Real-Time Head Pose Estimation by Tracking and Detection of Keypoints and Facial Landmarks

Jilliam M. Díaz Barros[1,2](✉), Bruno Mirbach[3](✉), Frederic Garcia[3](✉),
Kiran Varanasi[1](✉), and Didier Stricker[1,2](✉)

[1] Augmented Vision Department, German Research Center for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany
{Jilliam_Maria.Diaz_Barros,kiran.varanasi,Didier.Stricker}@dfki.de
[2] Computer Science Department, Technische Universität Kaiserslautern, 67663 Kaiserslautern, Germany
[3] IEE S.A., 5326 Contern, Luxembourg
{Bruno.Mirbach,Frederic.Garcia}@iee.lu
http://av.dfki.de/members/barros/

Abstract. We introduce a novel fusion framework for real-time head pose estimation using a tailored Kalman Filter. This approach estimates the pose from intensity images in monocular video data. The method is robust to extreme head rotations and varying illumination, with real-time capability. Our framework incorporates the head pose computed from a keypoint-based tracking scheme into the prediction step of the Kalman Filter and the head pose computed from a facial-landmark-based detection scheme into the correction step. The head pose from the tracking scheme is estimated from 2D keypoints tracked in two consecutive frames in the region of the head and their 3D projection on a simple geometric model. In contrast, the head pose from the detection scheme is estimated from 2D facial landmarks detected in each frame and their 3D correspondences retrieved through triangulation. In each scheme, the head pose results from the minimization of the reprojection error from the 3D-2D correspondences. In each iteration, we update the state transition matrix of the filter and subsequently the estimated covariance. We evaluated our approach on a publicly available dataset and compared with related methods of the state of the art. Our approach could achieve similar performance in terms of mean average error, while operating in real time. Furthermore, we tested our method on our own dataset, to evaluate its performance in the presence of large head rotations. We show good results even in cases where facial landmarks are partially occluded.

Keywords: Head pose estimation · Kalman Filter · Keypoints · Facial landmarks · Tracking · Detection · Real time

D. Bechmann et al. (Eds.): VISIGRAPP 2018, CCIS 997, pp. 326–349, 2019.
https://doi.org/10.1007/978-3-030-26756-8_16

1 Introduction

Head pose estimation (HPE) denotes the task of calculating the orientation and location of a person's head, *i.e.*, its pose, with respect to a given coordinate system. The estimation is generally performed for 6 degrees of freedom ($D.o.F.$), 3 rotation angles and 3 translation parameters. It is used for different purposes, either to increase the robustness of other computer vision tasks, such as face alignment [1], face recognition [2], facial expression recognition [3] or gaze estimation [4,5], or in a wide variety of applications, including human-computer interaction, driver monitoring [6,7] and augmented reality [8].

Our goal is to provide a head pose estimation method that can operate under realistic scenarios, given a set of restrictions: the method should be able to recover the pose for different users, regardless of age, gender or ethnicity, with no need of any additional calibration step; it should estimate the pose even for cases with large head rotations, where part of the face is occluded; the initialization should be performed as soon as the face is detected, without requiring the user to be in a specific initial pose, *i.e.*, facing the camera; and it should be able to estimate the pose in real time, with no power demanding devices like graphic hardwares.

For this task, the input data can be 2D images, such as intensity images, RGB or infrared (IR) data, depth images or a combination of both. Recently, more consumer RGB-D cameras have become available to the general public, causing an increase in the number of HPE methods using them [7,9,10]. We have opted to use intensity images, where the gaze could be extracted in a follow-up project. The advantage of not using additional sensors or depth images is that our method is suitable for applications where only 2D images are available.

We present a HPE approach that integrates two different pipelines operating in parallel: a keypoint-based HPE method, where the pose is computed from 2D tracked keypoints and using a simple geometric model, and a facial-landmark-based HPE method, where the pose is estimated from detected 2D facial landmarks and 3D facial landmarks refined through triangulation. Both pipelines are fused using a tailored Kalman Filter, which combines the strengths of both schemes: the robustness to handle large head pose variations of HPE from keypoints tracking and the precision and ability to recover from the facial-landmark based method. We show that our approach can perform in real time and is able to estimate the pose also for extreme head rotations.

Extending our previous works [11,12], the major contributions in this paper are:

- An updated facial-landmark-based head pose estimation technique, where the 3D facial landmarks are refined over time. The refinement is performed by triangulating 2D facial landmarks extracted from different frames. The 3D points obtained from triangulation are recursively added to a Kalman Filter, where for every new measurement, the observation noise covariance is updated with the covariance matrix computed during the triangulation.

- A re-evaluation of the head pose estimation approach with the updated 3D facial landmarks. We evaluated not only the fusion approach, but also the facial-landmark-based scheme individually. As before, we used the Boston University dataset for uniform illumination and compared to our previous results.
- A verification of the robustness of our approach under varying illumination. To do so, we evaluated our method on the Boston University video sequences under varying illumination and compared to other methods in the state of the art.
- A new dataset for head pose estimation under extreme head rotations, that we will be shared publicly. We included highly accurate groundtruth acquired with an optical tracking system.

2 Related Work

Following the classification proposed in [7,12], HPE methods can be classified in three main categories: model-based, appearance-based and 3D head model registration approaches. It should be noted that some methods might fall in more than one category.

Model-Based Approaches. These approaches are characterized for using rigid or non-rigid face models, facial landmark detection and/or any other prior information regarding the geometry of the head. La Cascia *et al.* in [13] proposed a HPE method based on registration of texture map images with a cylindrical head model. Choi and Kim [14] used templates for HPE, combining a particle filter with an ellipsoidal head model (EHM). Sung *et al.* [15] combined active appearance model (AAM) with a CHM. An and Chung [2] used an EHM to formulate the HPE as a linear system, assuming a rigid body motion under perspective projection. Kumano *et al.* [3] used a face model given by a variable-intensity template with a particle filter, for simultaneous HPE and facial expression recognition. Jang and Kanade [16,17] designed a user-specific CHM-based framework, by combining into a Kalman Filter the estimated motion and a pose retrieved from a dataset of SIFT feature points. In [4,5], Valenti *et al.* used a CHM for simultaneous HPE and eye tracking, based upon a crossed feedback mechanism, which compensated the estimated values and allowed to re-initialize the head pose tracker. Asteriadis *et al.* [18] used a facial-feature tracker with Distance Vector Fields (DVFs) for HPE. In [19], Prasad and Aravind computed the pose using POSIT from the 3D-2D correspondences from a parametrized 3D face mask and SIFT feature points. Diaz *et al.* [20], used random feature points and a CHM to estimate the pose by minimizing the reprojection error of the 3D features and the 2D correspondences. On the other hand, Vicente *et al.* [6] used facial landmarks and a deformable head model, namely parameterized appearance models, to minimize the reprojection error for HPE. Yin and Yang [21] used a pixel intensity binary test for face detection, with pose regression along with local binary feature for face alignment. From a rigid head model, the pose was retrieved by solving the 2D-3D correspondences. Wu *et al.* in [22] presented a pipeline for simultaneous facial landmark detection, HPE and deformation estimation using a cascade iterative procedure augmented with model-based HPE. Similarly, Gou *et al.* [23] proposed a Coupled Cascade Regression (CCR) framework for simultaneous facial landmark detection and HPE. In [11], we presented a first approach to combine the head pose estimated from facial landmarks with the head pose estimated from salient features. 3D points for both type of features were recovered from the intersection on a simple geometric head model. The estimated poses were integrated into a linear Kalman Filter as new measurements in the correction stage. In [12], we introduced a second approach to fuse both estimated poses. In this case, 3D facial landmarks were extracted from a reference head mesh and used through the entire video sequence. Similarly to this work, head pose estimated from keypoints was used to update the state transition matrix at the prediction stage, while the pose computed from facial landmarks was used as a new measurement at the correction stage.

Appearance-Based Approaches. These HPE methods are based on machine learning, using visual features of the face appearance. Even though they are robust to extreme head poses, usually the output corresponds to discrete head poses, thus assigning the pose to specific ranges instead of continuous estimation. These approaches usually have a higher performance for low-resolution face images [24,25]. In [26], Fanelli *et al.* used random regression forests for HPE and facial feature detection, from depth data. Patches from different parts of the face were used to recover the pose through a voting scheme. For the training, it was necessary a large dataset with annotated data. Wang *et al.* presented in [27] a head tracking approach from invariant keypoints. Simulation techniques and normalization were combined to create a learning scheme. Ahn *et al.* [24] introduced a deep-learning-based approach for RGB images, with a particle filter to refine and increase the stability of the estimated pose. In [28], Liu *et al.* used convolutional neural networks, where HPE was formulated as a regression problem. The network was trained using a large synthetic dataset obtained from rendered 3D head models. [24,28] used a GPU to reach real-time capabilities. Tulyakov *et al.* introduced in [29] a person-specific template scheme using a depth camera, which combined template-matching-based tracking with a frame-by-frame decision-tree-based estimator. Borghi *et al.* [7] presented a real time deep-learning-based approach for HPE from depth images, using a regression neural network, POSEidon, which integrated depth with motion features and appearance. In [30], Schwarz presented a deep learning method for HPE which fused IR and depth data with cross-stitch units. Derkach *et al.* [31] proposed a system intended for depth input data, which integrated three different approaches for HPE, two based on landmark detection and one on a dictionary-based method for extreme head poses.

3D Head Model Registration Approaches. These methods register the measured data to reference 3D head models. Meyer *et al.* [32] combined particle swarm optimization and the iterative closest point (ICP) algorithm to register a 3D morphable model (3DMM) to a measured depth face. Yu *et al.* [33] extended this with an online 3D reconstruction of the full head, to handle extreme head rotations. Ghiass *et al.* [34] estimated the pose through a fitting process with a 3D morphable model and RGB-D data. Papazov *et al.* [35] introduced triangular surface patch descriptors for HPE from depth data. The pose was computed from a voting scheme resulting from matching the descriptors to patches from synthetic head models. Jeni *et al.* [36] presented an approach for 3D registration of a dense face mesh from 2D images, through a cascade regression framework trained using a large database of high-resolution 3D face scans. Tan *et al.* [10] used RGB-D data to regress the 3D head pose using random forest in a temporal tracking scheme.

Other methods define HPE as an optimization problem. That is the case of [37], where Morency *et al.* presented a probabilistic scheme, namely Generalized Adaptive View-based Appearance Model (GAVAM), using an EHM. The pose was estimated by solving a linear system with normal flow constraint (NFC). Baltrusaitis *et al.* presented in [38] an extension, which combined head pose tracking with a 3D constrained local model, using both depth data and intensity information. Saragih *et al.* introduced in [39] a HPE approach which fits a deformable model using an optimization strategy through a non-parametric representation of the likelihood maps of landmarks locations. Drouard *et al.* [25] used a Gaussian mixture of locally-linear mapping model to map HOG features extracted on a face region to 3D head poses.

One of the issues of most tracking-based methods is that their robustness to initial HPE when the head is not frontal is not clear [31]. For facial-landmarks-based HPE methods, the accuracy of the head pose relies on the precision of the estimated facial landmarks. Since they strongly depend on the detection of facial landmarks, the misalignment of the landmarks in a frame might lead to erroneous estimations. Hence, these methods might be sensitive to extreme head poses, partial occlusions, facial expressions and low resolution images.

In this work, we introduce a model-based HPE approach based on intensity images. Two independent pipelines are fused on a Kalman Filter for pose estimation, extending the working range to large head rotations. The proposed method does not have any constraint for initialization, as facing the camera for the first frame, and is suitable for real time applications, making it useful for HPE in realistic scenarios.

3 Proposed HPE Pipeline

Several methods of the state of the art rely on facial landmarks for HPE. Even though they might be a reliable source for HPE for frontal and near-frontal faces, facial landmarks are sensitive to extreme head rotations and (self-) occlusions, where important reference regions of the face such as the eyes or nose are partially or totally occluded. In order to tackle this problem, we propose to integrate the head pose computed from a set of keypoints that can be tracked continuously, even when the facial landmarks are not visible. Although a keypoint-based HPE approach could be used alone, it might suffer from drifting in long sequences [16,17,20]. Accordingly, a mechanism to reinforce and correct the head pose from keypoint tracking, using the facial-landmark HPE scheme must be included.

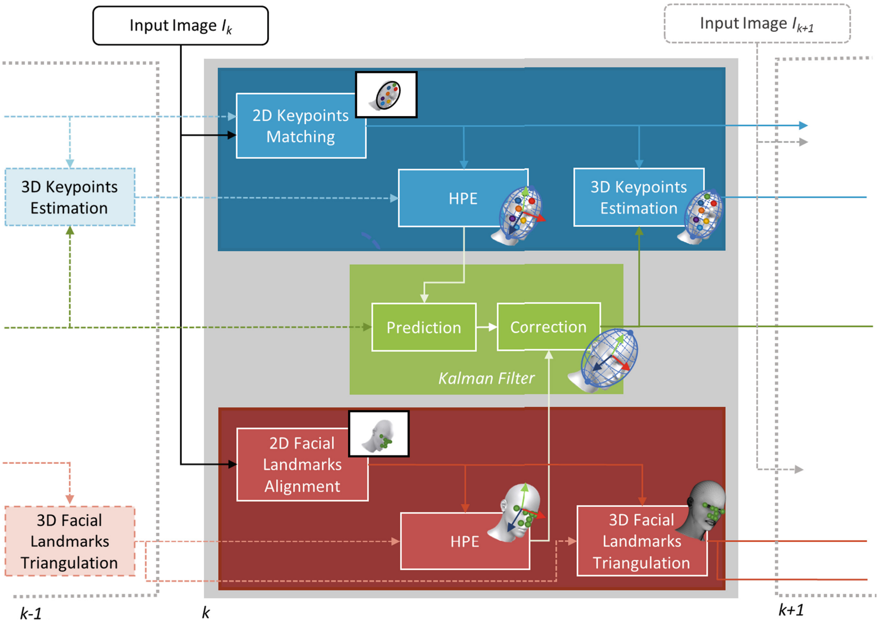


Fig. 1. Proposed HPE pipeline, from Keypoints (blue) and Facial Landmarks (red). (Color figure online)

A diagram of the proposed approach is presented in Fig. 1. On top, inside the blue rectangle, the HPE scheme from keypoints is depicted. At the bottom, inside the red rectangle, is the facial-landmark-based HPE method. The Kalman filter used to fuse both schemes is depicted in the center, inside the green rectangle.

For an input image at time k, I_k, we compute separately the head pose using keypoints and the head pose using facial landmarks. For the keypoint-based method (see Fig. 1), we use a temporal tracking scheme, where the correspondences of 2D keypoints are estimated pairwise using optical flow. Then, these 2D keypoints are projected on a simple geometrical head model, to recover 3D keypoints. From the 2D keypoints at the current input image at time k and the 3D keypoints from the previous frame, $k-1$, we obtain an estimation of the head pose.

For the facial-landmark-based HPE scheme, we align 2D facial landmarks in every input image, independently of the previously alignments (see Fig. 1). 3D facial landmarks are refined by triangulating 2D facial landmarks detected over time. Afterwards, we compute the head pose from the 2D facial landmarks at the current frame and the refined 3D facial landmarks at a fixed pose.

The head poses from both independent frameworks are later integrated into a Kalman filter as follows: HPE from keypoints is included at the prediction stage, while HPE from facial landmarks is used as a new measurement at the correction stage of the filter. As the two different strategies for HPE run in parallel independently of each other, time consumption of the algorithm can be reduced considerably.

The head pose is represented with a transformation, composed of a rotation $\mathbf{R}$ and a translation $\mathbf{t}$. The pose of every 3D point $\mathbf{P}$ in the head is updated following a rigid transformation. $\mathbf{R}$ can also be denoted by the rotation angles $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]$ with respect to the X, Y and Z axes of a known coordinate system. ω_x, ω_y, and ω_z are usually termed as pitch, yaw and roll angles. For our framework, the calibration of the camera is required in advance.

3.1 Facial Landmarks

We refer to facial landmarks as a specific set of feature points in the area of the face. Since the head is modeled as a rigid body, we chose a set of fiducial features that are robust to non-rigid motions, including facial expressions and blinking. This set is composed of 13 features, which encompasses the corners of the eyes, points in the nasal bridge and points around the nostrils, as shown in Fig. 2.

In this document, the set of n 2D facial landmarks is denoted by $\{\mathbf{p}_F\}_{i=1}^{n}$, while the corresponding set of n 3D facial features is denoted by $\{\mathbf{P}_F\}_{i=1}^{n}$.

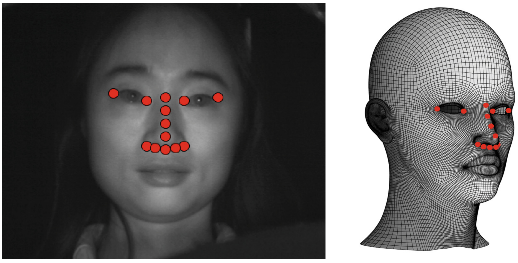


Fig. 2. Facial Landmarks in 2D (left) and 3D (right).

2D Facial Landmarks Detection. For every input image, the set of 2D facial landmarks is detected using the method proposed by Kazemi and Sullivan [40]. This method aligns the facial landmarks by using an ensemble of regression trees, from a sparse subset of intensity values indexed to an initial estimate of the shape. The resulting facial landmarks are depicted in Fig. 2 (left).

It should be noted that the face is not always detected in every frame, thus the facial landmarks cannot be properly aligned to the input image. In those cases, the head pose is only estimated from the keypoints, as detailed in Sect. 3.4.

3D Facial Landmarks Triangulation. Given the set of robust 2D facial landmarks described before, the corresponding 3D facial landmarks are extracted and refined progressively. For the initial frame, we use 3D points that were retrieved offline on a reference head mesh, as shown in Fig. 2 (right). These pre-defined 3D features were manually annotated from an open-source 3D face model [41].

Afterwards, as new 2D facial landmarks are detected along the video sequence, 3D facial landmarks are refined using triangulation. This is possible as the extrinsic camera parameters, *i.e.* the camera pose, is known. Following the linear triangulation method described in [42], from two camera poses $\mathbf{C}$ and $\mathbf{C}'$ and the corresponding sets of 2D facial landmarks, $\{\mathbf{p}_F\}_{i=1}^n$ and $\{\mathbf{p}'_F\}_{i=1}^n$, the relation of each 2D point x and its corresponding 3D point X is defined by $x \propto \mathcal{P}X$, where $\mathcal{P}$ is the 3×4 camera projection matrix. Given two views, we can re-write for each point an equation in the form $AX = 0$, as detailed in [42]. This equation is then solved using the Jacobi's method for finding eigenvalues of symmetric matrices [43,44].

By using the Jacobi's method, we apply singular value decomposition on matrix A, as follows:

$$A = U\Sigma V^T, \tag{1}$$

where U is an orthogonal matrix, Σ is a diagonal positive definite matrix and V is an orthogonal matrix. From this decomposition, we can compute the covariance matrix of A, given by:

$$\mathbf{cov} = V\Sigma^2 V^T. \tag{2}$$

To incorporate new camera poses, and thus refine the 3D facial landmarks, we include a linear Kalman filter where the 3D points are corrected when a new measurement is obtained. This measurement corresponds to the output of the triangulation method from two camera views described before. In each iteration, we use the covariance computed in Eq. (2), as the observation noise covariance of the Kalman filter.

With the previous procedure, we avoid the time-consuming process of manual [36] or semi-automatic facial landmarks annotation [38] on large datasets of 3D face scans, yet providing a robust estimated head pose as long as the facial features are visible in the image.

3.2 Keypoints

We denote the set of 2D keypoints by $\{\mathbf{p}_K\}_{i=1}^m$ and the set of 3D keypoints by $\{\mathbf{P}_K\}_{i=1}^m$. The number of keypoints per frame, m, is not fixed as in the facial landmarks, since it depends on the number of feature points tracked between two frames.

2D Keypoints Extraction. In every frame, we extract 2D keypoints in the area of the head and find their correspondences from the previous frame. These keypoints can be located in the area of the face, but also on the back side or top of the head, for large head rotations.

2D keypoints are detected using the Features from Accelerated Segment Test algorithm [45], also known as FAST. With this method, we are able to detect robust 2D features with low computation time.

Given the set of keypoints extracted from the previous frame, we find the 2D correspondences at the new input image using optical flow. To that end, we use the pyramidal Lucas-Kanade feature tracker detailed in [46].

3D Keypoints Computation. In contrast to 3D facial landmarks estimation, where points are first extracted offline on a 3D head mesh and later refined using triangulation, 3D keypoints are recovered by using a simple geometric model. Similarly to our previous works [11,12], this model corresponds to an ellipsoid as it resembles the shape of the head. Other methods of the state of the art use 3D morphable models for HPE [32,33,38]. These complex head models can be computationally expensive, requiring the use of graphics hardware. We demonstrate in Sect. 4 that the ellipsoid yields good results for the tracking task.

As depicted in Fig. 3, each 3D keypoint $\mathbf{P}_K$ results from the intersection of the projection line $\mathbf{l}$ on the ellipsoidal head model (EHM). This line passes through the optical center of the camera $\mathbf{C}$ and the corresponding 2D keypoint $\mathbf{p}_K$ in the image plane $\mathbf{I_0}$, The orientation, position and dimension of the ellipsoid on the 3D space are known in advance, as explained in Sect. 3.3.

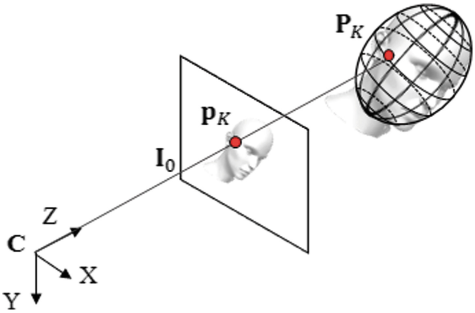


Fig. 3. Computation of 3D keypoints. Source: [12].

The equation of the projection line is given by $\mathbf{l} = \mathbf{C} + \lambda \mathbf{d}$, where $\mathbf{d}$ represents a line parallel to $\mathbf{l}$ and λ is a scalar retrieved from the quadratic equation of the ellipsoid [11,12] as follows:

$$|\mathbf{a}|^2 \ \lambda^2 + 2\left(\mathbf{a} \cdot \mathbf{b}\right)\lambda + |\mathbf{b}|^2 - 1 = 0 \tag{3}$$

Given an ellipsoid with radii $\{\frac{1}{r_x}, \frac{1}{r_y}, \frac{1}{r_z}\}$, rotation matrix $\mathbf{R}$ and having its center at $\mathbf{E_0}$, $\mathbf{a}$ and $\mathbf{b}$ are defined as $\mathbf{a} = \mathbf{G}\mathbf{R}^T\mathbf{d}$ and $\mathbf{b} = \mathbf{G}\mathbf{R}^T(\mathbf{C} - \mathbf{E_0})$, with $\mathbf{G}$ being a 3×3 diagonal matrix of the inverses of the ellipsoid radii.

Area for 2D Keypoint Detection. For every new input image, 2D keypoints are extracted exclusively from a defined area of the head. As mentioned earlier, this area might not only be on the face, but also on top or on the back side of the head. Some methods propose to extract salient features from the area given by a face detection algorithm [47]. Besides being time consuming, this approach would fail when the face is not detected, *i.e.*, for large head rotations.

We propose to continuously update the area for feature detection, by projecting the 3D ellipsoidal head model on the image plane in every frame, as shown in

Fig. 4. To that end, we first estimate the plane π parallel to the horizontal axis of the image plane and to the vertical axis of the ellipsoid, and which divides it in two parts. We then find the elliptical surface that results from the intersection of π and the ellipsoid. Finally, this surface is projected in the image, assuming a perspective camera model.

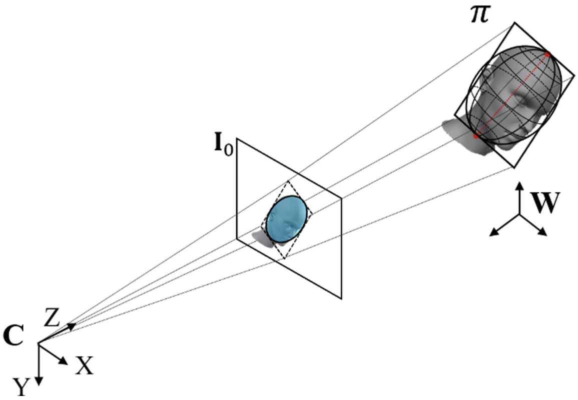


Fig. 4. Update area for keypoint detection. Source: [11].

3.3 Initialization

Similarly to [12], we adjust the dimension of the ellipsoid according to the size of the user's head. The 2D facial landmarks detected in the first frame are used for this step. Since the calibration of the camera is known, we can use the relation between the interpupillary distance of the eyes in pixels, δ_{px}, extracted from the input image and an approximate distance between a person's eyes in cm, δ_{cm} for the initialization. Measurements reported in [48,49] found that the averaged interpupillary distance for men is around 6.47 cm, while for women is 6.23 cm. We assumed this distance to be of 6 cm in our experiments.

As shown in Fig. 5, we define the 2D bounding box of the detected head by points $\{\mathbf{p}_{TL}, \mathbf{p}_{TR}, \mathbf{p}_{BL}, \mathbf{p}_{BR}\}$. Additionally, the corresponding 3D bounding box is defined by points $\{\mathbf{P}_{TL}, \mathbf{P}_{TR}, \mathbf{P}_{BL}, \mathbf{P}_{BR}\}$. The radii of the ellipsoid on the X and Z axes, r_x and r_z, are set equal to half of the width of the 3D bounding box, *i.e.*, $\frac{1}{2}|\mathbf{P}_{TL} - \mathbf{P}_{TR}|$ and are computed directly from Eq. (4) [12]. On the other hand, the radius r_y of the ellipsoid is given by half of the height of the 3D bounding box, *i.e.*, $\frac{1}{2}|\mathbf{P}_{TL} - \mathbf{P}_{BL}|$ and is calculated from Eq. (5) [12].

$$r_x = r_z = \frac{1}{2}|\mathbf{p}_{TR} - \mathbf{p}_{TL}| \cdot \frac{\delta_{cm}}{\delta_{px}} \tag{4}$$

$$r_y = \frac{1}{2}|\mathbf{p}_{TL} - \mathbf{p}_{BL}| \cdot \frac{\delta_{cm}}{\delta_{px}} \tag{5}$$

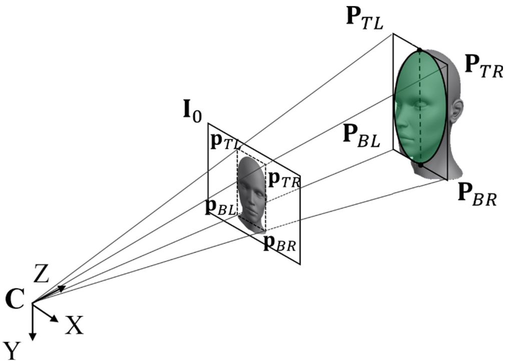


Fig. 5. Initialization of the ellipsoid. Source: [11].

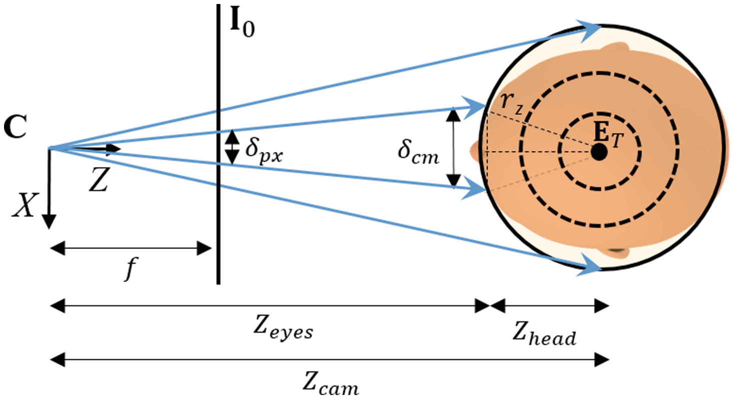


Fig. 6. Estimation of the ellipsoid's depth. Source: [12].

The estimation of the initial depth of the ellipsoid with respect to the camera's optical center $\mathbf{C}$ is depicted in Fig. 6. Z_{cam}, the distance between $\mathbf{C}$ and $\mathbf{E}_T$, the center of the ellipsoid, is given by $Z_{cam} = Z_{eyes} + Z_{head}$. Z_{eyes} is the distance from $\mathbf{C}$ to the eyes' baseline and is computed from Eq. (6) [11,12], where f is the focal length of the camera. Z_{head} corresponds to the distance from the eyes' baseline to $\mathbf{E}_T$ and is given by Eq. (7) [11,12].

$$Z_{eyes} = f \cdot \frac{\delta_{cm}}{\delta_{px}} \tag{6}$$

$$Z_{head} = \sqrt{{r_z}^2 - (\delta_{cm}/2)^2} \tag{7}$$

3.4 Head Pose Estimation

The head pose estimated from each independent pipeline is computed by minimizing the reprojection error between the 3D features points $\{\mathbf{P}\}_{i=1}^{\eta}$ on the image plane and the 2D correspondences on the image $\{\mathbf{p}\}_{i=1}^{\eta}$ at time k. For the keypoint-based scheme, the 3D and 2D features correspond to $\{\mathbf{P}_K, \mathbf{p}_K\}$ respectively, $\eta = m$ and $\{\mathbf{P}_K\}_{i=1}^{m}$ is given at time $k-1$. For the facial-landmark-based scheme, the 3D and 2D features correspond to $\{\mathbf{P}_F, \mathbf{p}_F\}$ respectively, $\eta = n$ and $\{\mathbf{P}_F\}_{i=1}^{n}$ is given at the initial frame. The minimization is expressed by Eq. (8) [12], where $\pi(\mathbf{P}) : \mathbb{R}^3 \mapsto \mathbb{R}^2$ denotes the perspective projection operator and i the index of the i-th feature point. Equation (8) [12] is minimized in the least squared sense with respect to the rotation $\mathbf{R}$ and translation $\mathbf{t}$, using Levenberg-Marquardt iteration.

$$\arg\min \sum_{i=1}^{\eta} \|\pi(\mathbf{R}\mathbf{P}_i + \mathbf{t}) - \mathbf{p}_i\|_2^2 \tag{8}$$

Before introducing the fusion scheme to combine both estimated head, it is important to understand the difference between both estimates.

For the keypoint-based scheme (Sect. 3.2), we calculate the head pose resulting from tracking keypoints in two consecutive frames. This implies that we compute a transformation from the frame at time $k-1$ to the frame at time k. This frame-to-frame transformation is referred to as a local transformation and is denoted by $\mathbf{R}_{k-1}^{k}$ and $\mathbf{t}_{k-1}^{k}$.

Head pose estimated from the facial-landmark-based scheme (Sect. 3.1) is computed with respect to 3D facial landmarks fixed to an initial head pose, $\mathbf{R}_0$ and $\mathbf{t}_0$. Although the 3D facial landmarks are refined over time, the pose is not updated. Therefore, the pose retrieved from this scheme maps the head pose from the first given frame at time k_0, to a pose at time k. This transformation is referred to a global transformation and is denoted by $\mathbf{R}_0^k$ and $\mathbf{t}_0^k$.

Given a local and a global transformation, we need to formulate our Kalman Filter in a way that we can integrate both head poses. We define the state vector $\mathbf{x}$ of the filter to be composed of the rotation and translation from the first given frame, *i.e.*, the global head pose. The head rotation is represented using a quaternion $\mathbf{q} = [q_x, q_y, q_z, q_w]^T$, where q_w is the scalar part and $\{q_x, q_y, q_z\}$ the vector part. The translation is denoted in homogeneous coordinates as $\tilde{\mathbf{t}} = [t_x, t_y, t_z, \mathit{1}]^T$. Consequently, the state vector is given by $\mathbf{x} = \left[\mathbf{q}^T, \tilde{\mathbf{t}}^T\right]^T$, with a dimension of 8×1.

Initial HPE. For the first given frame, the head pose $\mathbf{R}_0$ and $\mathbf{t}_0$ is computed from facial landmarks only, as 3D keypoints are not available. The pose is recovered by minimizing Eq. (8) with the 3D facial landmarks $\{\mathbf{P}_F\}_{i=1}^{n}$ extracted offline from the reference head model and the 2D facial landmarks $\{\mathbf{p}_F\}_{i=1}^{n}$ aligned at the first frame. We initialize the Kalman Filter using the computed head pose.

HPE for the Other Frames. We define a linear Kalman Filter to fuse the pose estimated from keypoints and the pose estimated from facial landmarks. This is possible, as a linear process model can be built by representing the rotation with quaternions and the translation with homogeneous coordinates. The predicted or *a priori* state estimate of the filter $\hat{\mathbf{x}}_k^-$ at time k is calculated using Eq. (9), where $\mathbf{A}$ represents the state transition matrix of the process model, with a normal distributed process noise with covariance $\mathbf{Q}$.

$$\hat{\mathbf{x}}_k^- = \mathbf{A}_k \hat{\mathbf{x}}_{k-1} \tag{9}$$

In order to integrate the keypoint-based HPE into the prediction step of the filter, we update $\mathbf{A}$ in every iteration with the pose estimated at the tracking scheme, as shown in Eq. (10) [12]. The resulting $\mathbf{A}$ is a matrix of size 8×8.

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_\rho & 0 \\ 0 & \mathbf{A}_\mathbf{t} \end{bmatrix} \tag{10}$$

$\mathbf{A}_\rho$ corresponds to the state transition sub-matrix to project the rotation ahead and is computed from the local rotation $\mathbf{R}_{k-1}^k$. We convert this rotation matrix to a quaternion $\boldsymbol{\rho} = [\rho_x, \rho_y, \rho_z, \rho_w]^T$, and calculate $\mathbf{A}_\rho$ as follows [12]:

$$\mathbf{A}_\rho = \begin{bmatrix} \rho_w & -\rho_z & \rho_y & \rho_x \\ \rho_z & \rho_w & -\rho_x & \rho_y \\ -\rho_y & \rho_x & \rho_w & \rho_z \\ -\rho_x & -\rho_y & -\rho_z & \rho_w \end{bmatrix} \tag{11}$$

$\mathbf{A}_\mathbf{t}$ represents the state transition sub-matrix to update the translation, and is defined by (12) [12]. The new translation estimate $\mathbf{t}_0^{k-}$ is given by (13) [12].

$$\mathbf{A}_\mathbf{t} = \begin{bmatrix} \mathbf{R}_{k-1}^k & \mathbf{t}_{k-1}^k \\ \\ 0 & 1 \end{bmatrix}, \tag{12}$$

$$\mathbf{t}_0^{k-} = \mathbf{R}_{k-1}^k \mathbf{t}_0^k + \mathbf{t}_{k-1}^k. \tag{13}$$

The covariance P^- at the prediction step is computed from Eq. (14).

$$\mathsf{P}_k^- = \mathbf{A}_k \mathsf{P}_{k-1} \mathbf{A}_k^T + \mathbf{Q} \tag{14}$$

The measurement model of the Kalman Filter is given by Eq. (15). The new measurement $\mathbf{z}_k$ at time k corresponds to the head pose retrieved from the facial-landmark-based scheme, *i.e.*, the global head pose. $\mathbf{H}$ is a 7×8 matrix that relates the current state $\mathbf{x}_k$ to the measurement and is given by $\mathbf{H} = [\mathbf{I}_7 \; 0]$, where $\mathbf{I}_7$ is a 7×7 identity matrix. $\mathbf{v}_k$ denotes the measurement noise in the observation with covariance $\mathbf{R}$.

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \tag{15}$$

The updated or *a posteriori* state estimate $\hat{\mathbf{x}}_k$ is calculated from Eq. (16), where $\mathbf{K}_k$ represents the Kalman gain.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \tag{16}$$

The covariance is updated at the correction step using Eq. (17).

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \tag{17}$$

Occlusion Handling and Pose Recovery. One challenge in HPE is to compute the head pose when the face is occluded due to large head rotations or when it is not detected at all. The facial-landmark-based scheme fails to provide an estimation in these cases, so our fusion approach uses only the head pose computed from the keypoint-based scheme. This implies that the head pose calculated at the Kalman Filter is given only from the prediction step, as no new measurement is available. Thereby, our approach is able to provide an estimated pose even when the face has not been detected.

When the face is detected again, the output of the facial-landmark-based scheme is integrated again at the Kalman Filter to correct the estimated pose. This step is fundamental in our approach, especially if the face has not been detected for several consecutive frames. If the pose has been computed only from keypoints for a long sequence, the estimation might suffer from drifting, while increasing the state covariance (uncertainty) of the filter (from Eq. (14)) over time. When a new measurement is available, the recovery takes place rapidly, since the weight of the predicted state covariance is relatively small.

In contrast to our previous method presented in [12], we do not update in each iteration the covariance of the process noise and measurement noise, but set it fixed for all the sequences.

4 Experiments and Results

We have evaluated the proposed approach using a publicly available database for HPE and compared to other methods of the state of the art. Additionally, we have evaluated the performance of our approach under extreme head poses using our own dataset. We have also analyzed the contribution of each HPE scheme in our approach, by assessing them individually. We implemented both pipelines in C++ and tested the algorithms in an Intel® Xeon(R) W3520 processor with 8 Gb of RAM.

4.1 Comparison with Other HPE Methods

We evaluated our method using the Boston University (BU) head tracking database presented in [13]. This database is composed of short video sequences with subjects performing several head movements inside an office. The database is divided in two sets of videos, one recorded under uniform illumination and the other under varying illumination. The first set contains 45 video sequences from 5 different subjects, while the second set has 27 video sequences from 3 subjects. Ground truth was acquired using the Flock of Birds magnetic tracker attached to the head, reporting nominal accuracies of 0.5° in rotation and 1.8 mm in translation.

We used three metrics for comparison with other relevant HPE methods of the state of the art: root mean square error (RMSE), mean absolute error (MAE) and standard deviation (STD). These three estimation errors were computed from Eqs. (18), (19) and (20) respectively, where n represents the number of frames, s_i the ground truth $\mathbf{s}$ at time i and $\hat{s}_i$ the estimate of the position or angle $\hat{\mathbf{s}}$, at time i.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(s_i - \hat{s}_i)^2} \tag{18}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|s_i - \hat{s}_i| \tag{19}$$

$$STD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}|(s_i - \hat{s}_i) - \mu|^2} \tag{20}$$

μ corresponds to the mean of $(\mathbf{s} - \hat{\mathbf{s}})$ and is given by, (21).

$$\mu = \frac{1}{n}\sum_{i=1}^{n}(s_i - \hat{s}_i) \tag{21}$$

Tables 1 and 2 present the results of our approach and related works on the BU database, with uniform and varying illumination, respectively. We have also included the average of the rotation MAE for each method.

The last three rows of each table present the angular accuracies from the individual HPE approaches, using only keypoints (K.P.) or facial landmarks (F.L.) and HPE from fusion. It should be noted that the latter performs better for every angle estimation, yaw, pitch and roll, than each individual method. Only for the varying illumination set, Table 2, the performance of the facial-landmark-based method is comparable to the fusion approach.

Regarding other methods of the state of the art, we can observe that the proposed approach has an outstanding performance. For the uniform illumination set, Table 1, only [36] presents a similar average error with a higher estimation rate. Other methods as [27,38] present lower MAE, but with much lower estimation rate. On the other hand, on the varying illumination set, Table 2, our fusion approach presents the best performance, with the lowest average of the MAE and real-time capability.

For the BU database, no calibration data is provided. For that reason, most of related work do not report their translation estimation errors. We have included our results in Tables 3 and 4, for the uniform and varying illumination sets, respectively. It can be noted that in average, the fusion scheme presents the lowest errors for both sets, with a similar performance of the keypoint-based HPE scheme. This can be explained as this pipeline uses frame-by-frame tracking, so the 2D estimation of the features is robust. In contrast, in the facial-landmark-based scheme the feature detection is not always precise and there are small displacements in the alignments between frames even when the person is not moving.

Table 1. Comparison with other methods of the state of the art on BU dataset with uniform illumination set.

Method	Year	RMSE ± STD			MAE				Time (FPS)
		Roll	Pitch	Yaw	Roll	Pitch	Yaw	Average	
La Cascia *et al.* [13]	2000	-	-	-	9.8	6.1	**3.3**	6.4	-
Sung *et al.* [15]	2008	-	-	-	3.1	5.6	5.4	4.7	-
Morency *et al.* [37]	2008	-	-	-	2.91	3.67	4.97	3.85	6
Jang and Kanade [16]	2008	-	-	-	2.1	3.7	4.6	3.46	-
An and Chung [2]	2008	-	-	-	2.83	3.95	3.94	3.57	-
Choi and Kim [14]	2008	-	-	-	2.82	3.92	4.04	3.59	14
Kumano *et al.* [3]	2009	-	-	-	2.9	4.2	7.1	4.73	-
Lefevre and Odobez [50]	2009	-	-	-	2.0	3.3	4.4	3.23	3
Asteriadis *et al.* [18]	2010	3.56	4.89	5.72	-	-	-	-	-
Prasad and Aravind [19]	2010	-	-	-	3.6	**2.5**	3.8	3.3	-
Jang and Kanade [17]	2010	-	-	-	2.07	3.44	4.22	3.24	-
Saragih *et al.* [39]	2011	-	-	-	2.55	4.46	5.23	4.08	8
Valenti *et al.* [4]	2012	3.00 ± 2.82	5.26 ± 4.67	6.10 ± 5.79	-	-	-	-	-
Wang *et al.* [27]	2012	-	-	-	**1.86**	2.69	3.75	2.76	15
Baltrusaitis *et al.* [38]	2012	-	-	-	2.08	3.81	**3.00**	2.96	-
Tran *et al.* [51]	2013	-	-	-	2.4	3.9	5.4	3.90	5
Vicente *et al.* [6]	2015	-	-	-	3.2	6.2	4.3	4.56	25
Jeni *et al.* [36]	2017	-	-	-	2.41	**2.66**	3.93	**3.0**	**50**
Wu *et al.* [22]	2017	-	-	-	3.1	5.3	4.9	4.43	-
Diaz Barros *et al.* [20]	2017	3.36 ± 2.98	4.46 ± 3.84	5.09 ± 4.56	2.56	3.39	3.99	3.31	56
Gou *et al.* [23]	2017	-	-	-	3.3	4.8	5.1	4.4	-
Diaz Barros *et al.* [11]	2018	3.36 ± 2.99	4.32 ± 3.62	5.25 ± 4.70	2.54	3.27	4.07	3.29	23
Diaz Barros *et al.* [12]	2018	3.06 ± 2.78	4.38 ± 3.76	4.93 ± 4.56	2.32	3.41	3.90	3.21	40
HPE from K.P.	2018	3.42 ± 3.05	4.53 ± 3.78	5.46 ± 4.81	2.61	3.45	4.27	3.44	-
HPE from F.L.	2018	2.46 ± 2.31	4.43 ± 3.61	5.23 ± 4.85	**1.92**	3.49	4.17	3.19	-
HPE from fusion	2018	**2.41 ± 2.20**	**4.11 ± 3.26**	**4.92 ± 4.40**	**1.91**	3.26	3.92	**3.03**	**40**

Table 2. Comparison with other methods of the state of the art on BU dataset with uniform illumination set.

Method	Year	RMSE ± STD			MAE				Time (FPS)
		Roll	Pitch	Yaw	Roll	Pitch	Yaw	Average	
Lefevre and Odobez [50]	2009	-	-	-	2.3	3.5	**4.1**	3.3	3
Jang and Kanade [17]	2010	-	-	-	2.8	4.25	5.92	4.32	-
Jeni *et al.* [36]	2017	-	-	-	2.24	**2.72**	4.87	3.27	50
HPE from K.P.	2018	3.44 ± 2.77	5.31 ± 4.38	6.67 ± 5.56	2.73	4.20	5.21	4.04	-
HPE from F.L.	2018	2.70 ± 2.17	4.14 ± 3.59	5.21 ± 4.51	**2.18**	3.22	**4.12**	3.17	-
HPE from fusion	2018	**2.66 ± 2.08**	**4.11 ± 3.42**	**5.15 ± 4.05**	**2.16**	3.23	**4.12**	**3.17**	40

Time Consumption Analysis. To estimate the time consumption of our approach, we have evaluated the runtime for each step on the BU dataset (see Table 5). A comparison with other methods of the state of the art is presented in the last column of Tables 1 and 2.

Table 3. Errors on the translation estimation on BU dataset with uniform illumination set.

Method	RMSE ± STD			MAE			
	Trans. X	Trans. Y	Trans. Z	Trans. X	Trans. Y	Trans. Z	Average
HPE from K.P.	4.09 ± 3.76	**2.10** ± **1.68**	1.58 ± 1.23	3.27	**1.64**	1.22	2.04
HPE from F.L.	4.30 ± 4.07	2.49 ± 2.05	1.45 ± 1.19	3.48	1.99	1.13	2.2
HPE from fusion	**3.89** ± **3.65**	2.31 ± 1.91	**1.40** ± **1.07**	**3.15**	1.86	**1.13**	2.04

Table 4. Errors on the translation estimation on BU dataset with varying illumination set.

Method	RMSE ± STD			MAE			
	Trans. X	Trans. Y	Trans. Z	Trans. X	Trans. Y	Trans. Z	Average
HPE from K.P.	3.17 ± **2.63**	**2.41** ± **1.82**	1.53 ± 1.09	2.58	**1.94**	1.24	1.92
HPE from F.L.	3.69 ± 3.26	2.66 ± 2.12	1.11 ± 0.89	3.00	2.18	0.91	2.03
HPE from fusion	**3.10** ± 2.85	2.76 ± 2.16	**0.96** ± **0.75**	**2.54**	2.29	**0.77**	1.87

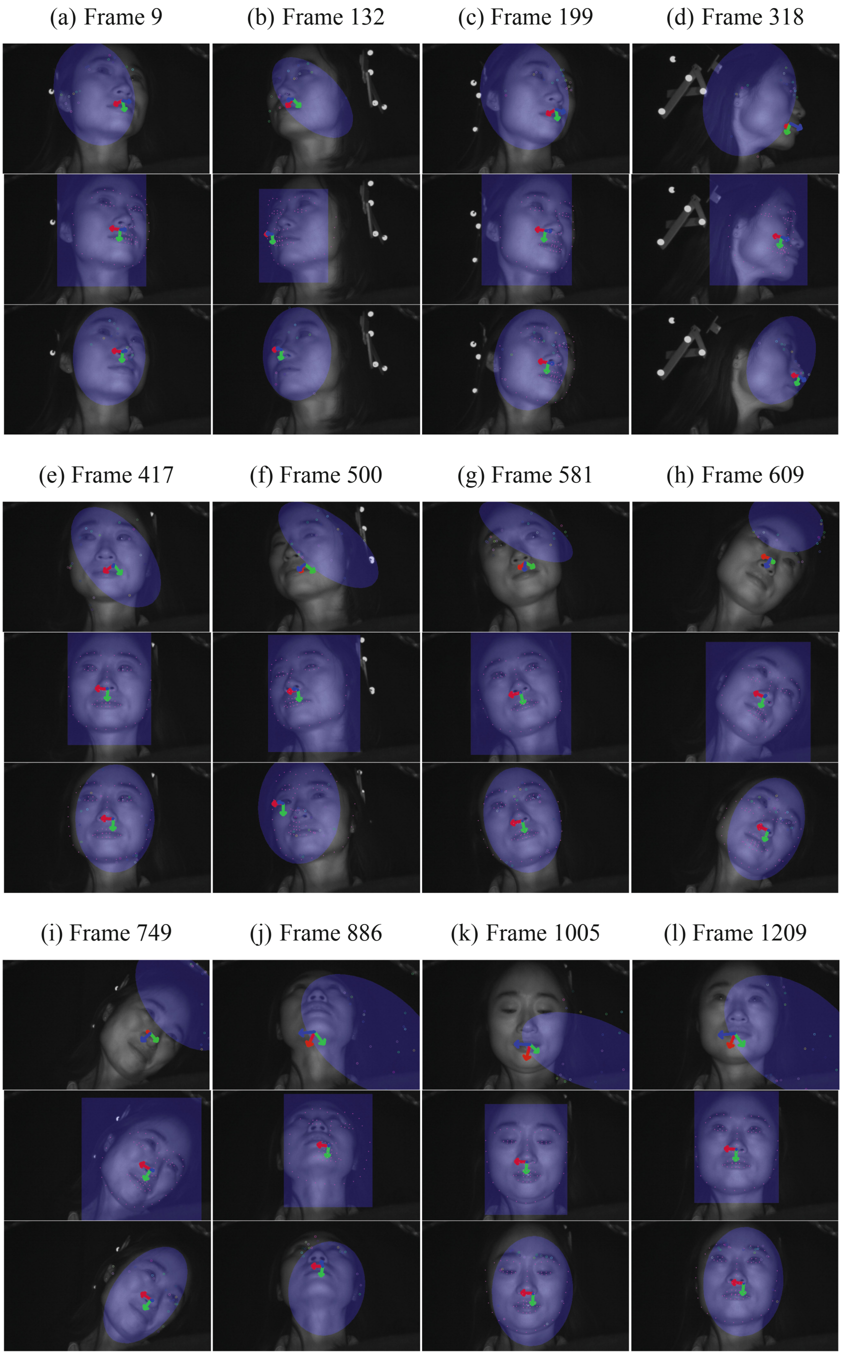
Table 5. Errors on the translation estimation on BU dataset with uniform illumination set.

Process	Time (ms)
Initial face detection	22.34
Initial head pose estimation	2.01
Total	24.35
2D feature detection and matching	21.27
Estimation of 3D keypoints	0.3
HPE and refinement of 3D facial landmarks	3.12
Total	24.68

As can be noted in Table 5, the HPE in our method for both the initialization step and the other frames took around 40 FPS. If we compare to previous works, only [36] and our previous approaches [12,20] could reach >40FPS, with estimation errors similar to the proposed approach. In contrast to [36], we did not need manual annotation on a large dataset of high-resolution 3D face scans.

4.2 Experiments with Our Own Dataset

We have also evaluated the HPE fusion approach and both independent schemes on our own video sequence. We made this video publicly available for research purposes at [52], and included groundtruth with the respective calibration file. The video contains 1263 images of a person sitting, while moving her head with large rotations. In some frames, the head is partially self-occluded, due to rotations larger that 45° in the X or Y axes (pitch and yaw). The most challenging estimations are the yaw and pitch rotation, where facial landmark detectors usually present higher error.



(a) Frame 9 (b) Frame 132 (c) Frame 199 (d) Frame 318

(e) Frame 417 (f) Frame 500 (g) Frame 581 (h) Frame 609

(i) Frame 749 (j) Frame 886 (k) Frame 1005 (l) Frame 1209

Fig. 7. HPE from the provided sequences with keypoints (top), facial landmarks (center) and the fusion scheme (bottom) for each frame.

Table 6. Estimated pitch angle from HPE methods.

Method	RMSE ± STD			MAE			
	Roll	Pitch	Yaw	Roll	Pitch	Yaw	Average
HPE from K.P.	57.63 ± 30.20	55.04 ± 54.02	32.78 ± 25.15	52.94	46.02	25.70	41.55
HPE from F.L.	16.34 ± 16.16	16.66 ± 16.61	9.82 ± 9.73	9.72	12.61	5.66	9.33
HPE from fusion	14.87 ± 14.69	16.85 ± 16.81	9.69 ± 9.42	8.24	12.09	5.65	8.66

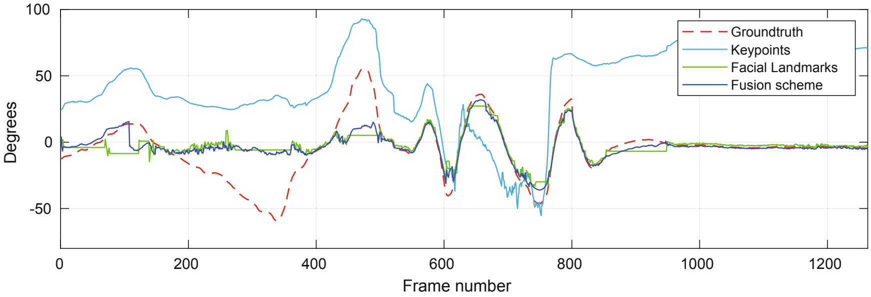


Fig. 8. Estimated roll angle from the three HPE methods.

Figure 7 presents the results for different frames from the three HPE schemes. For each frame, the results are shown as follows: on top is the keypoint-based method; the facial-landmark-based scheme is on the middle; and the fusion approach is at the bottom. The blue area around the face depicts the projection of the ellipsoid onto the 2D image, while the coordinate system representing the estimated pose is displayed with the RGB arrows. For the facial-landmark-based scheme, we define a plane parallel to the image plane with the dimensions of the head and we project it on the 2D image.

We also estimated the root mean square error (RMSE), mean absolute error (MAE) and standard deviation (STD) for the three rotation angles. Results are presented in Table 6. Similar to the Boston University dataset, the fusion scheme presents the lowest errors. The high errors in the keypoint-based scheme is due to the fact that the estimated head pose started drifting and it was not able to recover (see Fig. 7 from frame 417 (e) onwards).

The estimated angles in degrees for the three schemes are presented in Figs. 8, 9 and 10. For some frames, the pose from the facial-landmark-based method could not be updated from the last estimate, since the face was not detect. This can also be noted in Fig. 7 in frames 9 (a), 132 (b), 318 (d), 749 (i) and 886 (j). However, as soon as the face is correctly detected again, the facial-landmark-based approach is able to recover. On the other hand, the head pose from the keypoint-based method is continuous, but suffers from drifting in long sequences. The fusion scheme is able to estimate the pose through the entire sequence, even with large head rotations (frames 318 (d) and 886 (j)) and recover in case it starts drifting, exploiting the advantages from both pipelines.

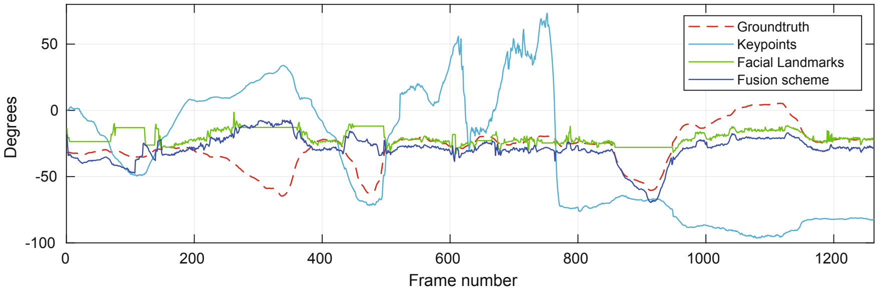


Fig. 9. Estimated pitch angle from the three HPE methods.

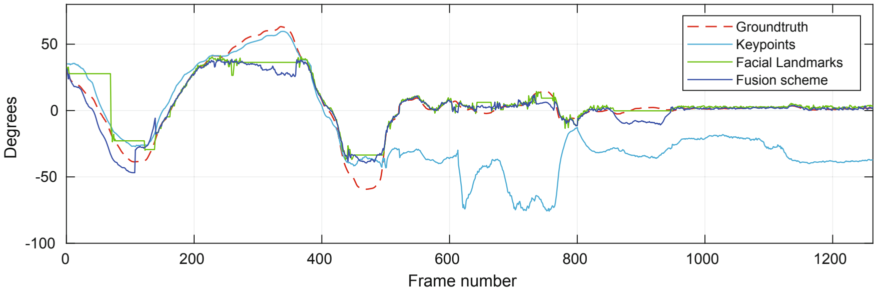


Fig. 10. Estimated yaw angle from the three HPE methods.

5 Conclusions

We have presented a method for head pose estimation in real time from monocular video data. The proposed approach integrates into a tailored Kalman Filter the head poses estimated from two different pipelines, one based on tracked keypoints and the other based on detected facial landmarks. Its particular strength is that it combines the advantages of both HPE methods. On the one hand, it benefits from the robustness of detecting keypoints, which is nearly always possible, regardless of the head pose and (limited) occlusions. On the other hand, it benefits from the absolute (i.e. not relative to a previous head pose) determination of the head pose based on facial landmarks, which is unaffected by drift or similar effects. Another great advantage is that the head pose from keypoints and the head pose from facial landmarks can be determined independently of each other, *i.e.*, the corresponding calculations can be performed in an arbitrary sequence, *e.g.* in parallel. These greatly help to provide a real-time head pose estimation, which is robust to large head rotations.

We have evaluated and compared our approach to other methods of the state of the art obtaining similar results, with an average runtime of 40FPS. We also demonstrated that our approach provides a reliable estimation even for extreme head poses and under varying light conditions.

For future work, we are interested in investigating a method to refine the 3D keypoints, in order to have a more robust estimation from the keypoint-based HPE scheme.

References

1. Xu, X., Kakadiaris, I.A.: Joint head pose estimation and face alignment framework using global and local CNN features. In: 12th International Conference on Automatic Face & Gesture Recognition (FG 2017), vol. 2, pp. 642–649. IEEE, May 2017
2. An, K.H., Chung, M.J.: 3D head tracking and pose-robust 2D texture map-based face recognition using a simple ellipsoid model. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 307–312. IEEE (2008)
3. Kumano, S., Otsuka, K., Yamato, J., Maeda, E., Sato, Y.: Pose-invariant facial expression recognition using variable-intensity templates. Int. J. Comput. Vis. **83**(2), 178–194 (2009). https://doi.org/10.1007/s11263-008-0185-x
4. Valenti, R., Sebe, N., Gevers, T.: Combining head pose and eye location information for gaze estimation. Trans. Image Process. **21**(2), 802–815 (2012)
5. Valenti, R., Yucel, Z., Gevers, T.: Robustifying eye center localization by head pose cues. In: International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 612–618. IEEE (2009)
6. Vicente, F., Huang, Z., Xiong, X., De la Torre, F., Zhang, W., Levi, D.: Driver gaze tracking and eyes off the road detection system. Trans. Intell. Transp. Syst. **16**(4), 2014–2027 (2015)
7. Borghi, G., Venturelli, M., Vezzani, R., Cucchiara, R.: Poseidon: face-from-depth for driver pose estimation. In: International Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2017)
8. Mohr, P., Tatzgern, M., Grubert, J., Schmalstieg, D., Kalkofen, D.: Adaptive user perspective rendering for handheld augmented reality. In: Symposium on 3D User Interfaces (3DUI), pp. 176–181. IEEE (2017)
9. Fanelli, G., Dantone, M., Gall, J., Fossati, A., Van Gool, L.: Random forests for real time 3D face analysis. Int. J. Comput. Vis. **101**(3), 437–458 (2013)
10. Tan, D.J., Tombari, F., Navab, N.: Real-time accurate 3D head tracking and pose estimation with consumer RGB-D cameras. Int. J. Comput. Vis. **126**, 1–26 (2017)
11. Diaz Barros, J.M., Garcia, F., Mirbach, B., Varanasi, K., Stricker, D.: Combined framework for real-time head pose estimation using facial landmark detection and salient feature tracking. In: Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), vol. 5, pp. 123–133. INSTICC, SciTePress (2018)
12. Diaz Barros, J.M., Mirbach, B., Garcia, F., Varanasi, K., Stricker, D.: Fusion of keypoint tracking and facial landmark detection for real-time head pose estimation. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 2028–2037. IEEE, March 2018
13. La Cascia, M., Sclaroff, S., Athitsos, V.: Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3D models. Trans. Pattern Anal. Mach. Intell. **22**(4), 322–336 (2000)
14. Choi, S., Kim, D.: Robust head tracking using 3D ellipsoidal head model in particle filter. Pattern Recogn. **41**(9), 2901–2915 (2008)

15. Sung, J., Kanade, T., Kim, D.: Pose robust face tracking by combining active appearance models and cylinder head models. Int. J. Comput. Vision **80**(2), 260–274 (2008)
16. Jang, J.S., Kanade, T.: Robust 3D head tracking by online feature registration. In: 8th International Conference on Automatic Face & Gesture Recognition (FG 2008). IEEE (2008)
17. Jang, J.S., Kanade, T.: Robust 3D head tracking by view-based feature point registration. People Image Analysis (PIA) Consortium, Carnegie Mellon University, Technical report (2010)
18. Asteriadis, S., Karpouzis, K., Kollias, S.: Head pose estimation with one camera, in uncalibrated environments. In: Workshop on Eye Gaze in Intelligent Human Machine Interaction, pp. 55–62. ACM (2010)
19. Prasad, B.H., Aravind, R.: A robust head pose estimation system for uncalibrated monocular videos. In: 7th Indian Conference on Computer Vision, Graphics and Image Processing, pp. 162–169. ACM (2010)
20. Diaz Barros, J.M., Garcia, F., Mirbach, B., Stricker, D.: Real-time monocular 6-DoF head pose estimation from salient 2D points. In: International Conference on Image Processing (ICIP), pp. 121–125. IEEE, September 2017
21. Yin, C., Yang, X.: Real-time head pose estimation for driver assistance system using low-cost on-board computer. In: 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry, vol. 1, pp. 43–46. ACM (2016)
22. Wu, Y., Gou, C., Ji, Q.: Simultaneous facial landmark detection, pose and deformation estimation under facial occlusion (2017)
23. Gou, C., Wu, Y., Wang, F.Y., Ji, Q.: Coupled cascade regression for simultaneous facial landmark detection and head pose estimation. In: International Conference on Image Processing (ICIP). IEEE (2017)
24. Ahn, B., Park, J., Kweon, I.S.: Real-time head orientation from a monocular camera using deep neural network. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9005, pp. 82–96. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16811-1_6
25. Drouard, V., Ba, S., Evangelidis, G., Deleforge, A., Horaud, R.: Head pose estimation via probabilistic high-dimensional regression. In: International Conference on Image Processing (ICIP), pp. 4624–4628. IEEE (2015)
26. Fanelli, G., Gall, J., Van Gool, L.: Real time head pose estimation with random regression forests. In: International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 617–624. IEEE (2011)
27. Wang, H., Davoine, F., Lepetit, V., Chaillou, C., Pan, C.: 3D head tracking via invariant keypoint learning. Trans. Circuits Syst. Video Technol. **22**(8), 1113–1126 (2012)
28. Liu, X., Liang, W., Wang, Y., Li, S., Pei, M.: 3D head pose estimation with convolutional neural network trained on synthetic images. In: International Conference on Image Processing (ICIP), pp. 1289–1293. IEEE (2016)
29. Tulyakov, S., Vieriu, R.L., Semeniuta, S., Sebe, N.: Robust real-time extreme head pose estimation. In: 22nd International Conference on Pattern Recognition (ICPR), pp. 2263–2268. IEEE (2014)
30. Schwarz, A., Haurilet, M., Martinez, M., Stiefelhagen, R.: Driveahead - a large-scale driver head pose dataset. In: International Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE (2017)
31. Derkach, D., Ruiz, A., Sukno, F.M.: Head pose estimation based on 3-D facial landmarks localization and regression. In: 12th International Conference on Automatic Face & Gesture Recognition (FG 2017), pp. 820–827. IEEE, May 2017

32. Meyer, G.P., Gupta, S., Frosio, I., Reddy, D., Kautz, J.: Robust model-based 3D head pose estimation. In: International Conference on Computer Vision (ICCV), pp. 3649–3657. IEEE (2015)
33. Yu, Y., Funes Mora, K.A., Odobez, J.M.: Robust and accurate 3D head pose estimation through 3DMM and online head model reconstruction. In: 12th International Conference on Automatic Face & Gesture Recognition (FG 2017), pp. 711–718. IEEE, May 2017
34. Ghiass, R.S., Arandjelović, O., Laurendeau, D.: Highly accurate and fully automatic head pose estimation from a low quality consumer-level RGB-D sensor. In: 2nd Workshop on Computational Models of Social Interactions: Human-Computer-Media Communication, pp. 25–34. ACM (2015)
35. Papazov, C., Marks, T.K., Jones, M.: Real-time 3D head pose and facial landmark estimation from depth images using triangular surface patch features. In: International Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2015)
36. Jeni, L.A., Cohn, J.F., Kanade, T.: Dense 3D face alignment from 2D video for real-time use. Image Vis. Comput. **58**, 13–24 (2017)
37. Morency, L., Whitehill, J., Movellan, J.: Generalized adaptive view-based appearance model: integrated framework for monocular head pose estimation. In: 8th International Conference on Automatic Face & Gesture Recognition (FG 2008), pp. 1–8. IEEE (2008)
38. Baltrušaitis, T., Robinson, P., Morency, L.P.: 3D constrained local model for rigid and non-rigid facial tracking. In: International Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2012)
39. Saragih, J.M., Lucey, S., Cohn, J.F.: Deformable model fitting by regularized landmark mean-shift. Int. J. Comput. Vision **91**(2), 200–215 (2011)
40. Kazemi, V., Sullivan, J.: One millisecond face alignment with an ensemble of regression trees. In: International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1867–1874. IEEE (2014)
41. Makehuman: Open source tool for making 3D characters (2017). http://www.makehumancommunity.org/. Accessed 31 May 2018
42. Hartley, R.I., Sturm, P.: Triangulation. Comput. Vis. Image Underst. **68**(2), 146–157 (1997)
43. Atkinson, K.E.: An introduction to numerical analysis. Wiley, New York (2008)
44. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C: The Art of Scientific Computing, Cambridge (1992)
45. Rosten, E., Porter, R., Drummond, T.: FASTER and better: a machine learning approach to corner detection. Trans. Pattern Anal. Mach. Intell. **32**, 105–119 (2010)
46. Bouguet, J.Y.: Pyramidal implementation of the affine Lucas-Kanade feature tracker description of the algorithm. Intel Corporation 5, 1–10 (2001)
47. Kun, J., Bok-Suk, S., Reinhard, K.: Novel backprojection method for monocular head pose estimation. Int. J. Fuzzy Logic Intell. Syst. **13**(1), 50–58 (2013)
48. Dodgson, N.A.: Variation and extrema of human interpupillary distance. In: Stereoscopic Displays and Virtual Reality Systems XI, vol. 5291, pp. 36–46. SPIE (2004)
49. Gordon, C.C., et al.: Anthropometric survey of U.S. army personnel: methods and summary statistics. In: Technical report 89–044, U.S. Army Natick Research, Development and Engineering Center, Natick, MA (1989)
50. Lefevre, S., Odobez, J.M.: Structure and appearance features for robust 3D facial actions tracking. In: International Conference on Multimedia and Expo, pp. 298–301. IEEE, June 2009

51. Tran, N.-T., Ababsa, F.-E., Charbit, M., Feldmar, J., Petrovska-Delacrétaz, D., Chollet, G.: 3D face pose and animation tracking via eigen-decomposition based bayesian approach. In: Bebis, G., et al. (eds.) ISVC 2013. LNCS, vol. 8033, pp. 562–571. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41914-0_55

52. German Research Center for Artificial Intelligence (DFKI): Head pose estimation dataset (2018). http://av.dfki.de/publications/real-time-head-pose-estimation-by-tracking-and-detection-of-keypoints-and-facial-landmarks/