3D Head Pose Estimation enhanced through SURF-based Key-Frames

Jorge Francisco Madrigal Diaz, Frédéric Lerasle, André Monin

2018 IEEE Winter Conference on Applications of Computer Vision

3D Head Pose Estimation enhanced through SURF-based Key-Frames

Francisco Madrigal[1], Frederic Lerasle[1,2], Andre Monin[1]
[1]CNRS, LAAS, 7 avenue du Colonel Roche, F-31400 Toulouse, France
[2]Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France
{jfmadrig,lerasle,monin}@laas.fr

Abstract

This work presents a method that incorporates 2D and 3D cues for the estimation of head pose. We propose the use of the concept of Key-Frames (KF), a set of frames where the position and orientation of the head is automatically calculated off-line, to improve the precision of pose estimation and detection rate. Each KF consists of: 2D information, encoded by SURF descriptors; 3D information from a depth image (both acquired by an RGB-D sensor); and a generic 3D model that corresponds to the head localization and orientation in the real world. Our algorithm compares a new frame against all KFs and selects the most relevant one. The 3D transformation between both, selected KF and current frame, can be estimated using the depth image and the Iterative Closest Point algorithm in an online framework. Compared to reference approaches, our system can handle partial occlusions and extreme rotations even with noisy depth data. We evaluate the proposal using two challenging datasets: (1) an dataset acquired by us where the ground-truth information is given by a commercial Motion Capture system and (2) the public benchmark Biwi Kinect Head Pose Database.

1. Introduction

Head Pose Estimation (HPE) has been a hot topic in recent years. The techniques developed for HPE have found applications in different areas such as driver assistance[17], human-machine interaction [14], expression recognition [3], augmented reality [10], among others. This task of detecting the head and estimating its orientation with a non-invasive system seems simple. Many today gadgets, such as smart phones or webcams, can detect faces in 2D images in real-time. However, multiple systems require that the head pose estimators provide higher quality results, good detection rate and high 3D accuracy. In this paper, we propose a new methodology (Fig. 1) to detect, robustly, the human head pose and to improve the accuracy of the estimations even with large head rotations. Our monocular-based proposal is aimed for driver assistance applications, focusing on improving both accuracy and detection rate. In this applicative context, a driver may exhibit erratic and abrupt rapid movements that a head pose system must handle, as missed detections or inaccurate estimates could lead to accidents. Therefore, we focus on increasing the estimation quality in terms of accuracy and robustness for a wide range of orientations. This is feasible by including a fast and non-invasive offline step that learns driver's appearance for some poses. This information is used to estimate online an accurate pose for a new frame. The appearance cue has been used in several works, [6, 13, 19] to estimate the head orientation by searching facial features, such as eyes, eyebrows, mount or nose. Those provide high accurate results, in real-time, for targets facing straight to the camera or with small head rotations. Recent works [3, 1] use depth information to estimate 3D facial features, allowing to detect poses with a wide range of head orientations. Other proposals [8, 15, 6] combine appearance and depth information to overcome the limitations of single cues. Some others [8, 4, 1] take advantage of a predefined 3D human face model which provides accurate results. Our work falls into these last two categories, we extract RGB features that describe target appearance and we combine them with depth information into a face model based system.

For such propose, we develop a framework that combines the best features of existing approaches into an original concept of Key-Frames (KF). Each KF contains: (1) a 3D human-face model describing the pose and position of the target's head for a relevant orientation; (2) a set of SURF descriptors, and (3) the depth image, which allows to know the 3D position of each 2D descriptor. These are learned in an offline step, see Fig. 1. In the online step, the input frame is compared to all keyframes, using the SURF descriptors. The best match is used to estimate, quickly and accurately, the head posture. The results are then re-

† This work was carried out in LAAS-CNRS and supported by Thales Group under the IKKY project.

fined using an algorithm based on Iterative Closest Point (ICP). Fig. 1 gives an overview of our proposal. The evaluations are done using: (i) the standard benchmark Biwi Kinect Head Pose Database [3] and (ii) our own dataset recorded with a Microsoft Kinect v1. BIWI dataset is, in the literature, the standard for evaluating head pose detectors [3, 4, 8, 11]. Each target is recorded with neutral expression, rotating the head at a slow-medium speed. This is aimed to frame-by-frame detection and not tracking. Inspired by this benchmark, we develop our own dataset but, in comparison with BIWI, the targets perform more natural movements as those expected in a real scenarios, *i.e*. drivers in a car. It consists of 4 sequences where targets show complex behaviors, such as: rapid head movements, self-occlusion, facial expression, among others. Our ground truth is created through a commercial Motion Caption (MoCap) system that uses passive markers, located in a helmet wear by the targeted person, to provide the pose and position of the head. Thanks to quantitative evaluations of challenging sequences, we highlight that our monocular RGB-D based approach outperforms current approaches in the state of the art. This paper is organized as follows: Section 2 discusses related work. The formulation of our proposed pose detector is presented in Section 3. In Section 4 we present both quantitative and qualitative evaluations and a discussion, comparing our proposal with respect to other two state-of-the-art approaches. Finally, conclusions and future work are presented in Section 6.

2. Related work

There have been many works aimed on the monocular-based system for head pose estimation [9, 2]. These can be categorized according to the cues used. Here, we mention some of the most relevant monocular proposals.

RGB-based approaches Multiple approaches make use of deformable models to approximate the shape of a human face [13] considering facial expressions. Some methods track the face in video sequences using the classic active-look model (AAM) such as Zhou *et al*. [21]. This work incorporates temporal matching constrains that enforce the inter-frame coherence in a fitting (cost) function. Kazemi and Sullivan [6] propose a random forest-based framework that performs face alignment quickly, achieving a detection speed of 1 ms. This proposal constructs a set of regression trees. Each is learned by a loss function, based on gradient boosting, with invariant feature selection. This state-of-the-art method can detect faces with high accuracy even with strong facial deformations or small head rotations. Those methods are more focused on face detection, although the pose can be inferred once the model fitting is done. Some other methods rely on the use of 2D information to detect specific facial features, *i.e*. eyes, nose. Valenti *et al*. [19] use the RGB images to detect simultaneously the head pose and eyes location. This method assumes that the head follows a cylindrical shape, the face features are learned from the images and then projected onto a cylinder. Then, the model can be used to detect and track that specific person on the scene. The previous estimation of both eye and pose are used as feedback between them to improve performance. However, 2D-based techniques are very sensitive to the lack of features, (self) occlusions, illumination changes or limited head rotation.

Depth-based approaches The depth cue is used in most of today state-of-the-art methods. Breitenstein *et al*. [1] propose a framework for head pose estimation that processes depth images in real time using Graphics Processing Units (GPUs). First, in an offline step, the proposal calculates a generic 3D model of a human face. This model is rotated, and stored, with different orientations. Then, the GPU finds the best match between the stored models and the input depth image. In a similar way as [1], Fanelli *et al*. [3] generate a set of 3D face models with different orientations that are used to train a set of regression trees. Each leaf of the trees votes for the position and orientation of the nose using only a patch of the training model. The proposal provides high quality results and it has been used as a baseline to compare the performance of other methods. Also, Papazov *et al*. [11] propose a 3D invariant descriptor that represents facial landmarks. The head pose is inferred by feature matching. This method requires that the target has similar characteristics to those of the training set. In addition, facial deformation reduces the estimation performance. These three methods [11, 3, 1] depend on an offline training phase, which is executed on a set of synthetically generated 3D head models. Therefore, in order to achieve optimal results, we need to provide a comprehensive dataset with head samples in a wide range of orientation. On the other hand, Ghiass *et al*. [4] perform HPE by fitting a 3D morphable model over the 3D point cloud of the target. As the previous methods in state of the art, this proposal has of an offline training/learning where a person-specific model is learned. The model fitting is done by minimizing a cost function, which includes pose and depth data. Although RGB-based methods, such as Kazemi's proposal, are more accurate when a person sees directly to the camera, this type of methods have a higher detection rate even in fast motion.

RGBD-based approaches Other works propose to combine the color and depth cues [16]. The work of Kaymak and Patras [5] is similar to the approach of Fanelli, it uses random forests with tensor regression methods to model large variations of head pose. Smolyanskiy *et al*. [15] include a new constraint on the AAM model fitting that considers the depth. Li *et al*. [8] propose a template-model

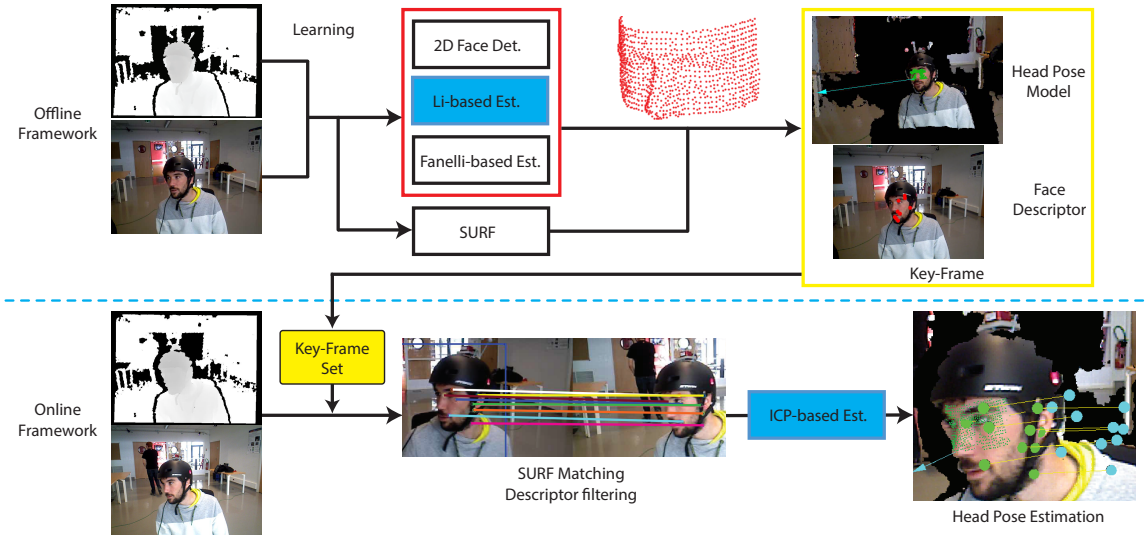

Figure 1. Pipeline of the proposed head pose tracking framework.

based framework. It uses the Kinect output (color and depth frames) to calculate a 3D point cloud. The template model is matched using the ICP algorithm. In a parallel process, the template is updated online using depth information. In addition, Li *et al*. propose several considerations to improve accuracy and computational cost, such as self-occlusions (head rotations that conceal part of the head) and handle extern occlusions (*i.e*. glasses) using color information. Also, to increase accuracy, the approach includes RGB-based information from the 2D facial landmark detector from Viola and Jones [20]. The landmark detections, *i.e*. eyes, are projected to 3D world and are used to weight the matching model-current frame used by ICP. This leads the template to zones more likely to be part of the face, reducing the computational cost. It adapts well to different targets. The method does not detect a face as fast as Fanelli's proposal but it is more precise and it estimates poses better under large head rotations. Vacchetti *et al*. [18] propose a Key-Frames method to detect the pose of static rigid objects using multiple views. In our case, the target is moving and not the camera. Also, targets are subjected to deformations (*i.e*., facial expressions) and (some) strong occlusions. The nature of the aforementioned proposals allows them to perform well in specific scenarios, *i.e*. high accuracy in frontal view (Kazemi [6]), good detection rate (Fanelli [3]) or high precision for a large rotation range (Li [8]). Those methods complement each other limitations but the real-time constrain is not achievable. BIWI dataset is, in the literature, the standard for evaluating head pose detectors [3, 4, 8, 11]. It consists of 24 sequences, each with a different person recorded with neutral expression, moving only the head with slow-medium speed. This dataset is aimed to frame-by-frame detection and not tracking.

Contributions. Following such insights, our contributions are: (i) Combination of existing approaches into an original concept of KF. There, we exploit the complementarity of the aforementioned proposals to learn, in an offline step, driver's appearance for some poses in a discretized orientation space. Each KF associates RGB information in 3D points. Those points are used to estimate a raw head pose that is later refined using ICP. (ii) Creation of a dataset more challenging that those in the literature. We recorded our own dataset, aimed to tracking, with more challenging situations than BIWI, such as facial deformation, self-occlusions, rapid movement and among others. (iii) A conclusive evaluation with respect to recent approaches.

3. Key-Frame-based pose estimator

Given the previous insights, we consider that Fanelli and Li proposals are the most relevant, the first has over 200 cites and even has been incorporated into the widely used Robot Operating System (ROS) library. The latter shows a better performance than Fanelli in publicly available sequences. We evaluate them and highlight their strengths and weaknesses. These proposals tend to fail in challenging target behaviors, such as extreme head orientations and rapid movements.

To overcome these issues, we propose the use of Key-Frames (KF), in the vein of Vacchetti *et al*. [18] but for head pose instead of object localization or Breitenstein *et al*. [1] but taking into account both color and depth using a generic partial-face model. Given these KFs, our work proposes the use of (natural) facial markers through SURF type descriptors, in a similar way as commercial MoCap systems that rely on artificial markers. Our motivation is

to improve the robustness of pose estimation and increases the accuracy and orientation range that the estimator can handle. Fig. 1 shows the pipeline of our proposal. At the top, we have the off-line framework that automatically estimates the KFs. Our proposal is designed for driver assistance. In our scenario, precision plays an important role that could be difficult to achieve given that target behavior could be random, spontaneous or abrupt. Therefore, taking an initial step for learning/estimating KF is not an inconvenience. KFs are formed from color and depth frames, provided by an RGB-D sensor, and a template 3D facial model. The model represents a human face using a 3D mesh (point cloud), where its pose represents the true head pose of the target. The pose is calculated using a computationally expensive but off-line procedure, red block of Fig. 1, which is described in the subsection below. Once a KF is selected, we use the appearance cue to characterize the targets using the SURF descriptors, yellow block in Fig. 1. We chose this type of descriptors since they are invariant to rotation and scale. The latter is very important in our case because a target may not be static in the seat, it could move closer of farther from the camera. The main idea is to perform a classic matching between current and reference descriptors but we limit that part as follows: we know the real head position from the template model, projected to image plane. We estimate descriptors only in the area surrounding the face. Then, the 3D position of the descriptors is estimated thanks to the depth cue. If the 3D points are in a distance bigger that a threshold, *i.e*. 1.5m, we consider them as part of the background and we remove them. Thus, we have a set of $D = \{ft_j, P_j\} \ \forall \ j = \{1 \dots J\}$ descriptors which associate 2D face information in 3D points. The 3D face model represents the head pose of the target. The model pose is very important and could be difficult to estimate, even manually. For this reason, we propose an automatic method for learning them.

3.1. Automatic Key-Frames learning

Our proposal considers key-frames that combine color and depth cues, this is one novelty of our approach. These cues provide information that makes our system more robust. First, we briefly describe how to accurately (but slowly) estimate the head pose. In order to accurately estimate the 3D head pose, we rely on an off-line framework formed by the combination of 3 face/head pose estimators in the state of the art (see section 2): 2D based face detector of Kazemi [6], Fanelli 3D based method [3] and RGB-D proposal of Li [8], see red block of Fig. 1. The pose is defined through a 3D facial (head) template. In our case, we employ a model generated by the Basel face model [12]. This proposal is based on PCA and it is trained with 200 subjects with neutral expressions. It can represent a human face using a 3D mesh and, by tuning some weight parameters, we can create faces with different shapes (young female faces, old man faces, skinny faces and so on). In our approach, we use a generic human face model and, as in [8], we take only the part between the forehead and the base of the nose. This selection is due to the fact that this region of the human face is not very affected by deformations from facial expressions. Thus, we have a model $M = \{p_1, \dots, p_m\}$ consisting of $m = 1000$ 3D points $p = \{x, y, z\}$.

Fig. 1 depicts this model as the output of the red block. The key idea is that each head pose estimator (Kazemi, Fanelli and Li) proposes a pose candidate $P = \{p, \theta\}$, representing the nose position $p = \{x, y, z\}$ and head orientation θ, in spherical coordinates. Thus, a frame t is selected as a KF if all the estimators provide a similar configuration. In other words, we calculate the mean $P^* = \{p^*, \theta^*\}$ and variance $Var(C)$ of the $C = \{P_{Kazemi}, P_{Li}, P_{Fanelli}\}$ proposals. If the variance $Var(C)$ is smaller that a threshold $th_1 = 0.1$, all the estimators are consistent with the same pose. Therefore, we keep this frame as KF setting the 3D model according to P^*. Also, we privilege each method according to the situation. For example: for frontal detections, the 2D detector is more accurate; Li can handle better large rotations and Fanelli is better with fast movement. Thus, our selection is as follows:

$$P_{KF} = \begin{cases} P_{Kazemi} & \text{if } ||p^* - p_{Kazemi}|| < th_d \\ & \quad \textbf{and } \theta^o < th_\theta \\ P_{Li} & \text{if } ||p^* - p_{Li}|| < th_d \\ & \quad \textbf{and } \theta^o > th_\theta \\ P_{Fanelli} & \text{if } ||p^* - p_{Fanelli}|| < th_d \\ & \quad \textbf{and } s^* < th_s \end{cases},$$

where $P_{\cdot}$ is the candidate pose for each proposal, th_d is a threshold for the pose error, θ^o is the angle between head pose θ^* and camera origin and $s^* = ||P^t - P^{t-1}||$ is the speed calculated between current and previous pose estimation. This process is done only on the offline learning phase, since the computational cost of the 3 algorithms make impossible to use them in a real time application. This weakly supervised process makes the user to move along until the system has recorded a certain number of KF. During this learning phase, a target performs simple head motions such as moving from left to right, up to down and in circles. To keep the number of KFs low and covering all possible orientations, we discretized the orientation space, *e.g*. 20 degrees for inclination and azimuth in spherical coordinates, as shown in Fig. 2. The yellow sphere depicts the discretized orientation. The goal is that the target covers all the possible discretized orientations, green regions in Fig.2. At the end, we have a set $S_{KF} = \{P^k_{KF}, D^k\} \ \forall \ k = \{1 \dots K\}$ of KF with pose and descriptors. In our implementation, we consider $\sim 30-40$ KF, which cover the possible discretized orientation space, see Fig. 2, each with approx. 50 descriptors.

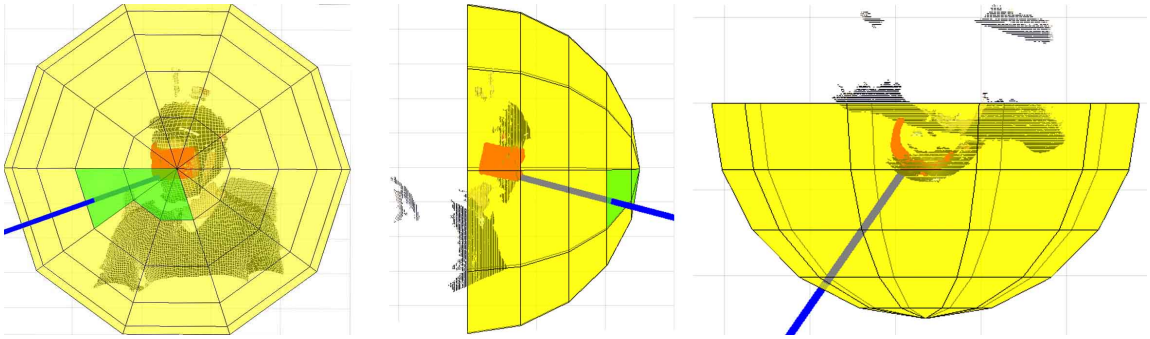

Figure 2. Example of Key-Frame learning. The sphere depicts the discretized orientation space : yellow are unvisited areas, green are zones that have learned a KFs. The red part is the 3D head model and blue line depicts its orientation, both are estimated by combining the results of the 3 head pose estimators.

Both Kazemi [6] and Fanelli estimators [3] are provided as part of public libraries, *i.e*. ROS and DLib [7]. We have implemented the framework of Li [8], which has two parts computed in parallel: (1) an ICP-based pose detector and (2) an on-line algorithm to update the template model to better fit the target. We exclude the model updating part to reduce computational cost. In practice, we observe that the proposal has problems to estimate head poses with large rotation. Also, it tends to fail when the eyes are not visible, due to the proposal relies in a 2D face landmark detector that might not detect one or both eyes. To address these problems, we propose to include a simple but yet relevant 3D feature, calculated from the previous estimation. We provide with this feature an small contribution to the original proposal. We know the 3D position of the nose from the previous estimation. In the next frame (Fig. 3), we consider that the new nose position should be close to the previous one with a similar orientation. Thus, the idea is to find a point, in the point cloud of current frame, that meets these characteristics. First, we select all neighboring points around the previous nose location, below a threshold. Then, we look for the furthest point in the previously estimated orientation. This point is a good candidate for being a nose and it is considered in the ICP algorithm. In Li's proposals, eyes are detected using Viola and Jones detector [20]. Those are projected to the 3D world using the depth image. Then, in the ICP algorithm, a weight factor between both detections and the eye region in the 3D face model is included to privilege this association. In our case, we incorporate our 3D nose feature in ICP algorithm in the same way. We evaluate and compare this approach that includes the nose feature against Li's original proposal. This version is used in KF learning. Our method is person-specific then, like in [4], the learning process must be done for each new target. Nevertheless, the time used in this step is rewarded by increasing the performance of the estimator.

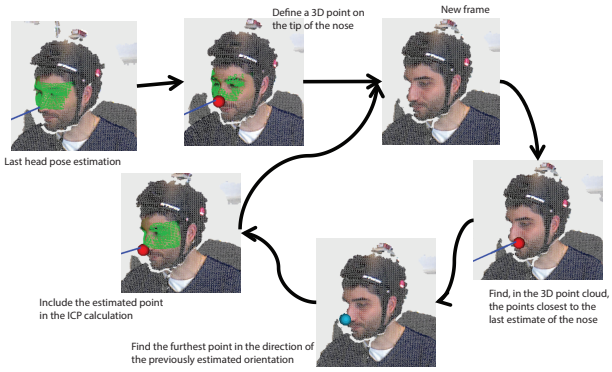

Figure 3. Example of nose feature estimation. The previous estimation of the nose is used as a feedback to guide ICP algorithm to a nose candidate.

3.2. Head Pose estimation

In this section we describe how to calculate the registration between KF and current frame. This is done in two steps. First, we use the 2D correspondences, between the SURF descriptors of KF and current frame , and the 3D face model to estimate an initial head pose. Then, we refine the estimation taking into account the 3D point cloud information. From the offline learning step we obtain K Key-Frames, each with RGB-D information and a robust head pose. For each KF, we calculate the SURF descriptors but only in a region of interest (RoI) $R = \{h, w\}$ around target face. This RoI is defined by projecting 3D face model back to image plane. As we can see in Fig. 3, 3D face model does not cover the entire face, leaving out some relevant information of the face. Therefore, we scale the size of this RoI by a fixed value $R^* = \alpha * R$ with $\alpha = 0.15$. Since each image pixel can be projected to 3D world using the depth image, we project the descriptors and remove those who have a distance greater than $1.5m$. Finally, we have a set of η SURF descriptors for each k KF in such a way that

$\hat{D}^k = \{d_1, \ldots, d_{\eta_k}\}$. For the current frame t, we calculate the SURF descriptors over the entire image because the target location is unknown. After filtering them by depth, we have D^t descriptors with 3D position. Now, we calculate the best key-frame D^b between descriptors D^t and $\hat{D}$ where $D^b = \arg\min_k f(\hat{D}^k, D^t)$ and f is the matching cost function of feature descriptors. In other words, we take the k KF with the best number of correspondences. Some matches between D^t and D^b may be inconsistent, *i.e*. a descriptor in the left eye matches the right eye, or spurious, *i.e*. mouth corner matches with the forehead edge, because, sometimes, those have a similar appearance. We can remove them by considering that coherent matches must have a similar distance and orientation (in 3D world coordinates). Thus, we filter the matches $M^{b,t}$ between the descriptors D^t and D^b as follows: First, we calculate the mean μ_o and variance σ_o of the orientation of $M^{b,t}$ and the mean μ_d and variance σ_d of the distance of $M^{b,t}$. Then, we use the Mahalanobis distance $Mah(\cdot)$ to remove a match i if:

$$Mah(\mu_o, \sigma_o, M_i^{b,t}) < th_m \quad \textbf{or} \quad Mah(\mu_d, \sigma_d, M_i^{b,t}) < th_m.$$

An initial head pose can be estimated by computing the rigid transformation of the 3D point of the matches $M^{b,t}$ using SVD. This estimation is easily and quickly computed. Then, we apply this transformation over the 3D face model to obtain an unrefined pose. Inspired by Li, we build the 3D Point Cloud, from the depth cue, and apply ICP between this and the initial face model. The constraints imposed in the ICP algorithm, like creating correspondences only at short distance, make it converge quickly (1 or 2 iterations). ICP refines the pose giving a more accurate estimation.

4. Experimental evaluations

We test our proposals, Li variant with/without nose feature and KF-based framework, on two challenging benchmarks: the publicly available Biwi Kinect Head Pose Database [3] and one dataset developed by us.

4.1. Datasets

Biwi dataset [3] contains 24 sequences with over 15K images of 20 different people. The targets move their heads covering a range about ± 75 degrees yaw and ± 60 degrees pitch. It provides a ground truth of head pose (3D location and rotation of the head) for each frame. Our dataset consists of 4 challenging sequences, see Fig. 4, each one with a different person with very different facial morphology. The sequences were created to test the performance of the head pose estimator in situations where the target shows complex behaviors such as rapid movements, extreme rotation of the head, distance change with respect to the camera and occlusions. Each sequence is more aimed to a specific behavior that the others. All sequences are recorded under laboratory conditions using a Microsoft Kinect v1 and a commercial Motion Caption (MoCap) system, which provides the ground-truth. The RGB-D images have a resolution of 640×480. The targets are located at a distance of about 80 cm, which is the expected distance of a seated driver. The MoCap system requires distinguishable marks, which reflect infrared light, in order to detect (with high accuracy) the location and orientation of a rigid body. Thus, we put 6 marks over a helmet that the target wears allowing the MoCap to measure the head pose at any time. Therefore, we have an accurate ground-truth. The specific characteristics of each sequence are presented in Tab. 1. The seq1 is easier of the whole set. It consists of 1890 frames. The target performs slow simple motions with a small range of head orientation and no (self) occlusions. Seq2 has a medium difficulty with 1083 frames. It has some fast motions, extreme head orientation range and multiple distance shifts with respect to the camera ($\pm 20cm$). Seq3, with 1535 frames, has more extreme head orientations and some self-occlusions. Seq4 is the fastest with wide range of head orientation and few distance shifts. It has 1929 pictures. The first sequence is simple and quite similar to other public available datasets, *i.e*. Biwi dataset. The others are more challenging and are acquired to test the viability of estimators under extreme circumstances.

Figure 4. 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.

Seq	Frames	Rot. Range	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

Table 1. Description of our own head pose sequences.

4.2. Evaluation criteria

We evaluate the performance of the proposals using two metrics: Mean Angular Error and Missed Detections. The first is only the angular error, in radians, between the ground-truth and the estimated orientation. We consider a

missed detection when the algorithm does not converge or the angular error is greater than a threshold, *e.g*. 45 degrees. In most cases, the head is detected with a relatively correct position. A poor or no detection is reflected in the accuracy of the orientation and missed detections. Therefore, like other works in the literature, we do not focus on the position but on the estimated rotation.

4.3. Results

The Fig. 5 shows some quantitative results which are the mean of 30 runs of each algorithm. All results have a variance of less than 0.01 that shows the repeatability of the results. The left figure presents the Mean Angular Error, in radians, of each proposal. We observe that the method of Fanelli presents the highest error. This is mainly because it can not handle well a wide range of orientations. This could be improved if Fanelli's proposal is better trained, although this requires more pre-processing. The rest of the approaches have similar accuracy. Our proposal (in purple) shows a small improvement of 0.1 - 0.2 radians (5-11 degrees). The right Fig. 5 depicts the missed detection rate. We observe in the figure that the KF proposal performs better in comparison with the others by a wide margin. Also, comparing the results of Fanelli and Li, we see that Fanelli has a better detection performance. On the other hand, Li does not detect well a pose under fast motions. Also, Li's proposal has problems handling poses with extreme orientation. The added nose feature improves this aspect and increases the detection rate, as we can see by comparing the green and red columns in Fig. 5. This results corroborate those shown by the literature, see section 2. We present in Fig. 7 the results on the Biwi dataset as the mean over the 24 sequences. We can observe that the nose-based heuristic, included as a new feature in the ICP algorithm, improves Li's proposal in terms of detection rate. The last column of both graphs shows our results. The mean orientation error is the smallest, this shows that the proposal provides better accuracy. Moreover, the missed detection rate of the KF-based framework is three and two times smaller than Fanelli and Li's proposals, respectively.

From those results, we show that our proposal can better estimate the head pose in challenging situations maintaining or even improving the accuracy in comparison with other proposals in literature. The distribution of missed detection is evaluated, see Fig. 6, using a 2d histogram of yaw and pitch rotations. The histogram represents the set of orientations in a discretized way and is constructed by counting how many times the proposal failed to detect a head for a specific orientation. We normalize the histogram with respect to the number of frames evaluated, this means that the number represented in the figure corresponds to the percentage of missed detections for a particular orientation. Each cell represents a possible discretized head orientation. Therefore, the center of the graph corresponds to a head facing forward the camera. This figure shows the results with 3 proposals evaluated using Seq2. We can observe that the higher values are located in the bottom-right corner, this means that most missed detections occur when the target is looking in that direction. Also, on the right side of Figs. 6(a) and 6(b) there is a red region. It means that this orientation is the most difficult for the approaches based on Li. This is not the case in Fig. 6(c) where the error is smaller.

Method	Fast motion	Orient. Range	Detection rate	Precision
Li [8]	+	++	+	++
Li Nose	++	++	++	++
Fanelli [3]	+++	+	+++	+
Our proposal	+++	+++	+++	++

Table 2. Evaluation summary of each head pose estimator.

4.4. Discussion

The evaluated proposals can handle different types of behaviors. We show a summary of the characteristics of each proposal in Table 2 grading them as (+) low, (++) good and (+++) excellent. Fanelli's approach can better deal with rapid head movements and detect a face most of the time. However, the estimations are not very accurate and the range of orientation is restricted. Also, it requires an exhaustive training step, which is susceptible to false positives and has trouble handling facial expressions. In contrast, Li's algorithm gives better results. It can estimate the head pose more robustly. Also, it can cover a wider rotation range, in comparison to Fanelli, with very accurate estimations. However, it is limited to slow/medium head speed due to the nature of ICP. Then, the detection rate is reduced during fast movements, *e.g*. Seq3 and Seq4. The new nose feature enhances Li's proposal by providing information in fast motion and in a wider orientation range. But, as shown in Fig. 6, neither can handle well extreme rotations. Our approach improves the detection rate with competitive accuracy compared to the literature proposals. Each sequence has a greater complexity than the previous one. From Fig. 5, we can see that our proposal has a higher gain than the others, as the sequences become more complex, leaving the missed detection almost constant compared to other approaches.

5. Conclusion and future work

We have presented a head pose estimation system that takes advantage of Key-Frames. We present a supervised methodology for automatically creating a KF set in an offline learning step. The KF allows to handle poses with rotations of large range and provides robust estimates. We compare our proposal with those in the literature and eval-

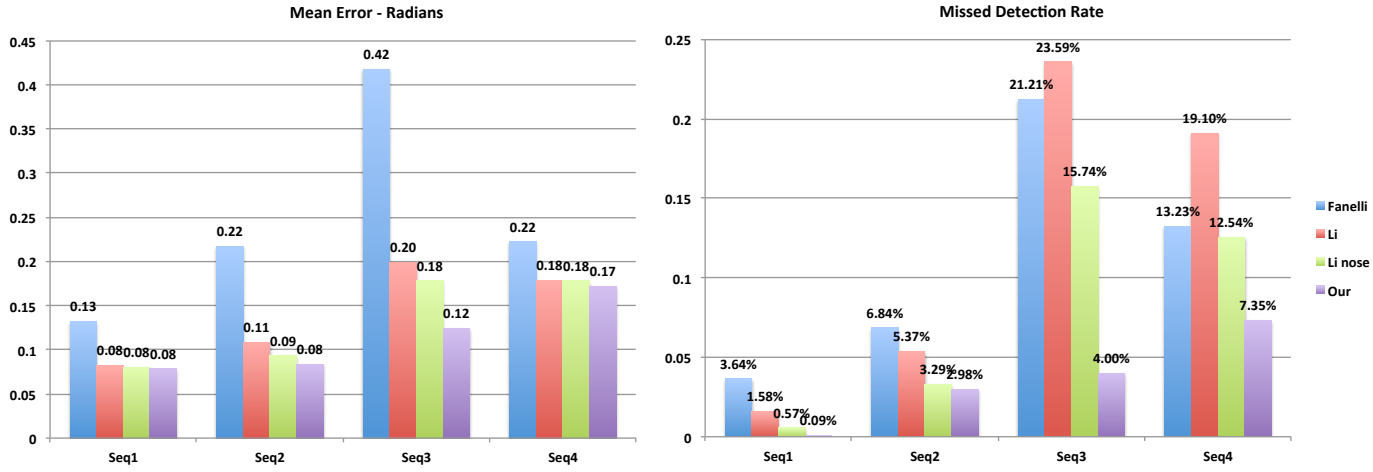

Figure 5. Results using our dataset: (blue) Fanelli, (red) Li simple approach, (green) Li proposal including of nose detection heuristic, and (purple) our descriptor-based method. Results are the mean of 30 runs with a variance lower than 0.01.

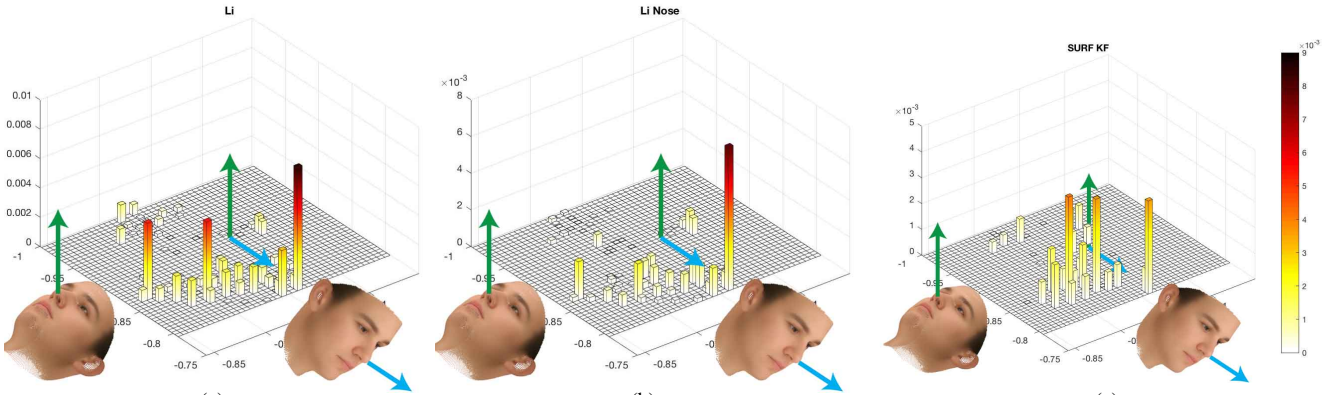

Figure 6. 2D Histogram of the missed detection distribution. Results using: (a) Li approach, (b) Li proposal including of nose detection heuristic, and (c) our descriptor-based method.

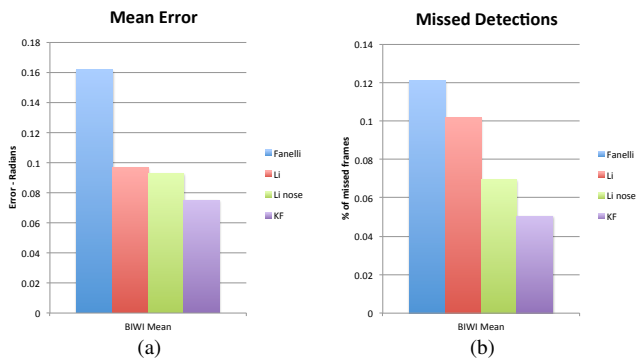

Figure 7. Mean of the results on the BIWI dataset. Left: Mean orientation error (in radians). Right: Percentage of missed detections.

uate them using two challenging benchmarks. Each sequence presents complex behaviors ranging from rapid head movements, extreme orientations, self-occlusions and targets moving away/forward from the camera. The experiments show how our approach efficiently manages these challenging situations that could lead to missed detections or loss of accuracy. As future work, we will improve the KF set by adding an online KF learning update. Also, we would like to use the previous estimation as a feedback such that: the new SURF descriptors are calculated around the previous pose and privilege the KF with the same orientation as the previously estimated pose. We will publish our dataset online for the benefit of the research community.

References

[1] M. D. Breitenstein, D. Kuettel, T. Weise, L. van Gool, and H. Pfister. Real-time face pose estimation from single range images. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[2] B. Czupryński and A. Strupczewski. *High Accuracy Head Pose Tracking Survey*, pages 407–420. Springer International Publishing, 2014.

[3] G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool. Random forests for real time 3d face analysis. *Int. J. Comput. Vision*, pages 437–458, 2013.

[4] R. S. Ghiass, O. Arandjelović, and D. Laurendeau. Highly accurate and fully automatic head pose estimation from a low quality consumer-level rgb-d sensor. In *Proceedings of the 2Nd Workshop on Computational Models of Social Interactions: Human-Computer-Media Communication*, HCMC '15, pages 25–34, New York, NY, USA, 2015. ACM.

[5] S. Kaymak and I. Patras. Exploiting depth and intensity information for head pose estimation with random forests and tensor models. In *Asian Conference on Computer Vision*, pages 160–170, 2012.

[6] V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Conf. on Computer Vision and Pattern Recognition*, pages 1867–1874, June 2014.

[7] D. E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[8] S. Li, K. N. Ngan, R. Paramesran, and L. Sheng. Real-time head pose tracking with online face template reconstruction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(9):1922–1928, Sept 2016.

[9] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation in computer vision: A survey. *Trans. on Pattern Analysis and Machine Intelligence*, 31(4):607–626, 2009.

[10] E. Murphy-Chutorian and M. M. Trivedi. Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):300–311, June 2010.

[11] C. Papazov, T. Marks, and M. Jones. Real-time 3d head pose and facial landmark estimation from depth images using triangular surface patch features. In *Conf. on Computer Vision and Pattern Recognition*, June 2015.

[12] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. In *Int. Conf. on Advanced Video and Signal Based Surveillance*, pages 296–301, Sept 2009.

[13] J. Saragih, S. Lucey, and J. Cohn. Deformable model fitting by regularized landmark mean-shift. *Int. J. of Computer Vision*, 91(2):200–215, 2011.

[14] S. Sheikhi and J.-M. Odobez. Combining dynamic head posegaze mapping with the robot conversational state for attention recognition in humanrobot interactions. *Pattern Recognition Letters*, 66:81 – 90, 2015. Pattern Recognition in Human Computer Interaction.

[15] N. Smolyanskiy, c. Huitema, L. Liang, and S. Anderson. Real-time 3d face tracking based on active appearance model constrained by depth data. *Image and Vision Computing*, 32(11):860 – 869, 2014.

[16] A. Strupczewski, B. Czupryński, W. Skarbek, M. Kowalski, and J. Naruniec. Head pose tracking from rgbd sensor based on direct motion estimation. *Procs. in Int. Conf. Pattern Recognition and Machine Intelligence*, pages 202–212, 2015.

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

[18] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3d tracking using online and offline information. *Trans. on Pattern Analysis and Machine Intelligence*, 26(10):1385–1391, 2004.

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

[20] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, May 2004.

[21] M. Zhou, L. Liang, J. Sun, and Y. Wang. AAM based face tracking with temporal matching and face segmentation. In *Computer Society Conference on Computer Vision and Pattern Recognition*, pages 701–708, June 2010.