EURASIP Journal on Image and Video Processing

RESEARCH Open Access

Robust head pose estimation based on key frames for human-machine interaction

Francisco Madrigal[1] and Frederic Lerasle[1,2*]

Abstract

Humans can interact with several kinds of machine (motor vehicle, robots, among others) in different ways. One way is through his/her head pose. In this work, we propose a head pose estimation framework that combines 2D and 3D cues using the concept of key frames (KFs). KFs are a set of frames learned automatically offline that consist the following: 2D features, encoded through Speeded Up Robust Feature (SURF) descriptors; 3D information, captured by Fast Point Feature Histogram (FPFH) descriptors; and target's head orientation (pose) in real-world coordinates, which is represented through a 3D facial model. Then, the KF information is re-enforced through a global optimization process that minimizes error in a way similar to bundle adjustment. The KF allows to formulate, in an online process, a hypothesis of the head pose in new images that is then refined through an optimization process, performed by the iterative closest point (ICP) algorithm. This KF-based framework can handle partial occlusions and extreme rotations even with noisy depth data, improving the accuracy of pose estimation and detection rate. We evaluate the proposal using two public benchmarks in the state of the art: (1) BIWI Kinect Head Pose Database and (2) ICT 3D HeadPose Database. In addition, we evaluate this framework with a small but challenging dataset of our own authorship where the targets perform more complex behaviors than those in the aforementioned public datasets. We show how our approach outperforms relevant state-of-the-art proposals on all these datasets.

Keywords: Head pose estimation, Key frames, RGB-D information, SURF descriptors

1 Introduction

The head pose provides rich information about the emotional state, behavior, and intentionality of a person. This knowledge is useful in several areas such as human-machine interaction [1], augmented reality [2, 3], expression recognition [4], and driver assistance [5], among others.

The task of correctly estimating the head pose with non-invasive systems might seem easy, and many current devices (smartphones or webcams) can detect human faces from videos or images in real time. Those are good for recreation, but they cannot handle all the difficulties in head pose estimation (HPE) such as (self) occlusion, extreme head poses, facial expressions, and fast movements.

Driver assistance scenario is a particular case where the user may exhibit complex behaviors such as zooming in/out of the steering wheel, wide range of head rotation, and fast movements. Here, the pose can verify if the user pays attention to the road allowing an autonomous system to assist the driver when necessary. Therefore, HPE algorithms should provide fast and robust information because missed detections or spurious estimates can lead to accidents.

Usually, HPE proposals [6–8] rely in RGB images to find specific 2D facial features, such as eyes, eyebrows, mouth, or nose. These heterogeneous features provide accurate estimations, but those are not available all the time, i.e., working with blurry images or light changes. Depth-based approaches, e.g., Fanelli et al. [4], can overcome some of the limitations of the 2D estimation allowing a better 3D HPE. Both methodologies perform well where the target's face is nearly frontal, but as mentioned above, this assumption cannot be guaranteed. Some applications use 3D models [9, 10] to retrieve the pose because they also

*Correspondence: lerasle@laas.fr
[1] CNRS, LAAS, 7 avenue du Colonel Roche, F-31400, Toulouse, France
[2] Univ. de Toulouse, UPS, LAAS, F-31400, Toulouse, France

provide semantic information, i.e., gaze estimation and facial expression.

We propose a framework that takes the best features of the aforementioned methodologies, combining 2D and 3D cues with a rigid 3D face model. It can handle challenging situations, such as large head poses, with a high detection rate and good accuracy for a wide range of orientations. Our approach follows an efficient key frame (KF) methodology with an offline learning phase and an online pose estimation step. Our fast and non-invasive offline step learns target's appearance and pose using a RGB-D sensor, in such a way that it creates a set of key frames (KFs) for that specific person, see Fig. 1. The KFs could be spurious or inaccurate; therefore, we propose a global optimization process based on bundle adjustment that improves the set of KFs and updates the 3D face model to better fix the target. This information is later used to estimate an accurate pose in the online step.

This process could be seen as a disadvantage because it needs to learn KFs for each new user, but our proposal incorporates an automatic learning system that only requires the user to perform simple movements in a short time before launching the online step. In several contexts, we can afford to perform this initialization stage. This is the case for driving assistance where learning could be done when the vehicle is stopped. Moreover, we might even suppose that the offline process conditions the start of the vehicle, allowing to verify in advance whether the user is in good conditions to drive.

We show how this key frame-based proposal provides competitive results to those in the state of the art. We evaluate our approach using the following: (i) the standard benchmark BIWI Kinect Head Pose Database [4], (ii) ICT 3D HeadPose Database, and (iii) our own dataset recorded with a Microsoft Kinect v1.

BIWI and ICT-3DHP datasets are, in the literature, standard benchmarks for evaluating head pose detectors with more than 240 and 200 cites, respectively [4, 10–12], where each target is recorded with neutral expression, rotating the head at a slow-medium speed. However, these datasets do not represent complex and challenging movements that a human could do. Therefore, we develop our own dataset where the targets perform more natural movements as those expected in real scenarios. It consists of four sequences where targets show complex behaviors, such as rapid head movements, self-occlusion, and facial expression, among others. Although we evaluate several datasets, all the examples shown in this paper use images from our "ICU" dataset to describe the different steps of our proposal. Thanks to quantitative evaluations of these challenging sequences, we demonstrate that our monocular RGB-D-based approach offers competitive results to current approaches in the state of the art.

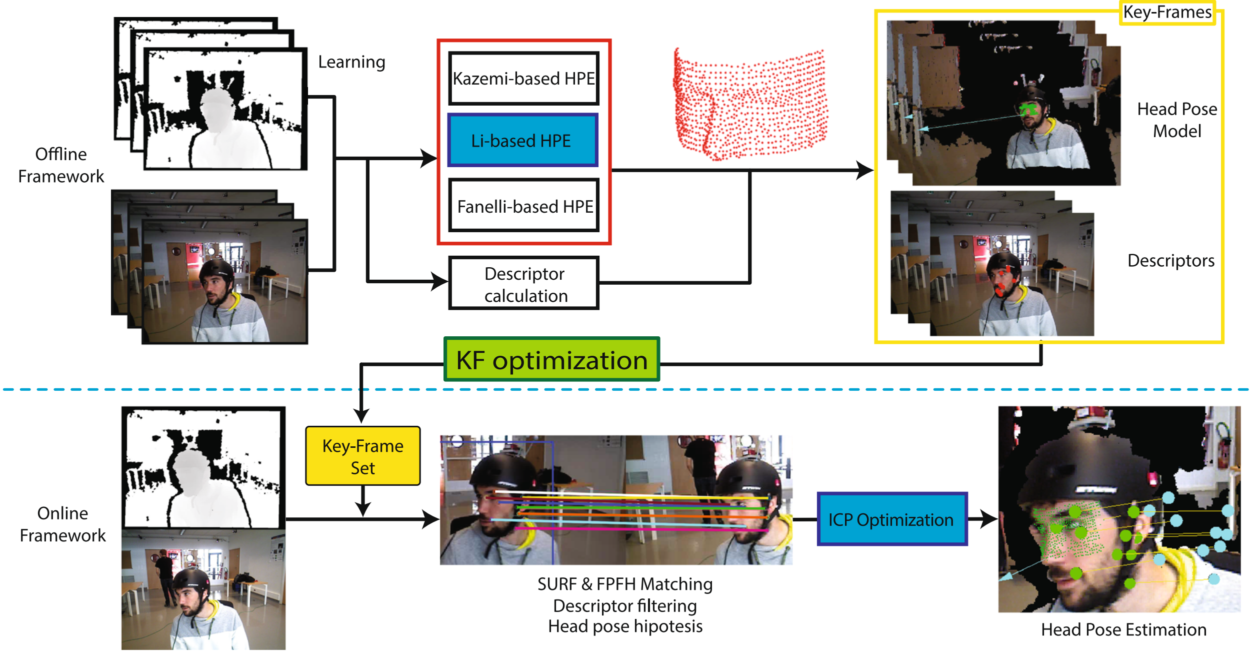

Fig. 1 Pipeline of the proposed head pose estimation method. Top: offline framework. The input is a RGB-D stream of the person to track. Rough pose estimation is done (red block), and most relevant frames are selected (yellow block). Then, an optimization process improves the pose accuracy (green block). Bottom: online pose estimation process. The current image is shown at left. The key frame with the best matching score is selected and depicted at right. Finally, ICP refines the estimation (right image)

The main contributions of this paper are as follows:

1. A key frame-based framework, with state-of-the-art accuracy, that consists of an original offline process with an automatic learning step with global consistency, a KF optimization step based on error propagation, and a 3D face model updating methodology. All the above learned information is considered during an online head pose estimation with a formulation that takes into account the descriptors, normal surface, and self-occlusion.
2. A new dataset exhibiting more complex behaviors to those present in the aforementioned datasets.

This paper has the following structure: We present the related work in Section 2. The formulation of our methodology for pose detection is given in Section 3. Section 4 presents the quantitative and qualitative results including a discussion where we compare our framework with respect to other two approaches in the state of the art. Last, Section 5 describes conclusions and future work.

2 Related works

In the fields of mobile robotics and computer vision, there are works focused on monocular systems for HPE, i.e., [13, 14], that can be categorized according to cue used. Hereafter, we mention a few of the most relevant ones.

2.1 RGB-based approaches

Some approaches tackle the HPE problem by using 2D deformable models that can approximate the human face shape [15, 16]. In [6], Kazemi and Sullivan propose a fast face alignment framework based on a random forest where each regression tree is learned by a gradient boosting-based loss function. This methodology allows to detect multiple faces with high accuracy at a speed of 1 ms per image even with complex expression (strong facial deformations) or small head rotations.

Other proposals seek specific facial features, i.e., eyes and nose, among others. Valenti et al. [8] learn the location of the eyes from a set of training images, and assuming that the head follows a geometrical shape, those are projected in a cylinder. This person-specific model is used then for detecting and tracking the target. Barros et al. [7] follow a similar strategy, but including motion information from optical flow to reinforce the estimation. Drouard et al. [17] propose a learning method based on histogram of oriented gradients (HoG). HoG features are mapped (through a Gaussian locally linear model) onto the head pose space, which is then used to predict a new head orientation. Chen et al. [18] achieve good results with RGB images of low resolution using a Support Vector Regression (SVR) classifier trained with a gradient-based feature. All these methods combine the information in a single model and achieve state-of-the-art (SoA) results when sufficient training data is provided.

Learning the appearance of a person with a single shot is not always possible due to problems such as changes in lighting or occlusions. Therefore, several works rely on the information coming from a set of relevant frames, called key frames (KFs)[19]. In [20], Vacchetti et al. propose a KF-based method that detects and estimates the 3D pose of static rigid objects using only RGB images. Each KF consists of a set of key points and a 3D model, projected to image plane using camera calibration. The proposal provides SoA results by considering both 2D-3D key frame matching and 2D-2D temporal matching. The work of Kim et al. [21] exploits the idea of KF for pose estimation and tracking of multiple 3D objects from 2D information. The methodology can obtain results in real time, i.e., 40 objects within 6 to 25 ms per frame. In the last two proposals, the camera is moving while the target remains static. Nevertheless, these methods can work in the opposite way, i.e., static camera with moving targets. Morency et al. [22] propose a generalized adaptive view-based appearance model (extension of the AVAM algorithm of [23]) that estimates the head pose for a specific image region. The final pose is inferred by merging the results of (1) a referential frame, (2) tracking between current and previous frame, and (3) matching against a KF.

A more recent method [24] uses deep learning to train a convolutional neural network (CNN) using RGB images. The results are provided in real time and can handle challenging issues such as different light conditions.

The 2D-based proposals perform well with nearly frontal views, but they have difficulty estimating an accurate head pose due to problems such as large poses, (self) occlusions, and changes in lighting. In this sense, depth cue is more efficient in such situations.

2.2 Depth-based approaches

Many of nowadays SoA methods are based on the depth cue because 3D information provides the shape of the head in a more distinctive way [4, 12, 25] .

In [25], the authors use the depth image to tackle some of the problem of pose estimation such as partial occlusion and head orientation variations. The proposal rotates a generic 3D human face model, and each rotation is transformed in a depth image, which is later used in the alignment process. This offline-learned set is compared to the input depth frame, and the best match provides the pose hypothesis. It achieves real-time results thanks to a framework based on graphics processing units (GPUs).

Fanelli et al. [4] train a random regression forest that allows to detect poses in real time through nose tip detection. The training data is generated in a similar way as [25] using a 3D face model set with several orientations. Each leaf of the regression tree votes for a possible nose position, and the final pose is inferred by considering all votes. The high quality of their results has converted it in a baseline to compare new proposals.

Papazov et al. [12] propose a new 3D invariant descriptor that encodes facial landmarks. The descriptors are learned in an offline training phase using a group of high-resolution meshes with triangular paths. A CNN is used in [26] to estimate head pose from pure depth data with the use a Siamese network (a couple of CNN) achieving high accurate results in real time.

2.3 RGB-D-based approaches

The combination of color and depth cues has shown high performance in challenging situations. In the work of [11], the pose is inferred by fitting a morphable 3D model on the target represented by a 3D point cloud. The model is learned for a specific person in an offline training step. Saeed and Al-Hamadi [27] use HoG features, extracted from both RGB and D cues, to train a classifier based on support vector machine (SVM). In [28], the authors present a similar method that combines 2D and 3D HoG features but to train a multi-layer perceptron classificator. In [29], the authors present an improvement to the constrained local model by including 3D information. Then, they train some SVM classifiers and logistic regressors using probabilistic features.

Some works enhance classical methods by including depth information. This is the case with [30], in which the authors use the depth cue in a visual odometry technique. Smolyanskiy et al. [31] add a depth-based constraint to an active appearance model fitting. However, this approach suffers from drift problems, where the final model is not well aligned with target's 3D position. Some other proposals propose to combine depth and color cues using random forest [32]. Here, tensor-based regressors allow to model large variations of head orientation.

In [10], Li proposes a method based on an energy minimization function that optimizes the distance between a 3D point cloud (current frame) and a rigid template model of the human face. The optimization is carried out using ICP algorithm, and the color cue is used in two ways: (1) to detect 2D facial landmarks, using the method of Viola and Jones [33], and (2) to remove outliers, using a k-means clustering algorithm. The detected landmarks, i.e., eyes, are projected to 3D world through the depth image and included in the energy function as a weight factor, which increased the accuracy and convergent speed of ICP. On the other hand, k-means allows to separate relevant 3D points (i.e., those belonging to the face) from the spurious ones (i.e., clutter). The face model is updated online in a parallel process using only the depth cues allowing to adapt to different kinds of faces. The proposal relies in the work of Fanelli et al. [4] to reinitialize the approach because ICP requires more time to infer a face pose from an initial position than from previous frame. Meanwhile, Fanelli et al.'s approach finds a face faster but with less precision. Yu et al. [9] propose a similar method that instead learns a 360° 3D morphable model, including a motion cue, based on optical flow, in the ICP optimization process.

2.4 Descriptors

Descriptors encode important information about the visual characteristics of the objects present in images [34], such as appearance [35, 36], motion [37], or geometry [38]. Therefore, they have been used in multiple contexts. Yan et al. [39] propose a FAST-like descriptor which considers the orientation of image intensity. Alioua et al. [35] propose a 2D head pose estimation framework using a combination of classic descriptors, e.g., HoG, SURF, and Haar. Yan et al. [36] uses two CNN features to model global and local appearance of the target and a 3D CNN which codify the motion.

The computational cost of some descriptors could be expensive, e.g., especially those based on deep learning [36], even using parallelization methods [37]. Therefore, we rely on robust features with fair computational cost.

2.5 Synthesis

The aforementioned proposals have some qualities that adapt well in specific scenarios. To mention some outstanding methods, we have the following: Kazemi and Sullivan [6] who use a RGB-based method with fast estimation and high accuracy in frontal view, Fanelli et al.'s [4] proposal which relies in depth information and provides good detection rate, and Li et al. [10] who can achieve accurate results for head poses with large rotation. A combination of these (or more) methods could face the challenges of estimating head pose, but a direct combination could not generate results in real time.

Finally, there are some datasets to evaluate the performance of HPE algorithms, such as BIWI dataset [4] and ICT-3DHP dataset [29], that are the standard benchmark used in several relevant papers [4, 9–12]. They consist of multiple sequences, each with a different person, where the target has a neutral expression, with slow-medium speed head rotation and (mostly) remaining in the same position.

From above, we can summarize our contributions as follows:

1. A robust HPE algorithm based on KF that combines 3D geometry information (point cloud), appearance, and shape (encoded through SURF and FPFH descriptors), exploiting all RGB-D channels.
2. A double mechanism consisted of (1) an offline learning phase that exploits the complementarity of aforementioned techniques to create a

person-specific set of KFs and (2) an online framework based on KF and ICP that estimates robustly and in real time the head pose.

3. A bundle adjustment process that improves the accuracy, in terms of performance and CPU cost, of the learned KFs in order that they are consistent between them.
4. An online update of both the KFs and 3D face model.
5. A new dataset with more challenging behaviors and situations that those in the literature consisting of four sequences with a ground truth generated from a tion (MoCap) system. It includes rapid head movements, facial deformation, self-occlusions, and position displacement, among others.
6. A rigorous and large-scale evaluation and comparison with relevant existing approaches in the state of the art.

3 Method

Our key frame-based approach is inspired by some works like [20, 22], and [25] but for the applicative context of HPE for human-machine interaction, i.e., human HPE instead of static objects considering both appearance and depth cues with a partial 3D face model. Each KF consists of a set of 3D appearance features (SURF descriptors projected to 3D world through the depth image), 3D-based features, and an approximate head pose, represented with a 3D template model. First, we describe the contents of each key frame to then show how they are learned consistently and subsequently used in a pose estimation system.

3.1 Key frame generation

3.1.1 3D face model

A 3D morphable face model (3DMFM) is a shape representation of a human face that can be used to provide accurate estimations for most of the head poses. Then, a face model M is a set of 3D vertex/points created as a linear combination of a mean shape μ with a weighted deformation basis DB as follows:

$$M = \mu + \sum_{i=1}^{V_n} \gamma_i \bar{\omega}_i DB_i. \qquad (1)$$

Here, γ_i and DB_i are the eigenvalue and eigenvector, respectively, learned from a set of 3D scans. In our approach, we use the Basel Face Model (BFM) [40], which has learned the DB values from the 3D face scans of 200 subjects, each with different age, gender, height, and width. Traditionally, 3DMFM fitting is an offline optimization step that finds the $\bar{\omega}_i$ values through the minimization of the distance between one (or more) 3D frame(s) and the model. This allows to create a model with a facial shape similar to a specific person, i.e., [9, 10].

Our offline key frame learning step uses a generic human face model M with average characteristics, i.e., age, weight, and gender. This model fits well in most of the cases, but it must be updated in order to fit some facial structures. Section 3.2.3 describes an efficient optimization scheme that does not rely in calculating $\bar{\omega}$ of Eq. 1 like other methods, but in an error propagation-based approach inspired by as bundle adjustment.

Even with a well-fitted model, some HPE algorithms have problems handling face deformation such as mouth movements or facial expressions. This is a common situation when a person is speaking with other one or reacting to external situations, i.e., music and other people's movements, to mention a few. We keep this in consideration and create a partial model with only the part between nasal base and forehead. This region does not deform much and provides results as accurate as more complete models.

In any case, we use Eq. 1 to build a partial face model $M = \{p_1, \ldots, p_m\}$ consisting of $m = 1000$ 3D points $p = \{x, y, z\}$; an example of the model is shown in Fig. 1 represented as the output of the red block.

3.1.2 Face descriptors

Our proposal relies and is based on natural facial landmarks encoded through SURF descriptors, which allow to estimate features invariant to rotation and scale, and Fast Point Feature Histogram (FPFH) descriptors, which include 3D information invariant to illumination changes. These descriptors enhance the robustness of the HPE and increase both accuracy and detection orientation range.

SURF descriptors SURF is a robust and reliable descriptor that has shown good performance in several topics such as SLAM, camera pose estimation, and image registration. In the context of HPE, SURF describes a specific person's face in a general way, avoiding the need to search specific features (e.g., eyes, nose). Therefore, any relevant characteristic is taken into account, regardless of its origin, i.e., beard, mustache, glasses, or other. In addition, these descriptors are invariant to scale and rotation allowing to detect no-static targets, i.e., drivers moving around in the cockpit and people interacting with robots, among others.

We use SURF in a similar way as in image registration: we calculate a set of η^α interest point in the foreground of image plane using the good features to track algorithm. Since each RGB pixel has associated a depth value, we define the background as any point farther than a threshold th_a. Thereby, we have a set of f^α features with their respective 3D position $p_j^\alpha = \{x, y, z\}$ as follows:

$$d_j^\alpha = \left\{ f_j^\alpha, p_j^\alpha \right\} \ \forall j \in \{1 \ldots \eta^\alpha\} \ : ||p_j^\alpha|| < th_{\text{bg}}. \quad (2)$$

From Eq. 2, we have a descriptor that encodes the appearance of a specific person in 3D world, and by grouping them, we get the set:

$$D^\alpha = \left\{ d_1^\alpha \ldots d_\eta^\alpha \right\}. \quad (3)$$

In practice, the parameters used in SURF get a $\eta^\alpha \approx 100 - 200$ descriptors. SURF descriptors are robust in cases with little luminosity changes and flat objects, and in our problem, they have proven to be useful for the pose estimation. Although certain changes of a 3D object, due to lighting or rotation, cannot be captured properly by these descriptors, we use a shape descriptor that reinforces the estimation.

FPFH descriptors Curvature estimates and surface normals are a basic representation of the geometry of an object, easy to compute and compare. Although the level of detail captured is not much, many points contain the same (or similar) feature information. Alternatives are the 3D descriptors, and they summarize the object's geometry taking into account the aforementioned features in an efficient manner.

Fast Point Feature Histogram (FPFH) descriptor, proposed by Rusu et al. [38], captures the normal surface variations around a point, resulting in a high hyperspace signature that is invariant to the 6D pose (rotation and position) and robust against the neighborhood noise. It is formulated as follows:

$$f_j^\beta = \text{FPFH}\left(p_j^\beta\right) = \text{SPFH}\left(p_j^\beta\right) + \frac{1}{|\mathbb{N}_j|} \sum_{i \in \mathbb{N}_j} \frac{1}{\kappa_i} \cdot \text{SPFH}(p_i), \quad (4)$$

where SPFH (Simplified Point Feature Histogram) computes the set of angular features of the PFH descriptor, κ_i is the distance between p_j^β and p_i, and $\mathbb{N}_j$ is the set of neighboring points of p_j^β. We build the set point to evaluate by considering (1) the 3D projection of the points computed by good feature to track methods, in the same way as in SURF, and (2) a downsampling of the target point cloud. The 3D frame descriptors are formulated in a similar way as in the previous section:

$$D^\beta = \left\{ d_1^\beta \ldots d_\eta^\beta \right\}, \quad (5)$$

where

$$d_j^\beta = \left\{ f_j^\beta, p_j^\beta \right\} \ \forall j \in \left\{1 \ldots \eta^\beta\right\} \ : ||p_j^\beta|| < th_{\text{bg}}. \quad (6)$$

Finally, each KF contains these three elements: appearance and shape signatures and a 3D face model, together with the depth image. In practice, the number of descriptors $\eta^\beta \approx 200$.

3.2 Offline key frame learning

In this section, we describe how the KFs are learned from a RGB-D stream, see workflow in Fig. 1. First, the target pose is roughly estimated using a robust but computational expensive system based on three state-of-the-art

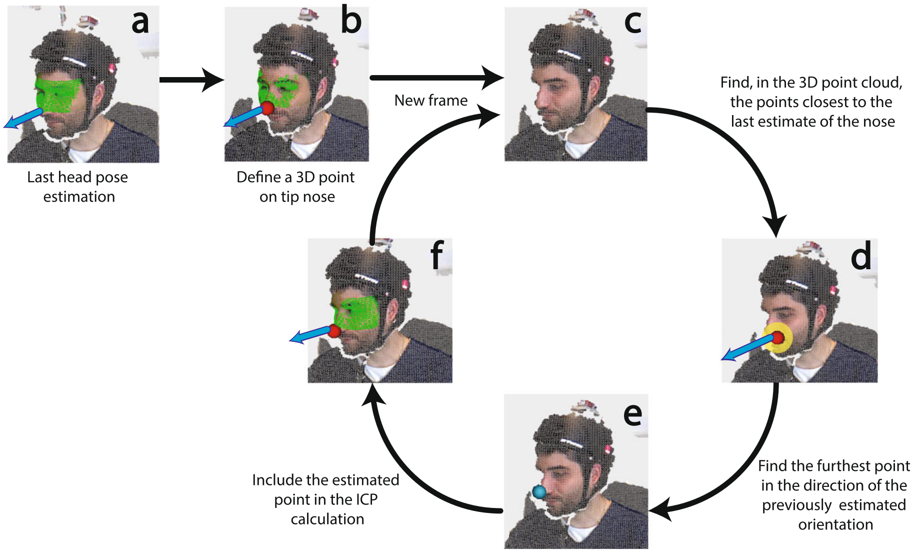
a
Last head pose estimation
b
Define a 3D point on tip nose
New frame
c
Find, in the 3D point cloud, the points closest to the last estimate of the nose
d
Find the furthest point in the direction of the previously estimated orientation
e
Include the estimated point in the ICP calculation
f

Fig. 2 Estimation of nose feature. From a previous pose (**a**), we define a 3D point on the tip of the nose (**b**). Then, From a new frame (**c**), we find all neighboring points from last estimation (**d**) and select the furthest point in the direction of last pose estimation (**e**). This point is then included in the ICP calculation

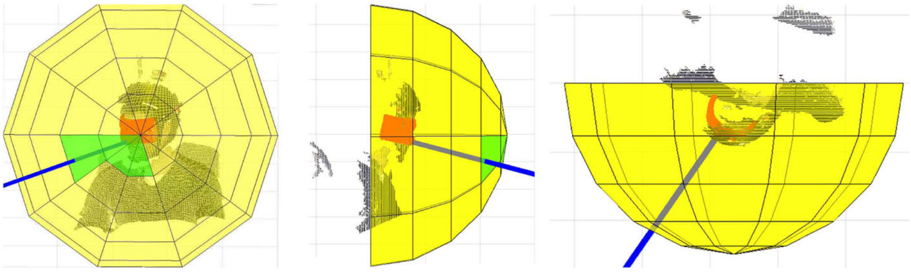

Fig. 3 Key frame learning process. Each figure represents the same target observed from three views. The discretized orientation is depicted as the region of the sphere. The color indicates visited areas (green) from regions without an associated pose (yellow). The set of points in red is the 3D model of the estimated pose. Target orientation at current frame is shown with the blue line

methods (red block in Fig. 1). Only the most relevant frames, according to the quality of the estimated pose and the descriptors, are selected as key frames, yellow block. Finally, an optimization process (green block) improves the KF-estimated poses and suppresses spurious frames, i.e., which are not consistent with any other.

3.2.1 *Rough pose estimation*

Some methods require the use of other algorithms for initialization or learning [9, 10]. Our proposal requires a rough estimation of the pose, or rough pose estimation, that is computed by combining three HPE systems that have a good accuracy/CPU-cost ratio: Kazemi and Sullivan [6] 2D face detector, Fanelli et al. [4] depth based, and Li et al. [10] RGB-D based method.

These proposals complement each other and provide a first good estimate on which we rely to create a more robust method. Kazemi and Sullivan's [6] proposal is a fast-facial feature detector and is part of a public library, DLib from [41]. Fanelli et al.'s [4] approach has over 200 cites and has been included as a module for the Robot Operating System (ROS) library. Li et al.'s [10] method brings more accurate results than that of Fanelli et al. for far-reaching orientations.

The work of [10] consists of two independent parts (computed in parallel): (1) a head pose tracking framework based on ICP and (2) a 3D model update system. This method is based on facial features that cannot handle well large head rotations, and therefore, the accuracy decreases when the 2D face landmark detector fails. Therefore, we propose a simple but reliable 3D feature, see in Fig. 2, that provides additional information for feature-based systems, i.e., [10].

Let's assume $q_{t-1} = \{x, y, z\}$ as the 3D position of nose tip estimated from previous frame and θ_{t-1} as the head orientation, red sphere and blue line in Fig. 2b, respectively. Assuming a slow movement of the target, the next nose point q_t should be close to previous estimation; we can find this new nose by analyzing the neighboring q_{t-1} in the current target' point cloud ψ_{t-1}:

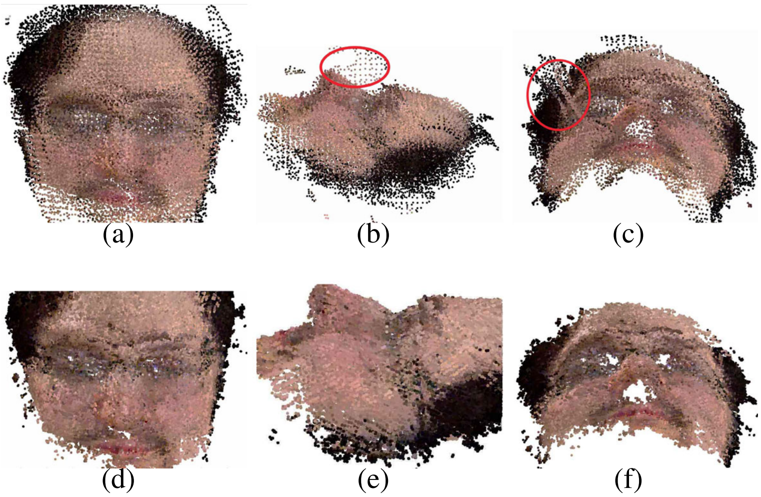

Fig. 4 Three views: (**a**, **b**) frontal, (**b**, **e**) profile and (**c**, **f**) side; of the point cloud of all KFs projected to a common frame before (top) and aJer optimization (bottom)

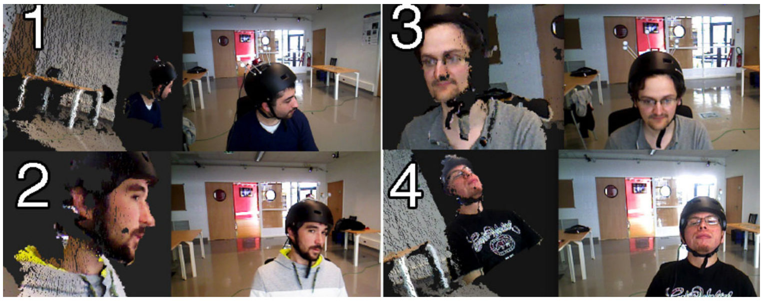

Fig. 5 Example of the 4 sequences of our dataset. The images show the 3D point cloud (on the left) and the RGB image (on the right) for each sequence. NOTE: panel label only enumerates the sequences

$$N_t = \{p \in \psi_t : ||p - q_{t-1}|| < r\},$$

where $r = 0.2m$ is the searching radio. In other words, N_t are the neighboring points of q_{t-1} and one of those is a good candidate to be the next nose tip ($q_t \in U_t$), see yellow area of Fig. 2d. From previous pose estimation, we define nose as the furthest point in the orientation θ_{t-1}:

$$\hat{P}_t = \arg\min_{p \in N_t} \{\upsilon(p, q_{t-1}, \theta_{t-1})\},$$

where $\upsilon(\cdot)$ computes the distance between point p and a line segment defined through q_{t-1} and θ_{t-1}. $\hat{q}_t$ is shown as the blue sphere in Fig. 2e.

In [10], the authors include the 3D eye positions, detected with Viola and Jones [33] algorithm and projected through the depth image, as a weighted factor in the ICP algorithm. We do the same with this nose feature $\hat{q}_t$; the correspondences between $\hat{q}_t$ and a 3D template model have a weight of 40, as indicated in [10]; and the rests are set to 1. This process guides the template to zones with high probability of been the target's face; Fig. 2f shows the final estimation.

This feature enhances the accuracy of the original proposal; thus, we use this nose-based framework in the KF learning. Like other person-specific methods [11], we must learn the appearance of each new target, but the process is worth it because, as detailed below, it improves the accuracy of the estimations.

3.2.2 Automatic frame selection

In some application context, e.g., driver assistance, we can take some time to perform the KF learning before starting the vehicle without any danger. Here, robust estimates of head pose are essential because inaccurate or missed detections can cause accidents. This could be difficult to achieve because the target behavior is sometimes complex with random or abrupt movements. We develop our proposal considering that the KFs can handle well these scenarios providing high-quality results. Therefore, we consider that it is justifiable to take a little time in order to learn a robust person-specific set of KFs.

First, we estimate the rough pose as described in Sec. 3.2.1 where the methods (Kazemi, Fanelli, and Li) propose each one a HPE $P_* = \{q_*, \theta_*\}$ where $q = \{x, y, z\}$ is the nose location and θ is the head orientation. Thus, we have at frame t three pose estimation candidates $C_t = \{P_{\text{Kazemi}}, P_{\text{Li}}, P_{\text{Fanelli}}\}$. In the best-case scenario, all the methods converge to a similar point, i.e., mean of the three poses $\bar{P}_t = \{\bar{q}_t, \bar{\theta}_T\}$ has a small variance $Var(C_t)$. If this is the case, we add $\bar{P}_t$ to the set of key frame pose S^{KF}. Otherwise, we select a pose according to the qualities of the methods. Kazemi is highly accurate with frontal view targets, Fanelli can detect poses even with rapid motion, and Li works better with heads that exhibit large orientation (looking to right/left, full profile). Therefore, we privilege these techniques according to each situation:

$$P_t^{KF} = \begin{cases} \bar{P}_t & \text{if } \; Var(C_t) < th_v \\ P_{\text{Kazemi}} & \text{if } \; ||\bar{q}_t - q_{\text{Kazemi},t}|| < th_d \\ & \quad \textbf{and } \; \theta^o < th_\theta \\ P_{\text{Li}} & \text{if } \; ||\bar{q}_t - q_{\text{Li},t}|| < th_d \\ & \quad \textbf{and } \; \theta^o > th_\theta \\ P_{\text{Fanelli}} & \text{if } \; ||\bar{q}_t - q_{\text{Fanelli},t}|| < th_d \\ & \quad \textbf{and } \; \bar{s} < th_s \end{cases},$$

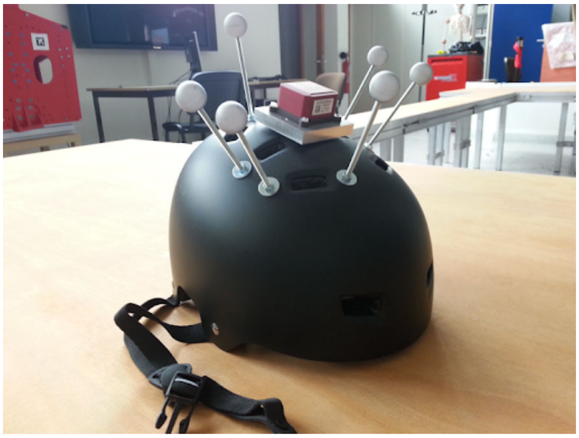

Fig. 6 Helmet used for the acquisition of the ground truth

where $th_d = 5$ cm and $th_\theta = 45^o$ are the pose and orientation thresholds, θ^o is the existing angle between camera origin and target pose, and $th_v = 0.5$ is the variance threshold. We define $\bar{s} = ||\bar{P}_t - P^{KF}_{t-1}||$ as the angular speed between two consecutive pose estimations with $th_s = 1 rad/s$ as the speed threshold.

Descriptor computation So far, the descriptions D^α and D^β are calculated in the foreground and, therefore, may include irrelevant non-face features. To remove spurious information, we simply rely in the rough estimate P^{KF}_t that defines the position of the 3D face model. We use this knowledge to filter out the points far enough from the template. Let us assume q^M as the nose position of the model zone and $L_2(d,p)$ as the Euclidean distance (norm L_2) between 3D points. Then, we filter the points according to a threshold th_e as follows:

$$\hat{D}^\alpha = \left\{ d^\alpha_j \in D^\alpha \ : \ L_2\left(p^\alpha_j, q^M\right) < th_e \right\} \tag{7}$$

$$\hat{D}^\beta = \left\{ d^\beta_j \in D^\beta \ : \ L_2\left(p^\beta_j, q^M\right) < th_e \right\} \tag{8}$$

Frame selection The accuracy of the estimation is related to the number of key frames. More KFs improve the results, but computational cost is also increased. We keep the number low by discretizing the orientation space through spherical coordinates discretized at 20°. An example is shown in Fig. 3 where a yellow polygon depicts the discretized orientation.

Once an estimate is close to the center of the discretized area, we keep the pose P^{KF}_t and compute the descriptors $\hat{D}^{KF}_t = \left\{ \hat{D}^\alpha, \hat{D}^\beta \right\}$ around it. We change the color of the visited areas to green in such a way that the user can observe the missing orientations (Fig. 3). Sometimes an area is visited more than once; in this case, we keep the best KF based on a fitting score (given by pose estimation algorithms) and number of descriptors. Finally, the KF set is defined as follows:

$$S^{KF} = \left\{ P^{KF}_k, \hat{D}^{KF}_k \right\} \ \forall \ k = \{1 \ldots K\}. \tag{9}$$

In this learning process, target should move its head at normal speed performing only head rotations, as recorded in BIWI and ICT-3DHP datasets. We consider around of $30 - 40$ KF, covering most of the orientation space, and 100 SURF/FPFH descriptors. The set S^{KF} can be used as it is; however, we can enhance the pose estimation of each KF by applying an optimization step.

3.2.3 Key frame pose optimization

The KFs provide rich information of the pose and appearance of the target. An automatic learning method provides a good initial estimation, but small errors in the set of KF limit the quality of new estimates. Moreover, it could include spurious frames (inconsistent estimate), red circle in Fig. 4. Therefore, we can overcome those issues by applying an optimization process that provides a global and simultaneous consistency between all KFs and the 3D face model.

Table 1 Description of our own head pose sequences

Seq	Frames	Rot. range (degrees)	Mean speed (rad/s)
Seq1	1890	± 60 yaw ± 40 pitch	0.94
Seq2	1083	± 80 yaw [+ 30, − 65] pitch	0.83
Seq3	1535	± 80 yaw ± 45 pitch	2.3
Seq4	1929	± 80 yaw [+ 20, − 80] pitch	2.51

To achieve this, we need to minimize the error between the 3D face model and all KFs. Let us assume M as the template model in a reference position (origin of 3D world with not rotation) and $\mathbb{K}_k$ as the point cloud of the k-th KF. We need to process only the points corresponding to the face. This position is known from the estimated poses P^{KF}_k, and therefore, we filter the points p of $\mathbb{K}_k$ keeping only those around 20 cm of the pose estimation, i.e., $\mathbb{H}_k = \{p \in \mathbb{K}_k : L_2\left(p, p^{KF}_k\right) < 0.2m\}$.

Hence, the goal is to find the transformation parameters $\tau_k = \{\mathbf{R}_k, \mathbf{t}_k\}$ that minimize two aspects: (1) the local error between the paired points of the human face model M and the KF point cloud $\mathbb{H}_k$,

$$\mathbb{P}_k = \{(h,p) : h \in \mathbb{H}_k, p \in M\},$$

and (2) the global error between the rest of the KF facial point cloud $\mathbb{H}_*$,

$$\mathbb{Q}_{ik} = \{(h_i,h_k) : h \in \mathbb{H}_i, h_k \in \mathbb{H}_k\}.$$

This can be achieved by minimizing the following cost function:

$$\begin{aligned} \arg\min_{\tau} \sum_{k=1}^{K} \frac{1}{|\mathbb{P}_k|} \sum_{(h,p)\in\mathbb{P}_k} ||p - \mathbb{T}(h,\tau_k)||^2 + \\ \sum_{i \neq k}^{K} \frac{\lambda_i}{|\mathbb{Q}_k|} \sum_{(h_i,h_k)\in\mathbb{Q}_{ik}} ||\mathbb{T}(h_k,\tau_k) - \mathbb{T}(h_i,\tau_i)||^2, \end{aligned} \tag{10}$$

Table 2 Evaluation of our proposal considering the different elements

Variants	Descriptors		Global	3D model	Weighted
	SURF	FPFH	optimization	update	ICP
			(Eq. 10)	(Eq. 11)	(Eq. 15)
KFv1	Yes	No	No	No	No
KFv2	Yes	No	Yes	Yes	No
kFv3	Yes	Yes	Yes	Yes	Yes

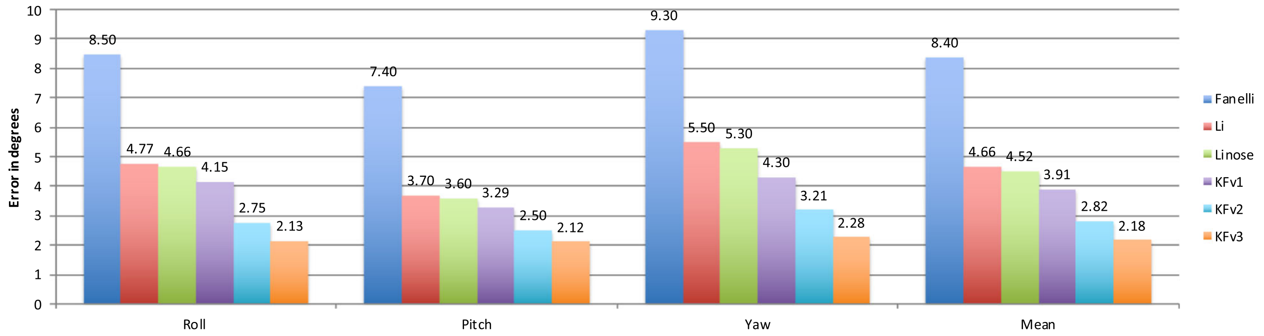

Fig. 7 Mean of the results on the BIWI dataset. Mean orientation error (in degrees)

where $\mathbb{T}(\cdot)$ apply the geometric transformation of a point h with respect to τ_*, $|\cdot|$ is the cardinality, and $\tau = \{\tau_1 \dots \tau_K\}$ is the set of all transformations. The variable λ_i weights the contribution of the ith KF ($\mathbb{H}_i$) to evaluate and is derived from the percentage of paired points between the face model M and the ith KF point cloud:

$$\lambda_i = \frac{|\mathbb{P}_i|}{|M| \cup |\mathbb{H}_i|}.$$

We can observe that λ_i is close to zero when the number of paired points($\mathbb{P}_i$) is small, meaning this KF is not a good match to work with because it is desalinated or is a spurious frame. At each iteration, we remove the KFs with a low weight $\lambda_i < 0.25$ because we cannot guaranty that those are a real part of the face or point cloud coming from bad estimates.

We optimize Eq. 10 following an iterative scheme such as ICP. First, we select a KF k and perform the optimization, and we repeat this process with the rest until convergence. Figure 4 shows the KFs (projected to a reference frame) before and after optimization, from which the target's face can be seen more clearly. Finally, we recalculate the poses and filter 3D points of the model.

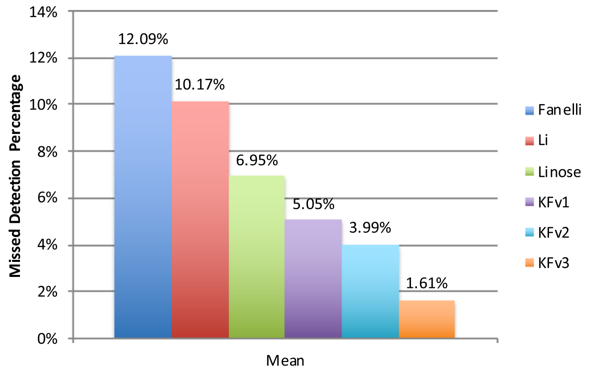

Fig. 8 Mean of the results on the BIWI dataset. Percentage of missed detections

3D facial model update Some persons could have facial features more different than generic model, i.e., a person with mustache or beard, causing a bias in the estimate. We can overcome this issue by adjusting the model according to the refined 3D point clouds. Let us assume $\hat{\mathbb{H}}$ as the union of all 3D mesh projected in the reference position:

$$\hat{\mathbb{H}} = \bigcup_{k=1}^{K} \mathbb{T}(\mathbb{H}_k, \tau_k),$$

This point cloud $\hat{\mathbb{H}}$ is seen as a scattered data, and through an interpolation algorithm based on Delaunay triangulation [42], we create a mesh $\mathbb{F} = \text{Delaunay}(\hat{\mathbb{H}})$ that describes the facial surface of the target. Then, the new model $\hat{M}$ is estimated from the paired vertex $(M^i, \mathbb{F}^i)$ by minimizing the cost function:

$$\sum_{i \in M} ||\hat{M}^i - \mathbb{F}^i||^2 + \frac{\gamma}{|N_i|} \sum_{j \in N_i} ||\hat{M}^i - M^j||^2, \qquad (11)$$

where N_i are the neighboring vertex of $\hat{M}^i$ and γ weights the similarity of the original model. This equation updates the points of M with respect to $\mathbb{F}$ allowing the generic template to evolve in a model $\hat{M}$ more similar to the target.

3.3 Online head pose estimation

In this section, we present our original framework that exploits the characteristics of the KFs, in comparison with other existing approaches. We have a set S^{KF} with appearance and shape descriptors (associated to 3D points) and a robust pose estimation. As mentioned above, descriptors are computed only on the area around the 3D model, so we have $\hat{D}_k = \{d_1, \dots, d_{\eta_k}\}$ for each k KF. We apply a similar process for the current frame.

3.3.1 Pose estimation

Initialization For a new frame t, we first compute the descriptors following the steps as mentioned in Section 3.1.2, sampling over the whole foreground images

Table 3 Results on BIWI dataset in Euler angles

Method	Yaw	Pitch	Roll	Mean
Fanelli et al. [4]*	9.3 ± 8.8	7.4 ± 8.1	8.5 ± 9.9	8.40 ± 8.93
Li et al. [10]*	5.5 ± 4.6	3.7 ± 5.9	4.77 ± 5.2	4.66 ± 5.23
Li Nose*	5.3 ± 3.5	3.6 ± 5.5	4.66 ± 4.9	4.52 ± 4.63
Saeed and Al-Hamadi [27]+	3.9 ± 4.2	5.0 ± 5.8	4.3 ± 4.6	4.4 ± 4.9
Baltrušaitis et al. [29]+	14.80	12.03	23.26	16.69
Venturelli et al. [26]+	2.8 ± 3.3	2.3 ± 2.7	2.1 ± 2.2	2.4 ± 2.73
Yang et al. [28]+	8.9 ± 8.2	9.1 ± 7.4	7.4 ± 4.9	8.5 ± 6.9
Papazov et al. [12]+	3.0 ± 9.6	2.5 ± 7.4	3.8 ± 16.0	4.0 ± 11.0
Ahn et al. [24]+	2.8 ± 2.4	3.4 ± 2.9	2.6 ± 2.5	2.9 ± 2.6
Yu et al. [9]	2.54	**1.45**	**2.10**	**2.03** ± 3.0
KFv1	4.3 ± 2.8	2.8 ± 2.9	4.15 ± 3.19	3.91 ± 3.19
KFv2	3.21 ± 1.4	2.5 ± 1.5	2.75 ± 2.77	2.82 ± 2.08
KFv3	**2.28** ± 1.7	2.12 ± 1.17	**2.1** ± 1.46	2.18 ± **1.44**

Results with the best performances are presented in bold
*Estimation that we calculated
+Results taken from author's papers

because we do not know the location of the target. Thus, we have extracted the descriptors $D_t = \left\{D_t^{\alpha}, D_t^{\beta}\right\}$. Although in some cases it may not be necessary to use both types of descriptors, the use of both allows to compensate any problem that the other has, for example, drastic changes in the lighting affect SURF.

Key frame selection We need to find the KF $S_b^{KF} = \{P_b, D_b\}$ that matches better with the current frame. Let us assume f is a vector with the feature part of the SURF D^{α} and D^{β} FPFH descriptors. Then, for paired features $\left\{f_k^{(j)}, f_t^{(j)}\right\}$ (KF and current frame, respectively), we compare D_t against each KF descriptor $\hat{D}_k^{KF}$ as follows:

$$\arg\min_{k} \frac{1}{\rho_k} \sum_{j} \mathtt{dist}\left(\hat{f}_k^{(j)}, f_t^{(j)}\right), \tag{12}$$

where `dist` computes the distance between two features and ρ_k is the number of correspondences. After optimization, we set the k-th KF as the best candidate for the current t frame, i.e., $S_b^{KF} = S_k^{KF}$. Finding the best KF is a time-consuming process, but our proposal achieves real-time results by considering the previous estimation. We evaluate first those KFs close to the last estimated pose, and we accept it as the best frame if the number of correspondences is enough (i.e., > 20). This selection reduces considerably the computational cost.

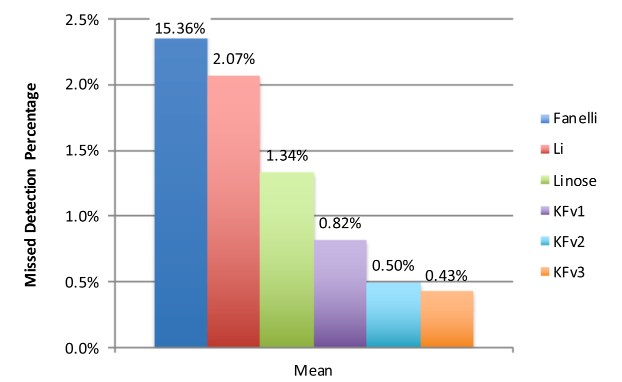

Fig. 9 Mean of the results on the ICT dataset. Percentage of missed detections

Nevertheless, the correspondences between D_t and $\hat{D}_b$ could be inconsistent due to the symmetry of the face (i.e., eyes) or matching between different parts with similar appearance (i.e., mustache and eyebrow). Coherent matches must share similar geometrical characteristics such as distance and orientation in 3D coordinates.

Descriptor filtering Let us assume $\mathbb{M}_{b,t}^{*}$ as the correct match set between D_b and D_t, and $\hat{p}$ as a vector containing the 3D position of both appearance (D^{α}) and shape (D^{β}) descriptors. We compute the mean and variance between the KF points $\hat{p}_b^{*}$ and current frame $\hat{p}_t^{*}$ in terms of distance and orientation, and then, we remove atypical points as follows:

$$\begin{aligned} \mathbb{M}_{b,t} = \{m_{b,t} \in \mathbb{M}_{b,t}^{*} :& \\ Mah(m_{b,t}, \mu_d, \sigma_d) &< th_m \\ Mah(m_{b,t}, \mu_{\theta}, \sigma_{\theta}) &< th_m\}, \end{aligned} \tag{13}$$

where $Mah(\cdot)$ calculates the Mahalanobis distance, $th_m < 1.$ is its associated threshold, and μ_{*} and σ_{*} are the mean and variance, respectively, of (1) Euclidean distance

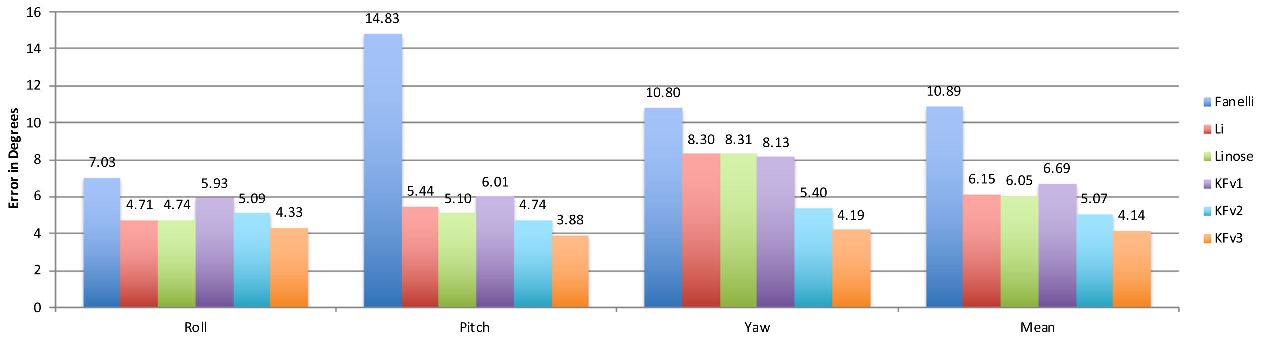

Fig. 10 Mean of the results on the ICT dataset. Mean orientation error (in degrees)

between $\hat{p}_b^*$ and $\hat{p}_t^*$, and (2) orientation of $\hat{p}_b^*$ with respect to $\hat{p}_t^*$.

Initial pose We use the points p of the correspondences $\mathbb{M}_{b,t}$ to compute a rigid transformation from D_b to D_t in order to get an initial head pose P_b. The relative transformation $\hat{\tau}_t = \{\hat{\mathbf{R}}_t, \hat{\mathbf{t}}_t\}$ is estimated by minimizing the cost function:

$$\arg\min_{\hat{\mathbf{R}}_t,\hat{\mathbf{t}}_t} \sum_j \hat{\omega}_j ||\hat{\mathbf{R}}_t \hat{p}_b^{(j)} + \hat{\mathbf{t}}_t - \hat{p}_t^{(j)}||^2, \tag{14}$$

where ω_j is the confidence weight of the matched pair, calculated based on the distance between their corresponding features as follows:

$$\hat{\omega}_j = \exp\left(\frac{-\mathtt{dist}\left(f_k^{(j)}, f_t^{(j)}\right)^2}{\sigma_1}\right).$$

Thus, reliable features contribute more in the estimate of the transformation $\hat{\tau}_t$. This pose is enhanced by considering additional information such as occlusion of current frame. Now, let us assume M_t as the model M after applying this rigid transformation. We improve the pose by aligning now the points p_m of the model M_t with the corresponding p_t points of the current frame, which is done by minimizing the next point-to-plane cost function:

$$\arg\min_{\mathbf{R}_t,\mathbf{t}_t} \sum_j \omega_j \left(\left(\mathbf{R}_t n_m^{(j)}\right)^T \left(\mathbf{R}_t p_m^{(j)} + \mathbf{t}_t - p_t^{(j)}\right)\right)^2, \tag{15}$$

where $n_m^{(j)}$ is the normal surface of point $p_m^{(j)}$. The weight ω_j encodes the affinity between correspondences based on their normals, distance, and orientation with respect to the camera. We formulate it as follows:

$$\omega_j = c_1\omega_j^1 + c_2\omega_j^2 + c_3\omega_j^3,$$
$$s.t.\ c_1 + c_2 + c_3 = 1,$$

where

Table 4 Results on ICT-3DHP dataset in Euler angles

Method	Yaw	Pitch	Roll	Mean
Fanelli et al. [4]*	10.80 ± 7.8	14.83 ± 9.4	7.03 ± 8.5	10.89 ± 8.1
Li et al. [10]*	8.3 ± 8.0	5.44 ± 4.3	4.71 ± 5.2	6.15 ± 5.85
Li Nose*	8.31 ± 7.3	5.10 ± 5.2	4.74 ± 5.4	6.05 ± 5.92
Saeed and Al-Hamadi [27]+	5.1 ± 5.4	4.9 ± 5.3	4.4 ± 4.6	4.8 ± 5.1
Baltrušaitis et al. [29]+	6.9	7.06	10.48	8.15
Venturelli et al. [26]+	9.8 ± 10.1	4.5 ± 4.6	4.4 ± 4.5	6.23 ± 6.4
KFv1	8.13 ± 8.7	6.01 ± 5.8	5.93 ± 5.2	6.69 ± 6.49
KFv2	5.40 ± 6.4	4.74 ± 4.1	5.09 ± 5.7	5.07 ± 5.44
KFv3	**4.19** ± 4.8	**3.88** ± 4.2	**4.33** ± 4.9	**4.14** ± 4.47

Results with the best performances are presented in bold
*Estimation that we calculated
+Results taken from author's papers

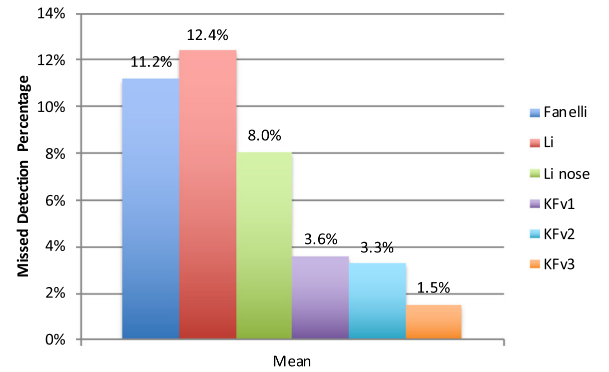

Fig. 11 Mean of the results on the ICU dataset. Percentage of missed detections

$$\omega_j^1 = \exp\left(\frac{\text{ang}\left(n_m^{(j)}, n_t^{(j)}\right)}{\sigma_{a1}}\right), \qquad (16)$$

$$\omega_j^2 = \exp\left(\frac{\text{ang}\left(n_m^{(j)}, o_j\right)}{\sigma_{a2}}\right), \qquad (17)$$

$$\omega_j^3 = \exp\left(\frac{L_2\left(p_m^{(j)}, p_t^{(j)}\right)}{\sigma_{a3}}\right), \qquad (18)$$

$$\text{ang}(a_1, a_2) = \text{acos}\left(\frac{a_1 \cdot a_2}{||a_1||\ ||a_2||}\right). \qquad (19)$$

Equation 16 measures the angle between the normals of points $p_m^{(j)}$ and $p_t^{(j)}$, respectively. Equation 17 considers that the model itself could occlude some correspondences, which happens when the normal of the point $p_m^{(j)}$ and its orientation with respect to the camera (i.e., a normal vector centered at $p_m^{(j)}$ pointing to the camera) o_j have a large angle. Finally, Eq. 18 weights the correspondences according to their distances.

Sometimes it is not possible to find a suitable KF for a given frame, i.e., the number of matches is not enough. In this case, we use the last KF-based estimation as a temporal KF, and thus, we continue the pose estimation without interruptions.

We optimize Eqs. 10, 14, and 15 through an ICP scheme with classic termination criteria, i.e., maximum number of iteration (10) and mean square error in terms of translation and rotation. Thus, we obtain the final pose $P_t = \{p_t, \theta_t\}$, which corresponds to the nose tip and orientation, respectively, of the model after the transformation $\tau_t = \{\mathbf{R}_t, \mathbf{t}_t\}$.

3.3.2 Key frame updating

Our system does not require to learn all the 50 discretized orientation in order to be launched, but it benefits the more KFs there are. Therefore, the online system begins when it has 20 frames, and then, new KFs could be added from the current estimates of our proposal. This is done by checking the current estimated pose P_t; if the orientation θ_t does not have a KF associated in discretized space, we include it in the set following the considerations of Section 3.2.2. Otherwise, we compare the fitness score of current frame with the closest KF. The score checks the average distance between the model and point cloud, the number of descriptors, and the feature distance, and we keep the one with more descriptors and smaller distance. The optimization described in Section 3.2.3 is carried out when enough KFs have been added or modified, i.e., 5 frames. Since this operation is performed in parallel and only when necessary, no additional time is added to the online estimate.

4 Experimental evaluations

We evaluate our KF-based proposal, Fanelli's method, and Li' approach with the variant of the 3D nose feature, see Section 3.2.1, on two public benchmarks: ICT-3DHP dataset [29] and BIWI Kinect Head Pose Database [4]. Also, we create a more realistic dataset with complex behaviors that challenge these pose estimation frameworks.

4.1 Datasets

The BIWI Kinect Head Pose Database [4] is a baseline for evaluating HPE algorithms. It consists of 24 sequences with 20 persons of different gender, age, and facial characteristics. It has over 15K RGB-D images aiming to frame-by-frame detection and not tracking because there are many sequences with some missed frames. Each sequence has a single target rotating his/her head, with a range of ± 75 and ± 60° for yaw and pitch, respectively, slowly with a neutral expression. Head pose annotations are estimated using a tracking system.

The ICT-3DHP Dataset is proposed in [29]. It is divided into 10 sequences containing about 14K RGB-D frames with both color and depth images. The targets perform a similar head motion as in BIWI dataset, but some targets present facial expressions, self-occlusion (e.g., hair), and small change of position. The ground truth is generated through a *Polhemus FASTRAK* flock of birds tracker, which is a commercial system that estimates head pose from sensors located over a white sport cap.

Our ICU-Head Pose Dataset consists of 4 sequences each with a unique person, see Fig. 5. The targets have different facial morphology and features, i.e., glasses, mustaches, or beards. The sequences are created to test the performance of HPE algorithms under challenging scenarios. Therefore, targets perform complex behaviors including change of head position, large range head orientation, self-occlusions, fast motion, and facial deformation.

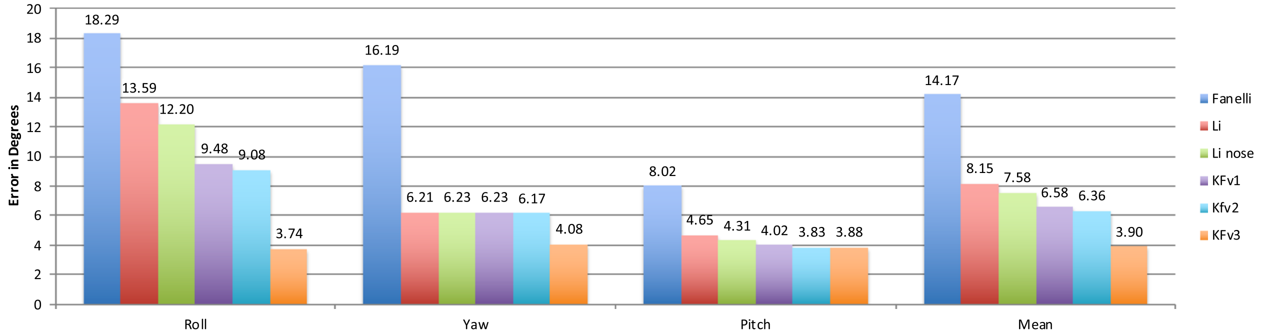

Fig. 12 Mean of the results on the ICU dataset. Mean orientation error (in degrees)

We collect the sequences with a Microsoft Kinect v1 under controlled conditions with a resolution of 640 × 480. The ground truth is automatically annotated through a commercial motion caption (MoCap) system with a total of 6 marks (reflective spheres) fixed over a bicycle helmet using metallic bars of 10cm, see Fig. 6. The MoCap detects these markers as a rigid object and estimates the location and orientation of the helmet and therefore the target's head, with high precision.

Each target performs a different set of behaviors with unique characteristics such as speed. A summary of the sequences is presented in Table 1. The details of each sequence are as follows: In Seq1, the target performs simple actions at slow speed. It presents small range over the head orientation with a complexity similar to the public BIWI and ICT-3DHP datasets. We rate Seq2 as medium difficulty because it presents a large orientation range and fast motions. Also, the target changes its head position several times, approaching and moving away to the camera. Seq3 and Seq4 are the most challenging of the whole set. In Seq3, the target performs extreme head orientation and multiple self-occlusion. Finally, Seq4 depicts fast head movements in orientation and position. Throughout the article, we show several examples using our dataset.

4.2 Evaluation criteria

We evaluate the performance of the HPE algorithms through standard metrics such as missed detection, Euler angle error (roll, pitch, and yaw), and mean angular error. A head pose is labeled as missed detection whether the estimation algorithm does not converge to a solution, according to the termination criteria, or the proposed pose has an error of more than 45°. We learn the KFs for each sequence using the system described in Section 3.2.2, and those frames are not considered in the evaluation step.

Table 5 Results on ICU dataset in Euler angles

Method	Yaw	Pitch	Roll	Mean
Fanelli et al. [4]*	16.19 ± 8.5	8.02 ± 4.6	18.29 ± 12.1	14.17 ± 8.4
Li et al. [10]*	6.21 ± 6.9	4.65 ± 2.6	13.59 ± 6.7	8.15 ± 5.4
Li Nose*	6.23 ± 6.6	4.31 ± 2.1	12.20 ± 4.8	7.58 ± 4.5
KFv1	6.23 ± 6.1	4.02 ± 2.66	9.48 ± 8.47	6.58 ± 6.2
Kfv2	6.17 ± 5.5	**3.83** ± 1.9	9.08 ± 4.1	6.36 ± 3.8
KFv3	**4.08 ± 4.6**	3.88 ± 2.1	**3.74 ± 3.6**	**3.90 ± 3.5**

Results with the best performances are presented in bold
*Estimation that we calculated

We evaluate and report the results of three proposals: (1) Fanelli's method [4], using the open source code; (2) an implementation of Li's proposal [10]; and (3) *Li Nose* that includes our nose-based feature in the approach of Li. We analyze different parts of our proposal separately creating three variants; an overview is shown in Table 2. Recall that **KFv1** is our a basic version, published in [43], which only uses the SURF descriptors.

We only report the results with respect to the orientation because an incorrect position estimate is reflected in the orientation error as well.

4.3 Results

First, we analyze the BIWI dataset, Fig. 7 reports the mean error in all sequences per proposal, and Fig. 8 shows the missed detection percentage. The mean error in the Li-based approaches (red and green columns) is almost the same, but the number of missed detection has decreased substantially when we incorporate the nose feature (green column). The last three columns (purple and cyan) depict the results of our proposal. The performance, in both accuracy and detection rate, is improved after we apply the optimization process over the KFs. Also, Table 3 reports the results and compares them against other methods in the state of the art. Our proposal has the best accuracy in terms of pitch and yaw; meanwhile, Venturelli et al.'s approach [26] has a similar performance for roll. Nevertheless, the variance of our KFv3 proposal is smaller in all cases, making this approach more stable.

Similarly, we evaluate the proposals with the ICT-3DHP dataset and we show the results in Figs. 9 and 10. The mean error is almost the same for both Li's approach and KFv1 proposal, but we can observe that the optimized approach KFv3 is more accurate with a missed detection rate of less than 0.5%. We compare our results with other approaches in Table 4. KFv3 method gives the best results, with a smaller variance of all the techniques, meaning it is more stable.

Figures 11 and 12 and Table 5 show the results using our dataset. Fanelli's approach has the biggest error, Li-based proposals have a similar mean error around 8°, and KF-based approaches have the smallest error. By observing Fig. 11, we point out a great improvement with respect to missed detection because our KF-based approach handles better fast motions and occlusions. In Fig. 13, we see a qualitative example for Seq3 where, for a given frame, we estimate the pose depicted with a blue line and the 3D template model in green. We observe that Fanelli and Li have limitations detecting the pose; meanwhile, our approach can detect a sufficient part of the face to infer a correct pose.

From all the results, we observe that Fanelli's approach has the bigger error in most of the cases. This is because it is difficult to find the point of the nose when the face is in full profile, which makes the nose barely distinguishable. A better training could improve this aspect, but that requires more pre-processing.

In general, Li's basic approach has a better performance than Fanelli's, but in our dataset, Li's proposal has problem detecting the pose. Figure 14 shows more detailed results of each sequence. We can observe how sequences 1 and 2 have a performance similar to those of the previous public datasets; nonetheless, in sequences 3 and 4, the missed detection rate of Li is higher than the rest. These sequences present fast motion with both targets wearing glasses; therefore, the images are blurred, and in some occasions, the light is reflected in the glasses. This makes it difficult for the 2D face landmark detector to find the eyes, forcing Li's proposal to use ICP without any additional information. If we compare the red and green column, we observe an improvement, meaning that the addition of the 3D nose feature overcomes the aforementioned problems.

In Figs. 15 and 16, we analyze the results in terms of missed detection. These figures are 2D histograms of the discretized orientation for pitch and yaw. When a frame is labeled as missed detection, we use the ground truth and increase a counter of the corresponding pose. The histograms are normalized considering the number of frames, so each cell (for a specific orientation) depicts the percentage of missed detections. The histogram center, highlighted with the green and blue arrows, represents a target in frontal view (looking to the camera). Following over x-axis means the head is moving from left to right (blue arrow) or from up to down with the y-axis. In Fig. 15, we report the results of sequence 14 of BIWI dataset where we can observe how Fanelli's proposal (left image) cannot detect well a pose at full profile. In other words, it has problems to handle a target looking up on

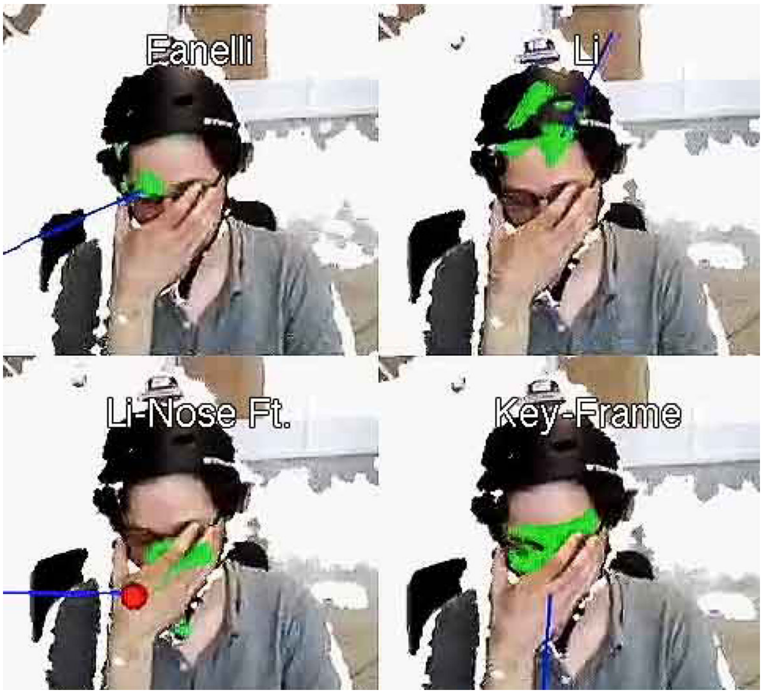

Fig. 13 Qualitative results. Head pose estimated by 4 different proposals using the same frame

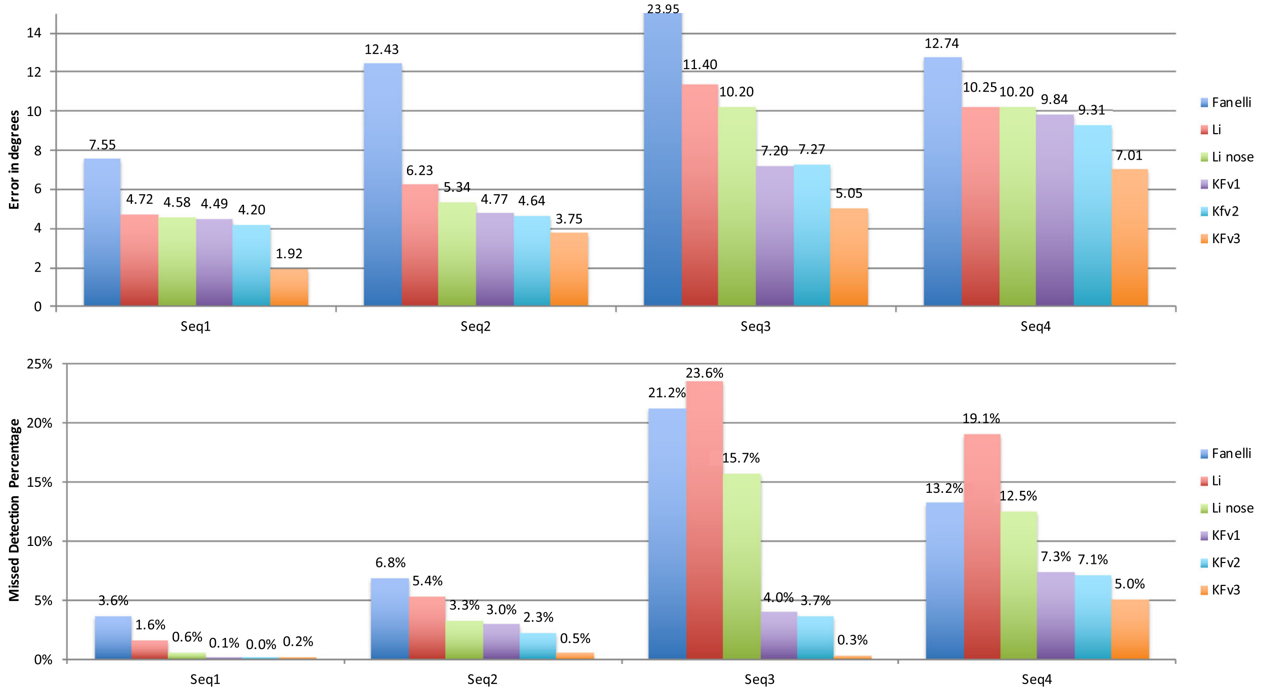

Fig. 14 Results of ICU sequences using (blue) Fanelli, (red) Li's simple approach, (green) Li's proposal including of nose detection heuristic, and (purple) our descriptor-based method without and with optimization (cyan). The first row shows the mean angular error, and the second the missed detection. For clarity, each graphic only shows two sequences

the right. The rest of the proposals (Li, Li with nose feature, and KFv3) perform well in this sequence. Figure 16 shows other cases but with BIWI dataset using sequence 24. Both approaches based on Li (first two images at the left) do not detect well the head when it is looking a little to the upper right corner. The third figure shows the results with our KF-based method without optimization (KFv1). Most of the undetected frames happen when the target is looking upward. On the contrary, this does not happen with the KFv3 because it improves the detection rate in that orientation.

The previous results show how our approach improves the HPE performance under challenging scenarios. In some cases, other proposals provide a little more accurate result, but in all cases, the KF-based approach is more stable, and it does not require a specific architecture (i.e., GPUs) with a reasonable computation time. This makes the approach more reliable and robust.

4.4 Discussion

Our learning step uses the output of two state-of-the-art HPE methods, e.g., Fanelli and Li, but several proposals in Tables 3 and 4 outperform them. The intuitive question is why we privilege those instead of more accurate proposals. This can be answered by observing Table 6 that summarizes some features of the most relevant approaches. The proposals of Ahn, Saeed, and Venturelli [24, 26, 27] are more accurate

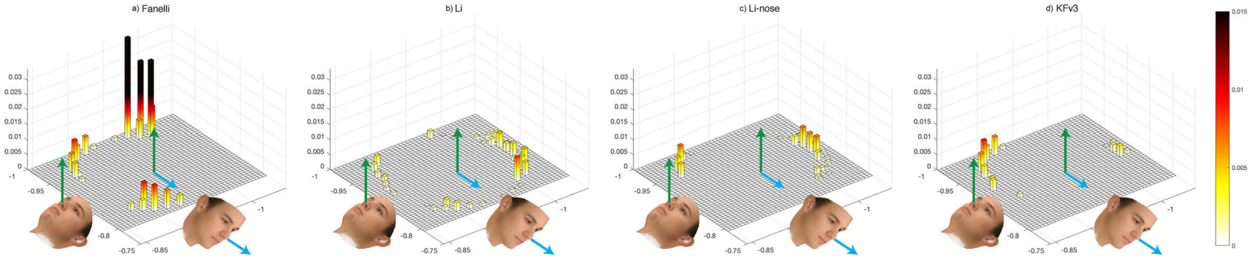

Fig. 15 2D histogram of the missed detection distribution on sequence BIWI 14. Results using **a** Fanelli, **b** Li's approach, **c** Li's proposal including of nose detection heuristic, and **d** our descriptor-based method, KFv3

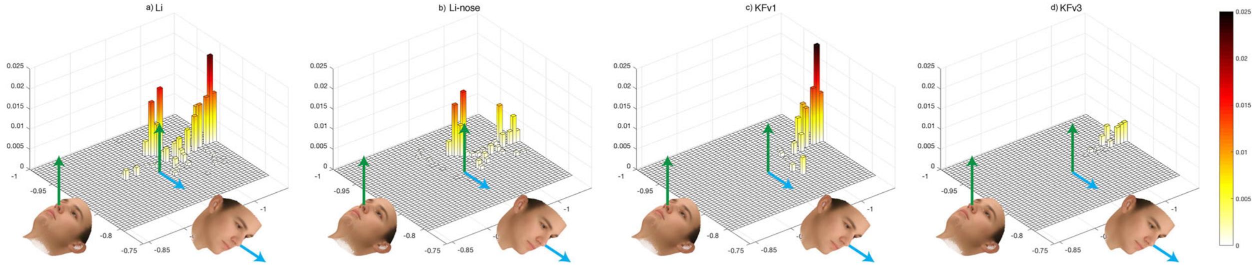

Fig. 16 2D histogram of the missed detection distribution on sequence BIWI 24. Results using **a** Li's approach, **b** Li's proposal including of nose detection heuristic, and **c** our KF-based method without optimization (KFv1) and **d** our KFv3 proposals

and faster, but they require the use a GPU card. This makes them more expensive and complex to use in embedded systems. The other proposals, e.g., [9, 12, 28], require more computational time with high variance in their estimate, i.e., [12] has a variance of 16 and 9.6° for yaw and pitch, respectively.

Our proposal has a computational cost of ≈ 10 fps, which is reasonable for most applications. One characteristic is that most of our proposal is highly parallelized, so we can improve calculation times if necessary.

When comparing the results of each dataset, we observe that in the simplest sequences, our proposal obtains results with equivalent precision. Also, the results with the most complex sequences (i.e., ICU dataset) show that our proposal has a better performance both in accuracy and missed detection percentage.

If we compare the three versions of KF, we observe how the versions with global optimization (KFv2 and KFv3), described in Section 3.2.3, improve the stability of the performance in comparison with the KFv1. The accuracy is further improved in KFv3 by including (1) the descriptor distance as weighting factors in the optimization process and (2) an adaptive model to the target's face.

We give a qualitative evaluation of the tested methods in Table 7, based on our personal experience. Here, we grade them according to our impression in each aspect as follows: (+) low, (++) good, and (+++) excellent.

As shown in the first row, Li does not handle well fast motions. In this case, blurry images affect directly two appearance-based aspects of the proposal: the 2D (eye) landmark detector and the color-based *k*-means, which remove no-face correspondences of the ICP algorithm. This makes it unstable in fast situation, and therefore, it gives a low detection rate. On the other hand, it can detect poses in a wide range of orientations with a good precision.

Li's proposal improves when more features are available. The inclusion of the nose feature enhances the accuracy of the estimations and reduces the missed detection rate. This is because the 3D feature is based on depth information, which is not much affected by blurry images. In general, the orientation range and accuracy are better than the classic implementation but still need more improvement.

Fanelli's approach deals better with fast motions because depth information is not distorted by movement. In contrast, it has a more restricted detection range because the nose tip, the key element of Fanelli's method, is undistinguished at images of full profile. In other words, there is not enough evidence to distinguish the nose tip from the edge of the face. The rest of the time, it has no problem detecting a pose in short time and this is why Li used this method to initialize its proposal. Nevertheless, the accuracy of the results is low.

In most of the fast motions, our proposal could find enough features to estimate the pose. Also, it estimates the pose even with targets at full profile (i.e., looking to the left of right) with an excellent orientation range. From these two aspects, it has less problems detecting the target most of the time with competitive results to those in the state of the art.

From the results, we observe how the use of KF-based approach improves the estimation, and those are

Table 6 Evaluation of computational cost of each pose proposals

Method	Time (ms per frame)	Architecture
Kazemi and Sullivan [6]*	12.0 ± 1	CPU
Fanelli et al. [4]*	20.1 ± 2	CPU
Li et al. [10]*	62.4 ± 5	CPU
Li Nose*	64.2 ± 5	CPU
KFv3*	89.9 ± 10	CPU
Papazov et al. [12]+	122	CPU
Yang et al. [28]+	∼ 100	CPU
Yu et al. [9]+	∼ 250	CPU
Ahn et al. [24]+	0.98	GPU
Venturelli et al. [26]+	10	GPU
Saeed and Al-Hamadi [27]+	> 45	GPU

Results with the best performances are presented in bold
*Time that we calculated
+Results taken from author's papers

Table 7 Evaluation summary of each head pose estimator

Method	Fast motion	Orient. range	Detection rate	Precision
Li et al. [10]	+	++	+	++
Li Nose	++	++	++	++
Fanelli et al. [4]	+++	+	+++	+
KFv3	++	+++	+++	++

Results with the best performances are presented in bold

enhanced by applying the global optimization process. The inclusion of the descriptor weights (KFv3) helps to estimate more robustly the pose because it reduces the importance of weak correspondences, which may not be a good match (great distance between descriptors), and prioritizes strong matches.

5 Conclusion and future work

This paper has presented a framework for HPE based on key frames, which includes information of appearance, shape head pose hypothesis. This includes an original offline learning proposal consisting of two stages: (1) an automatic KF learning step and (2) an original post-processing step that minimize globally the error between KFs and the 3D face model, enhancing the accuracy and consistency of the KF set. We evaluated this person-specific approach in two public benchmarks, and we have shown that the use of the KF provides robust estimates for a wide range of orientations in reasonable time. Also, we presented a more challenging dataset with complex behaviors that includes self-occlusions, fast motion, change of the head position, and extreme head orientation. The results in this dataset showed that our approach can estimate a pose even in complex situations, contrarily to other approaches. At the same time, we have shown that our proposal is more stable than others and with a gain in precision as the complexity of the datasets increases.

We have compared against several works and considered classic benchmark datasets. Regarding the benchmarked datasets, the results have shown how the KF-based approach, learned from weaker estimation algorithms, provides good performance and how those are enhanced after optimization. Furthermore, our approach maintains a competitive CPU cost with respect to other applications.

A natural investigation track is to relax the offline stage (to leave a mostly online system) by learning only a couple of KFs of the target, with neutral pose and looking into the camera direction. Then, we perform our pose estimation algorithm where we learn more KFs as soon as new estimates are available. The set of KF is updated as described in Section 3.3.2.

Abbreviations

3DMFM: 3D morphable face model; BFM: Basel face model; CNN: Convolutional neural network; FPFH: Fast Point Feature Histogram; GPU: Graphic processing units; HoG: Histogram of gradients; HPE: Head pose estimation; ICP: Iterative closest point; KF: Key frame; SoA: State of the art; SURF: Speeded Up Robust Feature; SVM: Support vector machine

Acknowledgments
This work was carried at LAAS-CNRS and supported by the project CORAC SEFA IKKY ("Integration KockKpit et de ses sYstèmes").

Authors' contributions

All authors participated in the work described in this paper and have read and approved the final manuscript.

Funding
Project CORAC SEFA IKKY ("Integration KockKpit et de ses sYstèmes").

Availability of data and materials

Please contact author for data requests.

Ethics approval and consent to participate

Approved.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 26 June 2019 Accepted: 31 December 2019
Published online: 23 March 2020

References

1. S. Sheikhi, J.-M. Odobez, Combining dynamic head pose–gaze mapping with the robot conversational state for attention recognition in human–robot interactions. Pattern Recog. Lett. **66**, 81–90 (2015). Pattern Recognition in Human Computer Interaction
2. E. Murphy-Chutorian, M. M. Trivedi, Head pose estimation and augmented reality tracking: an integrated system and evaluation for monitoring driver awareness. IEEE Trans. Intell. Transport. Syst. **11**(2), 300–11 (2010)
3. E. Marchand, H. Uchiyama, F. Spindler, Pose estimation for augmented reality: a hands-on survey. IEEE Trans. Visual. Comput. Graphics. **22**(12), 2633–2651 (2016)
4. G. Fanelli, M. Dantone, J. Gall, A. Fossati, L. Van Gool, Random forests for real time 3D face analysis. Int. J. Comput. Vision, 437–458 (2013). https://doi.org/10.1007/s11263-012-0549-0
5. A. Tawari, S. Martin, M. M. Trivedi, Continuous head movement estimator for driver assistance: issues, algorithms, and on-road evaluations. IEEE Trans. Intell. Transport. Syst. **15**(2), 818–830 (2014)
6. V. Kazemi, J. Sullivan, in *Conf. Comput. Vision Pattern Recogn.* One millisecond face alignment with an ensemble of regression trees, (2014), pp. 1867–1874. https://doi.org/10.1109/cvpr.2014.241
7. J. M. Barros, B. Mirbach, F. Garcia, K. Varanasi, D. Stricker, in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Fusion of keypoint tracking and facial landmark detection for real-time head pose estimation, vol. 00, (2018), pp. 2028–2037. https://doi.org/10.1109/WACV.2018.00224
8. R. Valenti, N. Sebe, T. Gevers, Combining head pose and eye location information for gaze estimation. IEEE Trans. Image Process. **21**(2), 802–815 (2012)
9. Y. Yu, K. F. Mora, J. M. Odobez, Headfusion: 360° head pose tracking combining 3D morphable model and 3D reconstruction. IEEE Trans. Pattern Anal. Mach. Intell., 1–1 (2018). https://doi.org/10.1109/TPAMI.2018.2841403
10. S. Li, K. N. Ngan, R. Paramesran, L. Sheng, Real-time head pose tracking with online face template reconstruction. IEEE Trans. Pattern Anal. Mach. Intell. **38**(9), 1922–1928 (2016)
11. R. S. Ghiass, O. Arandjelović, D. Laurendeau, in *Proceedings of the 2Nd Workshop on Computational Models of Social Interactions: Human-Computer-Media Communication. HCMC '15*. Highly accurate and fully automatic head pose estimation from a low quality consumer-level rgb-d sensor (ACM, New York, 2015), pp. 25–34
12. C. Papazov, T. K. Marks, M. Jones, in *Conf. on Computer Vision and Pattern Recognition*. Real-time 3d head pose and facial landmark estimation from

depth images using triangular surface patch features, (2015). https://doi.org/10.1109/cvpr.2015.7299104

13. E. Murphy-Chutorian, M. M. Trivedi, Head pose estimation in computer vision: a survey. Trans. Pattern Anal. Mach. Intell. **31**(4), 607–626 (2009)
14. B. Czupryński, A. Strupczewski, in *Active Media Technology: 10th International Conference, AMT 2014, Warsaw, Poland, August 11-14, 2014. Proceedings*, ed. by D. Ślezak, G. Schaefer, S. T. Vuong, and Y.-S. Kim. High accuracy head pose tracking survey (Springer, 2014), pp. 407–420. https://doi.org/10.1007/978-3-319-09912-5_34
15. J. M. Saragih, S. Lucey, J. F. Cohn, Deformable model fitting by regularized landmark mean-shift. Int. J. Comput. Vis. **91**(2), 200–215 (2011)
16. M. Zhou, L. Liang, J. Sun, Y. Wang, in *Computer Society Conference on Computer Vision and Pattern Recognition*. AAM based face tracking with temporal matching and face segmentation, (2010), pp. 701–708. https://doi.org/10.1109/cvpr.2010.5540146
17. V. Drouard, S. Ba, G. Evangelidis, A. Deleforge, R. Horaud, in *2015 IEEE International Conference on Image Processing (ICIP)*. Head pose estimation via probabilistic high-dimensional regression, (2015), pp. 4624–4628. https://doi.org/10.1109/icip.2015.7351683
18. J. Chen, J. Wu, K. Richter, J. Konrad, P. Ishwar, in *2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. Estimating head pose orientation using extremely low resolution images, (2016), pp. 65–68. https://doi.org/10.1109/ssiai.2016.7459176
19. Z. Dong, G. Zhang, J. Jia, H. Bao, Efficient keyframe-based real-time camera tracking. Comput. Vis. Image Underst. **118**(Supplement C), 97–110 (2014)
20. L. Vacchetti, V. Lepetit, P. Fua, Stable real-time 3d tracking using online and offline information. Trans. Pattern Anal. Mach. Intell. **26**(10), 1385–1391 (2004)
21. K. Kim, V. Lepetit, W. Woo, in *2010 IEEE International Symposium on Mixed and Augmented Reality*. Keyframe-based modeling and tracking of multiple 3d objects, (2010), pp. 193–198. https://doi.org/10.1109/ismar.2010.5643569
22. L.-P. Morency, J. Whitehill, J. Movellan, Monocular head pose estimation using generalized adaptive view-based appearance model. Image Vision Comput. **28**(5), 754–761 (2010). Best of Automatic Face and Gesture Recognition 2008
23. L. P. Morency, P. Sundberg, T. Darrell, in *2003 IEEE International SOI Conference. Proceedings (Cat. No.03CH37443)*. Pose estimation using 3D view-based eigenspaces, (2003), pp. 45–52. https://doi.org/10.1109/amfg.2003.1240823
24. B. Ahn, J. Park, I. S. Kweon, in *Computer Vision – ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part III*, ed. by D. Cremers, I. Reid, H. Saito, and M.-H. Yang. Real-time head orientation from a monocular camera using deep neural network (Springer, Cham, 2015), pp. 82–96
25. M. D. Breitenstein, D. Kuettel, T. Weise, L. van Gool, H. Pfister, in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. Real-time face pose estimation from single range images, (2008), pp. 1–8. https://doi.org/10.1109/cvpr.2008.4587807
26. M. Venturelli, G. Borghi, R. Vezzani, R. Cucchiara, in *VISIGRAPP*. From depth data to head pose estimation: a Siamese approach, (2017). https://doi.org/10.5220/0006104501940201
27. A. Saeed, A. Al-Hamadi, in *2015 IEEE International Conference on Image Processing (ICIP)*. Boosted human head pose estimation using kinect camera, (2015), pp. 1752–1756. https://doi.org/10.1109/icip.2015.7351101
28. J. Yang, W. Liang, Y. Jia, in *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. Face pose estimation with combined 2D and 3D hog features, (2012), pp. 2492–2495. https://ieeexplore.ieee.org/abstract/document/6460673
29. T. Baltrušaitis, P. Robinson, L. P. Morency, in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 3D constrained local model for rigid and non-rigid facial tracking, (2012), pp. 2610–2617. https://doi.org/10.1109/cvpr.2012.6247980
30. A. Strupczewski, B. Czupryński, W. Skarbek, M. Kowalski, J. Naruniec, Head pose tracking from rgbd sensor based on direct motion estimation. Procs. Int. Conf. Pattern Recog. Mach. Intell., 202–212 (2015)
31. N. Smolyanskiy, C. Huitema, L. Liang, S. E. Anderson, Real-time 3D face tracking based on active appearance model constrained by depth data. Image Vis. Comput. **32**(11), 860–869 (2014)
32. S. Kaymak, I. Patras, in *Asian Conference on Computer Vision*. Exploiting depth and intensity information for head pose estimation with random forests and tensor models, (2012), pp. 160–170. https://doi.org/10.1007/978-3-642-37484-5_14
33. P. Viola, M. J. Jones, Robust real-time face detection. Intl. J. Comput. Vision. **57**(2), 137–54 (2004)
34. C. Yan, L. Li, C. Zhang, B. Liu, Y. Zhang, Q. Dai, Cross-modality bridging and knowledge transferring for image understanding. IEEE Trans. Multimed. **21**(10), 2675–85 (2019). https://doi.org/10.1109/TMM.2019.2903448
35. N. Alioua, A. Amine, A. Rogozan, A. Bensrhair, M. Rziza, Driver head pose estimation using efficient descriptor fusion. EURASIP J. Image Video Process. (2016). https://doi.org/10.1186/s13640-016-0103-z
36. C. Yan, Y. Tu, X. Wang, Y. Zhang, X. Hao, Y. Zhang, Q. Dai, Stat: spatial-temporal attention mechanism for video captioning. IEEE Trans. Multimed., 1–1 (2019). https://doi.org/10.1109/TMM.2019.2924576
37. C. Yan, Y. Zhang, J. Xu, F. Dai, J. Zhang, Q. Dai, F. Wu, Efficient parallel framework for hevc motion estimation on many-core processors. IEEE Trans. Circ. Syst. Video Technol. **24**(12), 2077–2089 (2014). https://doi.org/10.1109/TCSVT.2014.2335852
38. R. B. Rusu, N. Blodow, M. Beetz, in *2009 IEEE International Conference on Robotics and Automation*. Fast point feature histograms (FPFH) for 3D registration, (2009), pp. 3212–3217. https://doi.org/10.1109/ROBOT.2009.5152473
39. C. Yan, H. Xie, J. Chen, Z. Zha, X. Hao, Y. Zhang, Q. Dai, A fast uyghur text detector for complex background images. IEEE Trans. Multimed. **20**(12), 3389–3398 (2018). https://doi.org/10.1109/TMM.2018.2838320
40. P. Paysan, R. Knothe, B. Amberg, S. Romdhani, T. Vetter, in *Int. Conf. on Advanced Video and Signal Based Surveillance*. A 3D face model for pose and illumination invariant face recognition, (2009), pp. 296–301. https://doi.org/10.1109/avss.2009.58
41. D. E. King, Dlib-ml: a machine learning toolkit. J. Mach. Learn. Res. **10**, 1755–1758 (2009)
42. I. Amidror, Scattered data interpolation methods for electronic imaging systems: a survey. J. Electron. Imaging. **11**(2), 157–176 (2002)
43. F. Madrigal, F. Lerasle, A. Monin, in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 3D head pose estimation enhanced through surf-based key-frames, (2018), pp. 75–83. https://doi.org/10.1109/WACV.2018.00015

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.