

A MULTISCALE MORPHOLOGICAL COPROCESSOR FOR LOW-POWER FACE AUTHENTICATION

Patrick Stadelmann, Jean-Luc Nagel, Michael Ansorge and Fausto Pellandini

Electronics and Signal Processing Laboratory,
Institute of Microtechnology, University of Neuchâtel,
Rue A.-L. Breguet 2,
CH-2000 Neuchâtel, Switzerland

Phone: +41 32 718 3412; Fax: +41 32 718 3402
e-mail: Patrick.Stadelmann@unine.ch

ABSTRACT

This paper reports a low-power multiscale morphological coprocessor for mobile face authentication devices. In such a system, face verification is performed by extracting mathematical morphology features (erosions and dilations) using a set of structural elements of increasing size. The proposed architecture reduces the number of operations by reusing results, and lowers the amount of needed memory transfers by caching data locally. To improve processing speed at a given operating frequency, four morphological units operate in parallel. Structuring elements can be modified easily as they are described in a table. The architecture was validated using an in-house developed tool, the corresponding simulation results being presented at the end of the paper.

1. INTRODUCTION

An increasing number of functionalities are being offered by new generations of wireless communication devices. Indeed, additionally to vocal conversation, mobile phones provide remote access to a variety of data and teleservices. Moreover, Personal Digital Assistants are becoming more sophisticated, while holding data of raising sensitivity. Accordingly, the access to these devices is secured, e.g. by means of Personal Identification Numbers (PIN). It is however desirable to use biometric verification/authentication methods, since they protect from access by unauthorized persons disposing of the PIN code. Face verification is particularly appealing among biometrics authentication methods, because it is intrinsically non-intrusive, and does not require an extra sensor as it would be the case for fingerprint, assuming that the needed miniature camera is shared with regular video and image acquisition applications. Interestingly, commercially available mobile phones start to be equipped with a miniature camera.

However, face authentication requires intensive computations and a considerable amount of memory transfers that are to be tightly optimized in terms of needed resources and power consumption to comply with the stringent requirements of portable devices. With this respect, dedicated architectures undoubtedly provide much higher performance than general purpose digital signal processors.

The paper is organized as follows. Section 2 describes the low-power face authentication system containing the multiscale morphology coprocessor reported in this article. The

morphology coprocessor is then discussed in detail in Section 3, including presentation of the algorithm implementation, architecture organization, corresponding instruction set, simulation results, and complexity / performance figures. The conclusions are finally drawn in Section 4.

2. FACE AUTHENTICATION SYSTEM

2.1. Face authentication algorithm

The presented system relies on the face authentication method known as Elastic Graph Matching (EGM), which is thoroughly described in [1]. In this scheme, a reference grid is placed over an image of the reference face. At each of the 64 nodes of the grid, a set of mathematical morphology features is extracted by performing multiscale morphology (see Subsection 2.2). In the selected method, the number of extracted morphological features is 19. This set of features is then reduced to only 3 values, through Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [1], [3]. When a new face has to be verified against the reference face, the same grid is positioned on the test image, and the corresponding set of 64×3 features is obtained. A metric is then applied to calculate the distance between the test and reference features. In order to take account of the small face variations that can be observed over the time, e.g. due to expression changes, the grid is iteratively moved, firstly in *rigid* mode (the grid undergoing no deformation, all nodes being jointly displaced), and then in *elastic* mode (the nodes being individually displaced). After every step, the distance between the test and reference features is calculated, where graph deformations introduced during the elastic mode induce a penalty that increases the resulting distance. The minimal distance obtained in the course of this process is then compared to a predefined threshold in order to determine whether the test and reference faces belong to the same person or not.

2.2. Multiscale morphology

A binary structuring element (SE) is basically a two-dimensional shape, cf Fig. 1a. Finding the eroded value of a grayscale image at location (x,y) simply consists in placing the SE over the image, with its center positioned at (x,y) , determining the lowest value among the pixels covered by the SE and assigning this value to the resulting image at (x,y) . The dilated value is obtained similarly by determining the highest value under the SE. A set of multiscale SEs is constituted of several SEs sharing the same shape, but at

different scales. Hence, multiscale morphology consists in performing morphological operations with SEs belonging all to a set of multiscale SEs. The selected face authentication method uses 9 erosion and 9 dilation levels obtained with circular SEs whose size ranges from 3×3 up to 19×19 pixels, corresponding to levels 1 to 9, respectively, cf Fig. 1a. The 19-feature vector mentioned in Subsection 2.1 is then obtained from the concatenation of these 18 morphological values and the original pixel value. A series of images resulting from multiscale erosions and dilations is shown in Fig. 1b, jointly to the related structuring elements.

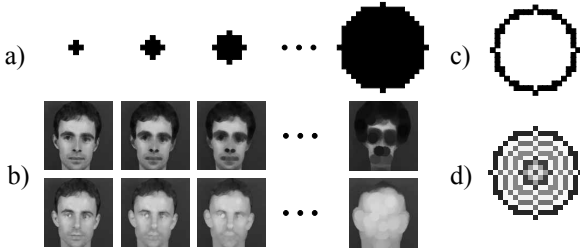


Figure 1: Multiscale morphology: **a)** Set of SEs; **b)** Example with corresponding erosions/dilations (upper/lower part); **c)** dSE_N ; **d)** Multiscale SE table.

2.3. System architecture

2.3.1. Task splitting

In the presented system, morphological features extraction and graph matching are executed separately by two dedicated coprocessors [3]. The reasons for proceeding so are multifold. Firstly, the operations required by these two tasks are very different: morphology uses mostly comparisons, while feature reduction and graph matching use Multiply-Accumulate (MAC) type operations. Secondly, when rigid matching is performed, every possible grid position is explored, and consequently, morphology features are extracted at least once for every pixel in the image. Therefore, computing morphology over the complete image beforehand doesn't involve any redundant operation. On the other hand, most pixels will be covered several times by a node as the grid moves. Extracting features only when they are needed would result in performing morphology several times for each pixel. Thirdly, by performing morphology all at once over the complete image, it is possible to reuse loaded pixels for neighboring positions [2]. The factorized implementation is therefore much more adapted to low-power applications. This paper focuses on the morphology coprocessor, while the feature reduction and graph matching coprocessor is presented in [3].

2.3.2. System organization

As depicted in Fig. 2, the complete system is composed of an image sensor, the morphology and matching coprocessors, and a RAM connected to all three system components by a shared bus. This bus can be controlled by any of the three components, which are in turn controlled by a master processor. Although using a color sensor would certainly prove very useful for preprocessing operations such as color-based face detection, morphology operations are carried out on grayscale images. The data flow proceeds as follows: 1) the face image captured by the sensor is stored in the RAM; 2) the morphology coprocessor reads the im-

age data, extracts the features, and stores the results back in the RAM; 3) the matching coprocessor performs feature reduction and graph matching, before returning the obtained measure score to the master processor.

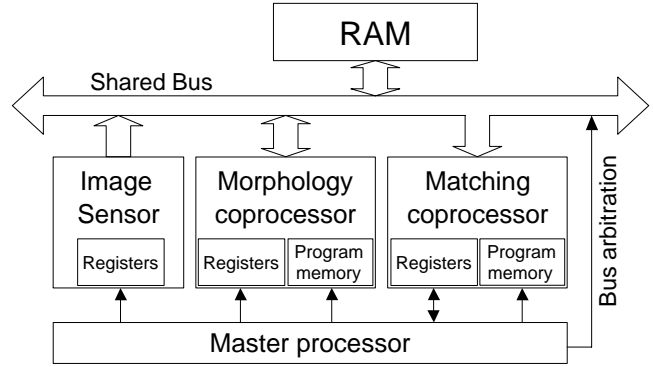


Figure 2: Face verification system.

3. MORPHOLOGY COPROCESSOR

3.1. Algorithm implementation

3.1.1. Differential multiscale morphology

Eroding and dilating an image with a given binary SE consists in finding the minimal and maximal pixel value under the given SE, respectively. If $\{SE\}$ represents the values of the pixels under SE, then the erosion E and dilation D can be expressed as:

$$E = \min(\{SE\}); D = \max(\{SE\}).$$

Decomposing SE in two parts SE' and SE'' , we have:

$$E = \min(\min(\{SE'\}), \min(\{SE''\})), \\ D = \max(\max(\{SE'\}), \max(\{SE''\})).$$

In case of multiscale morphology, we denote by SE_N the SE of level N , whereas E_N and D_N indicate the result of the image erosion and dilation over SE_N , so that:

$$E_N = \min(\{SE_N\}) = \min(\min(\{dSE_N\}), E_{N-1}) \quad (1a)$$

$$D_N = \max(\{SE_N\}) = \max(\max(\{dSE_N\}), D_{N-1}) \quad (1b)$$

where the right hand-sided terms are achieved assuming that the differential structural element dSE_N (cf Fig. 1c) verifies:

$$\{SE_N\} \equiv \{dSE_N\} \cup \{SE_{N-1}\}.$$

Multiscale morphology can thus be implemented by computing erosions and dilations using differential SEs. E_N and D_N are recursively obtained with $N \in [1, \dots, 9]$, cf Fig. 1d, whereas E_0 and D_0 are both equal to the value of the original pixel lying under the SE center.

3.1.2. Local image memory

Computing all morphological levels for one pixel P of the image requires accessing 253 pixels. As most of these pixels will be needed again when performing the morphology for the pixel next to P , they can be stored in a local memory to avoid re-reading them from external memory. When the process starts, the local memory is loaded with the 19×22 pixels window located on the top left corner of the face image, cf Fig. 3. As described in Subsection 3.2, the chosen

implementation processes 4 contiguous horizontal positions in parallel. Thus, the window moves by increments of 4 pixels, so that 4×19 new pixels have to be loaded before starting working on the new positions. These new pixels are written over the 4×19 leftmost pixels which are not needed anymore. To simplify the addressing scheme used to dispatch pixels to morphology units, new data are written to the local memory so that the whole window can be read sequentially by simply incrementing the address pointer. When the leftmost image column is reached, the window is moved down by one pixel, and the computation continues on the new line, from right to left, cf Fig. 3.

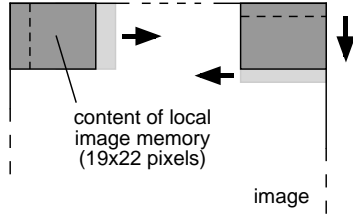


Figure 3: Local image memory updates.

3.1.3. Morphology implementation

Once the local image memory is updated, actual morphological operations are started. In this phase, during a typical instruction cycle, a pixel P is read from the local image memory output register Reg. 2, cf Fig. 4b. P is then processed by each EMU, according to the values of the parameters L and C originating from the SE memory and the instruction decoder, respectively, cf Fig. 4b.

L	C	Operation
15	–	NOP
14	–	$E_out \leftarrow P; D_out \leftarrow P$
0–8	0	$pE[L] \leftarrow \min(pE[L], P)$ $pD[L] \leftarrow \max(pD[L], P)$
0–8	1	$E_out \leftarrow \min(E_out, pE[L])$ $D_out \leftarrow \max(D_out, pD[L])$

Table 1: EMU operations.

According to Table 1, $L = 15$ is used when P does not belong to any dSE for the given location, thus performing a NOP; $L = 14$ specifies that P is the SE central pixel, and that it should be directly copied to the output registers. Otherwise, L is interpreted as the morphology level, and is used to select the partial results registers pE and pD . If $C = 0$, partial morphology is performed, otherwise P is ignored and the final results for level L are generated by combining the partial results $pE[L]$ and $pD[L]$ with the previous level results that are still available in output registers E_out and D_out . Once available, the final results are written back to external memory. Since no external read operations are executed during the morphology phase, external memory write operations can occur at any time.

Once morphology processing is complete for the current four pixels, the address registers and the local image memory are updated for the next four pixels, cf. Subsection 3.1.2.

3.2. Morphology coprocessor architecture

3.2.1. Elementary morphology unit (EMU)

Each elementary morphology unit (cf Fig. 4a) is composed of 2 banks of 9 8-bit registers holding the partial results for the 2×9 partial morphology operations being computed by the unit. It also comprises two output registers for erosion and dilation that store the final results to be saved in the external memory. Actual erosions and dilations are executed by two comparators (min and max, respectively). Each comparator is connected to one of the register banks and one of the output registers. Both comparators are also connected to the input pixel P . Control signals consist of the 4-bit morphology level selector L provided by the SE memory and by the C signal generated by the instruction decoder.

3.2.2. Morphology block

The morphology block encompasses 4 independent EMUs, all connected to the local image memory output register Reg. 2, cf Fig. 4b. Control signals are directly applied to the first unit, but are delayed by 1, 2 and 3 cycles before reaching units 2, 3 and 4, respectively. As each pixel is processed at the same time by the four units, the delays account for the different positions being worked on. Both outputs of each EMU are connected to a 2-1 multiplexer controlled by the E/D signal selecting between erosion or dilation. The four 2-1 multiplexers are connected to a 4-1 multiplexer controlled by the POS signal selecting one out of the four EMUs.

3.2.3. SE and local image memory

Data transfers from the SE memory and from/to the local and the external image memories always occur to/from a register. Memory read and write operations can thus last up to the entire clock cycle to complete, which allows the use of slower, less power consuming memories. The SE memory size is 1.5 kbits, used to store 370 4-bit words. This number is specified by the size of the multiscale SE table counting 19×19 values, cf Fig. 1d, supplemented by 9 additional values needed to combine the partial results according to Equations (1a,b). The L parameter is provided by a multiplexer selecting between the morphology level values read from the SE memory, and a NOP value, the purpose of the latter being to let the morphology block pipeline become empty. Finally, the local image memory capacity is 3.3 kbits, corresponding to 19×22 8-bit pixels.

3.2.4. Control adders

The morphology coprocessor contains three adders, used to update the various address registers. The first adder is a 10-bit adder whose results are provided modulo 418, and which is mainly used to increment the local image memory read and write addresses. The implicit modulo operation makes it possible to use the local image memory as a circular memory, which eliminates the extra code that would be needed to detect and deal with memory roll-overs. The second adder is a 9-bit incremter / decremter whose main duty is to perform offset updates for external read and write operations. The third and last adder is an 8-bit incremter exclusively used to update the SE memory pointer.

3.3. Instruction set

The morphology coprocessor instruction word is divided into 4 fields, each one controlling a different entity. The first field controls external load and store operations. The second one controls the operation of the morphology block. The third field deals with the modulo-adder of the addressing unit, and with the reset of the pE and pD register banks. The last one concerns the incrementer/decrementer. The total size of the instruction word is 14 bits: the four fields use 2, 2, 5 and 4 bit, respectively, the last bit being used to indicate the end of a loop. Loops are declared by a special instruction, where 9 bits in the instruction word are used to store the number of iterations to be executed.

3.4. Validation and simulation results

The proposed morphology coprocessor architecture was validated using a custom framework developed in C++, presented in [3]. Simulation results are reported in Table 2 for a 146×146 source image, resulting in 128×128×19 extracted features.

Number of executed cycles:	2'240'846
<i>external loads</i>	304'780
<i>external stores</i>	311'304
<i>local image memory read</i>	1'720'320
<i>min / max operations</i>	8'555'448
Static program size (# of instructions):	217

Table 2: Complexity estimation of the morphology coprocessor for a complete face verification.

It is interesting to note that the total number of external data loads corresponds to slightly less than 19 loads per pixel in the original image, or to less than one load per output result. Applying a full multiscale morphology on an image requires 2.3 million cycles. In [3], it is shown that elastic graph matching further needs 6.7 million cycles, resulting in a total of ca 10 million cycles to process a complete face verification. Assuming that for a secured access to mobile terminals or PDAs, a complete face verification

shall be handled within 1 second, an operating frequency of 10 MHz would be sufficient, thus avoiding the recourse to very fast and power consuming components.

4. CONCLUSIONS

A multiscale morphological coprocessor, foreseen for low-power face authentication, was presented in this paper. It was shown that, by using a dedicated architecture, it is possible to minimize the total number of comparisons, as well as to drastically reduce the number of external data loads. However, the proposed solution is not tied to a particular set of structuring elements. It is thus possible to modify them very easily, by simply updating the SE memory. More extensive changes (e.g. size of the image) can be implemented by updating the software program.

Future work would consist in carefully implementing the coprocessor in VHDL, permitting a real-time validation on FPGA, and a low-power VLSI implementation from which accurate power consumption and die area figures can be obtained.

ACKNOWLEDGMENTS

This work was part of the *SmartPix* project supported by the Swiss Center for Electronics and Microtechnology, Inc. (CSEM), Neuchâtel and Zurich, Switzerland, under Grant OIH3.

REFERENCES

- [1] C. Kotropoulos, A. Tefas and I. Pitas, "Frontal Face Authentication Using Discriminating Grids with Morphological Feature Vectors", *IEEE Trans. on Multimedia*, Vol. 2, No. 1, 2000, pp. 14-26.
- [2] I. Pitas, "Fast Algorithms for Running Ordering and Max/Min Calculation", *IEEE Trans. on Circuits and Systems*, Vol. 36, No. 6, June 1989, pp. 795-804.
- [3] J.-L. Nagel, P. Stadelmann, M. Ansoerge and F. Pellandini, "A Low-Power VLSI Architecture for Face Verification Using Elastic Graph Matching", submitted to *EUSIPCO 2002*, September 2002, Toulouse, France.

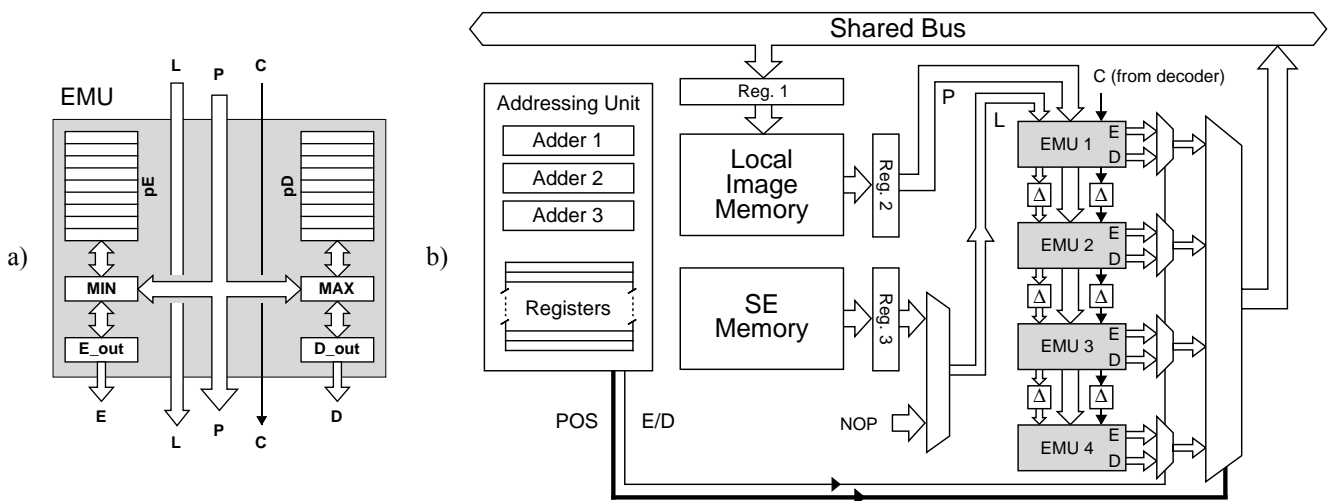


Figure 4: a) Elementary Morphology Unit; b) Morphology coprocessor (Δ symbolizes a one-cycle delay, corresponding to a register).