Head Pose Estimation Based on 3-D Facial Landmarks Localization and Regression

Dmytro Derkach, Adria Ruiz and Federico M. Sukno
Department of Information and Communication Technologies, Pompeu Fabra University, Barcelona, Spain

***Abstract*—In this paper we present a system that is able to estimate head pose using only depth information from consumer RGB-D cameras such as Kinect 2. In contrast to most approaches addressing this problem, we do not rely on tracking and produce pose estimation in terms of pitch, yaw and roll angles using single depth frames as input. Our system combines three different methods for pose estimation: two of them are based on state-of-the-art landmark detection and the third one is a dictionary-based approach that is able to work in especially challenging scans where landmarks or mesh correspondences are too difficult to obtain. We evaluated our system on the SASE database, which consists of ∼30K frames from 50 subjects. We obtained average pose estimation errors between 5 and 8 degrees per angle, achieving the best performance in the FG2017 Head Pose Estimation Challenge. Full code of the developed system is available on-line.**

I. INTRODUCTION

Human head-pose estimation has attracted a lot of interest because it is usually the first step of many face analysis tasks. It is an important aspect in facial motion capture, human-computer interaction and video conferencing, as well as a prerequisite for face recognition or facial expression analysis. Head pose estimation has traditionally been performed on RGB images, but recent advances in 3D geometry acquisition have led to a growing interest in methods that operate on 3D data. These methods are less sensitive to changes in illumination and viewpoint than 2D image-based approaches, which makes them more accurate and robust [1].

The goal of head pose estimation is to predict the relative orientation between the target head and the viewer or camera. It is usually parametrized by the heads pitch, yaw and roll angles. An early attempt to classify head pose estimation methods from a methodological perspective was presented by Murphy et al. [2], who proposed 8 categories including appearance template methods, flexible models, non-linear regression and tracking. While that classification included both 2D and 3D methods, in this paper we focus on head estimation based exclusively on depth information. This considerably reduces the number of categories to: geometric methods [3], [4], appearance methods [5], [6], [7], regression methods [8], flexible models [9], [10] and tracking methods [5].

An important aspect of 3D head pose estimation algorithms is whether RGB data or temporal information are used. Firstly, RGB data can provide complementary information to the one provided by depth data, especially at the detection stage, but it is likely to reduce the robustness to illumination that is inherent to 3D-only data. It is also very popular to make use of dynamic information to improve head pose orientation results. However, algorithms using tracking often benefit from the fact that test sequences usually start with near-frontal head orientations and, therefore, it is not clear their robustness to detect initial head poses other than frontal, which are arguably more challenging.

978-1-5090-4023-0/17/$31.00

In this paper, we present an approach for accurate static 3D head pose estimation which is able to perform head-pose estimation using only depth information from a single Kinect 2 frame of a person sitting in front of a camera. This setup has been specified in the FG2017 Head-Pose Estimation Challenge [11]. In contrast to most existing approaches, we base our system in the detection of 3D facial landmarks, whose positions are later used to derive geometry- and patch-based pose estimators. A key aspect of the proposed system is the use of Shape Regression with Incomplete Local Features (SRILF) [12] for landmark localization. This algorithm provides state of the art landmark localization accuracy with no prior initialization and is inherently tolerant to occlusions or missing data. The latter is very important when capturing moderate or large head rotations with a single-view depth sensor such as Kinect 2 since, in such cases, large parts of the face become unavailable due to self-occlusions. Our system is complemented with a secondary pose estimator based purely on patches sampled randomly on the head region to account for potential failures of the landmark-based estimation. Our tests on the SASE database [13] provided in the FG2017 Head-Pose Estimation challenge, showed average estimation errors of 7.82, 6.65 and 5.39 degree for pitch, yaw and roll angles, respectively.

II. RELATED WORK

As aforementioned, an important aspect of 3D head pose estimation algorithms is whether or not they use RGB data and tracking. Only few of methods has addressed this problem without the use of temporal information.

For instance, Sun and Yin proposed a geometric feature based pose estimation approach based on 3D facial models [3]. The pose orientation was estimated using a symmetry plane. Li and Pedrycz [4] developed a central profile-based 3D face pose estimation algorithm. The central profile is the intersection curve, that starts from forehead center, goes down through nose ridge, nose tip, mouth center, and ends at a chin tip. It is also called symmetry plane. They defined an objective function for conducting the Hough transform in parameter space that maps face profile to an accumulator cell. The face profile corresponding to the maximum accumulator cell was regarded as the central profile. Once the symmetry plane had been

completed, two angles (roll and yaw) were determined, since the objective function was based on three parameters. Based on the detection of central profile, nose tip was detected and pitch angle was estimated using the coordinates of three points nose tip, nose ridge point and nose bottom point. Valle et al. [14] also presented a free-tracking algorithm that estimates the head pose, but they estimated only one yaw angle from unrestricted 2D gray-scale images. In order to obtain a discrete head-pose estimation, they proposed a classification scheme, based on a random forest, where patches randomly extracted from the image cast votes for the corresponding discrete head-pose angle. Papazov et al. [5] presented a real-time system for 3D head pose estimation using a commodity depth sensor such as Microsofts Kinect. The proposed method consists of an offline training and an online testing phase. In both phases, 2D information was used for face detection. After that, a triangular surface patch (TSP) descriptor, which encodes the shape of the 3D face surface within a triangular area, was employed for final angle estimation. For testing, the authors utilized two approaches: tracking mode and detection mode (static).

Another free-tracking approach was presented in [6]. Breitenstein et al. developed an error function that compares the input range image to precomputed pose images of an average face model. In an offline step, range images of an average face were rendered for many poses, and the resulting reference pose range images were saved. For each pixel they computed signatures that are distinct for regions with high curvature, such as the nose tip. This yielded a set of candidate nose positions and orientations that were used as head pose hypotheses. Then they computed the error between the reference pose range images corresponding to the pose hypotheses and the input range image using a novel error function. The match with the lowest error yielded the final pose estimation and a confidence value. In [15], an approach was presented to estimate the 3D position and orientation of head from single RGB and depth images. 2D Scale-invariant feature transform (SIFT) features were used together with 3D histogram of oriented gradients (HOG) features, which were extracted in a pair of RGB and depth images captured synchronously. Random forests approach were then applied in order to formulate pose estimation as a regression problem, due to their power for handling large training data and the high mapping speed. Finaly, the mean-shift method was employed to refine the result obtained by the random forests.

Similarly, Fanelli et al. [8] used random forests to handle large training datasets and formulated a real-time head pose estimation as a regression problem for tracking purposes. In [7], authors proposed a fusion approach to address real-time head pose estimation. They constructed a system able to recover itself (in cases where the tracking was lost) by combining a frame independent decision tree based estimator with a personalized template tracker.

An alternate approach, using depth as well as intensity information, was presented by Baltrusaitis et al. [10]. The authors presented 3D Constrained Local Model (CLM-Z) for the facial feature tracking under varying pose. A two-step CLM fitting strategy was employed: performing an exhaustive local search around the current estimate of feature points leading to a response map around every feature point, and then iteratively updating the model parameters to maximize a posterior probability until a convergence metric is reached. For fitting, they used Regularised Landmark Mean-Shift (RLMS). Another relevant paper, by Padeleris et al. [16], estimated the pose of an input Kinect sensor depth map by finding the 3D rotation of a template that best matched the input. The proposed method searches for a view at which the rendered image matches the reference depth image obtained during an initialization phase. At run time, the method searches the 6-dimensional pose space to find a pose from which the head appears identical to the reference view. This registration was treated as an optimization problem that was solved through Particle Swarm Optimization (PSO). One more approache based on PSO was presented by Meyer et al. [9]. They performed pose estimation by registering a morphable face model to the measured depth data, using a combination of particle swarm optimization (PSO) and the iterative closest point (ICP) algorithm.

Martin et al. [17] presented approach for head pose estimation on consumer depth cameras that works without prior knowledge of the tracked person and without prior training of detector. To achieve this, they combined an algorithm to generate and track a model of the head with feature based head pose estimation. This algorithm was based on tracking a head model using the iterative closest point algorithm.

III. Proposed system

A block diagram of the proposed system is shown in Fig. 1. We start by approximately isolating the head region using clustering and use the obtained result to build a 3D mesh $\mathcal{M}$ that contains the head and a variable part of the shoulders. Mesh $\mathcal{M}$ is fed to the SRILF algorithm [18] with the aim to automatically detect 12 prominent facial landmarks. The SRILF algorithm performs both detection of the visible landmarks and estimation of potentially occluded landmarks. Thus, if successful, the algorithm always returns an estimate of the coordinates for all 12 targeted points. Landmark detection details are provided in Section III-A.

Once facial landmarks are available, we use two complementary approaches to estimate the head pose (Section III-B). Firstly, we perform a least-squares estimation of the eye-line and frontal-plane of the face which provide straight-forward *geometric* estimates of the head pose. The second estimate is based on regression over local surface descriptors (appearance) centered at the landmark points. While these two estimates are conceptually quite different, in practice, we will see that in practice they produce similar results (Section IV-A).

In a vast majority of cases ($\sim 90\%$) the above steps are sufficient to accurately estimate the head pose. The remaining 10% of cases are especially challenging scans, typically due to i) very large rotations, with self-occlusion of large portions of the face, and/or ii) low quality scans due to imaging artifacts. In such cases, we use an alternative estimate of the head

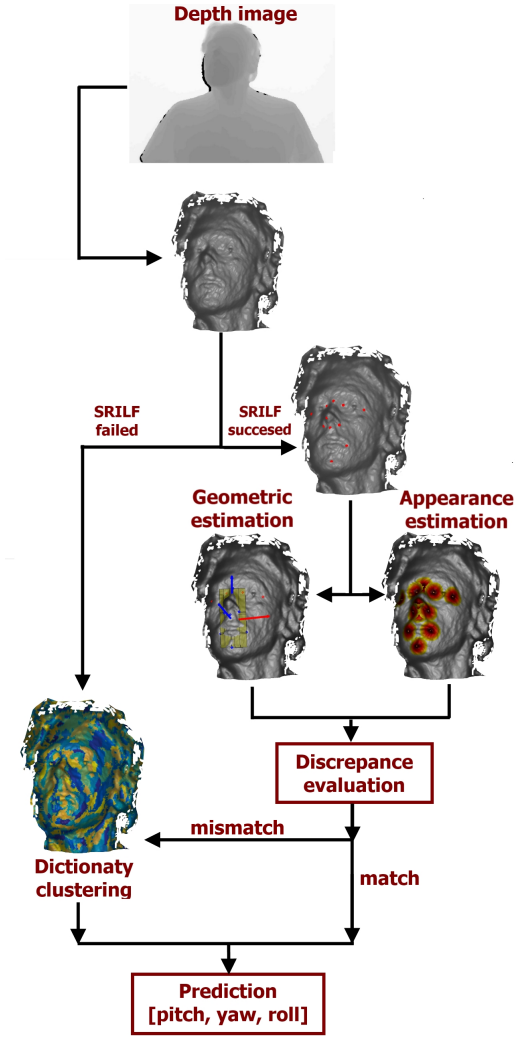

Fig. 1. Block diagram of the proposed head pose estimation method

pose based on dictionary learning (Section III-C). It should be emphasized that the system automatically chooses whether to use the landmark-based or dictionary-based estimates on a case-by-case basis, with the following rationale:

- If landmarks are accurately detected, their estimate of the head pose is more precise than the dictionary-based estimate.
- If the SRILF algorithm cannot produce a reliable estimate of landmark positions, the dictionary-based estimate is the only one available.
- If both landmark-based estimates (geometric and local descriptor regression) do not coincide, it is very likely that landmarks have been incorrectly detected. Thus, dictionary-based estimate should be used.

A. 3D Landmark Detection

We use Shape Regression with Incomplete Local Features (SRILF) [18] to locate the following 12 facial landmarks: inner and outer eye corners, nose corners, mouth corners, nose root, nose tip and chin tip. The SRILF algorithm combines the response from local feature detectors for each of the targeted landmarks with statistical constraints that ensure the plausibility of landmark positions on a global basis. The algorithm has three components: 1) selection of candidates through local feature detection; 2) partial set matching to infer possibly missing landmarks; 3) combinatorial search, which integrates the other two components.

1) Selection of candidates: The selection of candidates is performed independently for each targeted landmark. Given a mesh $\mathcal{M}$ and a landmark $\mathbf{x}_\ell$ to be targeted, a similarity score $s_\ell(\mathbf{v})$ is computed for every vertex $\mathbf{v} \in \mathcal{M}$; the set of *candidates* $\mathcal{C}_\ell$ for landmark $\mathbf{x}_\ell$ are the ϱ_ℓ highest scoring vertices:

$$\mathcal{C}_\ell = \{\mathbf{v} \in \mathcal{M} \,|\, \mathcal{O}(s_\ell(\mathbf{v})) \leq \varrho_\ell\} \tag{1}$$

where $\mathcal{O}()$ is the (descending) order function. The score $s_\ell(\mathbf{v})$ is based on the similarity of local surface descriptors with respect to a descriptor template derived at training time. The SRILF implementation currently available[1] uses Asymmetry Pattern Shape Contexts [19] as local descriptors.

As in many other algorithms, it is expected that one of these candidates will be close enough to the correct position of the landmark. Nonetheless, the number of false positives (i.e. vertices that produce high similarity scores even though they are far from the correct landmark location) can change considerably for different landmarks, as well as from one facial scan to another, making it difficult to choose the number of candidates that should be retained.

While many approaches try to retain large numbers of candidates to make sure that at least one will be reasonably close to the desired landmark position, SRILF determines the number of candidates as an upper outlier threshold from the distribution of false positives over a training set. This implies that, in the vast majority of cases, a candidate that is close enough to the target landmark will be detected, but a small proportion will be missed. Hence, for each targeted landmark there will be an initial set of candidates that may or may not contain a suitable solution and we need to match our set of target landmarks to a set of candidates that is potentially incomplete. This is analogous to the point-matching problem found in algorithms that search for correspondences. However, the human face is a non-rigid object and these point-matching algorithms are typically restricted to rigid transformations.

2) Partial set matching: The second component of the algorithm aims at dealing with the above problem. Based on the priors encoded in a statistical shape model, it uses a subset of the landmarks (i.e. those with suitable candidates) to infer the most likely position of the ones that are missing.

Let $\mathbf{x} = (x_1, y_1, z_1, x_2, y_2, z_2, \ldots, x_L, y_L, z_L)^T$ be a shape vector, constructed by concatenating the coordinates of the L targeted landmarks in 3D, and let $\overline{\mathbf{x}}$, $\mathbf{\Phi}$ and $\mathbf{\Lambda}$ be the mean shape, eigenvector and eigenvalue matrices, respectively. Given a shape for which we only know part of its landmarks, we could split it in the known (or fixed) part $\mathbf{x}^f$ and the unknown (to infer or guess) part $\mathbf{x}^g$. Thus, our objective is to

[1] http://fsukno.atspace.eu/Data.htm#SRILF_3dFL

infer the coordinates of landmarks $\mathbf{x}^g$ so that the probability that the resulting shape complies with the PCA model is maximized, ideally without modifying the coordinates in $\mathbf{x}^f$.

Let $Pr(\mathbf{x})$ be the probability that shape $\mathbf{x}$ complies with the model. Assuming that $Pr(\mathbf{x})$ follows a multi-variate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$ in PCA-space, this probability is proportional to the negative exponential of the Mahalanobis distance and it can be shown [18] that maximization of $Pr(\mathbf{x})$ with respect to $\mathbf{x}^g$ yields:

$$\mathbf{x}^g = \bar{\mathbf{x}}^g - (\mathbf{\Psi}^{gg})^{-1}\mathbf{\Psi}^{gf}\left(\mathbf{x}^f - \bar{\mathbf{x}}^f\right) \tag{2}$$

where $\mathbf{\Psi}^{gg} = \mathbf{\Phi}^g\mathbf{\Lambda}^{-1}(\mathbf{\Phi}^g)^T$, $\mathbf{\Psi}^{gf} = \mathbf{\Phi}^g\mathbf{\Lambda}^{-1}(\mathbf{\Phi}^f)^T$ and $\mathbf{\Phi}$ is split in $\mathbf{\Phi}^f$ and $\mathbf{\Phi}^g$ according to $\mathbf{x}^f$ and $\mathbf{x}^g$ (see [18]).

3) Combinatorial search: The third component of the algorithm integrates the two previous steps into a combinatorial search. It consists of analyzing subsets of candidates and completing the missing information by inferring the coordinates that maximize the probability of a deformable shape model.

Formally, let $\mathcal{F}$ and $\mathcal{G}$ be the sets of fixed and to-infer coordinates, respectively, with $\mathcal{F} \cap \mathcal{G} = \varnothing$ and $\mathcal{F} \cup \mathcal{G} = \{1, 2, \ldots, 3L\}$. The goal of the combinatorial search is to dynamically choose the splitting into $\mathcal{F}$ and $\mathcal{G}$ to minimize the localization error:

$$\underset{\mathcal{F}}{\operatorname{argmin}}\{\|\mathbf{x} - \hat{\mathbf{x}}\|^2\} \tag{3}$$

where $\mathbf{x}$ are the *true* landmark coordinates and $\hat{\mathbf{x}}$ is the algorithm's estimate. The key concept here is that only the coordinates in $\mathcal{F}$ will be based on image evidence (e.g. the candidates) and the rest will be treated as *missing data*. Thus, $\hat{\mathbf{x}}^g$ will be obtained by inference and it can be expressed as a function of $\hat{\mathbf{x}}^f$, making more apparent that the minimization looks for the optimal subset $\mathcal{F}$:

$$\underset{\mathcal{F}}{\operatorname{argmin}}\{\|\mathbf{x}^f - \hat{\mathbf{x}}^f\|^2 + \|\mathbf{x}^g - f(\hat{\mathbf{x}}^f)\|^2\} \tag{4}$$

with $f(\hat{\mathbf{x}}^f)$ as defined in eq. (2). Because the true coordinates $\mathbf{x}$ are unknown, we cannot explicitly compute the above errors and need an indirect estimate instead. The SRILF algorithm does this by minimizing (subject to statistical plausibility):

$$\underset{\mathcal{F}}{\operatorname{argmin}}\left(-|\mathcal{F}| - \exp\left(-\sum_{\ell\in\mathcal{F}} \min_{c\in\mathcal{C}_\ell} \|\hat{\mathbf{x}}_\ell - c\|^2\right)\right) \tag{5}$$

where $\mathcal{C}_\ell$ is the set of candidates for the ℓ-th landmark $\hat{\mathbf{x}}_\ell$. Intuitively, eq. (5) can be understood by noticing that the main component of the cost is the cardinality of $\mathcal{F}$, i.e. the number of landmarks that can be successfully included in $\hat{\mathbf{x}}^f$ while keeping the shape statistically plausible. Upon equality of $|\mathcal{F}|$ the cost function increases with the distance from $\hat{\mathbf{x}}$ to the nearest candidate per landmark. These distances to the nearest candidates have a different meaning for fixed and inferred landmarks and help understand the way the algorithm works.

Fixed landmarks $\{\hat{\mathbf{x}}_\ell\}_{\ell\in\mathcal{F}}$ are directly sampled from candidates to guide the combinatorial search. Thus, their nearest candidates are known beforehand and their distance to them is just the reconstruction error of the statistical shape model. For the remaining landmarks, $\{\hat{\mathbf{x}}_\ell\}_{\ell\in\mathcal{G}}$, positions are statistically inferred from eq. (2) independently from their candidate sets (Fig.3). It would be expected that better predictions generate inferred landmarks that are closer to their corresponding candidates, resulting in lower cost values.

The minimization in eq. (5) is addressed by testing all possible combinations of 4 candidates, which constitute the initial $\hat{\mathbf{x}}^f$. The shape is completed by inference of $\hat{\mathbf{x}}^g$ from eq. (5) and is checked against the statistical constraints of the shape model. As long as the generated shape is statistically plausible, candidates are added to $\hat{\mathbf{x}}^f$ from the remaining landmarks in a sequential forward selection strategy looking for the maximum possible $|\mathcal{F}|$.

An important aspect of the splitting between $\mathcal{F}$ and $\mathcal{G}$ is that it inherently provides tolerance to distorted or missing data (occlusions). Notice that there is no prior assumption regarding what landmarks can be in $\mathcal{F}$ or $\mathcal{G}$ nor the cardinality of the two sets and the splitting is performed dynamically on a case by case basis. This is an advantage in applications such as head-pose estimation with sensors like Kinect, which capture depth information from a single view. Under large head rotations, the generated depth maps will have large parts of the face missing due to self-occlusions and it is crucial to be able to exploit partial information.

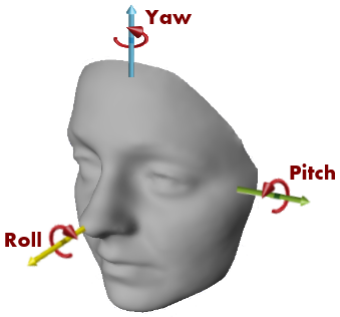

Fig. 2. Orientation of the head in terms of pitch, roll, and yaw angles

B. Landmark-based pose estimation

Once facial landmarks are extracted, we can estimate head pose, represented by three Euler angles also called as *yaw* (ϕ), *pitch* (θ) and *roll* (ψ) angles. Pitch (nodding) is the rotation around the horizontal axis, which in our case is the X axis. Yaw (shaking) is the rotation around the vertical axis of the body (Y axis). Roll (tilting) is the rotation around the axis perpendicular to two previous axes. In our case, this is the Z axis, which is perpendicular to the camera (Fig.2).

We derive two different landmark-based pose estimates, a geometric estimate and an appearance estimate:

1) Geometric estimate: It is based on least-squares estimates of simple geometric entities that can approximately describe the head pose. Specifically, we estimate the eye-line to determine the *roll* angle and a frontal-face plane for *yaw* and *pitch*.

The four landmarks of inner and outer eye corners are used to build the eye-line. Firstly, the eye-line is projected into the

XY plane, where it can be expressed as a linear equation of two variables $y = mx + b$. The roll angle is calculated as $\psi = \tan^{-1}(m)$.

The remaining landmarks, except the nose tip, are used to estimate a plane that will be a good approximation of the frontal-face region (see Fig. 1). Let the normal vector to this plane be $\mathbf{n} = [x_n, y_n, z_n]$. Due to the fact that angles can be obtained by rotations about its principal axes, we can compute the *yaw* and *pitch* angles as: $\phi = \tan^{-1}(x_n/z_n)$, $\theta = \tan^{-1}(y_n/z_n)$.

2) Appearance estimate: It is based on regression over the local appearance around landmark points. Specifically, for each detected landmark $\hat{\mathbf{x}}_\ell$ we compute a local surface descriptor $d(\hat{\mathbf{x}}_\ell)$ that will be the input to a multi-linear regressor $\mathbf{A}_\ell$ yielding an estimate for ϕ, θ and ψ. Thus, differently from the geometric estimate, the appearance estimate requires a training set to derive the regressors.

We use 3D Shape Contexts (3DSC) [20] as local descriptors, slightly modified to increase their sensitivity to viewpoint and robustness to noise. 3DSC are based on a spherical histogram computed on a neighbourhood of the interest point (in our case, landmark locations) and have been shown to perform well as descriptors of the facial surface [21]. Similarly to other popular descriptors 3D geometry [22], [23], [24], 3DSC use the surface normal at the interest point to appropriately orient the reference system of the local neighbourhood, aiming for rotational invariance[2]. Because our objective is to identify viewpoint, such normal-based orientation is not convenient, hence we will orient the reference systems of all local neighbourhoods based on the normal to the camera sensor. This choice avoids also the computation of surface normals, which are known to be especially sensitive to noise [5], [23].

Notice that, in principle, we will produce L different estimates for each angle (i.e. one per landmark). However, because of the potential presence of occlusions, it is not guaranteed that all estimated landmarks will actually lie on the mesh surface.[3] Indeed, when parts of the facial surface are missing, it is possible that some landmarks $\ell \in \mathcal{G}$ are estimated relatively far from the mesh $\mathcal{M}$, i.e. they are inferred in the position where we would statistically expect them to be, despite no surface has been captured there (Fig.3).

Therefore, we use the indicator function $\mathbb{1}(\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| < \epsilon)$ to filter out the estimates of landmarks that are estimated off the surface and produce our final appearance estimate as the average of the remaining ones:

$$(\phi, \theta, \psi)^T = \frac{\sum_k \mathbb{1}\big(\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| < \epsilon\big)\mathbf{A}_\ell d(\hat{\mathbf{x}}_\ell)}{\sum_k \mathbb{1}\big(\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| < \epsilon\big)} \tag{6}$$

where the distance from $\hat{\mathbf{x}}(\ell_k)$ to $\mathcal{M}$ is computed as the distance to the nearest mesh vertex:

$$\|\hat{\mathbf{x}}_\ell - \mathcal{M}\| = \min_{v_j \in \mathcal{M}} \|\hat{\mathbf{x}}_\ell - v_j\| \tag{7}$$

[2]Such invariance, however, is only partially achieved in 3DSC since the orientation of the surface normal still leaves one degree of freedom undefined (the sphere's azimuth [25])

[3]We consider that a landmark is *on the surface* when its distance to it is relatively small as compared to the mesh resolution.

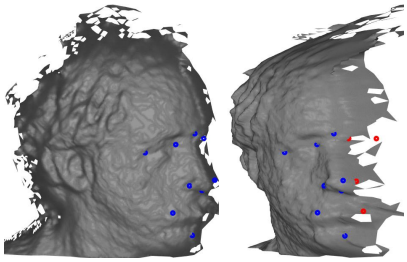

Fig. 3. Positions of the landmarks estimated automatically by SRILF in a head scan showing large yaw rotation. Two views of the same scan are provided: the original view (as seen from the camera) is shown to the left and a rotated view (to simulate a *frontal shot*) is shown to the right. Landmarks lying on the surface are indicated in blue color, while those off-the-surface (estimated by inference) are displayed in red.

C. Dictionary-based pose estimation

As mentioned before, for some small percentage of scans SRILF has difficulties to correctly locate the facial landmarks and, thus, the approaches described in Section III-B are not applicable to estimate the corresponding head pose. Typically, these are especially challenging scans, with big rotations, large parts of the head self-occluded and/or very poor quality. These difficulties, together with the failure of a state-of-the-art landmarker as SRILF, suggest the need for a landmark-free approach to tackle these scans. Thus, we employ an alternative dictionary-based strategy for the scenario where no explicit vertex-landmarks correspondences are found.

Inspired by the success of Bag-of-Words approaches in 3D shape retrieval [26], we represent each scan as a set of descriptors $\mathcal{D} = \{d(\mathbf{x}_1), d(\mathbf{x}_2), ..., d(\mathbf{x}_N)\}$ extracted vertices $\mathbf{x}_n = \{x_n, y_n, z_n\}$ randomly sampled on the mesh $\mathcal{M}$. Concretely, we use again 3DSC descriptors with fixed orientation coinciding with the camera axis (Section III-B) and a random sampling over the mesh with density of $7mm^{-1}$.

Given the sets $\mathcal{D}$ obtained from all the training scans, we use k-means clustering to learn a dictionary $\mathcal{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_K\}$ of 3D descriptors, where each $\mathbf{z_k}$ is a particular centroid and K is the total number of clusters considered. Intuitively, these clusters will represent different shapes typically appearing in face scans (e.g. nose tip, cheeks, eyes corners, etc.). This dictionary $\mathbf{Z}$ is then used to encode each 3D mesh as a vector $\mathbf{h} \in \mathcal{R}^K$ representing the frequency of each cluster $\mathbf{z}_k$ in the scan. For this purpose, we employ a Soft-Assignment approach [27], where each descriptor $\mathbf{x}_n$ is encoded as:

$$h_n^k = \frac{\exp(-||d(\mathbf{x}_n) - \mathbf{z}_k||^2)}{\sum_{j=1}^K \exp(-||d(\mathbf{x}_n) - \mathbf{z}_j||^2)}, \tag{8}$$

and the final vector representation is computed using a sum-pooling procedure as: $\mathbf{h} = \sum_{k=1}^K \mathbf{h}^k$. Finally, vectors $\mathbf{h}$ for all the training scans are used to train three different Least-Squares linear regressors for yaw, pitch and roll angles.

IV. Experiments

We used the recently published SASE 3D head-pose database [13], to assess the performance of our approach.

The data in SASE has been acquired with Microsoft Kinect 2 camera and contains RGB and depth images in pairs. The entire database includes 50 subjects (32 male and 18 female) in the range of 7-35 years old, with more than 600 frames per subject. For each person, a large sample of head poses are included, with wide range of yaw, pitch and roll variations.

For the Head Pose Challenge [11] organized at the International Conference on Automatic Face and Gesture Recognition (FG 2017), the SASE data has been divided in three sets: *Training* (comprising 28 subjects with a total of $\sim$ 17K images), *Validation* (12 subjects, $\sim$ 7K images) and *Test* (10 subjects, $\sim$ 6K images). Only the *Training* data was made available to challenge participants in order to investigate the performance of their algorithms prior to the final evaluation phase. Therefore, we first present detailed results of our system using only Training data (Section IV-A) and use them to choose the parameters that will be used for Validation and Test sets (Section IV-B). Notice that, although the SASE data contains both RGB and depth images, we only used depth data in order to comply with the participation requirements of the Head Pose Challenge.

A. Training

We started by splitting the training data into two subsets: *Development* and *Pre-test*. The Development set was composed of 840 images, by randomly choosing 30 images from each of the 28 training subjects. This set was used to train the landmarking algorithm and the dictionary-based estimate. The remaining images ($\sim$ 16K) were used as a preliminary test-set to asses system's performance and validate system parameters.

As explained in Section III, our system combines three different methods to estimate head pose: two of them are based on landmarks (geometric and appearance estimates) and the third one is dictionary-based. The three methods were developed independently and each has its advantages and shortcomings. Table I shows the results of each method applied separately on the entire Pre-test set. We can see that, as anticipated, landmark-based estimates are more accurate than dictionary-based estimates. On the other hand, for about 9% of the scans it was not possible to detect landmarks and only the dictionary-based estimates are available. Notice the comparatively large errors of the estimates in these scans, which confirm that these are especially challenging cases.

Within landmark-based methods, estimates based on appearance were slightly more accurate than geometric ones. However, we note that i) the geometric-based estimate is training-free while the appearance-based one requires a learning stage[4]; ii) combination of both estimates (by averaging) produced better results than each of them individually.

The dictionary-based approach was not as accurate as the landmark-based ones, but it was able to produce estimates in all cases. As explained in Section III-C, this method also requires learning, which was performed on our development set, fixing the number of clusters to $K = 500$.

[4]For the experiments reported in Table I, the appearance-based estimate was tested in a 10-fold cross-validation setting.

TABLE I
AVERAGE POSE ESTIMATION ERRORS ON THE TRAINING PART OF THE SASE DATABASE

Approach	**Pitch** (°)	**Yaw** (°)	**Roll** (°)	% **scans**
Scans with landmarks successfully detected				
Landmark-based geometric	6.34	6.42	7.33	90.8%
Landmark-based appearance	6.17	6.04	5.57	90.8%
Landmark-based combined	5.50	5.44	5.28	90.8%
Dictionary-based	8.74	8.06	5.89	90.8%
Scans with landmarks not detected				
Dictionary-based	14.74	14.10	9.83	9.2%
All scans				
Dictionary-based	9.29	8.61	6.25	100%
Combination	**6.33**	**6.10**	**5.46**	**100%**

The last line of Table I shows the final results of the system, obtained by combining the three methods. As indicated in Fig.1, landmark-based estimates were preferred over dictionary ones. However, if no landmarks were available or if geometric and appearance landmark-based estimates did not match, we used the dictionary approach. The rationale behind checking landmark-based estimates for agreement is that, if landmark are accurately located then both geometric and appearance estimates should produce similar results. On the other hand, if landmarks are located at incorrect positions, the estimates will also be incorrect but are unlikely to coincide among them, given the different nature of the estimators.

Therefore, given a head scan with geometric estimates $(\phi_G, \theta_G, \psi_G)$ and appearance estimates $(\phi_A, \theta_A, \psi_A)$, the system will use these estimates if and only if:

$$|\phi_G - \phi_A| + |\theta_G - \theta_A| + |\psi_G - \psi_A| < \tau \qquad (9)$$

Otherwise, the dictionary-based approach is used. Fig. 4 shows the variation of the estimation errors of the landmark- and dictionary-based approaches for different values of τ. The errors are displayed for each angle taking into account only the scans for which landmark-based estimates failed to comply with eq. (9). It can be seen that scans with larger differences between geometric and appearance estimates have larger pose estimation errors. Errors increase steadily $\forall\tau$ for landmark-based estimates and partially for the dictionary-based approach (approximately up to $\tau \leq 50$). The second observation from Fig. 4 is that, as expected, for large differences between geometric and appearance estimates the dictionary-based approach is more accurate. For the experiments reported in this paper, we have adopted a conservative value of $\tau = 50$ where Fig. 4 shows lower errors of the dictionary-based approach for all three estimated angles. As indicated by the black line in the plot, this represents replacing the landmark-based estimates by the dictionary ones in approximately 15% of the cases.

B. Validation and Test

Once we trained the necessary models and set the system parameters as described in the previous section, we submitted our estimates of head poses in the Validation and Test sets

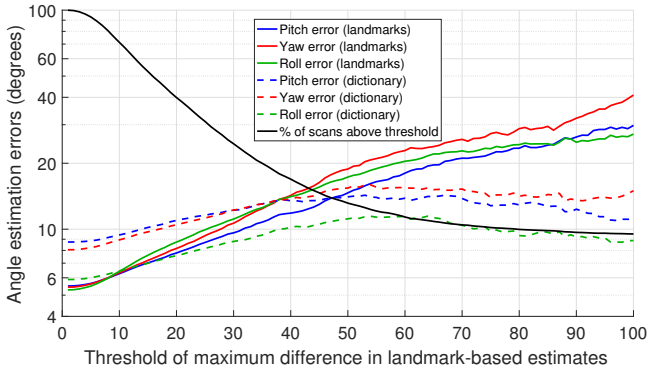

Fig. 4. Variation of estimation errors for landmark- and dictionary-based estimates as a function of the difference between geometric and appearance estimates. The black solid line shows the percentage of scans for which the difference in eq. (9) is above the value indicated in the horizontal axis. For those scans, the blue, red and green curves show the angle estimation errors based on landmarks (solid lines) and dictionary (dashed lines).

TABLE II
AVERAGE POSE ESTIMATION ERRORS ON THE SASE DATABASE

Subset	Pitch	Yaw	Roll	Sum
Training	6.33	6.10	5.46	17.89
Validation	7.82	6.65	5.39	19.86
Test	n/a	n/a	n/a	19.02

to the Head Pose Challenge. Table II summarizes our results, with which we obtained the first place in the challenge. It can be seen that the results at this stage were not too different from those obtained in training, indicating that there was not much over-fitting. Notice that, for the Test set, we can only provide the overall score (sum of average errors for all angles) since the ground truth for this part of the database was not made available by the challenge organizers.

Table III shows additional details about the performance of the proposed system in the validation and test sets. The average processing time reported correspond to tests on an Intel i7-770 processor at 3.4 GHz with 16 Gb or RAM. Most of the implementation was done in Matlab and it has not been optimized for speed. Full code of the system used to produce the reported results is publicly available.[5]

C. Comparison to other methods

Table IV shows the results reported by most relevant previous works addressing head pose estimation based on 3D data. Together with the estimation errors for each angle, we indicate whether the corresponding methods use tracking, RGB and/or depth data. Moreover, we show the specific database(s) used for testing in each case. As explained in Section II, most methods use temporal information to speed-up processing but also to avoid the need for initialization at every frame. This can considerably simplify the problem if sequences start from near-to-frontal shots, as is the case in most datasets used for head pose evaluation. However, this assumption do not need to

[5]https://github.com/DmytroDerkach/CMTech

TABLE III
DETAILED INFORMATION ABOUT THE PERFORMANCE OF THE PROPOSED SYSTEM ON THE VALIDATION AND TEST SETS

Overall statistics	
Total number of scans	13,885
Average processing time	8.42 s
Automatic landmarks	
Successfully detected	91.1 %
Average detection time	5.77 s
Landmark-based pose estimates	
Successfully computed	91.1 %
Agreement within $\tau = 50$	86.8 %
Average processing time	6.08 s
Dictionary-based pose estimates	
Computed for	13.2 %
Average processing time	3.29 s

be fulffiled in real-scenarios. Among the four methods listed in Table IV that do not rely on tracking, only one exclusively uses depth information (as our system) and the other three methods use both depth and RGB data.

Comparisons with our work using results depicted in Table IV are difficult and rather indirect considering the diversity of datasets and experimental setups that were used in the cited works. However, our results compares favorably to the best performing methods in the literature only relying on depth information. Moreover, performance of the proposed system is comparable to some algorithms that also use tracking for pose estimation. Finally, a detailed analysis of our results reveals that our average estimation errors are strongly influenced by the presence of outliers; e.g. our median absolute estimation errors were approximately 3.5 degrees per angle, considerably lower than the average absolute errors reported in Section IV. Analysis of these outlier cases revealed that they were typically scans with large rotation angles where the face was positioned quite oblique to the camera axis and the sensor could not capture it with sufficient quality.

V. CONCLUSIONS

In this work we present an approach for accurate head pose estimation from a single depth frame of consumer RGB-D cameras, such as Kinect 2. In contrast to most existing approaches, we base our system in the detection of 3D facial landmarks, whose positions are later used to derive geometry- and patch-based pose estimators. A key aspect of the proposed system is the use of state of the art landmark localization with no need for initialization and tolerance to occlusions or missing data. Our system is complemented with a secondary pose estimator based purely on patches sampled randomly on the head region to account for potential failures of the landmark-based estimation.

We evaluated our system on the SASE database, which consists of $\sim$ 30K frames from 50 subjects. We obtained average pose estimation errors between 5 and 8 degrees per angle, achieving the best performance in the FG2017 Head Pose Estimation Challenge. Our experiments also confirmed the initial hypothesis that the landmark-based estimates would

TABLE IV
AVERAGE ANGULAR ERRORS(IN DEGREES) FOR DIFFERENT EXISTING HEAD POSE ESTIMATION ALGORITHMS

Method	Errors			Tracking	Domain	Data
	ϕ	θ	ψ			
Valle [14]	12.6	-	-	No	RGB+ depth	AFLW/ AFW
Wang [15]	8.8	8.5	7.4	No	RGB+ depth	Biwi Kinect
Li [4]	8	12	4	No	depth	FRGC v2.0
Papazov [5]	2.5	1.8	2.9	No	RGB+ depth	Biwi Kinect
	3.0	2.5	3.8	Yes		
Meyer [9]	2.1	2.1	2.4	Yes	depth	Biwi Kinect
	2.9	2.3	-	Yes	depth	ETH
Padeleris [16]	2.4	3.0	2.8	Yes	depth	Biwi Kinect
Martin [17]	2.6	2.5	3.6	Yes	depth	Own
Baltrusaitis [10]	6.3	5.1	11.3	Yes	RGB+ depth	Biwi Kinect
	3.0	3.8	2.1	Yes		BU
	2.9	3.1	3.2	Yes		ICT-3DHP
Tulyakov [7]	4.7	7.6	5.3	Yes	RGB+ depth	Dali 3DHP
Fanelli [8]	8.9	8.5	7.9	Yes	depth	Biwi Kinect

be more accurate than correspondence-free approaches, such as the dictionary-based one that was adopted. Landmark-based estimates were successfully produced for $\sim 90\%$ of cases and the remaining ones were tackled by the dictionary-based approach. Our results compare well with those reported in the related literature, especially considering the added difficulty of not using tracking and RGB data to produce our estimates.

ACKNOWLEDGMENTS

This work is partly supported by the Spanish Ministry of Economy and Competitiveness under the Ramon y Cajal fellowships and the Maria de Maeztu Units of Excellence Programme (MDM-2015-0502).

REFERENCES

[1] E. Seemann, K. Nickel, and R. Stiefelhagen, "Head pose estimation using stereo vision for human-robot interaction," in *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*. IEEE, 2004, pp. 626–631.

[2] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 4, pp. 607–626, 2009.

[3] Y. Sun and L. Yin, "Automatic pose estimation of 3d facial models," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

[4] D. Li and W. Pedrycz, "A central profile-based 3d face pose estimation," *Pattern Recognition*, vol. 47, no. 2, pp. 525–534, 2014.

[5] C. Papazov, T. K. Marks, and M. Jones, "Real-time 3d head pose and facial landmark estimation from depth images using triangular surface patch features," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4722–4730.

[6] M. D. Breitenstein, D. Kuettel, T. Weise, L. Van Gool, and H. Pfister, "Real-time face pose estimation from single range images," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[7] S. Tulyakov, R.-L. Vieriu, S. Semeniuta, and N. Sebe, "Robust real-time extreme head pose estimation," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 2263–2268.

[8] G. Fanelli, T. Weise, J. Gall, and L. Van Gool, "Real time head pose estimation from consumer depth cameras," in *Joint Pattern Recognition Symposium*. Springer, 2011, pp. 101–110.

[9] G. P. Meyer, S. Gupta, I. Frosio, D. Reddy, and J. Kautz, "Robust model-based 3d head pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3649–3657.

[10] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "3d constrained local model for rigid and non-rigid facial tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2610–2617.

[11] I. Lüsi, J. C. S. Jacques Junior, J. Gorbova, X. Baró, S. Escalera, H. Demirel, J. Allik, C. Ozcinar, and G. Anbarjafari, "Joint challenge on dominant and complementary emotion recognition using micro emotion features and head-pose estimation: Databases," in *Automatic Face and Gesture Recognition, 2017. Proceedings. 12th IEEE International Conference on*. IEEE, 2017.

[12] F. M. Sukno, J. L. Waddington, and P. F. Whelan, "3-d facial landmark localization with asymmetry patterns and shape regression from incomplete local features," *IEEE transactions on cybernetics*, vol. 45, no. 9, pp. 1717–1730, 2015.

[13] I. Lüsi, S. Escarela, and G. Anbarjafari, "Sase: Rgb-depth database for human head pose estimation," in *Computer Vision–ECCV 2016 Workshops*. Springer, 2016, pp. 325–336.

[14] R. Valle, J. M. Buenaposada, A. Valdés, and L. Baumela, "Head-pose estimation in-the-wild using a random forest," in *International Conference on Articulated Motion and Deformable Objects*. Springer, 2016, pp. 24–33.

[15] B. Wang, W. Liang, Y. Wang, and Y. Liang, "Head pose estimation with combined 2d sift and 3d hog features," in *Image and Graphics (ICIG), 2013 Seventh International Conference on*. IEEE, 2013, pp. 650–655.

[16] P. Padeleris, X. Zabulis, and A. A. Argyros, "Head pose estimation on depth data based on particle swarm optimization," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 42–49.

[17] M. Martin, F. Van De Camp, and R. Stiefelhagen, "Real time head model creation and head pose estimation on consumer depth cameras," in *3D Vision (3DV), 2014 2nd International Conference on*, vol. 1. IEEE, 2014, pp. 641–648.

[18] F. M. Sukno, J. L. Waddington, and P. F. Whelan, "3-d facial landmark localization with asymmetry patterns and shape regression from incomplete local features," *IEEE transactions on cybernetics*, vol. 45, no. 9, pp. 1717–1730, 2015.

[19] F. Sukno, J. Waddington, and P. Whelan, "Asymmetry patterns shape contexts to describe the 3D geometry of craniofacial landmarks," *Computer Vision, Imaging and Computer Graphics - Theory and Applications. Communications in Computer and Information Science*, vol. 458, pp. 19–35, 2014.

[20] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *Proc. ECCV, LNCS vol. 3023*, 2004, pp. 224–237.

[21] F. Sukno, J. Waddington, and P. Whelan, "Comparing 3D descriptors for local search of craniofacial landmarks," in *Proc. 8th Int. Symp. on Visual Computing, Rethymnon, Crete, Greece. LNCS vol. 7432*, 2012, pp. 92–103.

[22] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, 1999.

[23] F. Tombari, S. Salti, and L. D. Stefano, "Unique signature of histograms for local surface description," in *Proc. ECCV*, 2010, pp. 356–369.

[24] R. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. ICRA*, 2009, pp. 3212–3217.

[25] F. Sukno, J. Waddington, and P. Whelan, "Rotationally invariant 3D shape contexts using asymmetry patterns," in *Proc. GRAPP*, 2013, pp. 7–17.

[26] X. Wang, L. Wang, and Y. Qiao, "A comparative study of encoding, pooling and normalization methods for action recognition," in *Asian Conference on Computer Vision*. Springer, 2012, pp. 572–585.

[27] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H. Van Nguyen, R. Ohbuchi *et al.*, "A comparison of methods for non-rigid 3d shape retrieval," *Pattern Recognition*, vol. 46, no. 1, pp. 449–461, 2013.