A Feasibility Study on the Use of Binary Keypoint Descriptors for 3D Face Recognition

Janez Križaj, Vitomir Štruc, and France Mihelič

Faculty of Electrical Engineering, University of Ljubljana,
Tržaška 25, SI-1000 Ljubljana, Slovenia
{janez.krizaj,vitomir.struc,france.mihelic}@fe.uni-lj.si

Abstract. Despite the progress made in the area of local image descriptors in recent years, virtually no literature is available on the use of more recent descriptors for the problem of 3D face recognition, such as BRIEF, ORB, BRISK or FREAK, which are binary in nature and, therefore, tend to be faster to compute and match, while requiring significantly less memory for storage than, for example, SIFT or SURF. In this paper, we try to close this gap and present a feasibility study on the use of these descriptors for 3D face recognition. Descriptors are evaluated on the three challenging 3D face image datasets, namely, the FRGC, UMB and CASIA. Our experiments show the binary descriptors ensure slightly lower verification rates than SIFT, comparable to those of the SURF descriptor, while being an order of magnitude faster than SIFT. The results suggest that the use of binary descriptors represents a viable alternative to the established descriptors.

Keywords: keypoints, descriptors, face recognition, 3D images

1 Introduction

Keypoint descriptors pervade in many computer vision tasks, such as object detection, object recognition, image stitching and retrieval and are becoming increasingly popular in the area of 3D face recognition as well. In 3D face recognition systems, the use of keypoint descriptors is motivated by the fact, that different sources of variability plaguing 3D face recognition, such as illumination, changes in facial expression or occlusions, are considered to be either local in nature or their effect is more easily eliminated at the local level. Thus, local keypoint descriptors present an appealing tool for 3D facial surface analysis.

Among descriptor-based techniques proposed for the problem of 3D face recognition, techniques relying on SIFT [12] and SURF [1] features dominate the literature [8,7,2], while only little attention is given to other alternatives, despite the fact that a lot of progress has been made in the area of image descriptors over recent years. Particularly interesting in this regard are the recently proposed binary descriptors such as BRIEF [4], ORB [18], BRISK [10] and FREAK [15], which were shown to represent a viable alternative to established descriptors such as SIFT or SURF for various computer vision task, but to the best of our knowledge have not yet been considered for the task of 3D face recognition.

In this paper we try to bridge this gap and present a feasibility study on the use of binary descriptors (i.e. BRIEF, ORB, BRISK and FREAK) for 3D face recognition. We assess the descriptors within a 3D face recognition framework similar to the one presented in [9]. We evaluate the performance of the descriptors on three publicly available datasets of 3D face images, namely, the FRGC, UMB and CAISA and show that the binary descriptors represent a viable alternative to SIFT and SURF, as they ensure comparable recognition performance but at the fraction of the computational burden. The findings of our analysis suggest that more research efforts should be devoted to techniques exploiting binary descriptors for 3D face recognition as they seem suitable for building high-performance low-computational cost recognition systems.

The rest of the paper is structured as follows. Section 2 summarizes related work on the topics of image descriptors that can be found in the literature. Section 3 discusses the methodology for the evaluating the descriptors. Section 4 presents the experimental evaluation and Section 5 concludes the paper with some final remarks.

2 Related Work

Computing a local keypoint descriptor typically requires two steps: *i)* detecting a point of interest in an image - the keypoint, and *ii)* computing a descriptor (i.e., a feature vector) in the detected keypoint. Below we briefly survey some of the techniques used for both steps, which together form a descriptor extraction procedure.

2.1 Keypoint detectors

Keypoint detectors generally find the interest points in an image by searching for points that have corner-like properties. One such approach is the Harris corner detector, proposed in [6], which is often used in conjunction with SIFT and SURF descriptors.

The SIFT descriptor typically relies on a keypoint detector that searches for maxima in the DoG (Difference of Gaussian) scale-space to detect corner-like and edge-like points. Here, the reason for using the DoG scale-space is to achieve scale invariance when searching for keypoints, while unstable edge-like points are removed by the Harris corner detector. The SURF [1] keypoint detector relies on a similar approach, but uses the determinant of the Hessian matrix to detect keypoints.

The more recent FAST [17] keypoint detector is very popular in real-time applications. To classify a pixel as a keypoint, FAST requires at lest p consecutive pixels in the s surrounding circle either being darker or brighter than the center pixel (generally $p = 9$ and $s = 16$). FAST does not produce multi-scale keypoints unless combined with a scale pyramid of an image. A graphical representation of the FAST-keypoint detector approach is shown in Fig. 1. An extension of the FAST was proposed in [13] in the form of the AGAST keypoint detector.

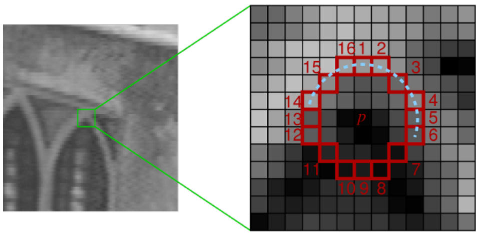


Fig. 1. FAST keypoint detector [17].

The FAST keypoint detector is typically used in conjunction with the BRIEF and ORB descriptors, while the multi-scale AGAST detector is commonly employed with the BRISK and FREAK descriptors.

2.2 Keypoint descriptors

One of the most popular representatives from the group of keypoint descriptors is SIFT. The SIFT descriptor is obtained from a grid of histograms of oriented gradients. SIFT proved to be highly descriptive and robust to affine image transformations, however, relatively slow to compute and match due to its high dimensionality (128 floating point values). The SURF descriptor is faster to compute and match than SIFT, while with similar matching performance. Similar to SIFT, SURF also relies on local orientation histograms, but uses sums of Haar-like features for histogram computation.

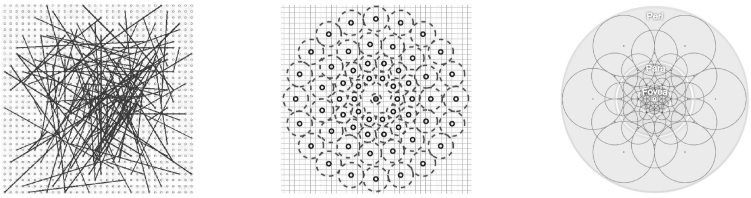
Binary descriptors are designed for high-speed descriptor computation and matching in real-time systems. Binary descriptors are generally obtained by concatenating the results of simple brightness comparison tests. Each test compares the smoothed intensity of two specific pixels on the patch centered at a keypoint, resulting in either 1 or 0, depending on which intensity value in the pair is greater. Different binary descriptors differ in the kernels used to smooth the patches before intensity differencing as well as in the spatial arrangement of the pixel pairs inside the keypoint patch.

The BRIEF descriptor, for example, pre-smooths the whole patch with the Gaussian kernel, while the pixel pairs are sampled from a Gaussian with the center at the keypoint and variance as shown in Fig. 2a. BRIEF is sensitive to scale variation as well as to in-plane rotations.

The ORB descriptor (oriented FAST and rotated BRIEF) is similar to BRIEF, with the difference that ORB also assures invariance to in-plane rotations.

The BRISK descriptor uses deterministic sampling from the points arranged in the sampling pattern shown in Fig. 2b where small circles denote sampling locations and larger dashed circles denote Gaussian kernels used to smooth the intensity values at the sampling points. BRISK is robust both to scale and to rotations.

The FREAK descriptor adopts a biologically inspired sampling pattern (see Fig. 2c). FREAK, similarly as ORB, selects the best pixel pairs from some



(a) BRIEF (b) BRISK (c) FREAK

Fig. 2. Binary descriptors: (a) random sampling of pixel pairs in BRIEF [4], (b) BRISK sampling pattern [10] and (c) FREAK sampling pattern [15].

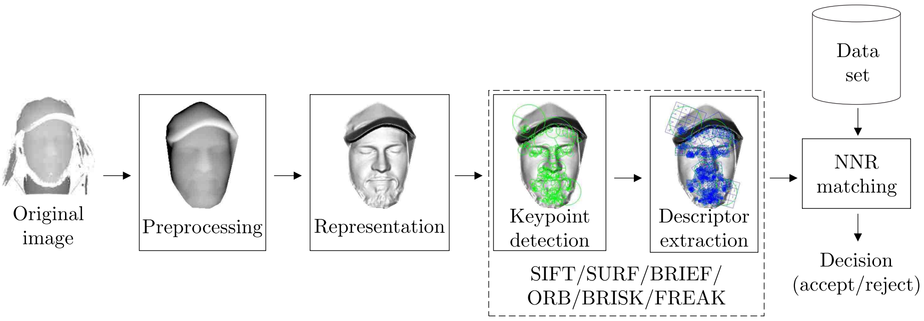


Fig. 3. Conceptual diagram of the evaluation system.

training data in a way that maximizes the variance of the descriptor elements and minimizes their correlation.

3 Methodology

In this section the employed framework for 3D face recognition is described. First, each 3D image is preprocessed, the facial region is localized and segmented from the 3D scan. Localized face images are then represented by various local surface shape metrics. Next, keypoints are detected and descriptors are extracted from each face image. Classification is based on the nearest neighbor ratio method as typically used with the SIFT descriptor [7,11]. Each step of the framework is described in more detail in the following sections, while the schematic diagram of the framework is presented in Fig. 3.

3.1 Image Preprocessing and Localization

Images are initially low-pass filtered to remove high frequency noise, while depth components (z values) are interpolated to a grid of 1.0 mm resolution on the (x, y) plane.

The face localization is similar to the one presented in [19] and is based on K-means clustering. Setting the number of clusters to $K = 3$, this method divides the 3D face image into three regions that most likely correspond to background,

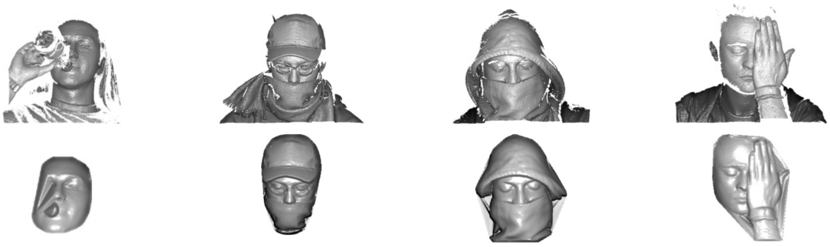
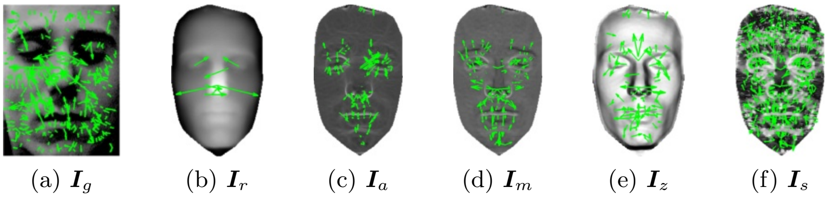


Fig. 4. Examples of original (top row) and localized (bottom row) 3D face images.



(a) $\boldsymbol{I}_g$ (b) $\boldsymbol{I}_r$ (c) $\boldsymbol{I}_a$ (d) $\boldsymbol{I}_m$ (e) $\boldsymbol{I}_z$ (f) $\boldsymbol{I}_s$

Fig. 5. Variation in the number of detected SIFT keypoints on a different data representations: (a) grayscale, (b) range, (c) maximum curvature, (d) mean curvature, (e) z components of the surface normals, (f) shape index.

body and face/head region. The face region is then selected as the cluster with the lowest average depth value.

This face localization procedure assures only rough localization of the facial region (see Fig. 4). However, it is computationally very simple and is able to localize a face even in the presence of severe occlusions and rotations.

3.2 3D Data Representation

Since the assessed keypoint detection and descriptor calculation methods are optimized for the use on 2D images, it is of major importance how 3D data is passed to the keypoint detection and the descriptor calculation module. With an inappropriate representation, the keypoint detector is unable to find a sufficient number of keypoints for the recognition procedure to work. Thus, the facial image needs to be represented in a reasonable form for our assessment to make sense.

In this regard, we consider different metrics of the surface shape, such as range images $\boldsymbol{I}_r$, shape index values $\boldsymbol{I}_s$, mean curvature values $\boldsymbol{I}_m$, maximum curvature values $\boldsymbol{I}_a$ and surface normal coordinates $\boldsymbol{I}_x$, $\boldsymbol{I}_y$ and $\boldsymbol{I}_z$ (see Fig. 5) to be used for the keypoint detection and descriptor calculation steps. In Section 4 we demonstrate how both the keypoint detection and the descriptor calculation steps, are affected by the employed 3D shape representation type. An illustration of the effect of different representations on the keypoint detection step is shown in Fig. 5.

3.3 Keypoints and Descriptors

For the keypoint-detection and descriptor-extraction steps of our framework, we consider all techniques presented in Section 2 - from the established SIFT and SURF descriptors, to the relatively novel binary descriptors like BRIEF, ORB, BRISK and FREAK. While SIFT and SURF methods have its own implementation of keypoint detectors, we use the FAST detector for BRIEF and ORB descriptors and the AGAST keypoint detector for the BRISK and FREAK descriptor.

3.4 Matching

In the last step of our evaluation framework a similarity matrix between the query and the target images needs to be generated, based on which classification of the query images is performed.

The similarity between the two face images, each represented with a number of descriptors, is generally measured based on the number of matching descriptors. Each descriptor from a given query image is matched independently against all descriptors extracted from one target image. Several methods exist that define when the two descriptors match. In this work, we use the most common method called nearest-neighbor ratio (NNR), which was originally introduced in [14]. With this technique, a descriptor from a query image is matched to its nearest neighbor in a target image if the distance ratio between the first and the second nearest neighbor is below some predefined threshold. In this way, ambiguous matches are typically eliminated. Eventually, the number of matching descriptors between the two face images serves as similarity measure.

Matching of binary descriptors is typically performed using the Hamming distance (bitwise XOR followed by bit count), which can be computed very efficiently on today architectures [10]. Note that it is of paramount importance that the correct (i.e. Hamming) distance is used with the binary descriptors. Our preliminary experiments in fact showed that the recognition results using the Euclidean distance are significantly worse than those achieved by applying the Hamming distance.

4 Experiments

For the experimental assessment of the descriptors, we utilize three datasets, i.e. the FRGCv2 [16], UMB-DB [5] and CASIA dataset. The FRGCv2 dataset serves for evaluating the recognition performance ensured by the descriptors in the case of a large number of subjects with near frontal orientations and major expression variations; the UMB-DB dataset is used to examine the robustness of the descriptors (within our framework) to occlusions; while the robustness to pose variations is assessed on the CASIA dataset.

We mainly focus on the overall performance of the face recognition system and thus do not use the otherwise more commonly used metrics for evaluation of detectors and descriptors, i.e. recall and precision, defined in [14]. Therefore, the experimental results show the verification performance and are presented in

Table 1. Influence of different 3D data representation techniques on the keypoint detection step and the recognition rate (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	$\boldsymbol{I}_r$	$\boldsymbol{I}_z$	$\boldsymbol{I}_a$	$\boldsymbol{I}_s$	$\boldsymbol{I}_m$
SIFT	21.5 (6)*	90.0 (72)	82.2 (62)	**94.3** (396)	81.6 (81)
SURF	52.1 (12)	90.9 (107)	88.9 (111)	**91.1** (243)	86.4 (107)
BRIEF	1.9 (3)	86.9 (200)	72.8 (134)	**89.9** (337)	66.8 (132)
ORB	9.6 (3)	89.9 (200)	82.0 (144)	**91.0** (345)	83.7 (140)
BRISK	0.0 (0)	87.9 (179)	89.0 (182)	**91.0** (301)	87.4 (191)
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 face image

the form of the verification rate (true accept rate, TAR) at a 0.1% false accept rate (FAR).

4.1 Data Representation Assessment

In the first series of experiments we aim at selecting the most appropriate 3D surface-shape representation that eventually serves as input for the keypoint detection and descriptor extraction steps. Recall, that this test is necessary to establish which facial representation is best suited for keypoint detection. From the results in Table 1, where the results of this series of experiments is presented, we can see, that keypoint detection from the shape index representation $\boldsymbol{I}_s$ gives the best recognition results. We argue that this is due to increased variability in shape index representation, resulting in much more detected keypoints and thus better description of the face. Likewise, the highest recognition rate is achieved if descriptors are extracted from the shape index representation, as can be seen in Table 2. This can be explained by increased robustness of descriptors, resulting from the invariance of shape index to scale, translation and rotation [2].

4.2 Robustness Evaluation

In this series of experiments we evaluate the robustness of the keypoint descriptors to expressions, occlusions and pose variations. Next to the comparison of different keypoint descriptor methods, we also compare the standard NNR matching to an alternative matching approach where descriptors are first modeled by the Gaussian mixture models (GMM) via maximum a posteriori adaptation of universal background model, while matching is performed based on the GMM parameters using support vector machines (SVMs) (see [9] for details). Note that the second matching approach is a variation of the Bag-of-Words model, which is commonly used for the task of object recognition, with the difference that the $k-$means clustering step is replaced with the GMM modeling. This second

Table 2. Influence of different 3D data representation techniques on the descriptor computation step and the recognition rate (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	$\boldsymbol{I}_r$	$\boldsymbol{I}_z$	$\boldsymbol{I}_s$
SIFT	12.6	79.3	**94.3**
SURF	78.3	88.4	**91.1**
BRIEF	50.3	82.4	**89.9**
ORB	61.2	81.2	**91.0**
BRISK	53.3	67.3	**91.0**
FREAK	72.8	85.6	**92.4**

matching procedure is introduced here as one could voice misgivings that the NNR technique is better suited for the SIFT and SURF descriptors than it is for the binary descriptors.

The results of this series of experiments are presented in Table 3. The upper half of the table summarizes the performance of the NNR framework, while the lower part of the table resumes the GMM-SVM performance. In the presence of expression variation, we can see that GMM-SVM classification outperforms the NNR matcher. On more challenging images from the UMB-DB and CASIA dataset the NNR matcher is mostly superior despite the reputation of the GMM-SVM classifier for its generalization strength that reflects in the robustness to occlusions, missing data and expression variations.

The SIFT-based method for the most part outperforms the other methods, but at the expense of higher computational cost as is observed in Section 4.3. When using the NNR matching approach, FREAK generally performs better than other binary descriptors and also better than SURF. As is expected, the BRIEF descriptor, being scale and rotation sensitive, achieves the lowest performance when using the NNR matcher. However, when coupled with the GMM-SVM classifier, BRIEF surprisingly gives the best results among the assessed methods. We argue that this is due to the fixed pattern of BRIEF descriptor (no orientation and scale normalization). Oriented keypoints present a more uniform appearance to descriptor computation, therefore, the descriptor variance is reduced resulting in diminished discriminativeness of the descriptors and poor estimation of GMM models.

All in all, the results suggest that while the SIFT descriptor still ensures the highest recognition rates, the binary descriptors (especially the FREAK descriptor) are not far behind. As will be shown in the next section, the binary descriptors have a significant edge when it comes to the matter of computational complexity.

Table 3. TAR (%) at a 0.1% FAR of the assessed methods in the presence of expression, occlusion and orientation variations.

	Dataset			Method					
Cls.	Name	Target	Query	SIFT	SURF	BRIEF	ORB	BRISK	FREAK
NNR	FRGC	*neutral*	*non-neut.*	81.2	71.1	66.5	73.5	72.5	77.8
	UMB	*non-occl.*	*occluded*	78.2	64.3	48.2	56.7	70.9	75.0
	CASIA	*frontal*	*non-fron.*	53.1	32.4	21.5	36.9	32.6	34.2
SVM	FRGC	*neutral*	*non-neut.*	84.4	73.4	85.3	75.8	78.8	68.8
	UMB	*non-occl.*	*occluded*	64.4	49.3	50.1	47.4	49.6	42.6
	CASIA	*frontal*	*non-fron.*	36.6	30.9	34.5	24.7	20.6	18.5

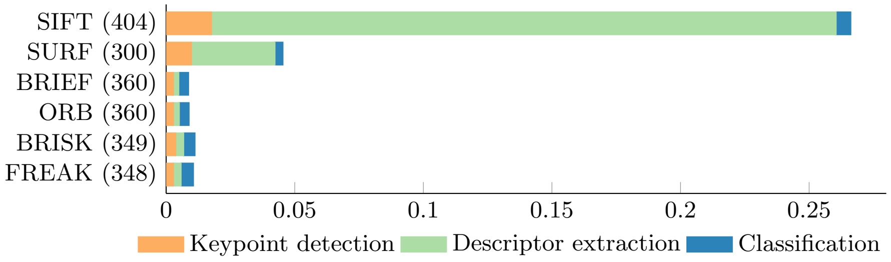


Fig. 6. Average running times (in seconds) of the assessed methods for the verification of one face image (numbers in brackets denote the number of detected keypoints).

4.3 Time Complexity

In the last series of experiments we measure time needed by our framework to verify a test image given different descriptors. All experiments are performed on an Intel Xeon CPU @ 2.67 GHz personal desktop computer with 12 GB of RAM. The implementation of keypoint detection and descriptor computation procedures are taken from OpenCV [3] and assessed through the recently introduced Matlab wrapper [20]. As can be seen from Fig. 6, the keypoint detection and descriptor extraction times of the binary methods are much lower than those of SIFT and SURF methods. However, the matching times seem to be similar for all methods, which is most likely a consequence of us using the Matlab wrapper. With a purely compiled implementation (without the wrapper) of the matching procedure, the binary descriptors are expected to have an edge in this regard as well.

5 Conclusion

We have assessed the relative usefulness of binary descriptors for the task of 3D face recognition. Among the assessed descriptors, SIFT still exhibits the best performance on all datasets. However, when a significant time efficiency is required (e.g. on mobile devices), binary descriptors are viable option, especially FREAK descriptor giving the best trade-off between performance and speed.

Acknowledgments

The work presented in this paper was supported in parts by the national research programe 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. Bay, H., Tuytelaars, T., Gool, L.: SURF: Speeded Up Robust Features. In: ECCV, LNCS, vol. 3951, pp. 404–417. Springer (2006)
2. Bayramoğlu, N., Alatan, A.: Shape Index SIFT: Range Image Recognition Using Local Features. In: Proc. ICPR. pp. 352–355 (2010)
3. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
4. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary Robust Independent Elementary Features. In: ECCV, LNCS, vol. 6314, pp. 778–792. Springer (2010)
5. Colombo, A., Cusano, C., Schettini, R.: UMB-DB: A database of partially occluded 3D faces. In: ICCV Workshops. pp. 2113–2119 (2011)
6. Harris, C., Stephens, M.: A Combined Corner and Edge Detector. In: In Proc. of Fourth Alvey Vision Conference. pp. 147–151 (1988)
7. Huang, D. et al.: 3-D Face Recognition Using eLBP-Based Facial Description and Local Feature Hybrid Matching. IEEE TIFS 7(5), 1551–1565 (2012)
8. Inan, T., Halici, U.: 3-D Face Recognition With Local Shape Descriptors. IEEE TIFS 7(2), 577–587 (2012)
9. Križaj, J., Štruc, V., Dobrišek, S.: Combining 3D Face Representations using Region Covariance Descriptors and Statistical Models. In: IEEE FG. pp. 1–7 (2013)
10. Leutenegger, S., Chli, M., Siegwart, R.: BRISK: Binary Robust invariant scalable keypoints. In: Proc. ICCV. pp. 2548–2555 (2011)
11. Lo, T.W.R., Siebert, J.P.: Local Feature Extraction and Matching on Range Images: 2.5D SIFT. Comput. Vis. Image Underst. 113(12), 1235–1250 (2009)
12. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
13. Mair, E. et al.: Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. In: ECCV, LNCS, vol. 6312, pp. 183–196. Springer (2010)
14. Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. IEEE TPAMI 27(10), 1615–1630 (2005)
15. Ortiz, R.: FREAK: Fast Retina Keypoint. In: Proc. CVPR. pp. 510–517. Washington, DC, USA (2012)
16. Phillips, P. J. et al.: Overview of the Face Recognition Grand Challenge. In: Proc. CVPR. pp. 947–954 (2005)
17. Rosten, E., Drummond, T.: Machine Learning for High-speed Corner Detection. In: Proc. ECCV Part I. pp. 430–443. Springer (2006)
18. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An Efficient Alternative to SIFT or SURF. In: Proc. ICCV. pp. 2564–2571 (2011)
19. Segundo, M., Queirolo, C., Bellon, O.R.P., Silva, L.: Automatic 3D Facial Segmentation and Landmark Detection. In: Proc. ICIAP. pp. 431–436 (2007)
20. Yamaguchi, K.: http://www.cs.stonybrook.edu/~kyamagu/mexopencv/