3D Face Recognition

Algorithms, Application and Comparison with 2D Approach

Technical Report - FIT - VG20102015006 - 2014 - 01

Ing. Štěpán Mráček

Faculty of Information Technology, Brno University of Technology

December 23, 2014

Abstract

This technical report describes the 3D Face recognition as well as the classical 2D approach. We have proposed a face recognition method that is utilizing individual weak recognition units that are further combined with Support Vector Machine-based score-level fusion. In order to select only those recognition units that have a positive impact on the recognition performance, a greedy hill-climbing selection is employed. We have evaluated our recognition method on the Face Recognition Grand Challenge v 2.0 database as well as on our database captured with low-cost SoftKinetic DS325 sensor.

Contents

1 Introduction

The 2D face biometric has become together with fingerprints and iris a part of biometric passports in the European Union and all member states of the ICAO (*International Civil Aviation Organization*). The face was recommended as the primary biometrics, mandatory for global interoperability in passport inspection systems, while the finger and iris were recommended as secondary biometrics to be used at the discretion of the passport-issuing state [1].

The three-dimensional face recognition brings several advantages compared to the two-dimensional recognition. Mainly because it provides new facilities of discrimination, because depth data are added.

Another advantage of the three-dimensional face recognition is related to the data capture technique. Many three-dimensional cameras operate in the infrared spectrum [22] that is independent to lighting conditions, such that direction of light and shadows on the face doesn't negatively affect face recognition.

A three-dimensional face model could be presented in various forms. Most often pointclouds [23], meshes [20] and range images [11] are used. Although all of these representations could be mutually converted to other representations, face recognition technique depends on the form of a three-dimensional model.

The common used representation of three-dimensional models is a range image. Because of their implementation, range images are sometimes called depthmaps. It is an array where each element represents the distance from camera, therefore this array could be stored as greyscale bitmap images. Range images are not universal for all types of three-dimensional models. Mainly because they can't store information about points that are hidden by other points. Using range image representation on three-dimensional faces is not affected by this limitation, because the frontal view of a face doesn't contain much points that are hidden by other parts of face.

2 Principles of Face Recognition

Face recognition is in principle a pattern recognition. Each face is represented as a vector that could be located in a multi-dimensional face space, e.g. in the two-dimensional face recognition a face could be represented as an image with resolution 150×100 pixels. This produces 15,000-dimensional space in which each face scan is stored. Face scans of the same person should be situated close to each other, while face scans of another person are further away. Calculating distances between the face scans in this multi-dimensional space, thus comparing faces, is very time consuming due to multi-dimensionality of the space. Moreover, there is a lot of unwanted information stored, such as a background, hair or clothes. Therefore, various techniques that decrease number of dimensions were invented. The best known are the *Principal Component Analysis* (PCA) [28], the *Linear Discriminant Analysis* (LDA) [3], and the *Independent Component Analysis* (ICA) [13].

2.1 Principal Component Analysis

Principal component analysis covers mathematical methods which reduce the number of dimensions of given multi-dimensional space. The dimensionality reduction is based on the data distribution. The first principal component describes best the data in a minimum-squared-error sense. Other succeeding components describe as much of the remaining variability as possible.

The eigenface method is an example of application of the principal component analysis. It is a holistic face recognition method which takes grayscale photographs of persons that are normalized with respect to size and resolution and represented as column vectors.

First, the mean face $\overline{\mathbf{x}}$ from the set of p training images $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$ is calculated:

$$\overline{\mathbf{x}} = \frac{1}{p} \sum_{i=1}^{p} \mathbf{x}_i \tag{1}$$

The mean face image $\overline{\mathbf{x}}$ is subtracted from each training image $\mathbf{x}_i$ after that:

$$\mathbf{x}_i \leftarrow \mathbf{x}_i - \overline{\mathbf{x}} \quad \forall i \in (1, 2, \ldots, p) \tag{2}$$

The covariance matrix C is constructed:

$$\mathbf{C} = \mathbf{A}\,\mathbf{A}^T = [\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_p]\,[\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_p]^T \tag{3}$$

where $\mathbf{A}$ stands for a matrix where each column i contains a corresponding vector $\mathbf{x}_i$ and $\mathbf{A}^T$ stands for transposed matrix $\mathbf{A}$.

The next step is the calculation of the eigenvalues and eigenvectors of the covariance matrix. This could be achieved by standard linear algebra methods [8]. Given a matrix $\mathbf{M}$, a non-zero vector $\mathbf{v}$ is defined to be an eigenvector of the matrix if it satisfies the eigenvalue equation

$$\mathbf{M}\,\mathbf{x} = \lambda\,\mathbf{v} \tag{4}$$

for some scalar λ. In this situation, the scalar λ is called an eigenvalue of $\mathbf{M}$ corresponding to the eigenvector $\mathbf{v}$ [18].

However, the covariance matrix might be very large and thus the computation of it's eigenvectors and eigenvalues would be time and memory consuming. If the amount of training images p is sufficiently smaller then size (dimensionality) n of training images, eigenvectors and eigenvalues could be retrieved from matrix $\mathbf{C}'$.

$$\mathbf{C}' = \mathbf{A}^T\,\mathbf{A} \tag{5}$$

The size of matrix $\mathbf{C}'$ is determined by the size of the training set and it is $p \times p$. The first p sorted eigenvalues of matrix $\mathbf{C}'$ are also eigenvalues of matrix $\mathbf{C}$. The eigenvectors of matrix $\mathbf{C}$ are calculated by multiplying matrix $\mathbf{A}^T$ by matrix $\mathbf{W}'$. $\mathbf{W}'$ is the matrix containing in each row one eigenvector $\mathbf{w}'$ of the matrix $\mathbf{C}'$.

$$\mathbf{W} = \mathbf{A}^T \mathbf{W}' = \mathbf{A}^T \begin{bmatrix} w'_{11} & w'_{12} & \cdots & w'_{1p} \\ w'_{21} & w'_{22} & \cdots & w'_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ w'_{p1} & w'_{p2} & \cdots & w'_{pp} \end{bmatrix} \tag{6}$$

The resulting matrix $\mathbf{W}$ contains one eigenvector of the covariance matrix $\mathbf{C}$ in each row. These eigenvectors define a set of mutually orthogonal axes within facial space, along which there is the most variance. The corresponding eigenvalues represent the degree of variance along these axes.

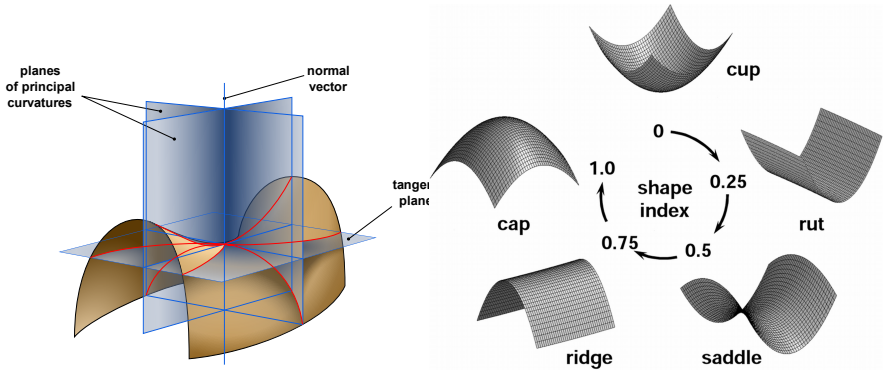
2.2 PCA for 3D Face Recognition

The eigenfaces method applied on three-dimensional face recognition is very similar to it's two-dimensional variant. The mean face and afterwards the calculation of the eigenvectors and eigenvalues from covariance matrix is performed on the range images or other image representation of three dimensional surface.

Since the three-dimensional face model can be described as a surface in the three-dimensional space, curvature analysis can be applied on it. Curvature is the amount by which surface derivates from being flat.

Assume a parametric curve $\gamma(S)$, where S is a parameter which determines tangent vector $T(S)$ and normal vector $N(S)$ at each point of the curve. This parameter also determines curvature $k(S)$ and radius of curvature $R(S) = \frac{1}{k(S)}$.

For the analysis of the face surface, the principal curvatures are important. At a given point of a surface they measure, how the surface bends by different amounts in different directions at that point. An illustration is in Figure 1a. Planes of principal curvatures k_1 and k_2 and the tangent plane are orthogonal.



(a) Principal curvatures. (b) Shape index shapes.

Figure 1: Principal curvatures (a) and shape index shapes (b) [21].

To express a surface curvature characteristics with only one value at each point on the surface, several options are available.

Gaussian curvature K of a point on a surface is defined as the product of principal curvatures k_1 and k_2] [24].

$$K = k_1 k_2 \tag{7}$$

Mean curvature H is the average of principal curvatures k_1 and k_2 [24].

$$H = \frac{1}{2}(k_1 + k_2) \tag{8}$$

Shape index S is for the classification of the surface into categories [20]. See Figure 1b.

$$S = \frac{1}{2} - \frac{1}{\pi} \tan^{-1} \frac{k_1 + k_2}{k1 - k2} \tag{9}$$

where the principal component k_1 is greater than k_2.

In Figure 2, the original range image, Gaussian curvature, mean curvature, and shape index are shown.

Figure 2: From left to right: the original range image, corresponding Gaussian curvature, mean curvature, and shape index on the face surface.

2.3 Filter Banks

The image filter banks are widely used technique in the area of texture analysis, segmentation, and classification. Individual filters in the bank are used in order to remove unwanted components or features. And, at the same time, emphasize significant information.

2.3.1 Gabor Filter Bank

The complex Gabor filter is defined as the product of a Gaussian kernel and a complex sinusoid:

$$g(x, y, \omega, \theta, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x'^2 + y'^2}{2\sigma^2}} \left(e^{i\omega x'} - e^{-\frac{\omega^2 \sigma^2}{2}} \right) \tag{10}$$

where x and y are coordinates within the Gabor kernel, $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$. It is often used as an image edge detector with respect to specific frequencies and orientations. The Gabor space is very useful in image processing applications such as optical character recognition [9], iris recognition [6] and fingerprint recognition [12]. The complex sinusoid is known as the *carrier* and the Gaussian-shaped function is known as the *envelope*. The rotation as well as the frequency of the carrier is controlled through the parameters θ and ω respectively. The parameter σ controls the envelope size.

The Gabor filter is usually controlled with just two discrete-value parameters – orientation $o \in (0, 1, \ldots, 7)$ and scale $s \in (1, 2, \ldots, 7)$. The parameters ω, θ, and σ are set to: $\omega \leftarrow \frac{\pi}{2}\sqrt{2}^{-s}$, $\sigma \leftarrow \frac{\pi}{\omega}$, and $\theta \leftarrow \frac{o\pi}{8}$. The example of Gabor filter bank is in Figure 3.

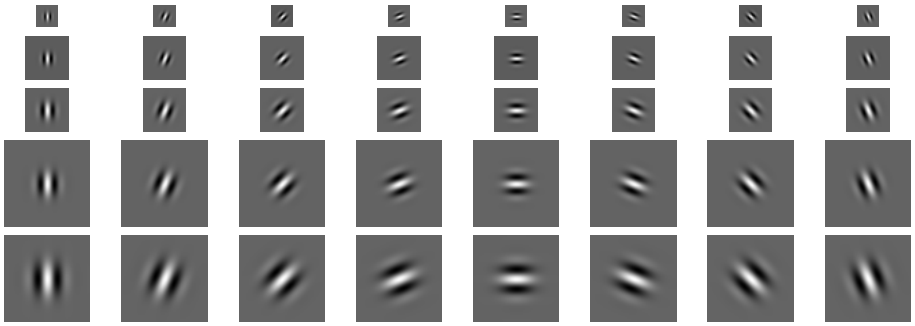
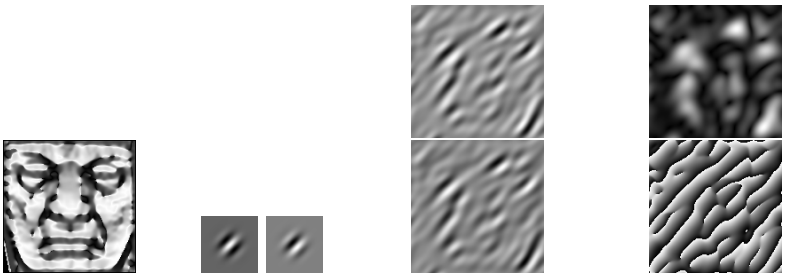


Figure 3: Gabor filter bank.

The Figure 4 shows the application of the complex Gabor filter on the input shape index image.



(a) (b) (c) (d)

Figure 4: Application of the complex Gabor filter on the input shape index representation of the face surface. From left to right: input image (a), real kernel and imaginary kernels (b), real response and imaginary responses (c), and absolute response (magnitude) with angle response (d).

2.3.2 Gauss-Laguerre Filter Bank

The Gauss-Laguerre wavelets are polar-separable functions with harmonic angular shape. They are steerable in any desired direction by simple multiplication with a complex steering factor and as such they are referred to self-steerable wavelets [2]. Our Gauss-Laguerre filter bank consists of 35 filters that were created with parameters $n \in (1, 2, 3, 4, 5)$, $k = 0$, $j = 0$ with sizes 16×16, 24×24, 32×32, 48×48, 64×64, 72×72, and 96×96 pixels. The θ has been set to $\theta \leftarrow \text{atan2}(x, y)$ and $r \leftarrow \sqrt{x^2 + y^2}$.

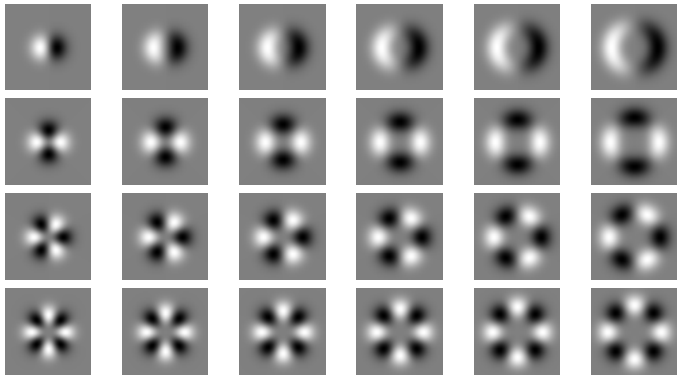


Figure 5: Gauss-Laguerre filter bank.

2.3.3 Local Binary Patterns

The *Local Binary Pattern* (LBP) [27, 26, 10] operator labels the pixels within image by thresholding the 3×3 neighborhood with the center value and considering the result as a binary number. At a given pixel, LBP is defined as an ordered set of binary comparisons of pixel properties between the center pixel and its eight surrounding pixels. The decimal form of the resulting 8-bit word (LBP code) is used to represent the detail property of the center pixel. The example of application of LBP filter is in Figure 6.

Local binary patterns are often used with spatial histograms – the image is divided into grid and histograms are calculated within each cell. Concatenated histograms may form the feature vector directly or they can be further processed with some subspace projection.

3 Algorithm Proposal

This chapter describes the generalized recognition pipeline that takes the input face mesh and normalizes it, such that the rotation is compensated. After that, some image representations of the surface, texture, and curvature are generated

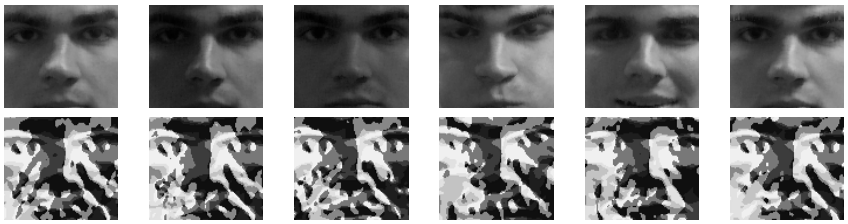


Figure 6: The example of application of the LBP filter. The input images are in the first row, while the results after filter application are on the second row.

from the normalized mesh. The pipeline continues with the application of specific image filters. Finally, subspace projections are used in order to extract features.

The main idea of proposed method is the score-level fusion of involved individual recognition units. By the application of some filter bank, e.g. Gabor filter bank, on the input image we obtain m new images. However, the number of filters within the bank is quite high, therefore some selection method is needed in order to improve speed as well as remove redundancy. We employ hill-climbing selection and the optimization criterion is fusion *Equal Error Rate* EER [16].

The similar approach has been proposed by Yang et al. [29]. Yang is using AdaBoost to select a small set of Gabor features (weak classifiers) in order to form a strong classifier. Moreover, he proposed intra-face and extra-face difference space to transform a multi-class classification to binary decision. The task is to assign the input two images to intra-personal or extra-personal space.

Su et al. [25] came with an algorithm exploiting both local and global features. The Global features are extracted from the whole face images by keeping the low-frequency coefficients of Fourier transform, which encode the holistic facial information, such as facial contour. For local feature extraction, Gabor wavelets are applied on the face image patches. The resulting classifier is based on the hierarchical feature-level fusion utilizing *Linear Discriminant Analysis*. However, both methods from Yang as well as from Su are designated for 2D face recognition. Our proposed method is able to profit from both 2D texture and 3D shape data.

The recognition pipeline suitable for face recognition is depicted in Figure 7. Image data are extracted from the aligned face mesh. Mesh is transformed to the range image or texture representations. Range image is further optionally processed in order to gain 4 new curvature representations – mean curvature, Gaussian curvature, eigencurvature and shape index image.

The alternative to image filters approach is the iso-geodesic curves [15, 14, 4]. A set of curves centering at a given point is retrieved from the mesh and converted to set of points.

The next step, common to both iso-geodesic curves as well as image filters, is some subspace projection of the input data. Image matrix is transformed to

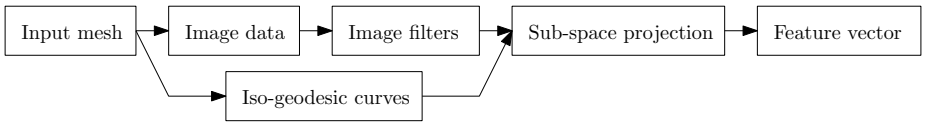


Figure 7: recognition pipeline for 3D face recognition.

the column vector representation and projected to the low-dimensional space after that. In case of iso-geodesic curves, set of 3D points is transformed to the simple column vector in the same manner.

The task of hill-climbing unit selection is to select from given set of units $U = (u_1, u_2, \ldots, u_n)$ a non-empty subset S that achieves possibly the best recognition performance. The exhaustive search of the relatively small set U is simple. However, when the number of units exceeds certain threshold, the exhaustive search becomes impossible.

We are employing a hill-climbing (sometimes called greedy or wrapper) selection [17]. This approach has been originally developed for feature selection, however, it can be also used for unit selection in multi-algorithmic fusion.

The optimization criterion is achieved EER (EER_i) of particular unit i. The algorithm is as follows:

1. Select the unit b that achieves the best EER_b, remove it from set U and add it to the set of selected units S.
2. For each remaining unit j in U measure fusion EER of set $S \cup j$
3. Select the best unit and add it to S. If there was no improvement of fusion EER exit. Otherwise return to 2.

There is a potential drawback of the hill-climbing selection – selected units might be too specific for the training set and can't generalize. Validation on the separate set is recommended therefore. However, our further experiments did not reveal any significant performance drop between the training and testing data. It has emerged that the hill climbing is a good choice for recognition unit selection.

4 Evaluation

The presented algorithm will be evaluated in this chapter. In order to compare the achieved results with other available state-of-the-art algorithms, evaluation is performed on the Face Recognition Grand Challenge version 2.0 [23]. The face recognition algorithm was also tested on the database obtained with low-cost 3D sensor – SoftKinetic DepthSense DS325[1]. The expansion of personal depth sensors related with the new ways of the human-computer interaction in recent years markedly lowered the price of 3D acquiring devices for personal use. However, the biggest challenge of the face recognition based on the low-cost depth sensors is the quality of acquired scans.

[1] http://www.softkinetic.com/products/depthsensecameras.aspx

4.1 3D Face

4.1.1 FRGC

FRGC database contains scans captured in spring 2003, fall 2003, and spring 2004. The spring 2004 portion (2,114 scans) was divided into five parts, such than each part contains the same count of subjects. No subject is present in more than one part. The first part of Spring2004 was used for training of the face alignment and training of individual parameters of subspace projections. Second part was used for validation of selected parameters and for training of final fusion. The last three parts were used for evaluation purposes.

There were evaluated 1,720 different units. Partial results are in Table 1. The best unit consists of following pipeline. The Gaussian curvature representation of the surface is blurred with the Gaussian kernel with size 7×7 pixels. This image is further processed with local binary patterns filter. The resulting image is divided to 10 horizontal and 9 vertical cells. Histogram of values within each cell of size 10×10 pixels is calculated. Set of histograms is further processed with PCA in order to reduce correlation of values as well as number of feature vector components.

Table 1: Partial results of units evaluation - the best representatives of each unit type.

Rank	Type	Input data	Applied filters	EER
1	LBP histogram	Gaussian curvature	gaussBlur(7); lbp(); histogram(10,9)	0.0247
7	Gauss-Laguerre	Range image	gaussLag(48,1,0); scale(0.5)	0.0267
12	Gabor	Eigencurvature	gaborAbs(1,2); equalize; scale(0.5)	0.0295
34	LBP	Mean curvature	gaussBlur(11); lbp(); scale(0.5)	0.0361
73	Plain	Range image	scale(0.5)	0.0416
759	Iso-geodesic curves		5 curves centered at the nosetip	0.0769

All recognition units were used as an input to the hill-climbing selector. The score-level fusion was provided by binary SVM classifier. The hill-climbing selector chose 13 units – see Table 2. In the first iteration, the unit employing the application of LBP histogram on the shape index image was selected. The subsequent iteration chose specific Gauss-Laguerre filter applied on the range (depth) image. The equalized texture followed by the application of Gauss-Laguerre filter was selected in the third iteration.

We have utilized the Spring2004 FRGC subset, such that it's evaluation part contains 1211 scans from 207 individuals designated for evaluation. This evaluation subset was divided into 3 partitions, as it was mentioned earlier. Table 3 shows achieved results with classifier utilizing SVM-based fusion.

4.1.2 SoftKinetic

We have captured all scans for the SoftKinetic database during spring 2014. It contains 398 scans from 52 individuals. There has been put emphasis on these points during capturing:

Table 2: Hill-climbing selection of individual units for SVM-based score-level fusion classifier

Iteration	Unit		Unit EER	Fusion EER
	Input data	Filters		
1	shape index	GaussBlur(7); lbp(); histogram(10,9)	0.0247	0.0247
2	range image	GaussLag(48,1)	0.0266	0.0114
3	texture	Equalize(); GaussLag(64, 4)	0.0963	0.0075
4	shape index	Gabor(3, 2); Equalize()	0.0585	0.0063
5	mean curvature	Gabor(2, 4); Equalize()	0.0657	0.0050
6	texture	Gabor(3, 2)	0.1391	0.0043
7	eigencurvature	Gabor(4, 2); Equalize()	0.0522	0.0035
8	shape index	Equalize(); GaussLag(48, 5)	0.0881	0.0030
9	iso-geodesic curves		0.1163	0.0028
10	range image	Equalize(); Gabor(7, 7)	0.1079	0.0022
11	Gaussian curvature	Equalize(); GaussLag(64, 1)	0.0564	0.00175
12	mean curvature	Gabor(7, 0); Equalize();	0.1014	0.00174
13	eigencurvature	Gabor(1, 0)	0.0642	0.00173

Table 3: Achieved results on the Spring2004 evaluation subset.

Evaluation partition	FNMR at given FMR		
	0.01	0.001	0.0001
1	0.0091	0.0388	0.0721
2	0.0098	0.0329	0.0596
3	0.0167	0.0589	0.1091
Median	0.0098	0.0388	0.0721

- Various lighting conditions.
- Various (but limited) facial expressions – we allowed subjects to have slight smile, lifted eyebrows or frowned face.
- Scanning of some subject was splitted to several sessions in different days.
- Effort was made to have diversity in gender, race, and age of scanned subjects.

Figure 8: Example of scans in the SoftKinetic database (processed, aligned, and cropped).

The example of same scans in the SoftKinetic database is in Figure 8. Contrary to the scans acquired with the Minolta Vivid scanner (FRGC database),

data captured with SoftKinetic DepthSense DS325 sensor suffer from high noise among the z axis [19, 7]

The multi-algorithmic fusion similar to the fusion used for FRGC was trained and evaluated. The hill-climbing optimization selected 10 units. Contrary to the FRGC fusion, no iso-geodesic or LBP-based recognition unit is present. The lower quality of SoftKinetic scans causes the absence of iso-geodesic curves unit. On the other hand, missing LBP-based unit is not so obvious. The selected units are in Table 4.

Table 4: Selected recognition units gained from the training part of the SoftKinetic dataset.

Unit	Input data	Filters
1	range image	GaussLag(72, 2)
2	texture	Equalize(); Gabor(5, 6)
3	range image	Gabor(3, 1)
4	texture	Equalize(); Gabor(6, 0);
5	eigencurvature	GaussLag(24, 5)
6	texture	Gabor(4, 2)
7	range image	GaussLag(48, 0);
8	shape index	GaussLag(96, 1)
9	shape index	Gabor(2, 5)
10	texture	Equalize(); Gabor(2, 7)

The DET curves from the evaluation of the training as well as testing parts of the SoftKinetic dataset are in Figure 9. The particular achieved FNMRs at given FMRs are in Table 5.

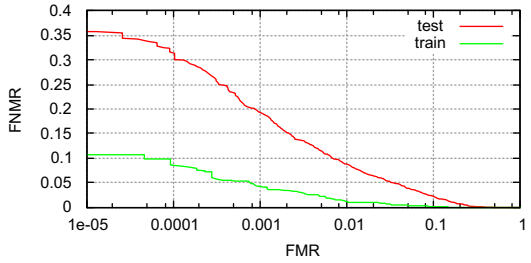


Figure 9: DET curves from the SoftKinetic evaluation.

4.2 2D Face

The 2D approach has been also tested on the FRGC database. However, only the texture information was used in order to recognize an individual. The hill-climbing selection of recognition units is in Table 6 and the DET describing the recognition performance is in Figure 10.

Table 5: Evaluation on SoftKinetic database.

Data	EER	FNMR at given FMR		
		0.01	0.001	0.0001
Train set	0.011	0.012	0.043	0.098
Test set	0.043	0.087	0.192	0.312

Table 6: Selected 2D face recognition units. dog(...) stands for Difference of Gaussians filter.

Unit	Filters
1	dog(5, 3); contrast(); gabor(4, 3); scale(0.5);
2	gaussBlur(11); lbp(); histogram(10, 9);
3	dog(5, 3); contrast(); gaussBlur(7); wld(10, 9);
4	dog(5, 3); contrast(); gabor(6, 0); scale(0.5);
5	dog(5, 3); contrast(); gabor(4, 1); scale(0.5);
6	dog(5, 3); contrast(); gabor(5, 7); scale(0.5);
7	dog(5, 3); contrast(); gaussBlur(11); lbp(); histogram(10, 9);
8	dog(5, 3); contrast(); gaussLaguerre(24, 4); scale(0.5);
9	dog(5, 3); contrast(); gaussBlur(11); lbp(); histogram(10, 9);
10	dog(5, 3); contrast(); gaussLaguerre(48, 4); scale(0.5);

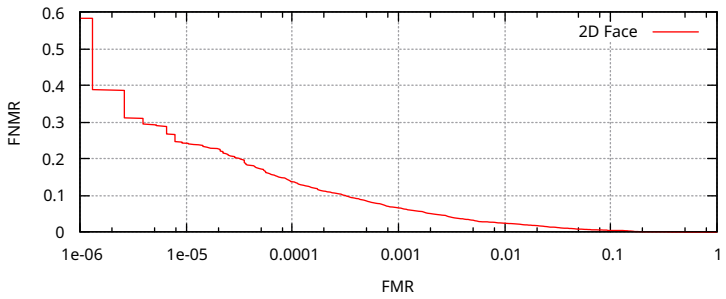


Figure 10: DET curve from the 2D face evaluation.

5 Algorithm Optimization and Real-time Application

In many face recognition applications, the speed is crucial. Sometimes, there is a need to process image data or video stream in a real-time. Nowadays, multi-processor architectures are common even in smartphones and tablets. This section will outline the basic ideas behind the parallel implementation of aforementioned algorithms in order to deploy them to the systems where real-time processing is necessary.

The two most computation expensive parts of the processing pipeline are the

mesh aligning and application of image filters. During mesh aligning, the input face mesh is aligned to the reference face mesh, such that the sum of square distances between corresponding points is minimized. This is done using an *Iterative Closest Point* (ICP) algoritm [5]. ICP requires in each step computation of corresponding points between the input mesh that is being aligned and the reference template mesh. This is quite computational expensive, however it can be easily parallelized.

The second part of the recognition pipeline, where the parallelization can be used, is the application of image filters and further feature extraction using PCA. Each recognition unit, previously selected by the iterative greedy hill-climbing process, is completely independent on other recognition units. Each units takes the input face image representation, applies its filters and projects the resulting image using PCA finally.

The easiest way to implement parallelization is the creation of multiple threads, such that each thread will obtain its portion of the input data. However, the process or thread creation is computationally expensive. It is more convenient to use thread pool pattern therefore. We have compared plain and parallelized implementations on several hardwares – reference core i7 CPU, passively cooled Intel Cerelon, Snapdragon ARM CPU and nVidia Tegra T30. Each machine has 4 CPUs or 2 CPUs with hyperthreading. The results of the benchmark are in Figure 11.

6 Conclusion

The 3D face recognition approach provides much higher security – at FMR = 0.0001 the FMR is 7% for 3D face and 14% for 2D face on the same database. However the biggest disadvantage of 3D approach is high acquisition costs. If we employ a low-cost 3D device, such is SoftKinetic DS325, the FMR raises to 31%.

On the other hand, depth camera provides liveness detection implicitly. It is much harder to spoof a biometric system where the depth sensor is involved.

References

[1] Machine Readable Travel Documents, ICAO doc 9303. Technical report, ICAO - International Civil Aviation Organization, 2006.

[2] H. Ahmadi and A. Pousaberi. An Efficient Iris Coding Based on Gauss-Laguerre Wavelets. In *Advances in Biometrics*, pages 917–926. 2007.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

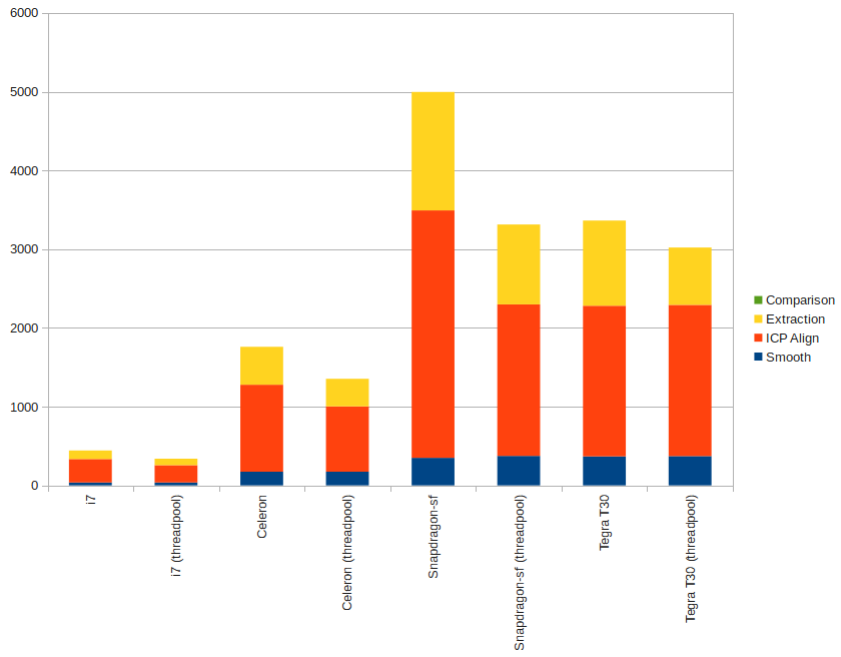


Figure 11: Benchmark of various architectures and plain implementation vs. thread pool. On the vertical axis is the processing time in milliseconds.

[4] S. Berretti, A. Del Bimbo, and P. Pala. 3D Face Recognition Using iso-Geodesic Stripes. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2162–2177, December 2010.

[5] P. J. Besl and H. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[6] J. Daugman. How Iris Recognition Works. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):21–30, January 2004.

[7] D. Falie and V. Buzuloiu. Noise Characteristics of 3D Time-of-Flight Cameras. In *International Symposium on Signals, Circuits and Systems*, page 4. IEEE, July 2007.

[8] M. Haag and J. Romberg. Eigenvectors and Eigenvalues. Technical report, 2009.

[9] Y. Hamamoto, S. Uchimura, M. Watanabe, T. Yasuda, Y. Mitani, and Shingo Tomita. A Gabor Filter-Based Method for Recognizing Handwritten Numerals. *Pattern Recognition*, 31(4):395–400, April 1998.

[10] G. Hermosilla, J. Ruiz-del Solar, R. Verschae, and M. Correa. A Comparative Study of Thermal Face Recognition Methods in Unconstrained Environments. *Pattern Recognition*, 45(7):2445–2459, July 2012.

[11] T. Heseltine. *Face Recognition: Two-Dimensional and Three-Dimensional Techniques*. PhD thesis, The University of York, 2005.

[12] L. Hong, Y. Wan, and A. Jain. Fingerprint Image Enhancement: Algorithm and Performance Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):777–789, 1998.

[13] A. Hyvärinen and E. Oja. Independent Component Analysis: Algorithms and Applications. *Neural Networks*, 13:4–5, 2000.

[14] S. Jahanbin, H. Choi, Y. Liu, and A. C. Bovik. Three Dimensional Face Recognition Using Iso-Geodesic and Iso-Depth Curves. In *2nd IEEE International Conference on Biometrics: Theory, Applications and Systems*. Ieee, 2008.

[15] S. Jahanbin, R. Jahanbin, and A. C. Bovik. Passive Three Dimensional Face Recognition Using Iso-Geodesic Contours and Procrustes Analysis. *International Journal of Computer Vision*, 105(1):87–108, June 2013.

[16] A. K. Jain, A. Ross, and S. Prabhakar. An Introduction to Biometric Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, January 2004.

[17] R Kohavi. Wrappers for Feature Subset Selection. *Artificial intelligence*, 97(97):273–324, 1997.

[18] G. A. Korn and T. M. Korn. *Mathematical Handbook for Scientists and Engineers: Definitions, Theorems, and Formulas for Reference and Review*. Dover Publications, 2nd revise edition, 2000.

[19] F. Lenzen, H. Schäfer, and C. Garbe. Denoising Time-Of-Flight Data with Adaptive Total Variation. In *Advances in Visual Computing*, pages 337–346, 2011.

[20] X. Lu, D. Colbry, and A. K. Jain. Three-Dimensional Model Based Face Recognition. In *ICPR 04: Proceedings of the Pattern Recognition, 17th International Conference on Pattern Recognition*, pages 362–366, 2004.

[21] M. H. Mahoor and M. Abdel-Mottaleb. Face Recognition Based on 3D Ridge Images Obtained from Range Data. *Pattern Recognition*, 42(3):445–451, March 2009.

[22] D. Modrow, C. Laloni, G. Doemens, and G. Rigoll. 3D Face Scanning Systems Based on Invisible Infrared Coded Light. *In Advances in Visual Computing*, pages 521–530, 2007.

[23] 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 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 947–954. IEEE, 2005.

[24] M. Segundo, C. Queirolo, O. Bellon, and L. Silva. Automatic 3D Facial Segmentation and Landmark Detection. In *Proceedings of the 14th International Conference on Image Analysis and Processing (ICIAP '07)*, pages 431–436, September 2007.

[25] Y. Su, S. Shan, and X. Chen. Hierarchical Ensemble of Global and Local Classifiers for Face Recognition. *Image Processing, IEEE*, 18(8):1885–1896, 2009.

[26] X. Tan and B. Triggs. Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions. *IEEE transactions on image processing*, 19(6):1635–1650, June 2010.

[27] H. Tang, B. Yin, Y. Sun, and Y. Hu. 3D Face Recognition Using Local Binary Patterns. *Signal Processing*, 93(8):2190–2198, August 2013.

[28] M. Turk and A. Pentland. Face Recognition Using Eigenfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 591, pages 586–591. IEEE Computer Society Press, 1991.

[29] P. Yang, S. Shan, W. Gao, S. Z. Li, and D. Zhang. Face Recognition Using Ada-Boosted Gabor Features. *Proceeding of sixth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 356–361, 2004.