on ground truth values that have been manually labeled or that been obtained using low accuracy methods. The absence of databases in which all the parameters, *i.e*. 3D pose and 2D ground truth, are under control prevent researchers from obtaining reliable results. The first contribution of this paper is the presentation of a synthetic video database for HPE and 2D tracking in which camera, head model and image data are provided. The objective of this database is to provide a framework in which both, alternative supervised learning and HPE methods can be trained and tested in a consistent fashion and under completely controlled conditions.

The second and main contribution is the submission of an optimal hybrid method for HPE based on POSIT by implementing new strategies for 2D landmark detection based on R-CR-C. The accuracy obtained is compared with state-of-art methods. Regarding the 3D head model, two proposals are tested, the Basel Face Model (BFM) proposed by the University of Basel [19] and the Surrey Face Model (SFM) proposed by the University of Surrey [10].

Section 2 describes the complete synthetic database, explaining its contents, structure and the simulation tool with which the database has been generated. Section 3 presents the combination of geometrical model and 2D tracking method proposed to perform the Head Pose Estimation. First, an improved tracking method is described. Secondly, two 3D head models are presented. In section 4, a HPE accuracy comparison using different 3D models and supervised descent regressors is made; a comparison between a HPE method using POSIT and other state-of-art methods is also performed. Section 5 includes the final remarks.

2. UPNA Synthetic Head Pose Database

Traditionally, HPE and supervised learning methods have been trained or evaluated using images or video sequences of real people performing a variety of head movements. The problem with those environments is that it is impossible to control the variables affecting the image and pose data acquisition or the labeling process for training.

In order to solve this problem, we have designed a simulator tool that allows us to create synthetic images or videos of head movements in which all those variables are controlled by the user and can be set in different manners depending on the goal of the study. New video sequences can be created at any moment if new requirements are considered for the application and if new studies want to be carried out, without the need of real subjects for the videos and the tedious task of setting up a new recording session.

Using this tool, we have created a synthetic video database that can be used to train or evaluate HPE and supervised learning methods among others according to learning-by-synthesis principles.

2.1. Simulator Tool

This tool can be divided into two main modules: the design of the simulation, where parameters that characterize the different variables of the simulation are specified according to the desired output; and the building of the simulation, where the previously defined parameters are used to generate the output as specified. These modules have been designed to be run sequentially and they will be described in detail in the following lines.

2.1.1 Simulation Designing

In this first step, the whole simulation is defined by setting different parameters that determine the output according to the needs of the user. The modifiable variables are the following:

- **Head model:** the simulator incorporates a generative 3D shape and texture model, the Basel Face Model (BFM) [19]. It is a publicly available 3D morphable face model. The model was built based on training data obtained from the 3D scans of 200 subjects, 100 females and 100 males between 8 and 62 years old, most of them Caucasian. All the scans contained a neutral facial expression and were registered using an Optimal Step Nonrigid ICP Algorithm [1] to ensure an optimized anatomical point correspondence between faces. The faces were parametrized as triangular meshes after registration, resulting in 53,490 vertices described by a coordinate vector $(x_i, y_i, z_i)^T \in R^3$ with an associated colour $(r_i, g_i, b_i)^T \in [0,1]^3$. Principal component analysis (PCA) was then applied to create an orthonormal basis of 199 principal components of texture and shape, which allows us to generate new observations as linear combinations of those components. It is thus a face generator in which, just by assigning the PCA coefficients for the principal components of shape and texture, we can create new faces at any moment. Note that, if all coefficients are set to zero, the mean face of the PCA is obtained.

  The simulator is thus able to create new users from a meta-database of 3D faces. Besides, a certain set of 3D facial points (3D landmarks) can be passed to the simulator so that their projection on the image plane is stored (2D landmarks). This allows us to obtain a 2D ground truth for any video sequence generated with the simulator, which is very useful for training purposes or for point tracking algorithm evaluation.

- **Camera parameters:** we can set the parameters that define the image or the video that would produce a real camera: the image resolution, the frame rate and the intrinsic camera parameters (*i.e.* focal length, principal point, radial and tangential distortion, and skew). The images or videos the simulator will retrieve, will correspond to what a camera of those characteristics would acquire.

- **Motion:** the motion parameters define the head movements that will compose the created video. We can define the length of the sequence as a number of frames (the frame rate has already been defined), and the head movements in the 6 DOF (*i.e.* translation in X, Y, Z; and *roll, yaw,* and *pitch* rotations) that will determine the head pose in each of the frames. Rotation matrices **R** can be obtained from the elemental rotation matrices $\mathbf{R}_{\theta_z}, \mathbf{R}_{\theta_y}$ and $\mathbf{R}_{\theta_x}$ with Euler angles θ_z, θ_y and θ_x around the Z, Y and X axes, *i.e.* from *roll, yaw* and *pitch* angles as:

$$\mathbf{R} = \mathbf{R}_{\theta_z}\mathbf{R}_{\theta_y}\mathbf{R}_{\theta_x}. \tag{1}$$

2.1.2 Simulation Building

This step consists in running the simulation according to the parameters defined in the previous step. The head model is generated based on the model parameters, and transformed in each frame with the corresponding rotation and translation values.

Having *N* 3D points $\mathbf{p} = [x_1, y_1, z_1, \ldots, x_N, y_N, z_N]^\mathbf{T}$, where $\mathbf{p}_n = [x_n, y_n, z_n]^\mathbf{T}$ are the coordinates of the n^{th} point, each point can be mapped to a new position $\mathbf{q}_n = [x_n, y_n, z_n]^\mathbf{T}$ using the motion parameters defined, by the rigid transformation:

$$\mathbf{q}_n = \mathbf{R} \cdot \mathbf{p}_n + \mathbf{t}, \tag{2}$$

where **R** is the 3×3 rotation matrix calculated by Equation 1, and $\mathbf{t} = [t_x, t_y, t_z]^\mathbf{T}$ is the spacial translation vector.

The simulator also calculates for each video frame which part of the model is visible and which is not, applying the Hidden Point Removal (HPR) algorithm [11]. It consists in transforming a point cloud according to the viewpoint and extracting the points that reside on the convex hull, which leads to determining the visible points in the cloud. The model is then projected onto the image plane using the shape and texture information in the corresponding pose. To follow, each 3D landmark is projected to obtain the 2D ground truth landmarks.

Using all the previous information, the output of the simulation is generated: a video file is created and all the parameters that define the simulation and the 2D ground truth are stored in the corresponding files.

2.2. Structure and Content

Using the tool described above, we have created the UPNA Synthetic Head Pose Database. The basic idea behind this has been to reproduce the database proposed by

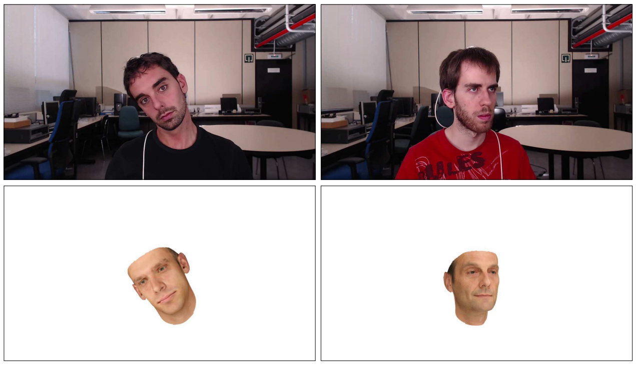


Figure 2. Frames from the real database (top level) and their equivalent in the synthetic database (bottom level). Synthetic user shown in bottom-left frame is created using the head generator; bottom-right one is from scans provided along with the BFM.

Ariz *et al*. [2], taking advantage of the simulator tool and the possibility to control every variable in play.

Similarly to the UPNA Head Pose Database, the synthetic video database consists of 120 videos of 10 different users. The first 5 users have been created using the head generator described in Section 2.1.1, assigning random coefficients to the principal axes in the PCA basis of the BFM. A large set of heads have been created using random shape and texture PCA coefficients from a Gaussian distribution with zero mean and standard deviation of one, from which the 5 users have been visually selected in order to include a certain variety among the faces regarding the gender, size, age, and facial appearance. The last 5 users in the database have been selected from the 10 example scans provided along with the BFM in their webpage [18]. These scans have also been registered but have not been included in the training set of the morphable model, and are not therefore exactly reproducible by a certain set of PCA coefficients.

The ten synthetic heads reproduce the exact pose variations of the real users using the ground truth pose files of the UPNA Head Pose Database. This allows us to assure certain realism in the synthetic database; rotations and translations at a constant speed were originally tried, and the visual effect was that of a robotic movement. Moreover, by copying the ground truth of the real database, we assure the same translation and rotation ranges are represented.

The frames provided in Figure 2 show two example frames from the real database (top level) and the corresponding frames from the synthetic database (bottom level). The user shown in bottom-left frame is one of the users created using the head generator while the user shown in bottom-right frame is one of the example scans provided along with the BFM. If we compare the synthetic frames with the real ones, we can observe that the head pose matches, the heads have similar proportions in the image, the appearance of the synthetic faces resembles reasonably that of a real face, and the main difference resides on the background, inexistent for the synthetic database.

Twelve videos per user have been thus generated, which include 6 guided-movement sequences and 6 free-movement sequences. The videos have been generated with a 1280×720 pixel resolution, at 30 frames per second. In the guided sequences, the user follows a specific pattern of movement: 3 pure translations (X, Y, Z) and 3 pure rotations (*roll, yaw, pitch*). Movement ranges include translations going up to more than 200mm in any axis from the starting point, and rotations up to 30°. In the free sequences, the user moves the head at free by combining translations and rotations along the 3 spatial axes. The order of videos is:

Video 01:	pure translation along the X axis.
Video 02:	pure translation along the Y axis.
Video 03:	pure translation along the Z axis.
Video 04:	pure rotation around the Z axis (roll).
Video 05:	pure rotation around the Y axis (yaw).
Video 06:	pure rotation around the X axis (pitch).
Videos 07-12:	free translations and rotations.

Each video is associated with three text files. One contains the 2D projections (in pixels) of the annotated 3D facial points, what we will call the 2D ground truth landmarks. The other two which contain the head pose with respect to the camera. These pose files are the same as those in UPNA Head Pose Database. Translations are given in millimeters and rotations in degrees. The difference between the two files is that one contains the originally acquired head pose, whereas the other one contains the equivalent ground truth beginning with '0-rotation'. Basically, it is the original pose transformed to get an exact zero rotation for the three angles in the first frame. This transformation is done by multiplying the inverse rotation matrix of the initial pose to the rotation matrix of each frame pose:

$$\mathbf{R}_0^{(i)} = \mathbf{R}^{(i)}\mathbf{R}_0^{\mathbf{T}}, \tag{3}$$

where $\mathbf{R}_0^{\mathbf{T}}$ is the 3×3 inverse rotation matrix of the initial pose, $\mathbf{R}^{(i)}$ is the 3×3 rotation matrix of the i^{th} frame pose, and $\mathbf{R}_0^{(i)}$ is the 3×3 zeroed rotation matrix of the i^{th} frame pose.

As a result, UPNA Synthetic Head Pose Database provides the 120 videos with their 2D ground truth landmarks projections, their corresponding head pose ground truth (zeroed and non-zeroed), the 3D head model of each user and the camera parameters with which the videos have been generated. UPNA Synthetic Head Pose Database is available by contacting the authors.

3. Methods

As mentioned above, the POSIT requirements are: the image features detected by the face tracking software (2D

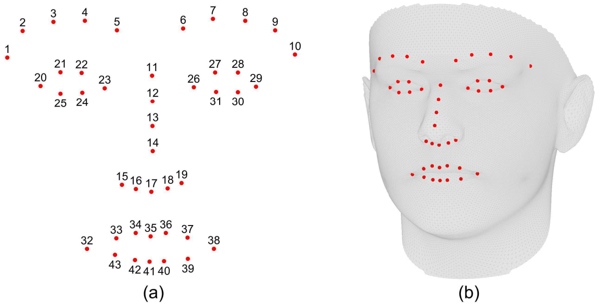


Figure 3. Set of 43 landmarks used; a subset of the 68 points used by IBUG [24]: (a) 2D landmarks detected by the tracking system; (b) 3D landmarks calculated for BFM.

landmarks), a set of head 3D model vertices that corresponds with the landmarks detected by the face tracking software (3D landmarks) and the camera parameters. In this section we will describe an improved tracking method and the 3D head models used to perform the Head Pose Estimation using POSIT algorithm.

The landmarks chosen to estimate the head pose are a subset of 43 landmarks from the 68 points used by IBUG [24]. Figure 3a represents the 2D landmarks detected by the tracking system and Figure 3b represents the 3D correspondences calculated for the BFM. Obtaining this correspondence is not a trivial process since manual labeling is highly uncertain thus, the method used to obtain the 2D-3D landmark correspondences will also be explained.

We choose this subset because, on the one hand the jaw landmarks occlude when images shows yaw rotations and, on the other, IntraFace detects a similar subset.

3.1. Tracking method improvements

To perform the 2D landmark detection we use *4DFace*. 4DFace is a face tracking software developed by Patrik Huber, a PhD student in the Centre for Vision, Speech and Signal Processing of the University of Surrey. The software is written in modern C++ and developed on GitHub [9]. Initially, we evaluated this software using the UPNA synthetic database and we determined that, for videos showing high roll values (*e.g.* videos 04), the tracker fails completely. In order to improve the tracking and obtain a more robust and accurate detection, we propose a method using a combination of two procedures: a roll normalization and a training data augmentation using synthetic images.

3.1.1 Roll normalization

The main idea behind the roll normalization is to provide an image that shows a lower roll value than the original one and to do the landmark detection upon these images.

For the first frame, we detect the face bounding box using Viola-Jones algorithm [29]. Using R-CR-C we can detect the 2D landmarks and perform an initial pose estimation using the *Gold Standard Algorithm* of Hartley & Zisserman [8, 10] implemented in the 4DFace software. We use this initial estimate to correct the roll rotation of the next frame in order to get a nearly 0-roll pose and prevent the tracker from getting lost. To follow, the landmark detection is performed. Once the landmarks are detected, the corresponding inverse roll rotation must be applied in order to calculate the position of the 2D face points in the original frame. In next frames, a similar procedure is applied.

3.1.2 Data augmentation

The face tracking software employed includes a pre-trained regressor [9] to which we shall refer as Surrey Supervised Descent Regressor (SSDR). SSDR has been trained with a set of 3283 images from AFW [34], HELEN [12], IBUG [23] and LFPW [4] databases. Nevertheless, landmarks used are re-annotated by using the IBUG semi-automatic annotation methodology [22, 24] followed by an additional manual correction. Images and landmarks used are available at [21]. SSDR detects the 68 IBUG landmarks, but we only use the 43 ones detailed in Figure 3.

The aim of adding synthetic images to the regressor training process is to increase the training data with a set of images that shows high rotation values. The synthetic images and the 2D ground truth landmarks used for this purpose have been obtained by using the UPNA Synthetic Head Pose Database presented in this paper.

We have trained two regressors using synthetic images, one using the same real images as SSDR and the other using a balanced number of real and synthetic images. Therefore, the regressors with which HPE accuracy is studied are:

1. **SSDR:** Regressor included on the face tracking software.

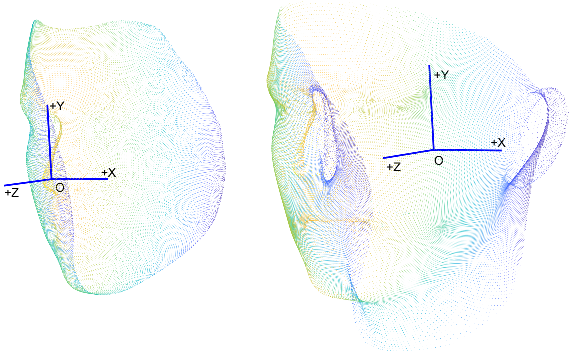
2. **MIX_1157:** Trained with 1157 real images from AFW and HELEN databases and 1200 synthetic images that show high yaw and pitch rotation values (videos 05 and 06 from UPNA Synthetic Head Pose Database)

3. **MIX_3283:** Trained with 3283 real images (same as SSDR) and 1200 synthetic images that show high yaw and pitch rotation values (same as MIX_1157).

A summary of how regressors have been trained is shown in Table 1. The number of real images and databases used is shown in the second column, and the number of synthetic images extracted from UPNA Synthetic Head Pose Database is shown in the third column. The type of rotation displayed on the images chosen for training is also detailed.

	Real images		Synthetic images			
Regressor	**N°**	**Databases**	**N°**	**Roll**	**Yaw**	**Pitch**
SSDR	3283	AFW, HELEN, IBUG, LFPW	-	x	x	x
MIX_1157	1157	AFW, HELEN	1200	x	✓	✓
MIX_3283	3283	AFW, HELEN, IBUG, LFPW	1200	x	✓	✓

Table 1. Trained regressors. The number of real images and databases used is shown in the second column, and the number of synthetic images extracted from UPNA Synthetic Head Pose Database is shown in the third column. The type of rotation displayed on the images chosen to training is also detailed.



(a) SFM

(b) BFM

Figure 4. Differences between (a) SFM and (b) BFM. Both models have different coordinate systems than the camera system ones (WCS) and the origin of the BFM and SFM do not coincide.

3.2. Head Models

We have studied the HPE accuracy using two different 3D models: the Basel Face Model (BFM) previously described and the Surrey Face Model (SFM) [10]. SFM was built based on training data obtained from the 3D scans of 169 subjects. In this case Non-Caucasian people are well-represented and a significant number of subjects from other races are included. The model comes in three different resolution levels with different number of vertices: the smallest model consists of 3448 vertices, the middle one consist of 16,759 and the full model consists of 29,587. To be able to obtain the best correspondence between 2D and 3D landmarks, the full model is used.

In both BFM and SFM, PCA coefficients are set to zero *i.e.* we employ the mean faces. Both models have different coordinate systems than the camera system ones (WCS), as it can be seen by comparing the Figures 1 and 4. The axes Y and Z are inverted and the origin of the BFM and SFM coordinate systems do not coincide. Thus, a coordinate system unification is required. We have inverted the Y and Z axes and set the BFM origin as the reference one.

As we said, obtaining the correspondence between the 2D landmarks detected by each tracking system (or trained regressor) and the 3D vertices of each model is not a trivial process. To achieve this, we synthetically generate an image in a known 3D position using the simulator tool. Then, in order to minimize the effect of jitter, we detect the 2D landmarks a hundred times and calculate the mean landmarks. Knowing the mean 2D landmarks, the ground truth 3D position and the camera parameters with which the synthetic image has been generated, we are able to calculate geometrically the 3D landmarks (Figure 3b) by means of Ray-Triangle intersection. In this manner, the correspondences between the 2D landmarks detected by the tracking software and the 3D model vertices are calculated.

4. Results

We evaluated the HPE accuracy on the database proposed by Ariz *et al.*: the UPNA Head Pose Database [2]. Thus, any database or image used for training is not used for testing. As we said, this database contains a set of 120 videos which correspond to 10 different subjects and 12 videos each: 6 guided-movement and 6 free-movement. Each video has 300 frames (30fps, 10 seconds in length).

Head pose estimation error is given by the mean and standard deviation ($\mu \pm \sigma$) of the absolute difference between the *zeroed* estimate and the *zeroed* ground truth. This time, the original pose is transformed to get an exact zero rotation as well as an exact zero translation. For this, we generate the 4×4 pose matrix

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}, \tag{4}$$

and the 4×4 inverse pose matrix

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{R^T} & \mathbf{\text{-}R^Tt} \\ 0 & 1 \end{bmatrix}, \tag{5}$$

where rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$ are obtained using POSIT algorithm.

Zeroed pose transformation is done by multiplying the inverse pose matrix of the initial pose to the pose matrix of each frame:

$$\mathbf{M}_0^{(i)} = \mathbf{M}^{(i)}\mathbf{M}_0^{-1}, \tag{6}$$

			Error	
Regressor	**RN**	**3D Model**	**Translation (mm)**	**Rotation (°)**
IF_126	-	**BFM**	9.14 ± 9.99	0.82 ± 0.96
		SFM	10.08 ± 10.58	0.85 ± 0.98
SSDR	*x*	**BFM**	10.21 ± 12.56	1.01 ± 1.25
		SFM	11.40 ± 13.30	1.05 ± 1.30
SSDR	✓	**BFM**	9.88 ± 11.29	0.98 ± 1.15
		SFM	10.60 ± 11.08	0.98 ± 1.12
MIX_1157	✓	**BFM**	8.90 ± 10.05	0.83 ± 1.01
		SFM	10.99 ± 12.12	0.97 ± 1.20
MIX_3283	✓	**BFM**	9.01 ± 10.21	0.90 ± 1.04
		SFM	10.17 ± 10.60	0.93 ± 1.06

Table 2. Methods comparison.

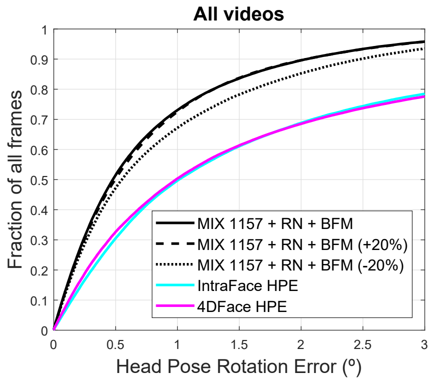


Figure 5. Comparison between methods and focal length variation impact. The horizontal axis shows the rotation error while the vertical one represents the fraction of total frames below each error value

where $\mathbf{M}_0^{-1}$ is the 4×4 inverse pose matrix of the initial pose, $\mathbf{M}^{(i)}$ is the 4×4 pose matrix of the i^{th} frame pose, and $\mathbf{M}_0^{(i)}$ is the 4×4 zeroed pose matrix of the i^{th} frame pose. In order to prevent a bad pose initialization and perform a more robust estimation, first frame 2D landmark detection is made a hundred times and mean 2D landmarks are calculated. $\mathbf{M}_0^{-1}$ is generated using this mean 2D landmarks.

Using $\mathbf{M}_0^{(i)}$ we can obtain the zeroed rotation angles and the zeroed translation from estimate and ground truth. Head pose estimation error is given by subtracting the zeroed estimate and the zeroed ground truth. This zeroed error can be defined as a differential error, which is the one that best describes the performance of a head pose estimator in a real application, where we have an estimation method and no ground truth. In such a situation, a calibration procedure where the user would be asked to face the camera in order to set the frontal reference position and to calculate the inverse pose matrix with zero-rotation $\mathbf{M}_0^{-1}$, is assumable.

4.1. Methods comparison

Table 2 presents a HPE accuracy comparison using POSIT and different head 3D models and supervised descent regressors. The first column specifies the regressor employed, the first one (IF_126) is the one included in the version 1.2.6 of the IntraFace software [5]; the other three regressors are the ones presented in section 3.1.2. As we said, IntraFace detects a similar set of landmarks but, in order to have comparable results, estimation is made only using the 43 ones defined in Figure 3. In the second column it is detailed if the roll normalization (RN) has been carried out or not. In the case of using IntraFace software, we can not implement the roll normalization. The third column specifies the head 3D model used among the two 3D models presented in section 3.2 (mean models). Finally, the fourth and fifth columns present the translation and rotation errors respectively.

If we compare 3D head models, we observe that HPE accuracy its quite similar using both models, although BFM seems to presents slightly better results. Regarding the roll normalization (RN), if we compare the two SSDR rows, we can see an improvement when we apply the roll normalization. This improvement is due to the tracking robustness against videos showing high roll values.

Concerning the regressors, we observe a clear improvement as we go from the one trained with real images only (SSDR) to the ones trained using both real and synthetic images (MIX_1157 and MIX_3283). If we compare our trained regressors with the IntraFace one, we can see that using MIX_1157, roll normalization and BFM, provides quite similar results to IntraFace.

Finally, if we compare the effect of using different 3D head models with the effect of using different supervised descent regressors, we can conclude that sensitivity of the HPE with respect to the 2D tracking accuracy is considerably higher than the one due to inaccuracies in the head 3D model. Therefore, optimizing the landmark tracking to do it more robust and accurate is highly important.

4.2. POSIT vs HPE software

IntraFace and 4DFace implementations provide their own pose estimation. Figure 5 depicts the difference in terms of HPE accuracy between our estimation method and the IntraFace and 4DFace ones. The graph shows which fraction of total frames is below a head pose rotation error. We refer only to head pose rotation error since IntraFace does not provide the translation estimate. We can see that, our method shows a 50% of the total frames below 0.5° of error while IntraFace and 4DFace present only a 30% of the total frames below the same error value. Furthermore, IntraFace results in a mean error of 2.07° and

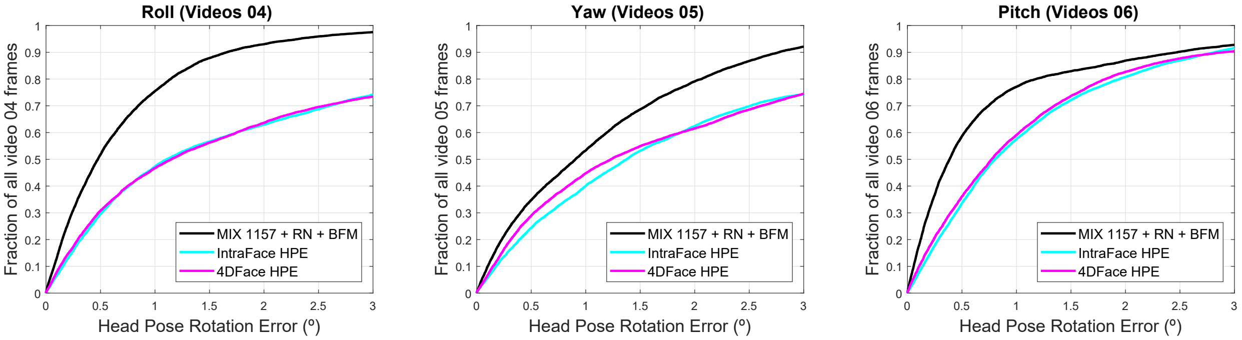


Figure 6. Comparison between methods grouping the results according to the type of head movement. Left graph depicts the HPE error in videos that show high roll angles, middle one refers to videos that show high yaw angles and the right one presents the HPE error in videos that show high pitch angles.

4DFace presents a mean error of 2.25° while our method reduces the mean error to 0.83°, entailing an improvement of 60%. Both, IntraFace and 4DFace estimations are uncalibrated HPE methods, meaning they do not require the camera parameters in order to produce a HPE, but estimate them from images. Nevertheless, we have tested the effect of focal length on the estimation of head rotation and we show that variations of ±20% in the focal length do not have a significant impact in the comparison between methods regarding head rotation error as it can be shown in Figure 5. Therefore it can be pointed out that moderate variations in the focal length have little effect on the estimation of the head rotation, *i.e.* the accuracy in the focal length estimation is not critical for head rotation estimation (but it is for translation). This allows us to compare our rotation estimate based on a calibrated method with the IntraFace and 4DFace ones.

We have also measured performance by grouping the results according to the type of head movement. As we said, videos 01 to 06 are guided sequences, the user follows a specific pattern of movement: three pure translations and three pure rotations. In Figure 6 we can see the difference in terms of HPE in the videos of pure rotations. A difference in accuracy between the three kind of videos can be observed. Videos showing high yaw angles (videos 05) present the worst results, high pitch angles ones (videos 06) show slightly better results and videos that show high roll angles (videos 04) present the best results. The videos of pure translation have also been studied but no differences have been found.

5. Conclusion

We have presented a simulator tool that allows us to create synthetic images or videos of head movements in which all the variables are controlled by the user and can be set in different manners depending on the goal of the study. Further, we have presented a synthetic video database created using this tool that can be used to train or evaluate HPE and supervised learning methods among others.

On the other hand, we have compared different 3D head models, as well as multiple facial-tracking systems to determine which combination provides a better estimation of a head pose. We have seen that the sensitivity of the HPE with respect to the 2D tracking accuracy is considerably higher than the one due to inaccuracies in the head 3D model. Therefore, optimizing the landmark tracking to do it more robust and accurate is highly important. Adding synthetic images to the regressor training process and performing a roll normalization increase the robustness and accuracy of the tracking systems. We have also compared our calibrated HPE method using POSIT with other state-of-art uncalibrated HPE methods and we have seen that using a meticulous combination of state-of-art techniques our method presents an improvement of 60% in terms of average HPE error. The performance achieved by our method using SDR show promising results.

As future work we propose to carry out a careful quantification of the sensitivity of the HPE regarding both, 3D model and 2D landmark tracking inaccuracies. Moreover we suggest to pursue in alternative training ways in order to improve the accuracy and robustness of the tracking stages. In addition, a more careful study of the head models is proposed.

6. Acknowledgement

We would like to acknowledge the Spanish Ministry of Economy, Industry and Competitiveness for their support under Contract TIN2014-52897-R.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

[1] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[2] M. Ariz, J. J. Bengoechea, A. Villanueva, and R. Cabeza. A novel 2d/3d database with automatic face annotation for head tracking and pose estimation. *Computer Vision and Image Understanding*, 148:201–210, 2016.

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

[4] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2930–2940, 2013.

[5] F. De la Torre, W.-S. Chu, X. Xiong, F. Vicente, X. Ding, and J. Cohn. Intraface. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–8. IEEE, 2015.

[6] D. F. Dementhon and L. S. Davis. Model-based object pose in 25 lines of code. *International journal of computer vision*, 15(1):123–141, 1995.

[7] Z.-H. Feng, P. Huber, J. Kittler, W. Christmas, and X.-J. Wu. Random cascaded-regression copse for robust facial landmark detection. *IEEE Signal Processing Letters*, 22(1):76–80, 2015.

[8] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[9] P. Huber. 4dface: Real-time 3d face tracking and reconstruction from 2d video. `https://github.com/patrikhuber/4dface`, 2016.

[10] P. Huber, G. Hu, R. Tena, P. Mortazavian, W. P. Koppen, W. Christmas, M. Rätsch, and J. Kittler. A multiresolution 3d morphable face model and fitting framework. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 1–8, 2016.

[11] S. Katz, A. Tal, and R. Basri. Direct visibility of point sets. In *ACM Transactions on Graphics (TOG)*, volume 26, page 24. ACM, 2007.

[12] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. Huang. Interactive facial feature localization. *Computer Vision–ECCV 2012*, pages 679–692, 2012.

[13] U. Lee and J. Tanaka. Finger identification and hand gesture recognition techniques for natural user interface. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction*, pages 274–279. ACM, 2013.

[14] M. Leo, G. Medioni, M. Trivedi, T. Kanade, and G. M. Farinella. Computer vision for assistive technologies. *Computer Vision and Image Understanding*, 154:1–15, 2017.

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[16] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):607–626, 2009.

[17] J. Navallas, M. Ariz, A. Villanueva, J. San Agustín, and R. Cabeza. Optimizing interoperability between video-oculographic and electromyographic systems. *Journal of Rehabilitation Research & Development*, 48(3):253–266, 2011.

[18] P. Paysan. Basel face model website. `http://faces.cs.unibas.ch/bfm/main.php`, 2009.

[19] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *Advanced video and signal based surveillance, 2009. AVSS'09. Sixth IEEE International Conference on*, pages 296–301. IEEE, 2009.

[20] Y. Qiao, X. Xie, T. Sun, and Y. Li. A design of human-computer interaction based on head tracker. In *Computational Intelligence and Industrial Application, 2008. PACIIA'08. Pacific-Asia Workshop on*, volume 2, pages 718–721. IEEE, 2008.

[21] C. Sagonas. Facial point annotations. `https://ibug.doc.ic.ac.uk/resources/facial-point-annotations`, 2013.

[22] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: Database and results. *Image and Vision Computing*, 47:3–18, 2016.

[23] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. 300 faces in-the-wild challenge: The first facial landmark localization challenge. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 397–403, 2013.

[24] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. A semi-automatic methodology for facial landmark annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 896–903, 2013.

[25] R. Subramanian, Y. Yan, J. Staiano, O. Lanz, and N. Sebe. On the relationship between head pose, social attention and personality prediction for unstructured and dynamic group interactions. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 3–10. ACM, 2013.

[26] A. Tawari, S. Martin, and M. M. Trivedi. Continuous head movement estimator for driver assistance: Issues, algorithms, and on-road evaluations. *IEEE Transactions on Intelligent Transportation Systems*, 15(2):818–830, 2014.

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

[28] M. F. Valstar, E. Sánchez-Lozano, J. F. Cohn, L. A. Jeni, J. M. Girard, Z. Zhang, L. Yin, and M. Pantic. Fera 2017-addressing head pose in the third facial expression recognition and analysis challenge. *arXiv preprint arXiv:1702.04174*, 2017.

[29] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[30] K. Wang, L. Xu, Y. Fang, and J. Li. One-against-all frame differences based hand detection for human and mobile interaction. *Neurocomputing*, 120:185–191, 2013.

[31] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 532–539, 2013.

[32] K. Yamaguchi, T. Komuro, and M. Ishikawa. Ptz control with head tracking for video chat. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pages 3919–3924. ACM, 2009.

[33] D. Zhu, T. Gedeon, and K. Taylor. Exploring camera viewpoint control models for a multi-tasking setting in teleoperation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 53–62. ACM, 2011.

[34] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886. IEEE, 2012.