SIFT vs. FREAK: Assessing the Usefulness of Two Keypoint Descriptors for 3D Face Verification

Janez Križaj, Vitomir Štruc and Simon Dobrišek
Faculty of Electrical Engineering
University of Ljubljana
Tržaška 25, SI-1000 Ljubljana, Slovenia
Email: {janez.krizaj, vitomir.struc, simon.dobrisek}@fe.uni-lj.si

Darijan Marčetić and Slobodan Ribarić
Faculty of Electrical Engineering and Computing
University of Zagreb
Unska 3, 10000 Zagreb, Croatia
Email: {darijan.marcetic, slobodan.ribaric}@fer.hr

***Abstract*—Many techniques in the area of 3D face recognition rely on local descriptors to characterize the surface-shape information around points of interest (or keypoints) in the 3D images. Despite the fact that a lot of advancements have been made in the area of keypoint descriptors over the last years, the literature on 3D-face recognition for the most part still focuses on established descriptors, such as SIFT and SURF, and largely neglects more recent descriptors, such as the FREAK descriptor. In this paper we try to bridge this gap and assess the usefulness of the FREAK descriptor for the task for 3D face recognition. Of particular interest to us is a direct comparison of the FREAK and SIFT descriptors within a simple verification framework. To evaluate our framework with the two descriptors, we conduct 3D face recognition experiments on the challenging FRGCv2 and UMB-DB databases and show that the FREAK descriptor ensures a very competitive verification performance when compared to the SIFT descriptor, but at a fraction of the computational cost. Our results indicate that the FREAK descriptor is a viable alternative to the SIFT descriptor for the problem of 3D face verification and due to its binary nature is particularly useful for real-time-recognition systems and verification techniques for low-resource devices such as mobile phones, tablets and alike.**

I. Introduction

For many computer vision tasks using local image descriptors has become the norm rather than an exception over the last decades [1]. Image descriptors, such as the SIFT [2], the SURF [3] or the HOG [4] descriptor, established themselves as state-of-the-art tools for solving various vision-related problems ranging from object detection, recognition and tracking to image stitching and retrieval.

Due to their popularity image descriptors have also found their way into the area of 3D-face recognition, where they were again shown to ensure state-of-the-art recognition results (see e.g., [5], [6]). However, most of the available literature on this topic focuses on established descriptors, such as the SIFT or SURF, and largely neglects more recent descriptors, such ORB [7], BRIEF [8] or FREAK [9], which unlike SIFT or SURF are binary in nature and, therefore, computationally much simpler. Whether this is a consequence of the superiority of the SIFT and SURF descriptors when applied to a recognition task or pertains to other factors remains an open question.

In this paper we try to address this question and present a comparative assessment of the SIFT and FREAK keypoint descriptors when applied to the task of 3D face recognition. We use the two descriptors within the 3D face recognition framework originally presented in [10], and apply the framework to images from the FRGCv2 [11] and UMB-DB [12] databases. We assess the descriptors in terms of computational speed and descriptiveness (reflected in the verification performance). The results of our experiments suggest that the FREAK descriptor represents an appealing alternative to the SIFT descriptor and is capable of ensuring a competitive verification performance at a fraction of SIFT's computational cost.

The rest of the paper is structured as follows. In Sections II and III we briefly describe the theory underlying the SIFT and the FREAK descriptors, respectively. In Section IV we introduce the 3D-face-recognition framework used in our experiments and present the experimental results and our main findings. We conclude the paper with some final comments and directions for future work in Section V.

II. Scale Invariant Feature Transform (SIFT)

The Scale Invariant Feature Transform (SIFT), introduced in [2], represents one of the most popular approaches to keypoint detection and subsequent descriptor calculation. The SIFT algorithm features four important steps: *(i)* scale-space extrema detection, *(ii)* removal of unreliable keypoints, *(iii)* orientation assignment, and *(iv)* keypoint descriptor calculation. In the remainder of this section we briefly describe all the four steps.

A. Extrema detection

In the first step of the SIFT algorithm, interest points (or keypoints) are identified in the given image by searching for pixels that represent extrema of the Difference-of-Gaussian (DoG) scale-space. Here, the DoG scale-space is defined as a function $D(x, y, \sigma)$ that is produced through convolution of a variable-scale Difference-of-Gaussian filter and the input image, $I(x, y)$ [13], [2]:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (1)$$

with

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}, \quad (2)$$

where σ denotes the standard deviation of the Gaussian $G(x, y, \sigma)$ and k stands for a scaling factor that controls the Gaussian's size.

Local maxima and minima of $D(x, y, \sigma)$ are identified by comparing the given sample point with its eight neighbors as well as the nine neighbors in the scale above and below. If the point represents a local extreme, it is selected as a keypoint candidate.

B. Removal of unreliable keypoints

Not all keypoints detected with the procedure described above are actually used for keypoint descriptor calculation. The final keypoints are selected based on different measures of stability. During this step keypoints with low contrast and keypoints with poorly determined locations along edges are discarded.

Two criteria are used for the detection of the unreliable keypoints. The first criterion evaluates the value of $|D(x, y, \sigma)|$ for each keypoint candidate. If the value is below some threshold, the keypoint is removed, as this indicates that the keypoint was detected in an image area of poor contrast. The second criterion evaluates the ratio of the principal curvatures across an edge and the principal curvature perpendicular to this direction. Note that this is necessary as the DoG function will have strong responses across edges regardless of whether the keypoints at the edge responses are stable (i.e., they have corner-like properties) or not. Thus, for unstable keypoints the ratio will be large and vice versa, for stable keypoints the ratio will be small. Consequently, all keypoints candidates with the ratio below some threshold are retained, otherwise they are discarded.

C. Orientation assignment

In the third step of the SIFT algorithm an orientation is assigned to each keypoint by building a histogram of gradient orientations $\theta(x, y)$ weighted by the gradient magnitudes $m(x, y)$ from the key-point's local neighborhood:

$$m(x, y) = \sqrt{A(x, y)^2 + B(x, y)^2}, \tag{3}$$

$$\theta(x, y) = \tanh \frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)}, \tag{4}$$

where $L(x, y)$ is a Gaussian smoothed image, where the scale of the Gaussian is determined by the scale that closest to the scale at which the keypoint was detected, and $A(x, y) = L(x+1, y)-L(x-1, y)$ and $B(x, y) = L(x, y+1)-L(x, y-1)$. By assigning a consistent orientation to each keypoint, the keypoint descriptor can be represented relative to this orientation and, therefore, can be made invariant to image rotation.

D. Keypoint descriptor calculation

Once the keypoint locations and orientation of each keypoint are determined, a descriptor can be computed for each of the detected keypoints. This fourth step of the SIFT algorithm calculates the SIFT descriptors by first computing the gradient magnitude and orientation at each image point of the 16×16 keypoint neighborhood (Fig. 1 - left). The keypoint neighborhood is weighted with a Gaussian and then used to compute orientation histograms of subregions of the neighborhood, each subregion having a size of 4×4 pixels (Fig. 1 - right), with the length of each arrow in Fig. 1(right) corresponding to the sum of the gradient magnitudes near that direction within the region [2]. Each histogram typically features 8 bins making the final keypoint descriptor comprising $4 \times 4 \times 8 = 128$ elements. The keypoint coordinates of the descriptor as well as the gradient orientations are rotated relative to the keypoint orientation to achieve orientation invariance and the descriptor is ultimately normalized to enhance invariance to changes in illumination [13], [2].

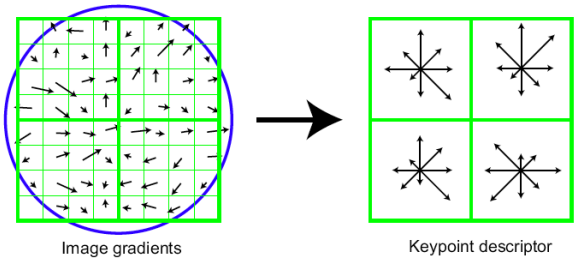

Fig. 1. In this figure the 2×2 subregions are computed from an 8×8 neighborhood, whereas in the experiments we use a 16×16 neighborhood and subregions of size 4×4 (image taken from [2]).

E. Matching

A procedure for matching the computed descriptors was also presented in [2] together with the SIFT algorithm. Consider a probe SIFT descriptor and some database of training descriptors[1]. The matching procedure first looks among the training descriptors for the the nearest neighbor of the probe descriptor. Generally, many descriptors do not have a good match among the training descriptors because they were either computed from different image features (and objects) or they arose from background clutter. To discard these descriptors a threshold is used based on which matches that are too ambiguous are discarded. The threshold is applied on the ratio between the distance to the descriptors closest neighbor and its second closest neighbor from the database of training descriptors. If the ratio is below a predefined threshold value the descriptors are declared a match.

III. Fast Retina Keypoint (FREAK) Descriptor

The FREAK (Fast REtinA Keypoint) [9] descriptor is a binary descriptor computed based on the results of brightness-comparison tests in a number of sampling locations around a keypoint. Unlike the SIFT algorithm, the FREAK approach does not include a keypoint detectior step, but relies on existing keypoint detectors - most often the AGAST corner detector [14].

A. Sampling Pattern

The sampling pattern adopted by the FREAK approach is biologically inspired by the retinal pattern in the eye. Thus, the sample points that form the basis for calculating the FREAK descriptor are arranged in the sample pattern shown in Fig 2.

[1]Typically the probe descriptor represents one descriptor computed from a probe image and the training descriptors represent all of the descriptors extracted from some training (or gallery) image. The goal here is to compare the probe and gallery images through descriptor comparisons.

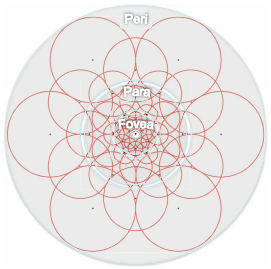

Fig. 2. FREAK sampling pattern (image taken from [9]). Red circles represents the standard deviations of the Gaussian kernels applied to the corresponding sampling points. A total of 43 sampling points are selected for the sampling pattern of the FREAK descriptor.

Before the descriptor is computed, the N sample points located around the given keypoint are smoothed with a Gaussian kernel. Here, the size of the kernel is varied with respect to the location of the sampling point to simulate the behavior of the human retina. In analogy to the human visual system, the smoothed image areas around the sampling points are referred to as receptive fields by the authors of [9]. The sampling points of the FREAK descriptor, hence, represent the centers of the receptive fields. Mathematically, this can be defined as follows:

$$P_i = P(x_i, y_i) = L_{r_i}(x_i, y_i), \tag{5}$$

where

$$L_{r_i}(x, y) = I(x, y) * G_{r_i}(x, y, \sigma_{r_i}). \tag{6}$$

In the above equations $I(x, y)$ stands for the input image, $G_{r_i}(x, y, \sigma_{r_i})$ denotes the Gaussian kernel for the i-th receptive field ($i = 1, 2, \ldots, N$) and $L_{r_i}(x, y)$ represents the smoothed version of the input image. The i-th sampling point P_i corresponding to the center of the i-th receptive field r_i and is defined with the predefined coordinates (x_i, y_i) from the sampling pattern, where $i = 1, 2, \ldots, N$.

B. Building the Descriptor

As indicated in the beginning of this section, the FREAK descriptor is constructed based on intensity comparisons between different pairs of smoothed sampling points (i.e., centers of receptive fields). Formally, this can defined as follows. Consider a pair of sampling points $P_a = (P_i, P_j)$, where $i, j \in \{1, 2, \ldots, N\}$ and $i \neq j$. The FREAK approach defines a binary encoded intensity comparison $s(P_a)$ on this pair as

$$s(P_a) = \begin{cases} 1, & \text{if } P_i > P_j, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

The presented comparison forms the basis for building the FREAK descriptor F as a N-dimensional bit string:

$$F = \sum_{0 \leq a < N} 2^a s(P_a). \tag{8}$$

The FREAK sampling pattern enables many pair-wise comparisons (binary tests) that would lead to a very large descriptor. However, since many of the pairs might not be useful for describing the content of an image, the authors of [9] run a training algorithm with the sampling pattern presented in Fig. 2 to identify useful pairs for building the descriptor. The final (trained) form of the FREAK descriptor, thus, defines 512 pairs from the sampling pairs that need to be tested in order to compute the bit-string in Eq. (8).

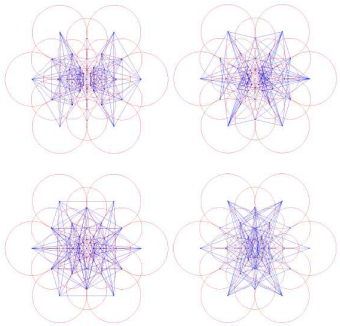

Fig. 3. Illustration of the four binary tests clusters composing the FREAK descriptor: peripheral receptive fields (top left), central (bottom right). The images were taken from [9]

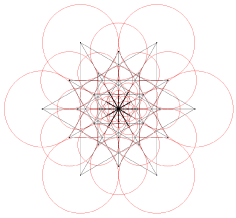

Fig. 4. Pairs selected to compute the keypoint orientation. The image was taken from [9].

Fig 3 shows the selected 512 sampling-point pairs grouped into four clusters of 128 pairs. Due to the orientation of the pattern along the global gradient, a symmetric pattern is captured in the clusters.

C. Orientation normalization

The orientation of the FREAK descriptor is estimated based on 45 selected sampling-point pairs that are arranged symmetrically with respect to the center of the sampling pattern (see Fig. 4). Let G be the set of all the selected pairs and assume that local gradients have been computed for all the selected sampling points, then the orientation o of the given keypoint can be computed as:

$$o = \frac{1}{M} \sum_{\substack{P_i, P_j \in G \\ i \neq j}} (P_i - P_j) \frac{T(P_i) - T(P_j)}{\|T(P_i) - T(P_j)\|}, \tag{9}$$

where M is the number of pairs in G and $T(P_i)$ denotes a function returning the 2D vector of the spatial coordinates of the center of receptive field, i.e., the vector of coordinates of the k-th sampling point $T(P_k) = [x_k, y_k]$.

D. Descriptor matching

The procedure for matching FREAK descriptors imitates the coarse-to-fine sacadic search of the human eye. Matching starts by considering only the first 128 bits of the FREAK descriptor carrying coarse information. If the computed Hamming distance is smaller than a predefined threshold, the matching proceeds by considering the remaining bits that represent finer information. With this procedure, more than

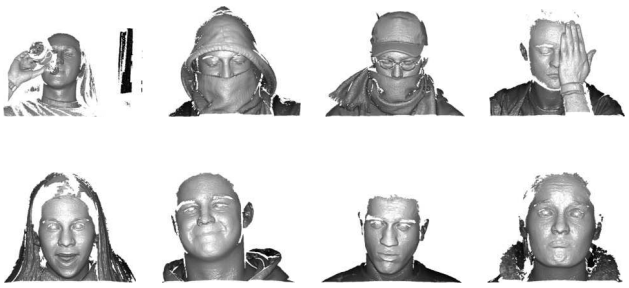

Fig. 5. Sample data from: the UBM-DB database (top row), the FRGCv2 database (bottom row)

90% of the candidate matches are discarded with the first 128 bits of the FREAK descriptor, resulting in an extremely rapid matching step.

IV. Experiments

A. Experimental databases

To assess the relative usefulness of the SIFT and FREAK descriptors for the task of 3D face recognition, we use two publicly available databases of 3D facial images, i.e., the FRGCv2 [11] and UMB-DB [12] databases. The FRGCv2 database serves for evaluating the recognition performance ensured by the descriptors within our 3D face recognition framework (presented in the next section) in the case of a large number of subjects with near frontal orientations and major expression variations. The UMB-DB databases is used to examine the robustness of our framework to occlusions given the two descriptors. Images from the two databases represent challenging problems for the existing 3D face recognition technology as evidenced by the sample images presented in Fig. 5. Here, the upper row depicts sample images from the FRGCv2 database and the lower row shows sample images from the UMB-DB database.

During our experiments we focus on the performance of our face recognition framework with respect to the two descriptors and do not use the otherwise more commonly used metrics for evaluation of detectors and descriptors (i.e. recall and precision), as suggested in [15]. Our experimental results are, therefore, mostly presented in the form of the verification performance (or true accept rate - TAR) at the 0.1% false accept rate (FAR) [16], [17], [18].

B. Experimental setup

For the experimental evaluation we use a similar recognition framework as presented in [10]. A diagram of the framework is presented in Fig. 6.

The processing chain of our framework starts by low-pass filtering the 3D scans to remove high frequency noise. The depth components (z values) are then interpolated and uniformly re-sampled on the (x, y) plane. After the re-sampling, face localization is performed with a technique relying on k-means clustering (similar to the one presented in [19]). With this technique, the facial area is extracted from the 3D scans by clustering the depth data into 3 distinct clusters that commonly correspond to the background, body parts and the face/head region. The cluster with the lowest average depth value typically corresponds to the facial area and is, therefore, retained for further processing. Note that the employed face localization procedure assures only a very rough localization of the facial region. However, it is computationally extremely simple and is able to localize the face even in the presence of severe occlusions, rotations and expression variations, where other localization techniques frequently fail [10].

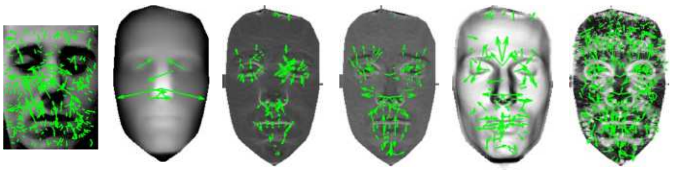

Fig. 7. Effect of data representation on the number of detected SIFT keypoints (with default parameters). From left to right: grayscale image, depth image, maximum curvature, mean curvature, z components of the surface normals, shape index (best viewed in color).

As the keypoint-descriptor-calculation methods are optimized for 2D images, it is of major importance in what form the 3D data is passed to the keypoint-detection and the descriptor-calculation procedure. With an unappropriate representation of the depth data, the keypoint detector will be unable to find a sufficient number of keypoints for the recognition procedure to work. Thus, the depth images need to be represented in a reasonable form for our assessment to make sense. Towards this end, we consider different representations of the surface shape to represent our depth data. Specifically, we use pure depth images $\mathbf{I}_r$, shape index values $\mathbf{I}_s$, mean curvature values $\mathbf{I}_m$, maximum curvature values $\mathbf{I}_{max}$ and surface normal coordinates $\mathbf{I}_{nx}$, $\mathbf{I}_{ny}$ and $\mathbf{I}_{nz}$ (see Fig. 7). An illustrative example of the effect of different representations on the keypoint detection step of the SIFT algorithm is shown in Fig. 7.

The keypoint-detection and descriptor-calculation steps of our framework, are implemented with the SIFT and the FREAK approaches. With the SIFT descriptor, the SIFT keypoint-detection procedure is used, while for the FREAK descriptor AGAST keypoint detector is adopted.

The final step in the processing chain of our framework is the matching stage. In this stage a similarity score (or matching score) measuring the similarity between the given probe and target images needs to be computed. Towards this end, each descriptor from the given probe image is matched independently against all descriptors extracted from the given target image. For the descriptor-matching procedure the technique proposed for the SIFT descriptor is used. Recall from Section II-E that the technique relies on the ratio between the distance to the nearest end second nearest neighbor [2], [15] - we will refer to this matching procedure as the "nearest-neighbor-ratio" matching in the reminder of the paper. Eventually, the number of matching descriptors between the two images serves as similarity measure for the given pair of probe and target images[2].

[2]Note that matching of the binary descriptors is performed using the Hamming distance (bitwise XOR followed by a bit count), which can be computed very efficiently on today architectures [20].

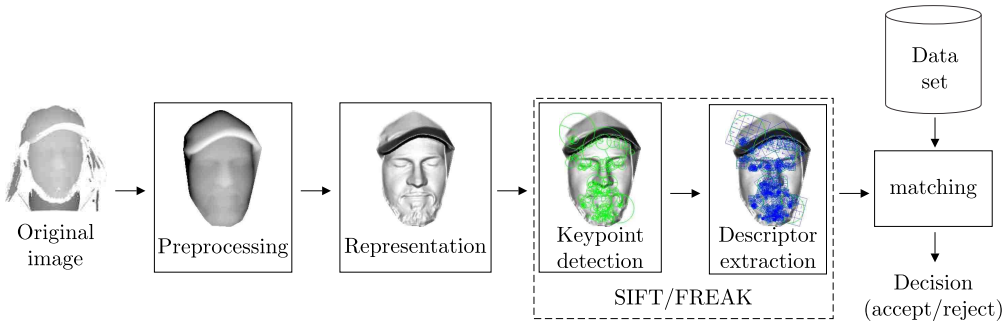

Fig. 6. Conceptual diagram of the 3D face recognition framework used in the experiments. During all of our experiments all steps were kept the same, except for the keypoint-descriptor-calculation step, which in one case was implemented with the SIFT algorithm and in the second with the FREAK approach.

TABLE I. INFLUENCE OF DIFFERENT 3D DATA REPRESENTATION TECHNIQUES ON THE KEYPOINT-DETECTION STEP AND THE VERIFICATION PERFORMANCE (TAR @ 0.1% FAR, FRGC V2, *neut. vs neut.*; ALL DESCRIPTORS ARE EXTRACTED ON THE SHAPE INDEX REPRESENTATION)

	Data representation for the keypoint detection				
Method	$\mathbf{I}_r$	$\mathbf{I}_{nz}$	$\mathbf{I}_{max}$	$\mathbf{I}_s$	$\mathbf{I}_m$
SIFT	21.5 (6)*	90.0 (72)	82.2 (62)	**94.3** (396)	81.6 (81)
FREAK	2.4 (3)	**92.5** (162)	78.1 (134)	92.4 (349)	76.8 (138)

* numbers in brackets denote the average number of detected keypoints per one 3D face image

TABLE II. INFLUENCE OF DIFFERENT 3D DATA REPRESENTATION TECHNIQUES ON THE DESCRIPTOR-CALCULATION STEP AND THE VERIFICATION PERFORMANCE (TAR @ 0.1% FAR, FRGC V2, *neut. vs neut.*; ALL KEYPOINTS ARE DETECTED ON THE SHAPE INDEX REPRESENTATION)

	Data representation for the descriptor extraction		
Method	$\mathbf{I}_r$	$\mathbf{I}_{nz}$	$\mathbf{I}_s$
SIFT	12.6	79.3	**94.3**
FREAK	72.8	85.6	**92.4**

C. Results

In the first series of our experiments we try to evaluate the impact of the selected 3D-surface-shape representation on the keypoint-detection and descriptor-calculation steps, and consequently on the recognition performance of our framework. For this series of experiments we use the FRGCv2 database and compute our performance metrics only on 3D facial images marked as neutral in the database. The experiments are conducted in a *all-vs-all* manner, thus, each image from the "neutral" subset of the FRGCv2 database is matched against all remaining images in this subset. Due to the selected setup the same images appear as probes and targets, which is common with this database [11], [10].

Table I presents the results of the first experimental run, where the keypoint-detection step is conducted on different 3D-shape-surface representations and the descriptors are computed from the shape-index representation $\mathbf{I}_s$[3]. Note that the best performance (considering both descriptor types) is achieved when both the keypoint-detection and descriptor calculation steps are conducted on the shape-index representation. We argue that this is due to an increased variability in the shape-index representation compared, for example, to the pure depth images, resulting in much more detected keypoints and thus better description of the face.

In the second experimental run of this series of experiments we conduct a similar experiment as in the first run, but this time use the shape-index representation to find the keypoints for the two descriptors and calculates the actual descriptors on different data representations. In Table II we present the results for the three top performing representations, all other representations were omitted from the table, as their performance was significantly worse from what is presented. Similar to the results in Table I the shape index is again the best representation, which can be explained by the increased robustness of the descriptors, resulting from the invariance of the shape index to scale, translation and rotation [21]. Based on the results of this series of experiments we select the shape-index representation for all our subsequent experiments.

In the second series of experiments we evaluate the robustness of the keypoint descriptors (within our framework) to variations in the facial expressions and presence to partial occlusions of the facial area. For this series of experiments we use both the FRGCv2 and UMB-DB databases. For the FRGCv2 database we match all images marked as neutral to all image marked as non-neutral. For the UMB-DB we match all non-occluded images against all occluded images. The results of the experiments are presented in Table III. The SIFT-based framework outperforms the FREAK-based one, but the differences in the performance is only around 3% on both databases. Furthermore, the SIFT-based framework is computationally much more expensive as shown in the graphs in Fig. 8, where the average time needed by our framework to process a single image (given a specific descriptor) is presented. Here, it has to be noted that our experiments were performed on a Intel Xeon CPU @ 2.67 GHz personal desktop computer with 12 GB of RAM. The implementation of the keypoint detectors and descriptor computation procedures is taken from OpenCV [22] and assessed through Matlab[4]. As can be seen from Fig. 8, the keypoint detection and descriptor calculation times for the FREAK-based framework are much lower than those of the SIFT descriptor.

[3]We have also experimented with settings, where the keypoint-detection and descriptor calculation steps were conducted on the same representation, but do not show the results here, as the performance was significantly worse than that tabulated in Tables I and II.

[4]http://www.cs.stonybrook.edu/~kyamagu/mexopencv/

TABLE III. TAR (%) AT A 0.1% FAR OF THE ASSESSED METHODS IN THE PRESENCE OF EXPRESSION, OCCLUSION AND ORIENTATION VARIATIONS.

Matching/classification	Data set			Descriptor	
	Name	Target	Query	SIFT	FREAK
nearest-neighbor-ratio	FRGC	*neutral*	*non-neut.*	81.2	77.8
	UMB-DB	*non-occl.*	*occluded*	78.2	75.0

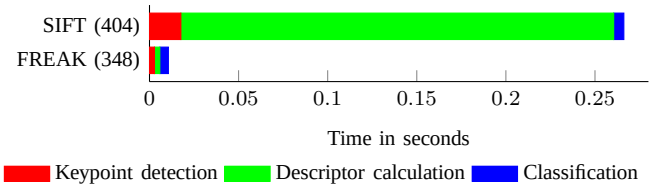

Fig. 8. Average times needed by different computational steps of the descriptors within our framework for the verification of one face image (numbers in brackets denote the number of detected keypoints).

All in all, the results of our experiments suggest that the FREAK descriptor represents a viable alternative to the SIFT descriptor for the task of 3D face recognition. Even with our simple recognition framework, both descriptors ensured high verification performance; in the case of the UMB-DB even comparable to the state-of-the-art (see, e.g., [23], [24]). When looking at the speed of computation, the FREAK descriptor definitely has an advantage compared to the SIFT descriptor and is also well suited for building recognition systems for low-resource devices such as mobile phones, tablets and alike.

V. CONCLUSION

We have assessed the relative usefulness of two keypoint descriptors, i.e., the SIFT and the FREAK descriptors, for the task of 3D face recognition. We have shown that despite its binary nature the FREAK descriptor is powerful enough to be used in 3D face recognition systems, where it ensures a recognition performance comparable to the SIFT descriptor, but at a fraction of the computational cost. For our future work, we plan to incorporate the FREAK descriptor into more elaborate recognition schemes, as the results of our experiments suggest that this represents a promising new research direction.

ACKNOWLEDGMENT

The work presented in this paper was supported in parts by the national research program P2-0250(C) Metrology and Biometric Systems, the European Union's Seventh Framework Programme (FP7-SEC-2011.20.6) under grant agreement number 285582 (RESPECT) and the post-doctoral project 3D-For-REAL (3D Face recognition in real world setting) funded partially by MIZŠ and the European Social Fund, M-1331E (H019001). The authors additionally appreciate the support of COST Actions IC1106 and IC1206.

REFERENCES

[1] H. Galoogahi, T. Sim, and S. Lucey, "Multi-Channel Correlation Filters," in *IEEE ICCV*, 2013. 1

[2] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004. 1, 2, 4

[3] H. Bay, T. Tuytelaars, and L. Gool, "SURF: Speeded Up Robust Features," in *ECCV*, ser. LNCS, A. Leonardis, H. Bischof, and A. Pinz, Eds. Springer Berlin Heidelberg, 2006, vol. 3951, pp. 404–417. [Online]. Available: http://dx.doi.org/10.1007/11744023_32 1

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005, pp. 886 – 893. 1

[5] 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, February 2013. 1

[6] G. Zhang and Y. Wang, "Robust 3D face recognition based on resolution invariant features," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 1009–1019, May 2011. 1

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," in *Proc. ICCV*, 2011, pp. 2564–2571. 1

[8] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *ECCV*, ser. LNCS, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin Heidelberg, 2010, vol. 6314, pp. 778–792. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15561-1_56 1

[9] R. Ortiz, "FREAK: Fast Retina Keypoint," in *Proc. CVPR*. Washington, DC, USA: IEEE Computer Society, 2012, pp. 510–517. [Online]. Available: http://dl.acm.org/citation.cfm?id=2354409.2354903 1, 2, 3

[10] J. Križaj, V. Štruc, and S. Dobrišek, "Combining 3D Face Representations using Region Covariance Descriptors and Statistical Models," in *FG Workshop*, 2013, pp. 1–7. 1, 4, 5

[11] 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 FRGC," in *CVPR*, 2005, pp. 947–954. 1, 4, 5

[12] A. Colombo, C. Cusano, and R. Schettini, "UMB-DB: A database of partially occluded 3D faces," in *ICCV W.*, 2011, pp. 2113–2119. 1, 4

[13] J. Križaj, V. Štruc, and N. Pavešić, "Adaptation of sift features for robust face recognition," in *Proceedings of the 7th International Conference on Image Analysis and Recognition (ICIAR 2010)*, Povoa de Varzim, Portugal, June 2010, pp. 394–404. 1, 2

[14] E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *Proceedings of the European Conference on Computer Vision (ECCV'10)*, September 2010. 2

[15] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE TPAMI*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2005.188 4

[16] B. Vesnicer and F. Mihelič, "The likelihood ratio decision criterion for nuisance attribute projection in gmm speaker verification," *EURASIP Jour. of Advances in Signal Processing*, vol. 2008, pp. 1–11, 2008. 4

[17] B. Vesnicer, J. Žganec Gros, and F. Mihelič, "Fusion of discriminative and generative scoring criteria in gmm-based speaker verification," in *Proceedings of Text, Speech and Dialogue (TSD 2011), Lecture Notes in Compute Science*, 2011, pp. 139–146. 4

[18] J. Žibert and F. Mihelič, "Fusion of acoustic and prosodic features for speaker clustering," in *Proceedings of Text, Speech and Dialogue (TSD 2009), Lecture Notes in Compute Science*, 2009, pp. 210–217. 4

[19] M. Segundo, C. Queirolo, O. R. P. Bellon, and L. Silva, "Automatic 3D Facial Segmentation and Landmark Detection," in *Proc. ICIAP*, 2007, pp. 431–436. 4

[20] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," in *Proc. ICCV*, 2011, pp. 2548–2555. 4

[21] N. Bayramoğlu and A. Alatan, "Shape Index SIFT: Range Image Rec. Using Local Features," in *Proc. ICPR*, 2010, pp. 352–355. 5

[22] A. Kaehler and G. Bradsky, *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc., 2013. 5

[23] A. Colombo, C. Cusano, and R. Schettini, "Three-dimensional occlusion detection and restoration of partially occluded faces," *J. Math Imag. Vis*, vol. 40, no. 1, pp. 105–119, 2011. 6

[24] N. Alyuz, B. Gokberk, and L. Akarun, "3-d face recognition under occlusion using masked projection," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 5, pp. 789–802, May 2013. 6