Embedded wavelet-based face recognition under variable position

Pascal Cotret, Stéphane Chevobbe, Mehdi Darouich

Embedded wavelet-based face recognition under variable position

Pascal Cotret[a], Stéphane Chevobbe[a] and Mehdi Darouich[a]

[a]CEA, LIST, Laboratoire Adéquation Algorithme Architecture, Gif-sur-Yvette, F-91191 France

ABSTRACT

For several years, face recognition has been a hot topic in the image processing field: this technique is applied in several domains such as CCTV, electronic devices delocking and so on. In this context, this work studies the efficiency of a wavelet-based face recognition method in terms of subject position robustness and performance on various systems. The use of wavelet transform has a limited impact on the position robustness of PCA-based face recognition. This work shows, for a well-known database (Yale face database B*), that subject position in a 3D space can vary up to 10% of the original ROI size without decreasing recognition rates. Face recognition is performed on approximation coefficients of the image wavelet transform: results are still satisfying after 3 levels of decomposition. Furthermore, face database size can be divided by a factor 64 (2^{2K} with $K = 3$). In the context of ultra-embedded vision systems, memory footprint is one of the key points to be addressed; that is the reason why compression techniques such as wavelet transform are interesting. Furthermore, it leads to a low-complexity face detection stage compliant with limited computation resources available on such systems. The approach described in this work is tested on three platforms from a standard x86-based computer towards nanocomputers such as RaspberryPi and SECO boards. For $K = 3$ and a database with 40 faces, the execution mean time for one frame is 0.64 ms on a x86-based computer, 9 ms on a SECO board and 26 ms on a RaspberryPi (B model).

Keywords: wavelet, face recognition, eigenfaces, embedded systems, high-constrained systems

1. INTRODUCTION

One of the key challenges in embedded systems design is to maintain a compromise between application efficiency (i.e. computing power) and systems resources consumption (in terms of power, area and memory footprint, etc.). In the image processing context, and especially in face recognition using subspace learning method such as eigenfaces,[1] the computing power is directly proportional to input images resolution. Thus, by limiting memory size and memory accesses, processing time must decrease in a significant manner. A previous work by Courroux et al.[2] revealed an efficient method to decrease the input image resolution based on the computation of a wavelet-based transform, such as the LeGall 5/3 wavelet, on the input image. By using only approximation coefficients as input, the amount of data to be processed is divided by a factor 2^{2K} (K being the wavelet decomposition level). This method can be hardened against illumination variation by the preprocessing of the ROI with a region-based equalization.[2]

Using this approach, the computing power required for a correct face recognition as well as the amount of memory required for database storage are strongly decreased. It allows us to consider the use of low-cost embedded systems, including low-performance embedded processors and small amounts of embedded memory. This work studies the robustness of a wavelet-based eigenfaces approach under several level of decomposition against 3D face position shifting. This work analyzes the impact of face position in relation to the snapshot used for learning purposes; face databases (Yale and a custom video) are also used to measure the robustness against light variation as well.

The paper is organized as follows. Section 2 presents related works. Section 3 describes the wavelet-based based recognition method used in this work. Section 4 studies the approach robustness (i.e. its tolerance in terms of face position). Section 5 summarizes several performance results. Finally, Section 6 gives some conclusions and persectives.

Pascal Cotret. E-mail: pascal.cotret@gmail.com, Telephone: +33 (0)2 99 84 45 77

[*] http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

2. RELATED WORKS

Face recognition methods are sensible to image variations such as illumination variation, subject misalignment or facial corruptions/occlusions. In the literature, many methods have been studied to overcome the misalignment issue by reconstructing an aligned Region Of Interest (ROI)[3] or using correlation filters to propose robust face recognition method.[4] However, in a real-time embedded context, the limited processing power prevent us to consider such complex methods.

In Shan et al.[5] work, a PCA-based face recognition method insensitive to illumination and expression variations is proposed and embedded implementations on a DSP and a NIOS II processor are performed. Due to the advanced methods used to enhance the recognition rate under variations, and despite several optimizations, the recognition of a face out of 16 learned subjects is done in 1 second, which is not acceptable real-time context. Endluri et al.[6] propose a PCA implementation on a TSK3000a (32-bit RISC embedded processor) that allows the recognition in real-time for a 2 users database, on 320x240 images. These studies show that the eigenface classification needs too many computing power to run on embedded system in real-time context.

However, no study is performed on the robustness of this system to image variations, and no data is given concerning the real performance and the ROI size. Pavan et al.[7] uses a weighted modular PCA-based method enhance the robustness of the recognition to facial variations. The proposed system, based on a NIOS processor coupled to a PCA hardware co-processor, runs at 26 fps on a Altera Stratix II and is able to learn up to 5 subjects (due to memory limitation).

3. WAVELET-BASED FACE RECOGNITION ALGORITHM DESCRIPTION

3.1 Face recognition based on eigenfaces

Method used in this work for face recognition is based on eigenfaces.[1] The work of Turk et al. describes a complete process for face recognition based on 2D calculations rather than complex 3D methods where a face should be entirely reconstructed. Using eigenfaces, recognition can be divided in two steps: learning and recognition (also known as testing).

Figure 1 shows a block diagram view of these two steps.
Learning is simply based on eigenfaces as it was described in the work of Turk et al.[1] First of all, faces must be acquired from snapshots with non-uniform background. In this first approach, it is considered that faces are clearly located in snapshots. It is considered a set of faces to be included in the learning database: these faces are ROIs centered on each subject to be learnt. Assuming that N ROIs of size $M * M$ have to be put in the database, the learning stage is performed in five steps:

- **Step 1:** An inline transformation is performed on the whole dataset. For a single ROI, it consists in a concatenation of its lines. Therefore, the size of the inline dataset (called D_{inline}) is a matrix of size $M^2 * N$.
- **Step 2:** The next step is to enhance image quality for recognition. A normalization is done on the line dataset using the global mean and standard deviation. Dimension of the result matrix $D_{inline,norm}$ is not affected.
- **Step 3:** Firstly, a multiplication between the transpose of $D_{inline,norm}$ and $D_{inline,norm}$ itself is done. The result is a covariance matrix C_{mat} of size $N * N$. Secondly, the algorithm computes eigenvalues (e_{val}) and eigenvectors (e_{vec}) of C_{mat}. e_{val} is a vector of size N while e_{vec} is a matrix of size $N * N$.
- **Step 4:** Then, a phase of sorting is done on eigenvalues and eigenvectors. Eigenvectors with null eigenvalues are erased and eigenvectors are sorted according to the ascending order of their corresponding eigenvalues. This operation does not change matrix orders as erased values are zero-padded ($size(e_{val,sort}) = size(e_{val})$ and $size(e_{vec,sort}) = size(e_{vec})$).
- **Step 5:** The final step of the learning stage is the computation of weights of each face in the dataset. This is done by a multiplication between the eigenvectors matrix and the inline database. The result, W_{mat}, is a matrix of size $N * N$.

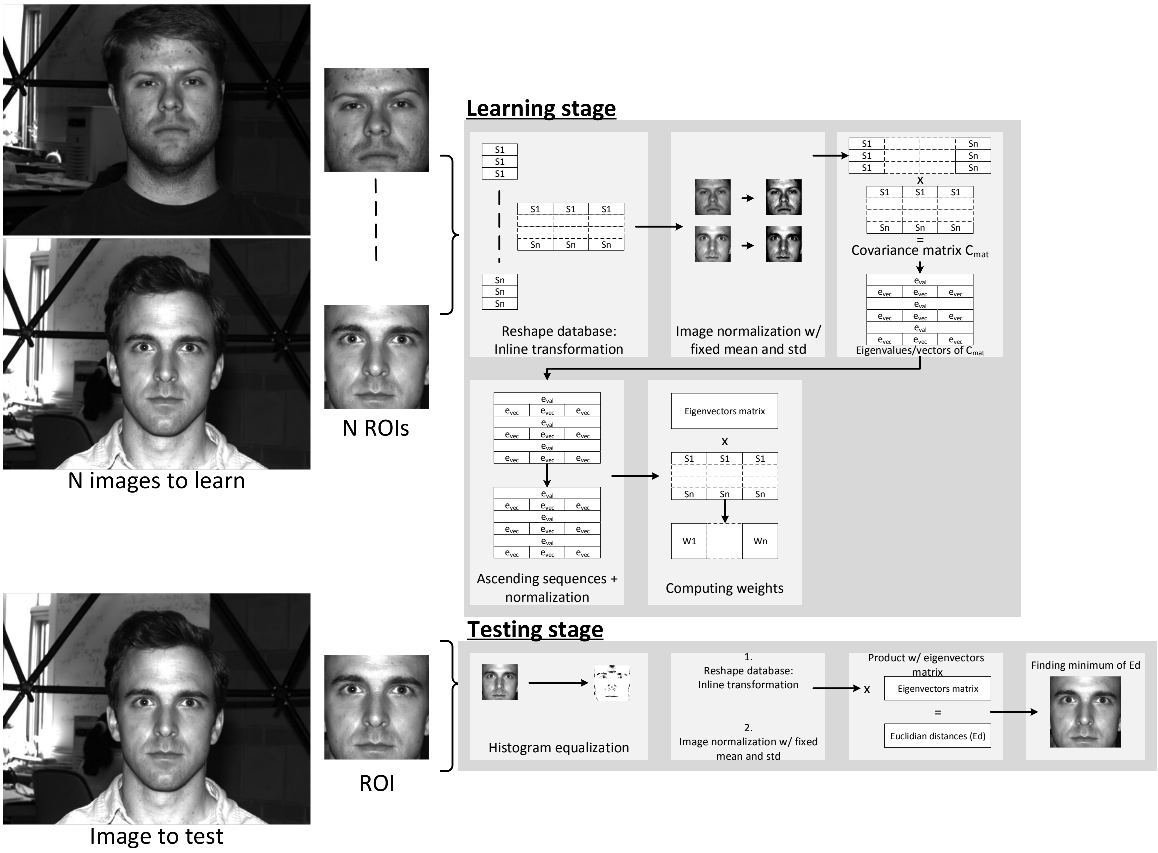

N images to learn
N ROIs
Learning stage
Reshape database: Inline transformation
Image normalization w/ fixed mean and std
Covariance matrix C_{mat}
Eigenvalues/vectors of C_{mat}
Ascending sequences + normalization
Eigenvectors matrix
Computing weights
W1
Wn
S1
Sn
e_{val}
e_{vec}
Testing stage
Image to test
ROI
Histogram equalization
1. Reshape database: Inline transformation
2. Image normalization w/ fixed mean and std
Product w/ eigenvectors matrix
Eigenvectors matrix
Euclidian distances (Ed)
Finding minimum of Ed

Figure 1: Basic face recognition flow based on eigenfaces

As outputs of the eigenfaces task, the algorithm produces three results:

- The overall eigenfaces mean.
- Weights of each face in the learning dataset (W_{mat}).
- A database including eigenfaces.

These data are processed in the testing stage of the algorithm. It also takes as inputs faces to be tested (a single face in case of a real-time implementation). This stage is done in three major steps:

- **Step 1:** The first operation is to compute a Regional Histogram Equalization (RHE) aiming to highlight face details and especially enhance borders (nose, eyes and so on).
- **Step 2:** Then, as it was done in the learning stage, the inline transformation and normalization is performed on the image currently tested.
- **Step 3:** Finally, a product is performed between this image and the eigenfaces matrix. Compared with the weight matrix W_{mat}, it give N Euclidian distances $E_{d,n}$ (also known as “scores”): the index of the minimum value in $E_{d,n}$ will identify the most similar face in the learning database.

In the context of embedded systems, two-dimensional wavelet transform is used to preprocess images before the eigenfaces task in order to decrease computation times and memory consumption.

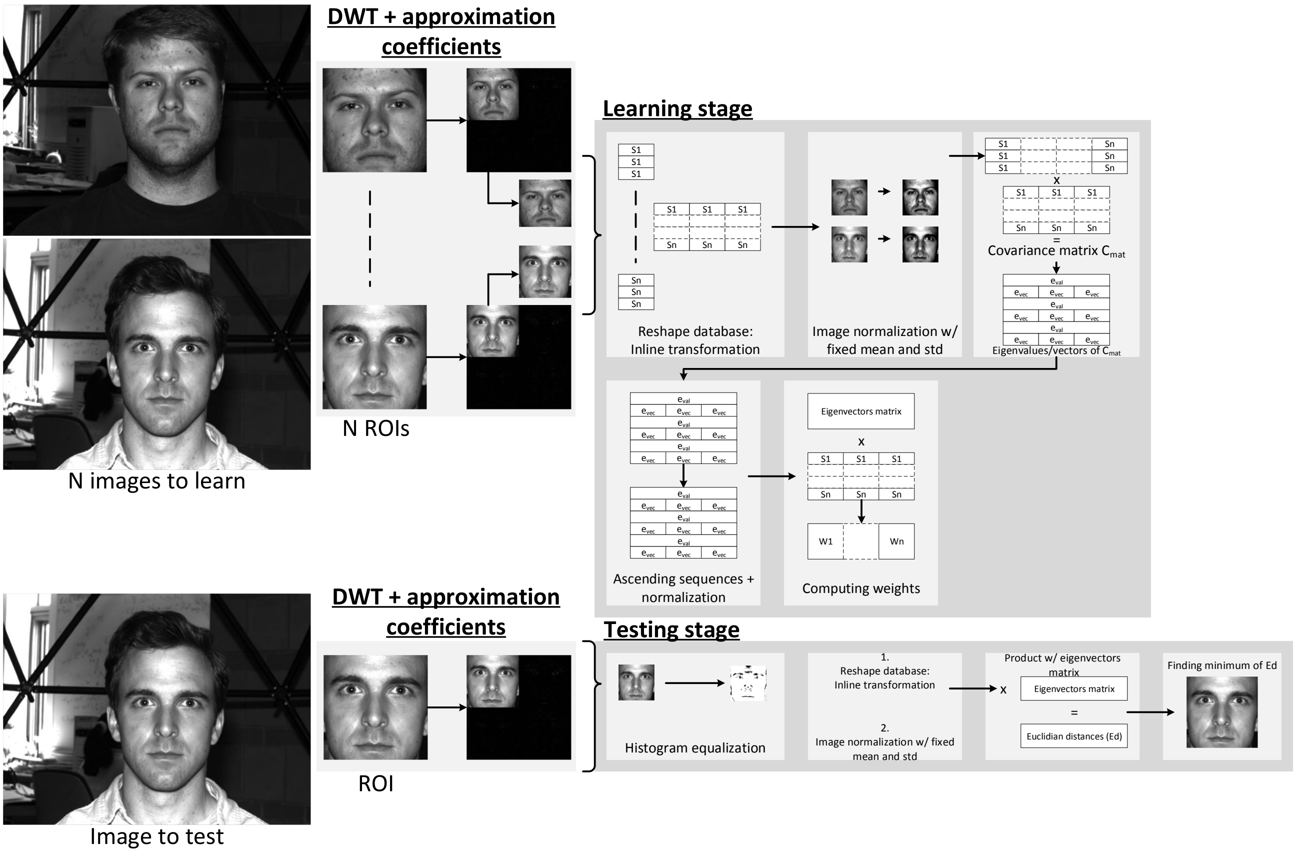

DWT + approximation coefficients
N ROIs
N images to learn
Learning stage
S1
Sn
Reshape database: Inline transformation
Image normalization w/ fixed mean and std
x
=
Covariance matrix C_{mat}
e_{val}
e_{vec}
Eigenvalues/vectors of C_{mat}
Ascending sequences + normalization
Eigenvectors matrix
W1
Wn
Computing weights
DWT + approximation coefficients
ROI
Image to test
Testing stage
Histogram equalization
1. Reshape database: Inline transformation
2. Image normalization w/ fixed mean and std
Product w/ eigenvectors matrix
Eigenvectors matrix
Euclidian distances (Ed)
Finding minimum of Ed

Figure 2: ROI-then-DWT recognition flow

3.2 ROI-then-DWT recognition flow

Based on the the flow described in Section 3.1, a 2D-discrete wavelet transform is implemented right after the ROI task. Figure 2 shows the so-called ROI-then-DWT recognition flow. In this approach, the wavelet transform is done after the ROI cropping. In our context, face recognition only takes care of approximation coefficients. Therefore, the ROI size is divided by a factor 2^{2K} (2^K ratio on each dimension, where K is the wavelet decomposition level). The wavelet transform used in this work is based on the work of Courroux et al.[2] The data amount processed in the algorithm is, de facto, decreased:

- In the learning stage, the DWT-processed ROIs imply some time decrease for the first three steps (eigenfaces computation).
- In the testing stage, DWT transformation has an impact on computation time for the first two steps (until the product with the eigenfaces matrix).

In the context of embedded systems, the integration of wavelet transform is a key point: as data amount is lowered, both computation time and memory consumption should be decreased. Section 5 presents some performance results. Section 4 also studies the impact of wavelet transform on face recognition rates.

4. ROBUSTNESS OF WAVELET-BASED FACE RECOGNITION

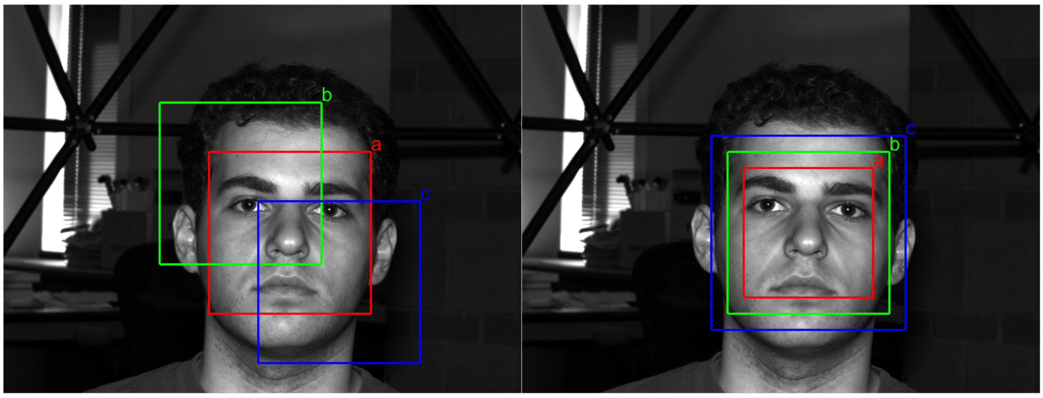
4.1 Experiments description

Using the wavelet-based method previously defined, several experiments are performed to measure the robustness of our approach regarding several parameters. Position of the subject regarding the image sensor (in three dimensions) is the main study.

4.1.1 Faces database

Efficiency of a given face recognition method partially depends on the images quality used for both learning and testing. As this work focuses on subject position and lightning conditions, the “Extended Yale Face Database B” [†][8] is used. This database provides a large set of images with fixed position of the subject regarding the image sensor with several lightning conditions.
For the learning database construction, 35 snapshots are acquired: one for each subject, with centered light source. For testing purposes, 210 snapshots can be processed. Each user is represented by 6 snapshots with slightly deviated light source, close to the centered position. Testing images stricly differ from learning ones. Following studies take into account ROIs of 200x200 pixels wide while the resolution of full-scale images is 640x480 pixels. ROIs built for face position shifting and distance from the sensor ([x,y] and [z] studies) take into account non-uniform background: even if snapshots are generally close to the user face, a few background elements may appear.

4.1.2 [x,y] shifting



(a) Face position shifting ROIs.
a: centered. b: upper-left bound.
c: lower-right bound.

(b) Scale variation ROIs.
a: 80%. b: 100% (i.e. equal to the learning database position). c: 120%.

Figure 3: ROIs for both face position shifting and scale variation studies.

For each subject from the learning database, a ROI frame is swept along all positions between b and c positions (see Figure 3a) in both horizontal (x) and vertical (y) directions: it defines a subspace where b is the upper-left bound and c the lower-right bound. Figure 3a shows a full-scale image taken from the database used in this work with three ROIs: b, c and a, centered on the user (equivalent to the learning position). This test database contains a large set of ROIs where the subject face may be partially out of the frame.
Robustness analysis against face position shifting is done in several steps :

- Firstly, a projection of each shifted ROI on learned eigenfaces is performed (testing stage of the ROI-then-DWT algorithm described in Figure 2).
- Then, projection values (also known as scores) are collected and displayed in a color map diagram giving 35 color maps for each subject to be tested (one color map for each image in the learning database).
- Finally, for each subject, a new color map containing the minimum scores of all projections except one of the current subject is created. A difference between the color map related to the current subject and color maps of the minimum score is computed. When this value is positive, the subject is correctly recognized. The thresholded map shows the area of shift variation tolerance. As each subject has its own tolerance in terms of shifting, an accumulation of all the tolerance areas is finally computed. These colormaps display the recogniton rate according to shift variation.

[†] http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

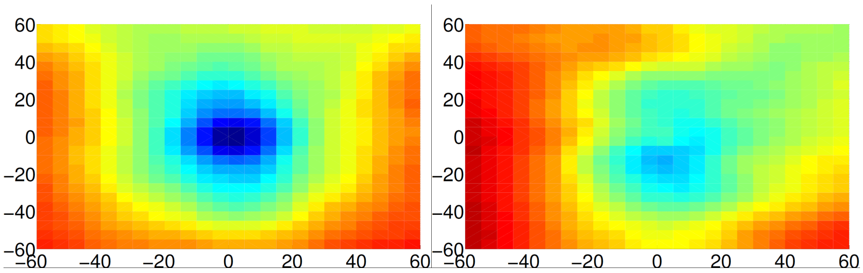
4.1.3 [z] scaling

Another study of this work demonstrates robustness against scale variation. It shows how the recognition algorithm reacts when a subject move away or close to the image sensor compared to the learning database position. Figure 3b shows a full-scale image with 3 ROIs. As snapshots are acquired at a fixed position, this work proposes to take ROIs with different sizes in order to feign scale variation. Then, a test database is built by interopolating all ROIs to the learned database size. Finally, we gather each ROI minimum score to compute a plot of recognition probability against scale variation.

4.2 Experiments results

4.2.1 [x,y] shifting

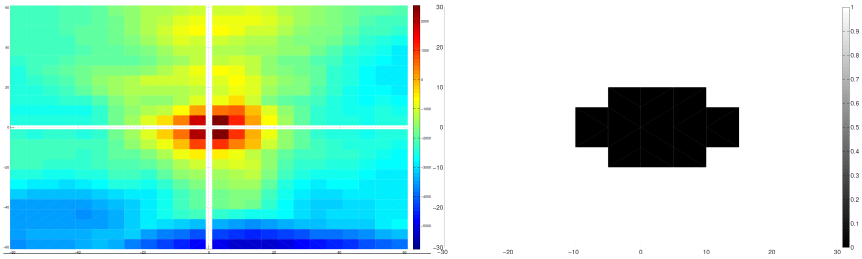
By applying the previous methodology on all learned subjects, minimal score is obtained when the ROI is centered on the subject face. Even on other projections, minimum score is obtained around the subject face center.



(a) Face #1 (currently tested). (b) Face #2.

Figure 4: Score colormaps of 2 faces.

As an example, Figure 4 shows score colormaps of subject #1 on two learned faces. Their coordinates represent the ROI center location ([0, 0] when centered on the subject face). Then, minima difference colormaps for all the subjects are computed.



(a) Colored (b) Binary/thresholded view

Figure 5: Minima difference colormaps for subject #1.

On both colormaps, shifting step is set to 5 pixels.In figure 5b, a tolerance of +/- 10% of the ROI size in each direction is obtained for a recognition rate over 75%.

4.2.2 $[z]$ scaling

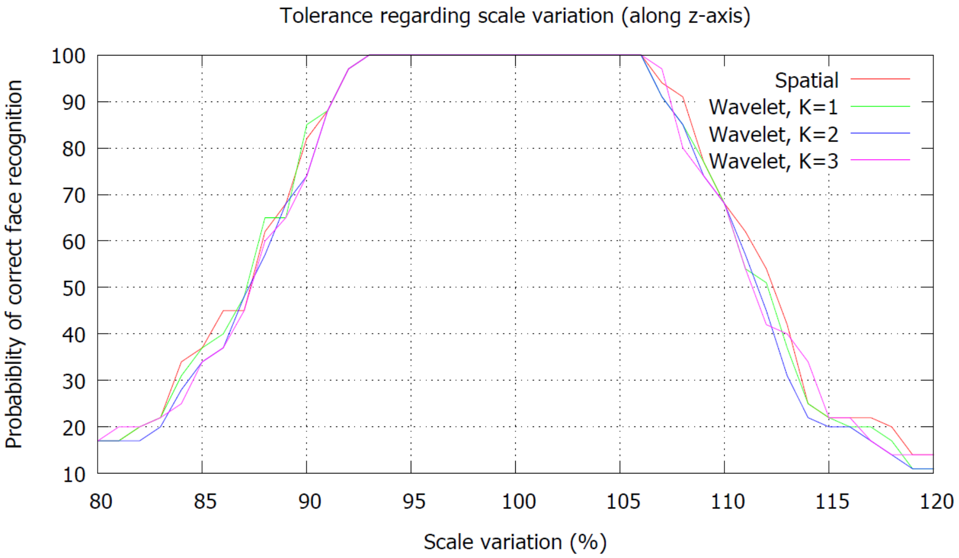


Figure 6: Probability of correct recognition (face identification) against scale variation of a centered face.

Figure 6 shows probability results for both spatial-based and wavelet-based methods at several decomposition levels. Wavelet decomposition does not alter the scale variation robustness: for both cases, the face recognition algorithm support a variation of +/- 10% in terms of scale with a correctness probability over 75% while a +/- 5% tolerance is admitted for a 100% rate.

4.2.3 Wavelet decomposition impact

From an implemention of the ROI-then-DWT flow described in Section 3.2, the work of Courroux et al.[2] studied the impact of the DWT levelon recognition rates of the Eigenfaces classifier. Even with partially obfuscated faces (first two sets of Yale database), rates of the wavelet-based approach were correct (over 80 %, with a decomposition level $K = 3$). The Regional Histogram Equalization implemented in the face recognition process gives more tolerance about faces illumination. It is one key point of the use of wavelet transform in the face recognition context: even if data amount is compressed by a factor 16 (between $K = 1$ and $K = 3$), recognition rates are still correct

5. PERFORMANCE RESULTS

The main advantage of using DWT at the input of the recognition pipeline is to drastically decrease the amount of processed pixels. As only the approximation coefficients are kept for regonition, the size of the input image is divided by a factor of $2^{(K+1)}$. We show in this section the impact on the processing time of the recognition pipeline of the wavelet decomposition level against number of faces in the database.

5.1 Measurement set up

The measures are conducted on three different hardware platforms:

- x86-based computer : an Intel Core i7 running at 3.4Ghz and cache size 8 MB.
- ARM-based computer : an iMX6 system from SECO[‡] with a quadcore Cortex-A9 running at 1GHz and cache size 32 KB.

[‡] `http://www.seco.com/prods/eu/boards/qseven-boards/quadmo747-x-i-mx6.html`

- ARM-based computer : a RaspberryPi model B with a single core processor from Broadcom running at 700MHz and cache size 32 KB.

The x86-based computer gives a reference for the output of the recognition pipeline and for its execution time. The two nano computers ARM-based give results on more embedded platforms. The core of the recognition pipeline is in C code with double precision for the computation of the eigenvalues. The main function is in C++ code. It manages the accesses to the images and outputs results of execution such as the two best scores of eigenface recognition procedure and the identified face for each frame. The image accesses are done with OpenCV functions. Source codes were compiled with standard options (mainly, -O3), to be executed on one core in each case.

All the executions to profile the code on the three platforms are done on the same video sequence and the same set of faces. The video sequence is composed of 1000 frames at resolution 640x480 with two faces changing in x, y and z positions. The video sequence was acquired with an IDS UI-3240CP[§]. The ROI size of the faces in the database is 200x200 pixels.

The tests are conducted on 4, 6, 8 ,10 20, 30 and 40 faces and with wavelet decomposition level (K) varying from 0 to 5. Each execution is done in two steps. First, the learning phase is executed to generate the eigenface basis at a given K. This step is not profiled. Second, the face classification using eigenfaces is executed and profiled. The execution time is a mean of 5 exections measured with *time* command. The profiling is obtained with the *gprof* tool.

x86 Core i7	K=0	K=1	K=2	K=3	K=4	K=5
4 faces	1,16	0.74	0.64	0.61	0.60	0.60
6 faces	1,41	0.78	0.64	0.61	0.60	0.60
8 faces	1,71	0.82	0.65	0.61	0.60	0.60
10 faces	2,02	0,84	0.66	0.62	0.61	0.60
20 faces	3,86	1.04	0.69	0.62	0.61	0.60
30 faces	4,35	1.23	0.74	0.64	0.61	N/A
40 faces	5,09	1.42	0.78	0.64	0.61	N/A

Table 1: Execution time per frame (in ms) on x86 Core i7 for K varying from 0 to 5 and number of faces in database varying from 4 to 40

5.2 Profiling analysis

The table 1, 2, 3 put together the execution mean times of the face classification, respectively, on x86 Core i7, on SECO board and on Raspberry Pi B for K varying from 0 to 5 and number of faces in database varying from 4 to 40. The mean execution times are normalized for one frame. Because the recognition rate for K = 5 and database with 30 and 40 faces are not relevant, the mean times are not given.

SECO board	K=0	K=1	K=2	K=3	K=4	K=5
4 faces	26	12	9	8	8	8
6 faces	33	13	9	9	8	8
8 faces	41	15	10	9	8	8
10 faces	48	16	10	9	9	8
20 faces	85	21	11	9	9	8
30 faces	121	26	12	9	9	N/A
40 faces	157	31	13	9	9	N/A

Table 2: Execution time per frame (in ms) on SECO board for K varying from 0 to 5 and number of faces in database varying from 4 to 40

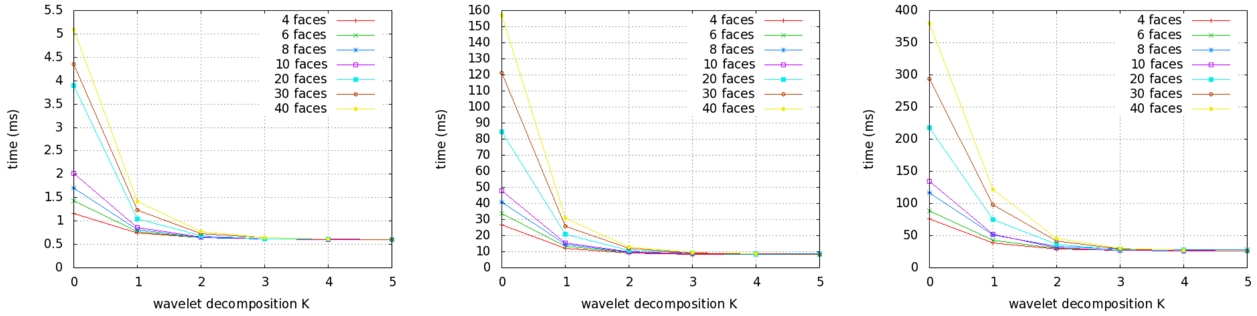
[§] `https://en.ids-imaging.com/store/ui-3240cp.html`

As expected the execution time on the intel core i7 varying from 0.6 ms to 5.09 ms is much more faster than on the two ARM based nano computers for SECO, varying from 8 ms to 157 ms and for RPI-B from 26 ms to 380 ms. Lots of hardware aspects such as, pipeline organization, floating point unit, vectorisation, etc. can explain this acceleration. But memory management and accesses are key factors, as the variations of the ratio between the execution times of the ARM based architectures and the intel core i7 follow closely the variations of the size of the database. This is even more true between the RPI-B and the SECO board. The difference between the execution times are mainly due to the memory accesses.

RPI	K=0	K=1	K=2	K=3	K=4	K=5
4 faces	78	38	29	27	26	26
6 faces	89	43	30	27	28	28
8 faces	116	51	33	29	28	28
10 faces	135	52	31	27	26	26
20 faces	218	75	38	28	26	26
30 faces	293	98	41	29	27	N/A
40 faces	380	121	46	30	27	N/A

Table 3: Execution time per frame (in ms) on RaspberryPi for K varying from 0 to 5 and number of faces in database varying from 4 to 40

The figure 7 shows the evolutions of the execution times on the three targets in function of K and the number of faces in the database. Even if, because the intel core i7 is very performant the impact of K and the number of faces is less significant than on the execution times of the ARM based nano computers, all the three architectures have the same behavior. The execution times drastically decrease when the decomposition level increases and, to a lesser extent increase with the number of faces in the database. Over than K equal to 3, the execution times stop to decrease and are even mostly independent of the number of faces in the database. We show, previously in,[2] the recognition rate is not significatively modified until K equal to 3. At this level, the execution on the intel core i7 is accelerated by a factor from 2 to 8 according to the number of faces in the database ; on the SECO board by a factor from 3 to 17 ; and on the RPI-B by a factor from 3 to 12.



(a) on x86 Core i7

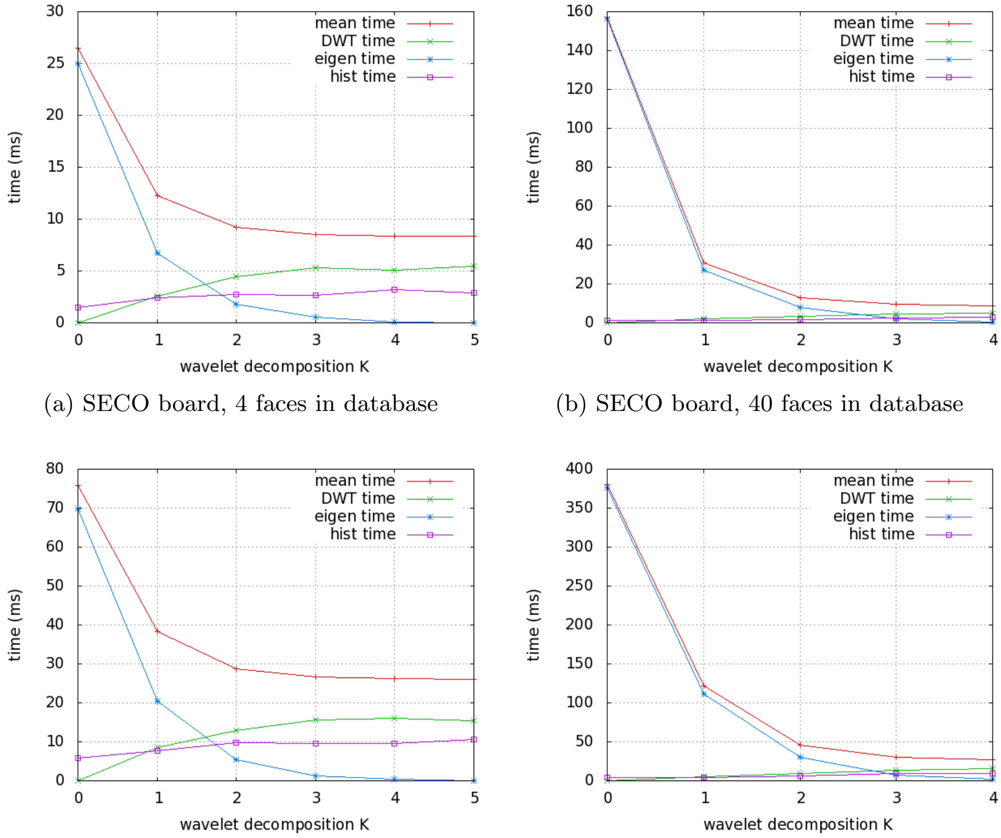
(b) on SECO board

(c) on RPI

Figure 7: Execution time in ms of the recognition pipeline per frame function of wavelet decomposition level (K) for database containing 4 to 40 faces.

The figures 8 show the evolution of the three main functions (histogramm equalization, wavelet decomposition, eigenface classification) of the recognition pipeline and the total execution time on the SECO board and RPI-B for 4 and 40 faces in database. While the eigenface classification execution time drastically decreases with K; the wavelet decomposition execution time increases in a more limited manner. The execution time of the histogramm equalization is more or less stable. So the total execution time decreases with K in the manner described above. The behaviors of the execution on the SECO board and on the RPI-B are very close.

As the eigenface classification execution time is dependant of the number of faces in the database, it decreases in a faster way for a database with 40 faces than a database for 4 users. So bigger the face database is, stronger the acceleration of the execution is with K increasing.



(a) SECO board, 4 faces in database

(b) SECO board, 40 faces in database

(c) RPI, 4 faces in database

(d) RPI, 40 faces in database

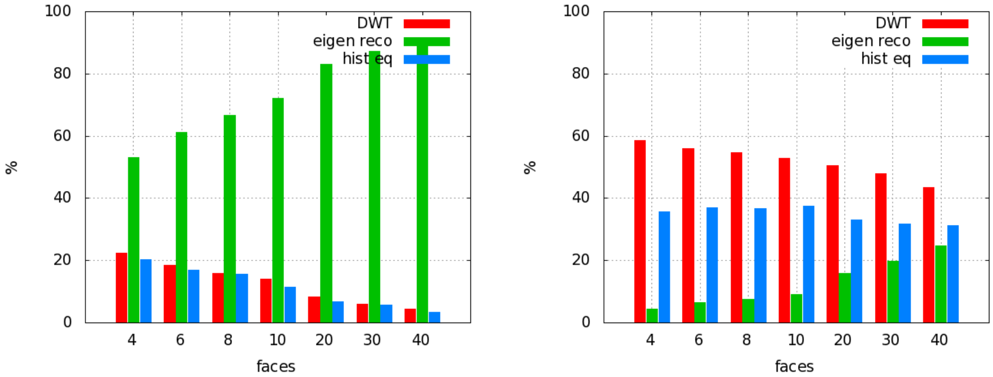
Figure 8: Execution time per frame for the main functions of the recognition pipeline (DWT, eigenface recognition, histogramm equalization) on SECO board, 4 faces in datatbase (a) and 40 faces in database (b) and on RPI 4 faces in database (c) and 40 faces in database(d) in function of K.

The figures 9 show the percentage of the total execution time on the RPI of the three main functions for K = 1 and K = 3 when the number of face in the database varies. In both case, K = 1 and K = 3, the contribution of the eigenface classification increases with the number of faces in the database while the contribution of the two other functions decreases. For K = 1, eigenface classification takes over than 50 % for 4 users to around 90 % for 40 users, as for K = 3 its contribution is between 5 % and 25 % for 4 to 40 users making the wavelet decomposition function becoming the main function in term of execution time.

6. CONCLUSION AND PERSPECTIVES

In our previous work, we studied the opportunity to use wavelet-based eigenface recognition in a low cost embedded context, i.e with low computing power and memory capacity. Applying wavelet to input images before performing the face recognition allows a high decrease of the computing time without lowering the recognition rate. In this work, we pushed forward the study of the method robustness and characterize its performances on existing embedded plateforms.

The first main results is the study of the wavelet-based eigenfaces robustness against face position shifting in both horizontal and vertical directions; and in scale, under several level of decomposition. For face position,



(a) wavelet decomposition level (K = 1) (b) wavelet decomposition level (K = 3)

Figure 9: Execution time per frame for the main functions of the recognition pipeline (DWT, eigenface recognition, histogramm equalization) on the RPI B.

a tolerance of +/- 10% of the ROI size is obtained with satisfying recognition rates (over 75%) and with a high level of wavelet decomposition. This level of robustness allows us to consider the use of a low complexity face detection stage compliant with the limited processing power available on embedded systems, as the location of the face can shift of less than 10% of the ROI size. It even enables the possibility of performing no detection if the user places his face in the indicated space, and the decision is made by accumulating the recognition results.

The performance analyzis extracted from the application execution on three systems (on Intel PC, and two ARM-based nano-computers) has shown that applying DWT transformation to the input image before performing the face recognition allows a speed gain of almost 3 for a 4 faces database, and more than 12 for a 40 faces database (from no DWT to a three level DWT), enabling realtime performances on considered nanocomputers, for instance 30 fps to recognize a face in a 40 faces database on Raspberry Pi. The application of a DWT of at least 3 levels highly decreases the computing power needed by the recognition phase which is no longer the major function (from 91% to 25%)for a 40 faces database. Further optimizations will target the DWT function which is now at 43% of needed computing power.

These results allow us to consider a face recognition application with smooth video feedback on an embedded low cost processing system. These results also confirms the interest of wavelet transform for face recognition with a high number of learned subjects, as the quantity of data to store is decrease by 64 for 3 wavelet decomposition levels. For instance, the size of the database for 10 faces is lowered form 1.5 MB to 24 KB.

In futur studies, we will consider the two major functions in term of complexity, which are DWT and histogram equalization.

REFERENCES

[1] Turk, M. and Pentland, A., “Eigenfaces for recognition,” *J. Cognitive Neuroscience* **3**, 71–86 (Jan. 1991).

[2] Courroux, S., Chevobbe, S., Darouich, M., and Paindavoine, M., “Use of wavelet for image processing in smart cameras with low hardware resources,” *Journal of Systems Architecture - Embedded Systems Design* **59**(10-A), 826–832 (2013).

[3] Yan, S., Wang, H., Liu, J., Tang, X., and Huang, T., “Misalignment-robust face recognition,” *Image Processing, IEEE Transactions on* **19**, 1087–1096 (April 2010).

[4] Sawides, M., Kumar, B., and Khosla, P., “"corefaces" - robust shift invariant pca based correlation filter for illumination tolerant face recognition,” in [*Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*], **2**, II–834–II–841 Vol.2 (June 2004).

[5] Shan, T., Bigdeli, A., Lovell, B. C., and Chen, S., “Robust face recognition technique for a real-time embedded face recognition system,” in [*Pattern Recognition Technologies and Applications: Recent Advances*], Verma, B. and Blumenstein, M., eds., 188–211, IGI Global, Hershey, PA, USA (2008).

[6] Endluri, R., Kathait, M., and Ray, K., "Face recognition using pca on fpga based embedded platform," in [*Control, Automation, Robotics and Embedded Systems (CARE), 2013 International Conference on*], 1–4 (Dec 2013).

[7] Kumar, A. P., Kamakoti, V., and Das, S., "System-on-programmable-chip implementation for on-line face recognition," *Pattern Recognition Letters* **28**(3), 342 – 349 (2007). Advances in Visual information Processing Special Issue of Pattern Recognition Letters on Advances in Visual Information Processing. (ICVGIP 2004).

[8] Lee, K.-C., Ho, J., and Kriegman, D., "Acquiring linear subspaces for face recognition under variable lighting," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **27**, 684–698 (May 2005).