Ensemble Methods for Robust 3D Face Recognition Using Commodity Depth Sensors

Florin Schimbinschi
Department of Computing and Information Systems
University of Melbourne, Melbourne, Victoria, Australia
Email: florinsch@student.unimelb.edu.au

Lambert Schomaker and Marco Wiering
Institute of Artificial Intelligence and Cognitive Engineering
University of Groningen, Groningen, The Netherlands
Email: l.r.b.schomaker@rug.nl and m.a.wiering@rug.nl

***Abstract*—In this paper we introduce a new dataset and a pose invariant sampling method and describe the ensemble methods used for recognizing faces in 3D scenes, captured using commodity depth sensors. We use the 3D SIFT keypoint detector to take advantage of the similarities between faces, which leads to a set of points of interest based on the curvature of the face. For all keypoints, features are extracted using a 3D feature descriptor. Then, a variable-sized amount of features are generated per each 3D face image. The first ensemble method we constructed uses a K-nearest neighbors classifier to classify each keypoint-sampled feature vector as belonging to one of the subjects recorded in our dataset. All votes over all keypoints are combined. In the second ensemble technique, the keypoints are clustered with K-means, using the feature vectors and approximated sampling positions relative to the face. This leads to a set of experts that specialize for a specific region. Then a K-nearest neighbors classifier is trained on the examples falling in each expert's specialized region. Finally, for a new 3D face image, votes from all experts are combined in a sum ensemble technique to categorize the 3D face. We also introduce 6 new "real world" datasets with different variances: 3 types of 3D rotations, distance to sensor, expressions, and an all-in-one dataset. The results show very high cross validation accuracies for the same type of variance. In addition, 36 variance specific pair-tests in which the system is trained on one dataset and tested on a completely different dataset also show encouraging results.**

I. Introduction

Face perception is perhaps the most highly developed visual skill in humans. The evolution of our perceptual systems has taken a very long time to reach current capabilities. Machines do not necessarily have to possess the same type of sensors as we do, hence we make use of commodity infrared depth sensors, which do not require intensive preprocessing for extracting shape cues. Furthermore, depth sensors such as the Kinect offer several advantages compared to normal cameras, such as robustness to different lighting conditions and extreme pose angles.

Algorithmically, there are two major schools of thought in traditional face recognition: appearance based and geometric methods. The classical examples from the former category, Eigenfaces and Fisherfaces [1], [2] are focused mainly on the illumination structure, which does not necessarily coincide with actual shape cues. The surveys [3], [4] also conclude that performance is limited by the variations in illumination [5] and pose. Even adding texture information can not describe shape alone in a 2D structure [6].

978-1-4799-7560-0/15/$31

For 3D image recognition, 3D morphable models [7] and normalization increase performance, while video sequences [8]–[11] only add a limited increase in performance while adding to the complexity of the problem. Hence, in our methodology, we consider the observations to be time-independent.

Previous research [12], [13] shows that surface reconstruction using algorithms such as Poisson reconstruction [14] or Marching cubes [15] is not a fruitful research path. Virtually all previous work on 3D methods [16]–[21] struggle with the key problem of pose normalization and subsequently use algorithms which are computationally intensive. Unlike such methods, we take advantage of the similarities between faces and use a 3D version of the Scale Invariant Feature Transform (SIFT) [22] for sampling, thus bypassing the normalization step and also coping with occlusions through non-holistic processing.

Contributions. This paper presents two ensemble methods for 3D face recognition. First, 3D SIFT is used to detect points of interest based on the curvatures of the face. For all keypoints, features are extracted using a 3D feature descriptor. This leads to a variable-sized amount of features generated per 3D face image. The first ensemble method uses a K-nearest neighbors classifier to classify each keypoint as belonging to one of the persons recorded in our dataset. Then all votes over all keypoints are combined. In the second ensemble technique, first keypoints are clustered using K-means clustering using the extracted feature vectors and the approximated position in the face. This leads to a number of experts that specialize for a specific region. Then a K-nearest neighbors classifier is trained on the examples falling in its specialized region. Finally, for a new 3D face image, all votes from all experts are combined in a sum ensemble technique to categorize the 3D face. We furthermore constructed a new real world scenario dataset by recording 3D scans, which generate occlusions at extreme pose angles. In some datasets, the pose variance is typically obtained by artificially rotating 3D scans, however this is not the case here. Although our dataset has lower resolution than previous datasets such as [23], the variance in pose, expression and distance to sensor is much higher. We perform cross validation tests and pair-tests for each type of variance, as recorded as mini-sets for each subject. There is a variable number of observations per each subject, averaging at 43 frames per subject over all sets. The results show that both methods perform very well for the cross validation experiments. The average recognition accuracy is around 96% for the datasets

containing 18 different persons. Furthermore, also the pair-tests where the systems are trained on a dataset containing one type of variance and tested on a different dataset show promising results.

Paper Outline. In Section II we describe how we acquired our dataset and which preprocessing steps we have used. In Section III we describe our ensemble techniques and the workings of our complete approach. Experimental results are presented in Section IV, and Section V concludes this paper.

II. Data Acquisition and Preprocessing

In this section we start with presenting the main challenges involved in unconstrained 3D face recognition using Kinect sensors. We present the technical limitations of such sensors and continue to describe the procedure used to record the new dataset. The preprocessing steps — explained in Section II-C — are used to partially filter out artifacts (e.g. neck, hair) and slightly normalize all the 3D images to a frontal view.

A. Goals

The main challenges in unconstrained 3D face recognition are robustness towards pose variance, facial expressions and occlusions. We formulate the problem of recognizing a person which is able to move freely in front of the sensor. As such, we also consider the distance to the sensor (which decreases resolution) as a factor; this has not been attempted before. For this purpose we record a new dataset which contains all types of possible variance, with separate variance categories for each set and finally an all-in-one set. This last set is not an union of the other sets. The pipeline in the current classification process is fully automatic and does not require user intervention.

B. The sensor

According to the manufacturer's specifications of the Microsoft Kinect 360 Manual [1] the field of view is 57° on the horizontal plane and 43° on the vertical plane. Experimental measurements reveal that the random error of depth measurements increases quadratically with the distance from the sensor. More precisely, it is only a few millimeters at 0.5 meters from the sensor $\sim$ 0.25 cm at 1 meter from the sensor and almost reaches $\sim$ 0.5 cm at 1.5 meter from the sensor [24]. The highest recorded error is 4 cm at the maximum range of 5 meters. Therefore, due to the decreasing density of point clouds and the noise in the measurements, for the current objective, the recorded datasets were constrained to a maximum distance of 1.5 meters from the sensor.

C. Preprocessing

While the current research does not focus on pose normalization, some preprocessing steps were necessary in order to capture only the face shape from the entire point cloud. In the process we also aimed to eliminate noise and unwanted artifacts. A real time algorithm for head detection and pose estimation using regression forests [25] implemented in the Point Cloud Library (PCL) [26] is used to provide the position of the head in the form of a centroid and a vector estimating the head pose. A fixed size cube with the edge of 10 cm is used to crop the face. The cube can only perform translation movements.

Fig. 1. Left: zoomed in original point cloud. The cube delimits a fixed size volume around the head. The blue arrow is an estimation of the pose. Right: the data within the cube are cropped, then rotated towards a frontal position and finally translated towards the origin.

After the segmentation of the head, the cropped data are rotated in space according to the pose estimation vector such as to achieve a frontal face view towards the camera. Figure 1 shows an illustration of how the head is segmented and then pose normalized. The blue arrow is an estimation of the pose, however it should be noted that this vector oscillates, even when the pose is not changed.

The segmented face sometimes contains disconnected clusters of points such as hair, parts of the neck, jewelry, etc. In order to remove these artifacts, Euclidean clustering was used to remove small clusters of points disconnected from the main block. After the segmentation process is finished the clusters which do not contain at least 1000 points are removed. While this does not eliminate all artifacts it results in frames with less irrelevant data.

The head detector stochastically returns false positives (non-faces). Furthermore, the pose estimate is not always accurate, hence the detection process is followed by a maximum of 50 steps of Iterative Closest Points (ICP) [16] which slightly reduces the rotation variance and lowers the number of false positives.

A downsampled generic 3D face model was used as a target norm for the ICP alignment. Initially, the distance is computed using a spatial nearest neighbor search for finding correspondences, after which the transformation parameters are estimated using the Mean Squared Error (MSE) cost function. If the initial alignment error returned by the cost function is above a qualitatively determined threshold, then further processing of the current frame is immediately stopped and the frame is discarded. This reduces the number of non-faces and speeds up processing.

D. Dataset

The established benchmarks such as the FRGC v2.0 [27] are based on datasets which contain complete frontal models with no occlusions, slight pose variance and only fixed distance to the sensor. Thus, the most challenging problems are eliminated or not considered. Furthermore, the data are recorded in high resolution and are by far not as noisy as those generated by a commodity depth sensor such as the Kinect. Unlike the method of capturing some high quality frontal 3D frames we aim to classify using training data that also contain partial views (which can also be considered occlusions) and varying expressions. As such, we have recorded a new real

[1] http://support.xbox.com/en-US/xbox-360/manuals-specs/

world scenario, low resolution dataset which captures one type of significant variance per each subset: rotation (roll, pitch, yaw), distance to the sensor (or z-translation), expression and finally an unconstrained, all-in-one set. A visualization from the roll set for eight classes can be seen in Figure 2. For the recording of all 6 datasets, the 18 subjects were seated in front of the sensor and the height of the chair was adjusted in order to have the nose initially pointing at the sensor. The distance to the sensor was kept at approximately 0.6 meters, except for the z-translation and the unconstrained all-in-one set.

Fig. 2. Best case scenario. Eight classes depicted. The normalization is approximately solved. Frames (individual observations) are overlapped for each class and are slightly rotated towards the right. In the center, all data are overlapped.

For the first four datasets — I yaw, II pitch, III roll, IV z-translation — the subjects were asked to keep a neutral facial expression and not talk during the recording procedure. For set I the subjects were asked to move their heads from left to right and *vice-versa*. For set II the subjects were asked to move their heads up and down, while for set III the subjects were asked to tilt their heads either left-right as much as possible. In all three cases the maximum angle was $\pm 45^\circ$ in either direction. For set IV the subjects were asked to keep the X and Y position of the head fixed while the chair was moved towards and away from the sensor. The range in movement along the Z axis was between 0.4 meters and 1.3 meters. For recording set V the subjects were asked to talk and display various facial expressions while keeping the position of the head relative to the sensor constant. The same procedure was repeated for the unconstrained all-in-one set VI, only that the sensor was moved following a spiral pattern around the head, starting far from the subject and gradually getting closer. Also, in set VI the normalization and preprocessing step was bypassed. In all cases the movements were slow and incremental in order to capture the gradual changes in pose, translation and expression. This resulted in a varying number of observations per person per set.

The dataset was recorded in real-time and contains a total number of 18 subjects and is available for download [2]. For each observation (frame) the data are stored in form of an ASCII text file containing a matrix with columns X, Y, Z, RGB — each line thus having the coordinates and color values for one point (3D pixel). The color information was kept although it is not used here. Each observation / text file has a variable number of points. In total there are 4675 observations captured over all sets, with an average number of 260 frames per person for all sets (relevant for Table I) and an average of 43 frames per person per each set, which is relevant for Tables II and III. Even though the dataset was recorded with color, there were no efforts during the recording in order to vary the lighting conditions. The code for the experiments and framework used to record the dataset in real time is available for download [3].

III. Ensemble of Face Region Experts

According to traditional face recognition methodologies [4], non-holistic approaches generally have more flexibility and allow further classification possibilities based on different criteria. This *modus operandi* suggests that the non-holistic processing of face parts (eyes, nose, mouth, eyebrows, etc.) — as observed in [28] along with configural information — should result in superior robustness and inherent redundancy and scalability.

In virtually all of the 2D and 3D methodologies, these regions are segmented as disjoint sets. This requires the precise alignment of faces — pose normalization, which is unfeasible for most images. In the current research, we aim to bypass the fine-tuned normalization step, and propose a non-holistic ensemble method based on sampling non-uniformly using the Scale Invariant Feature Transform (SIFT) [22].

Although a similar sampling method has been proposed in [29], we stress that we present face recognition, not face verification, our dataset is high-noise and low resolution, it contains five types of extreme variances and the matching and feature extraction is performed differently. In their method, the feature vectors are ranked and the number of similarities becomes the score.

By making use of the 3D SIFT keypoint detector, different numbers of feature vectors are sampled from each face image. We developed two ensemble methods for using this variable number of feature vectors. In the first ensemble technique, one single expert is used that uses the K-nearest neighbors algorithm as classifier. For each feature vector a vote is generated for one of the classes. Finally, all votes are combined using the sum rule. In the second technique, the feature vectors are first clustered according to the sampling position of a keypoint and the similarity between shape, resulting in an automatic separation of the parts of the face. Then in each specialized face region a K-nearest neighbors classifier is trained, and all votes are combined using the sum rule. The intuitive advantage of this second method is the unsupervised method of identifying face regions which also removes the need for highly specialized computer vision algorithms for the detection of specific face regions — which might not perform well in the case of occlusions, extreme pose and expressions. The total working of the system is illustrated in Figure 3. We will now explain all steps involved in more detail.

[2] http://www.ai.rug.nl/~mwiering/Kinect_face_dataset.html

[3] https://github.com/florinsch/Ens3DFRKinect

A. Keypoint Sampling

We use the 3D version of SIFT to detect keypoints where the curvature is higher than some predetermined threshold. These keypoints give most information about the face shape since they are very likely to be located at different key positions. When using 3D SIFT, the distribution of the keypoints depends on the parameters used for SIFT. According to the minimum scale, the number of octaves and the number of scales per octave, the result can have high precision and low variance or *vice-versa*, which affects the consistency of the data sampling location.

When the precision is high, the keypoints are detected more accurately around the same regions between frames — for example a keypoint is either detected or not in the middle of the eyebrow. This was the case when the minimum scale was set to 0.3 cm with 5 octaves and 10 scales per octave, which resulted in sampling an average of 15 keypoints per frame, mostly around the eyebrows and the nose, with some regions completely lacking keypoints. When the precision decreases, there is less consistency of the location of keypoints between frames, however this results in an average of 50 keypoints per frame — for example, three keypoints will be detected around the eyebrow. For the images in Fig. 4 the minimum scale was set 0.4 cm with 4 octaves and 5 scales per octave. In both cases the minimum curvature was set to 0.1 cm since this provides a large initial keypoint sample for SIFT, however it is also a source of error since it allows keypoints to be selected from noisy fractal-like regions.

B. Feature Descriptor

The PCL library contains several implementations of feature descriptors for 3D object recognition. Although these are not optimal for 3D face recognition, they can also be used for 3D face recognition. In future work we intend to study other descriptors and focus more on the representation. An object recognition experiment [30] on the accuracy and time performance of several 3D feature descriptors implemented in

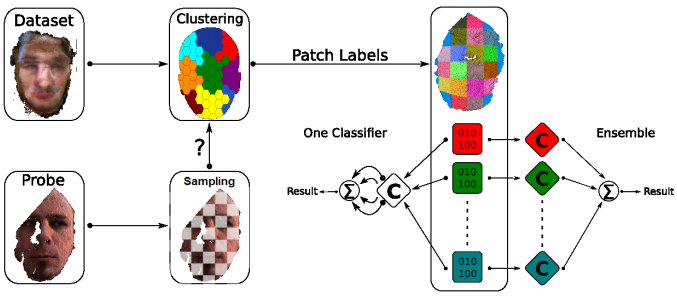

Fig. 3. Training (starts top left) *Step I*: Clustering is initially performed on the whole dataset, based on the SHOT feature vectors sampled using SIFT keypoints, face regions are learned. *Step II*: Supervised learning, an expert is trained per each face region (color rhomboids, each expert sees data only for a particular face region). A single classifier is also trained in parallel, regardless of face region. (white rhomboid, classifier sees all sampled data). Evaluation (starts bottom left) *Step I*: A probe arrives, keypoints are sampled using SIFT, the SHOT features are extracted, the clustering algorithm assigns each extracted feature vector to its specific face region expert. Some experts might not activate in case there is no data from that region due to occlusions, etc. *Step II*: Each expert outputs a probability vector with length equal to the number of classes. The results are combined using the sum rule (Eq. 4) to obtain the final class label.

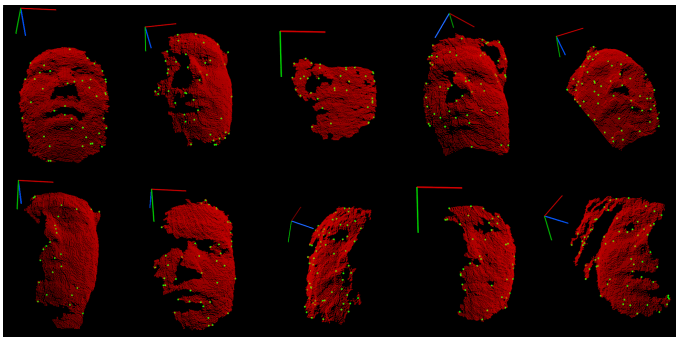

Fig. 4. Camera observations from several subjects. The SIFT keypoints (green) are detected in similar face regions. The camera observations come with missing data and are also pose variant. We use SIFT to take advantage of the similarities between faces. Features are extracted only in areas with high curvature, thus eliminating the need to compute features for the whole frame.

PCL gave some indications of possible descriptors. From these we tested three and selected the Signature of Histograms of Orientations (SHOT) [31], with feature vectors of $n = 352$ in length, which showed the best preliminary performance.

SHOT is based on computing a robust local reference frame using eigenvalue decomposition around a keypoint. A spherical grid is then centered on the same point and for each bin in the grid a weighted histogram of normals is computed according to a function of the angle between the normal at each point within the corresponding part of the grid bin and the normal at the keypoint.

The results are concatenated, the first 9 values represent the reference frame followed by 11 shape bins times the 32 bins — resulting from 8 azimuth divisions, 2 elevation divisions and 2 radial divisions of the spherical grid, with a total number of 352 values. To achieve robustness to variations of the point density, the whole descriptor is normalized to unit length.

C. Face Region Segmentation

For our second ensemble technique, we make use of the sampled keypoints returned by SIFT, and extract a feature vector from each keypoint using the SHOT descriptor. Once the keypoints are detected they are aligned using ICP to a downsampled version of the same 3D face template used during the preprocessing step. While pose normalization is computationally expensive for the whole frame, it can be efficiently done for the keypoints. Each feature vector is thus concatenated with the 3D coordinate of its extraction point. While this approach does not guarantee that the alignment is perfect, it allows us to take advantage of the approximate topological sampling location. It is evident that the normalization of the keypoints can not be entirely accurate since camera observations with extreme viewpoints or a large amount of missing data do not have enough correspondences between the keypoints and the template. However, while visualizing the pose-normalized keypoints, they showed higher consistency than the original topological positions. Finally, all keypoints with their approximated positions and their feature vectors are clustered using K-means clustering to create clusters representing different face regions in an unsupervised way.

The most challenging of the recorded datasets was VI which contains all types of variances. As such, the performance

of clustering using XYZ keypoint data or feature vectors for 36 clusters was compared. Furthermore, both the XYZ data and the feature vectors were length normalized and used as input for the K-means clustering algorithm. The performance was evaluated with leave one out cross validation using the sum rule of $E = 36$ KNN experts. The results showed that the concatenation of both keypoint location and the feature vector outperforms either method and was thus used in all further experiments.

D. Ensemble Learning

We use the K-nearest neighbors method as classifier, due to its speed and ease of scalability on multiple machines. The K-nearest neighbors (KNN) algorithm is a supervised non-parametric instance-based learning algorithm that has very strong consistency results which have been analytically proven [32]. The complexity of the decision boundary is a function of the number of neighbors K. The larger K is, the smoother the classification boundary.

It is common to use weights during the voting procedure, such that the closer the neighbor, the higher the contribution towards the average final vote. While this makes the decision boundary fuzzy, the weighting scheme is a way of un-biasing the classifier in cases where the number of examples for a particular class outnumbers the others.

The K-nearest neighbors algorithm is sensitive to the local structure of the data, hence the distance function is very important. In the case of high dimensional vectors, the Euclidean distance does not work well, since the distance to all neighboring points can be almost identical [32]. The sample negated correlation distance (Eq. 1) is a linear metric which is derived from the sample variance and covariance between two vectors and is a measure of multivariate independence. It is one if the vectors are statistically independent. This metric is often interpreted as an energy measure between two probability distributions. The distance function is applied as a metric for the KNN classifiers, as well as for K-means:

$$d_{corr}(a_i,\ b_i) = 1 - \frac{(a_i - \overline{a})^T\ (b_i - \overline{b})}{\sqrt{(a_i - \overline{a})^T\ (a_i - \overline{a})}\ \sqrt{(b_i - \overline{b})^T\ (b_i - \overline{b})}} \tag{1}$$

where a_i and b_i are two vectors and $\overline{a}$ and $\overline{b}$ denote the mean vectors from the datasets.

In ensemble learning, multiple experts are strategically combined to solve computational tasks such as classification or prediction. The effectiveness of such learning paradigms comes from the diversity of the experts. The distinction between experts can reside in the type of classification algorithm, the variance in parameters used during training of the same type of classifiers, in the type of features used for each classifier or using subsets of / re-sampling the training data (Bagging [33] or Boosting [34]).

By using disjoint subsets of the data per expert we aim to strengthen the overall performance, provided the convergence of the clustering algorithm. The clustering result should ideally represent different face regions, for any type of pose variation, or expression, etc. A favorable consequence of using subsets of training data is that we are less likely to require complex decision boundaries for the classifiers when increasing the number of examples N and the number of classes Ω. Since we want to observe the performance and impact of clustering and also compare the results of the ensemble, one single expert was trained using the same data. We call this one single classifier an ensemble since it makes use of the sum rule when making decisions. However, this ensemble does not take into consideration the clusters obtained in the clustering step.

Since the features correspond to specific areas of faces, in a similar way to bagging (although without replacement) the feature dataset $\mathcal{D}$ is partitioned in E distinct sets $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_E\}$ obtained during clustering with E denoting the number of clusters.

Even though the subsets $\mathcal{D}_{1\ldots E}$ are ideally disjoint, they are correlated by keeping track of the original camera observation $\mathcal{O}$ from where the features were extracted. Thus, from the complete dataset $\mathcal{D}$ with N examples, any camera observation $\mathcal{O}^i = \{\mathcal{O}^i_1 \in \mathcal{D}_1, \ldots, \mathcal{O}^i_E \in \mathcal{D}_E\}$ can be composed of a variable number of feature subsets. This happens since the clustering is not optimal and we can get multiple hits from one particular region per camera observation.

For each of the subsets $\mathcal{D}_\varepsilon$ with $\varepsilon \in [1, E]$, one expert C_ε is then trained. During testing, for any query frame, each expert can process more than one feature vector and thus return multiple results. In this case, the class posterior probability estimates P_ε are simply summed for each expert. Each camera observation $\mathcal{O}^i$ is selected and used as probe. Then, for each sampled feature vector the distance to each centroid is computed and the corresponding expert C_ε is activated for processing the feature vector.

Each expert C_ε returns a vector $P_\varepsilon(v) = \{P_\varepsilon(\omega_1|v), \ldots, P_\varepsilon(\omega_\Omega|v)\}$ with $v \in \mathcal{D}_\varepsilon$ which contains the posterior probability of the expert C_ε for each class ω. The posterior probability that a query feature vector v belongs to class ω is computed using equation 2 where $w_\Omega(k)$ is a weight vector computed as the inverse square distance $1/(d^2 + C)$ where C is a very small constant and d is the (negated correlation) distance between the query vector v and a neighboring vector:

$$P_\varepsilon(\omega|v) = \frac{\sum\limits_{k \in \mathcal{K}} w_\Omega(k)\ \Phi(v, k, \omega)}{\sum\limits_{k \in \mathcal{K}} w_\Omega(k)} \tag{2}$$

where:

$$\Phi(v, k, \omega) = \begin{cases} 1 & \text{if } k \in \mathcal{K}(v)\ ,\ \mathcal{K}(v) = \text{the closest neighbors} \\ & \text{and the class of } \ k = \omega \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The final ensemble decision is the class ω that receives the largest support after the sum rule is applied to the individual supports obtained by each expert:

$$\Omega(v) = \arg\max_{\omega} \sum_{\varepsilon=1}^{E} P_\varepsilon(\omega|v) \tag{4}$$

TABLE I. SINGLE KNN AND SUM RULE VS ENSEMBLES OF KNN EXPERTS. 10 FOLD CROSS VALIDATION RESULTS ON THE INDIVIDUAL SETS. HERE THE ACCURACY IS SHOWN AS THE PROPORTION OF THE NUMBER OF TEST EXAMPLES THAT ARE CORRECTLY CLASSIFIED. THE MOST CHALLENGING, AS EXPECTED IS THE ALL-IN-ONE DATASET (VI). THE NEXT MOST DIFFICULT SET IS YAW, WHERE THE TRANSLATION OF FRAMES IS SLIGHTLY SOLVED DURING NORMALIZATION. THE EASIEST SET IS ROLL WHERE THE FRAMES ARE ALMOST FRONTAL AND THE ROTATION VARIANCE IS SOLVED.

Dataset →	Yaw	Pitch	Roll	Z-Translation	Expressions	All-in-one	Average
N =4675	610	608	452	854	697	1454	779
Single	0.931	0.976	0.985	0.983	0.963	0.917	0.959
Ensembles	0.925	0.971	0.985	0.979	0.962	0.900	0.953

E. Parameters

In all cases the normalized feature vectors and approximate positions were used for clustering. Since the number of detected keypoints vary for each camera observation and features can be computed from keypoints which are in close proximity to each other (Fig. 4), this does not imply that we should have unique feature subsets per frame, that is, to have unique cluster "hits". The parameters for the KNNs and K-means clustering were selected during a 10 fold cross validation on set VI. Consequently, the number of neighbors was set to $K = 3$ to generate smoother decision boundaries which imply higher generalization on simpler sets hence less overfitting on the training set. The number of clusters was set to the average number of keypoints detected for this dataset, namely $E = 50$ in all cases. It should be noted that these are not necessarily the best parameters for all datasets.

IV. RESULTS

First, a 10 fold cross validation was performed using both methods depicted in Figure 3, with parameters $K = 3$ and $E = 50$. Although we have not included visualizations from each dataset for reasons of space, the challenge for each set is also consistent with the intuition that larger changes in pose, expression and distance to sensor imply a more difficult problem. This is also consistent with the results. Table I shows the cross validation results, in which the average error ranges between $\pm 2 \times 10^{-2}$ with a 99% confidence on all subsequent results.

The accuracies when using Ensembles of specialized experts are slightly lower in overall than with the single expert ensemble. We can notice a decrease of almost 2% accuracy on set VI (All-in-one), while the other results have decreased by 0.5% on average. Set VI has the lowest recognition rate and is also the dataset with the largest difference between using one single classifier and the ensembles method. The results show that there is no added benefit of using an ensemble of specialized classifiers. There could be several reasons for this. Firstly, the clustering might not have separated the face regions accordingly and in turn this can be due to the feature descriptor. Furthermore, each specialized expert in the second ensemble technique has much less data to learn from. This is because of the suboptimal splitting of the data into disjoint sets and as such the experts are trained with examples from a wrong face region, in effect leading those particular examples as unreliable. Since we do not use so many observations per class, the benefit of the specialization is diminished, because they have much less relevant data to learn from.

A. Single Expert Ensemble with Pair-tests

Although in most research only cross validation results are presented, we also computed the results of all 36 pair-tests in which the system was trained on one dataset and tested on a different one. With the right representation this would result in a transfer learning-like experiment. We first evaluated the ensemble method using the single expert. The results for all the pair-tests are displayed in Table II. The accuracy is reported with a 99% confidence interval, with the error in the order of $\pm 10^{-3}$ for all following pair-test tables.

The results show that the accuracies between the pairs are not symmetric. The last row, containing the average testing errors shows a similar pattern to the cross validation results: the lowest average *testing* accuracy is on all-in-one, followed by Yaw, with the highest on Roll. The absolute highest accuracy (bold) is observed when the Z-Translation dataset is used for *training* and Expressions is used for testing.

The lowest accuracy (last column, italic) was recorded when set VI was used for testing, since it contains all types of variances and is not pre-processed. The average accuracy (last column, 5th row) is also the lowest, for the same set. We also note that the overall accuracy using the negated correlation distance is almost 1.5% higher when compared to the results obtained using the Euclidean distance (not shown here). Finally, although the results of the pair-tests are much worse than those of cross validation, they are still quite good when we realize that the datasets are very different and that we are therefore testing the extrapolation power of the different methods.

B. Ensembles of Specialized Experts on Pair-tests

The results of the ensemble method with specialized experts are displayed in Table III where the overall accuracy is lower than when using the one single classifier architecture. There are however two cases when the accuracy is higher than the results in Table II, when training on Pitch and testing on Yaw and when training on Pitch and testing on Roll we can observe a 1% increase in accuracy.

Again, the overall lower accuracy is due to the fact that the features and the metric used do not cause the unsupervised identification of face regions to be optimal. Thus, the clusters do not represent perfect face regions. Furthermore, again each expert has much less training data to learn from.

V. DISCUSSION

We described a novel approach for recognizing faces from 3D recordings obtained with a Kinect sensor. The method makes use of 3D SIFT to sample keypoints with high curvature and uses a 3D feature descriptor to extract features describing the region around each keypoint. We compared two different algorithms that can deal with occlusions and the variable amount of extracted feature vectors. The simpler technique uses a classifier to classify each feature vector and uses the

TABLE II. SUM RULE OVER ONE KNN CLASSIFIER WITH $K = 3$ TRAINED USING THE NEGATED CORRELATION DISTANCE.

Testing → Training ↓ N =	Yaw 610	Pitch 608	Roll 452	Z-Translation 854	Expressions 697	All-in-one 1454	Average Train
Yaw	-	0.463	0.780	0.616	0.541	0.510	0.582
Pitch	0.349	-	0.765	0.717	0.741	0.324	0.579
Roll	0.495	0.652	-	0.684	0.672	0.355	0.572
Z-Translation	0.396	0.659	0.797	-	**0.840**	0.349	0.608
Expressions	0.415	0.636	0.631	0.683	-	*0.316*	0.536
All-in-one	0.602	0.476	0.536	0.521	0.602	-	0.548
Avg. Test	0.451	0.577	0.702	0.644	0.679	0.371	0.571

TABLE III. ENSEMBLES AND SUM RULE ACCURACY OVER PAIR-TESTS USING $K = 3$ NEIGHBORS AND $E = 50$ CLUSTERS. THE NEGATED CORRELATION DISTANCE WAS USED FOR K-MEANS AND THE 50 KNNS.

Testing → Training ↓ N =	Yaw 610	Pitch 608	Roll 452	Z-Translation 854	Expressions 697	All-in-one 1454	Average Train
Yaw	-	0.449	0.768	0.596	0.541	0.507	0.572
Pitch	0.357	-	0.773	0.715	0.725	0.320	0.578
Roll	0.499	0.645	-	0.677	0.650	0.328	0.560
Z-Translation	0.385	0.645	0.795	-	**0.839**	0.347	0.602
Expressions	0.413	0.604	0.616	0.673	-	*0.320*	0.525
All-in-one	0.569	0.460	0.518	0.515	0.450	-	0.502
Avg. Test	0.444	0.560	0.694	0.635	0.641	0.365	0.557

sum-rule to compute the final classification decision. The second approach uses K-means clustering to automatically generate face region experts, which then classify feature vectors only lying in their specialized region. We performed the experiments on a newly recorded real world scenario dataset containing 18 different persons. The results using both methods are very promising. For the easier cross validation experiments an average accuracy of 96% is obtained. For the more challenging transfer learning like setting — the pair-tests, we still obtain an average accuracy around 55-57% and furthermore, in some cases the accuracy reaches 84%. The experimental results also showed that the simpler ensemble technique slightly outperformed the more complex ensemble method. One reason is that the clustering of feature vectors did not result in perfect face regions. Another reason is that with the more complex technique each expert had less data to learn from. In future work, we are interested in learning better 3D representations for handling 3D face images. Sparse coding strategies have been extensively used for object recognition in video [11], [35], [36], and we also want to use these techniques. Furthermore, evaluations of purely appearance based methods tested under equal working conditions [37], [38] confirm that metrics are quite important. Hence, in the future, we also aim to include configural information — a validation of the agreement between the different face region experts. Finally, we remind the reader that this system can potentially be used for any face recognition task, even in complete darkness.

REFERENCES

[1] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Computer Vision and Pattern Recognition. Proceedings CVPR, IEEE Computer Society Conference on*. IEEE, 1991, pp. 586–591.

[2] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711–720, 1997.

[3] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 1, pp. 34–58, 2002.

[4] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *Acm Computing Surveys (CSUR)*, vol. 35, no. 4, pp. 399–458, 2003.

[5] W. Chen, M. J. Er, and S. Wu, "Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, no. 2, pp. 458–466, 2006.

[6] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, "Face recognition by humans: Nineteen results all computer vision researchers should know about," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1948–1962, 2006.

[7] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 9, pp. 1063–1074, 2003.

[8] S. Zhou, V. Krueger, and R. Chellappa, "Probabilistic recognition of human faces from video," *Computer Vision and Image Understanding*, vol. 91, no. 1, pp. 214–245, 2003.

[9] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1. IEEE, 2003, pp. I–313.

[10] I. Cohen, N. Sebe, A. Garg, L. S. Chen, and T. S. Huang, "Facial expression recognition from video sequences: temporal and static modeling," *Computer Vision and Image Understanding*, vol. 91, no. 1, pp. 160–187, 2003.

[11] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1470–1477.

[12] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Three-dimensional face recognition," *International Journal of Computer Vision*, vol. 64, no. 1, pp. 5–30, 2005.

[13] F. Schimbinschi, M. Wiering, R. E. Mohan, and J. K. Sheba, "4d unconstrained real-time face recognition using a commodity depth camera," in *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*. IEEE, 2012, pp. 166–173.

[14] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proceedings of the fourth Eurographics symposium on Geometry processing*. Eurographics Association, 2006, pp. 61–70.

[15] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *ACM Siggraph Computer Graphics*, vol. 21, no. 4. ACM, 1987, pp. 163–169.

[16] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Robotics-DL tentative*. International Society for Optics and Photonics, 1992, pp. 586–606.

[17] I. A. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis, "Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 4, pp. 640–649, 2007.

[18] W.-Y. Lin, K.-C. Wong, N. Boston, and Y. H. Hu, "3d face recognition under expression variations using similarity metrics fusion," in *Multi-*

media and Expo, 2007 IEEE International Conference on. IEEE, 2007, pp. 727–730.

[19] A. S. Mian, M. Bennamoun, and R. Owens, "An efficient multimodal 2d-3d hybrid approach to automatic face recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 11, pp. 1927–1943, 2007.

[20] T. C. Faltemier, K. W. Bowyer, and P. J. Flynn, "A region ensemble for 3-d face recognition," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 1, pp. 62–73, 2008.

[21] C. C. Queirolo, L. Silva, O. R. Bellon, and M. P. Segundo, "3d face recognition using simulated annealing and the surface interpenetration measure," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 2, pp. 206–219, 2010.

[22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[23] R. Min, N. Kose, and J.-L. Dugelay, "Kinectfacedb: A kinect database for face recognition," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 44, no. 11, pp. 1534–1548, 2014.

[24] T. Leyvand, C. Meekhof, Y.-C. Wei, J. Sun, and B. Guo, "Kinect identity: Technology and experience," *Computer*, vol. 44, no. 4, pp. 94–96, 2011.

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

[26] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[27] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, vol. 1. IEEE, 2005, pp. 947–954.

[28] Z. Li, J.-i. Imai, and M. Kaneko, "Robust face recognition using block-based bag of words," in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 1285–1288.

[29] D. Smeets, J. Keustermans, D. Vandermeulen, and P. Suetens, "meshsift: Local surface features for 3d face recognition under expression variations and partial data," *Computer Vision and Image Understanding*, vol. 117, no. 2, pp. 158–169, 2013.

[30] L. A. Alexandre, "3d descriptors for object and category recognition: a comparative evaluation," in *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal*, 2012.

[31] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 356–369.

[32] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.

[33] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[34] R. E. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, no. 2, pp. 197–227, 1990.

[35] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 490–503.

[36] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 2169–2178.

[37] B. A. Draper, K. Baek, M. S. Bartlett, and J. R. Beveridge, "Recognizing faces with PCA and ICA," *Computer vision and image understanding*, vol. 91, no. 1, pp. 115–137, 2003.

[38] K. Delac, M. Grgic, and S. Grgic, "Independent comparative study of PCA, ICA, and LDA on the FERET data set," *International Journal of Imaging Systems and Technology*, vol. 15, no. 5, pp. 252–260, 2005.