PRE-PROCESSING IMAGE ALGORITHM FOR FINGERPRINT RECOGNITION AND ITS IMPLEMENTATION ON DSP TMS320C6416

Farah Dhib[1] , Mohsen Machhout [2] and Aguili Taoufik[3]

[1]Department of Electrical Engineering, National School of the Studies of Engineer of Tunis, Tunis, Tunisia
[2]Faculty of Sciences of Monastir, Monastir, Tunisia
[3]National School of the Studies of Engineer of Tunis,Tunis, Tunisia

ABSTRACT

Fingerprint recognition technology is becoming increasingly popular and widely used for many applications that require a high level of security. We can meet several types of sensors integrated in the fingerprint recognition system as well as several types of image processing algorithm in order to ensure reliable and fast authentication of people. Embedded systems have a wide variety and the choice of a well-designed processor is one of the most important factors that directly affect the overall performance of the system. This paper introduces a preliminary treatment to the image in order to improve the quality, and then present a hardware implementation.

KEYWORDS

Fingerprint; Biometrics; Images Processing; Embedded Systems; DSP

1. INTRODUCTION

The Real time image processing is always in high demand for many applications used in security system, remote sensing, manufacturing process and multimedia, those require to have high performance. [1]

In the information time, security authentication is essential to human society. The reliable identification of people is becoming more and more necessary to solve problems of access and identity theft [2].

We can use several traditional authentication methods, such as identity authentication, key, etc, however, there are too many shortcomings. [3]

Personal identification based on biometrics is nowadays an emerging research subject. Several physiological and behavioral human characteristics have been suggested in the last decades as discriminating features to be used in personal recognition. Among them, fingerprint is the most widely used technique due to its high discriminating power. [4]

Biometric technology is the use of the inherent human physiological characteristics(e.g., fingerprints, face, retina, iris) and behavior characteristics (e.g., gait, signature) for identification of personal identity, including fingerprint recognition, iris recognition, face recognition, which are uniqueness, accuracy, graphs and non-invasive, etc.[3]

Each biometric has its strengths and weaknesses and the choice typically depends on the application. There are a number of desirable properties for any chosen biometric characteristic [5]. Using biometrics to recognize people has a very big importance in issues concerning security. Although biometrics do not solve all security problems, and is not appropriate for all applications, they represent a powerful means of identity verification in some applications when correctly implemented. Moreover, when combined with the other verification techniques (keys or passwords) a higher level of security can be reached. Such systems can be implemented in company entrances and banks; also there is a recent study of adding biometric information to passports on a radio frequency ID (RFID) for verifying that the passport isn't stolen or forged as the stored biometrics information should match the person holding the passport. [6]

Fingerprint recognition is a method to identify a person, based on the physiological characteristics of the finger. This technology is the oldest biometric technology, but still it is most widely used because it provides good levels of accuracy and simplicity. Fingerprint recognition is highly reliable for the recognition purpose because of their uniqueness and consistency over the time. Also, the fingerprint is fast biometric technique for more reliable and secure system.[5]
The goals of a biometric security system can be multiple. A security system can, for example, use the tools that biometrics makes available to manage the security of logical access of a system. This is probably the most common use of biometrics. But it can also be used to secure the physical access of a room containing sensitive data, from any place on which we choose to install a biometric lock. [7]

Fingerprint recognition algorithms are diverse and are based on different techniques in order to extract useful information from the input image. Obviously, the clearer and better the image, the more reliable and fast the system is.

An image sensor is composed of many photosites, each corresponding to a picture element, it is the "pixel". Each pixel of an image sensor records the amount of light to which it is exposed and converts it into a corresponding number of electrons. The stronger the light, the higher the number of electrons generated.

Sensors embedded in fingerprint recognition systems are diverse and based on several technologies, noting for example:

- CCD sensors: they have been used in cameras for over 30 years and have many advantages. In general, they still offer a slightly higher sensitivity to light than CMOS sensors and produce a little less noise. However, they are more expensive and more difficult to integrate into a camera.

- CMOS sensors: they reduce the overall price of cameras, as they contain all the elements necessary for the composition of cameras. They offer more integration possibilities and more functions than CCD sensors. They also offer shorter measurement times, lower power dissipation at the chip, and reduced footprint.

- Megapixel sensors (sensors containing one million pixels or more): They are integrated with megapixel cameras and are of a size equal to or only slightly larger than VGA sensors that provide a resolution of 640x480 (307,200) pixels. This means that the size of each pixel on a megapixel sensor is smaller than that on a VGA sensor. Therefore, although the megapixel camera provides higher resolution and more detail, it is less sensitive to light than its VGA counterpart since the pixel size is smaller and the light reflected from an object is spread over more pixels.

However, matching fingerprints in an automatic way is still a technical challenge. A compromise in the system performance exists: accuracy, security, real-time and cost are some of the critical points to face up to [4].

Whatever the technology used, the purpose of a fingerprint recognition system is to have reliable authentication and fast error rate as small as possible.
Many of today's recognition systems are realized on the PC-based platform, which has high power consumption and cannot be used with ease. But, the recently developed market for civil applications has more demand for system's portability and power consumption. So, the trend of miniaturizing and embedding the system for automatic fingerprint identification became more distinct.

Reliable extraction of features from input fingerprint image is the most challenging problem faced in the area of fingerprint recognition. However the performance of the Minutiae extraction algorithm relies heavily on the quality of the input fingerprint image. [5]

In order to achieve fast and intelligent image processing, image processing system is required to have powerful data processing capabilities and high flexibility. DSP with high-speed and high-performance is the key to achieve rapid processing [8].

In this article, we first present the image pre-processing procedure in order to improve and facilitate the extraction of biometric data, and then, we focused on the physical implementation of the system.

2. PRE-PROCESSING PROCEDURE FOR FINGERPRINT RECOGNITION

With the rapid development of image acquisition equipment, high-resolution images bring a great deal of data processing and a variety of complex algorithms bring large amount of calculation. Therefore, in some embedded applications that have harsh real-time requirements of system image processing speed often becomes a bottleneck affecting system's accuracy and usability. [8]
Thanks to the invention of IoT, the image processing is reaching up to a distinct level as IoT is becoming a major part of every one life. These systems are creating many applications in image processing field such as image filtering and processing. [9]

The most widely used technique among biometric techniques is fingerprint recognition (Figure 1). [10]

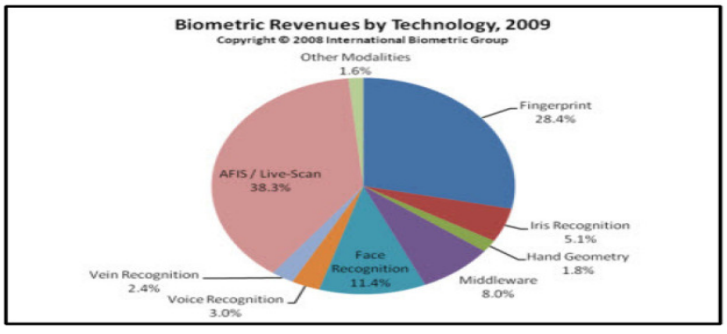

Figure 1. Biometrics market distribution

Fingerprints are natural unique patterns formed by friction on epidermal ridges (raised) and furrows (recessed), which appear on the pads of fingers and thumbs. Though these epidermal

ridges are found on fingers, palms and soles, they are popularly called just fingerprints. They have never been observed repeating in any human being in the history of Dactylography (Fingerprints are also called dactylograms and study of fingerprints is called dactyloscopy or dactylography).

Study and understanding of fingerprints have mostly helped forensic science to prove the presence of someone at a location. This has been associated mostly with the crime scenes and legal proceedings, until 21st century, when fingerprints started playing important role not only for identification of criminals, but also in day-to-day identification needs.

2.1. DESCRIPTION OF THE FINGERPRINT

A fingerprint is made up of a set of locally parallel lines forming a unique pattern for each individual. We can distinguish streaks (which are the lines in contact with a surface) and valleys (these are the spaces between two streaks). [11]

Each fingerprint has two types of singular points: the global singular points and the local singular points. The global singular points are the centers (the points of convergence of the striations) and the deltas (the points of divergence of the streaks) (figure 2). The local singular points are called minutiae (mainly termination and bifurcations).

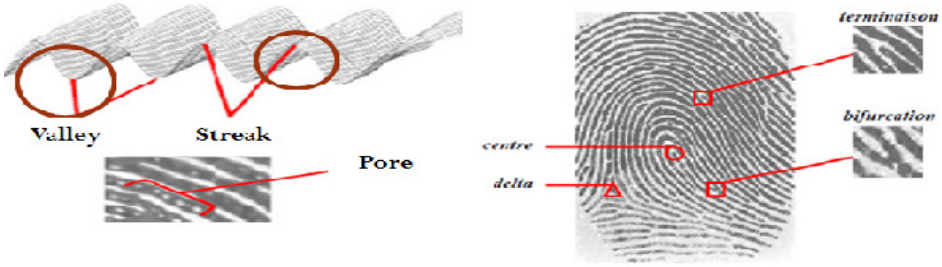

Figure 2. Fingerprint structure

When capturing a fingerprint image, several factors affect image quality such as:

- Scars in the finger, age of the person.
- Parasitic substances (water, grease, dirt, etc.)
- Capture environment: temperature (dilation, compression), humidity

2.2. PRE-PROCESSING ALGORITHM FOR FINGERPRINT RECOGNITION

To guarantee the reliability of the recognition, a pre-processing is important to eliminate the effect of these parasitic elements on the impression image. After this step, the image becomes clear and ready for the next step. The different pre-treatment phases are presented in Figure 3.

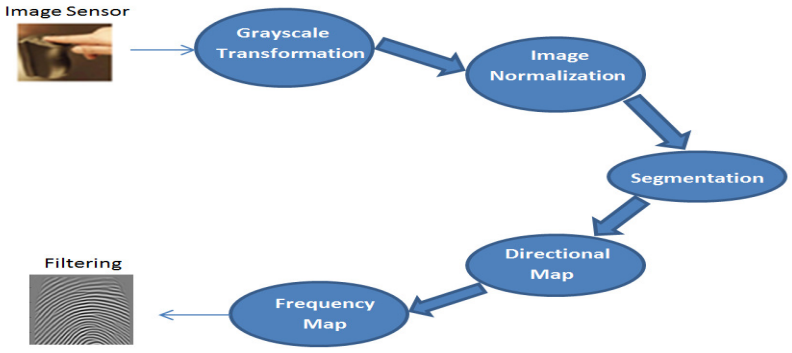

Figure 3. Preprocessing steps

Fingerprint recognition algorithms use several filtering methods in order to locate the useful points for each fingerprint. The most used filter is that of Gabor given its proven performance in this type of treatment.

However, in order to have reliable results, the input image of the filter must be as clear as possible, for this reason we propose a series of mathematical and morphological operations to improve the appearance of the image before pass to the filtering step.

2.2.1. GREY-SCALE TRANSFORMATION

The image becomes smaller because each pixel will be represented on 8 bits (from 0 to 255 grey levels) instead of 24 bits for the colour image (RGB).

2.2.2. NORMALIZATION

Normalization is used to standardize the intensity values in an image by adjusting the range of grey level values, the structure of the image does not change, and the variation in grey levels is standardized.

2.2.3. Segmentation

The segmentation makes it possible to eliminate the edges of the image as well as the too noisy zones

2.2.3. DIRECTIONAL MAP

The directional map makes it possible to specify the local direction of the constituent elements for each pixel (or block of pixel).

2.2.4. FREQUENCY MAP

The frequency map of the image consists of estimating the local frequency of the elements in each pixel.

2.2.5. GABOR FILTERING

The principle of filtering is to modify the value of the pixels of an image, generally in order to improve its appearance.

3. IMPLEMENTATION OF THE ALGORITHM IN A HARDWARE PLATFORM DSK TMS320C6416

Most of the image processing systems are developed based on Desktop PC which is more generic for image processing applications and these systems may not meet the requirement of real time performance[1] so the image processing and transmission system is increasingly developed along with the development of electronic technique, especially DSP and FPGA[12].

In order to achieve fast and intelligent image processing, image processing system is required to have powerful data processing capabilities and high flexibility. DSP with high-speed and high-performance is the key to achieve rapid processing [8]..

3.1. HARDWARE PLATFORM DSK TMS320C6416

DSK TMS320C6416 is a development platform for low cost applications and designed to quickly develop high performance applications. It is based on a DSP (Digital Signal Processor) of the Texas Instruments TMSC64x family [13].

The C6416 allows as to download and step through code quickly and uses Real Time Data Exchange (RTDX™) for improved Host and Target communications. The DSK utilities include Flash burn to program flash, Update Advisor to download tools, utilities and software and a power on self test and diagnostic utility to ensure the DSK is operating correctly. [14]

TMS320C6416 platform uses a USB connection which will later be transformed into a JTAG interface. The DSK features the C6416 DSP, a 600-MHz device delivering up to 4800 MIPS and designed to meet the needs of high-performing, memory-intensive applications such as networking, video, imaging and most multi-channel systems.

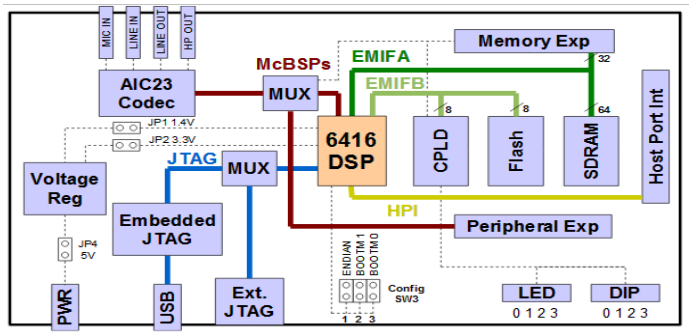

Figure 4. Block Diagram TMS320C6416 DSK

The core of the DSP is based on the Very Large Instruction Word (VLIW) architecture, which provides eight 32-bit instructions for every clock stroke, and with a maximum frequency of 1 GHz (generated by an internal PLL). Can therefore reach 8000 MIPS (Million of Instructions Per Second).

VLIW offers the advantages of superscalar implementations without the overhead expense of instruction scheduling hardware since all instructions are scheduled at compile time.
The instructions are grouped into 256-bit packets. There are two types of packages: "fetch" packages and "execute" packages. The first type has a fixed size of 256 bits, the packets are searched from the program memory, while the second type varies in size. It contains the instructions that will be executed in parallel, linked together by a bit to 1 in the LSB (Low Significant Bit) of each instruction, the instructions are said to be chained. The presence of a 0 in

the LSB of an instruction will cut the string and place the following instructions efficiently in another packet, one deduces that within an execute package the instructions are linked by 1 at the level Of the LSB.

A "fetch" packet can contain one or more "execute" packets (Figure 5):

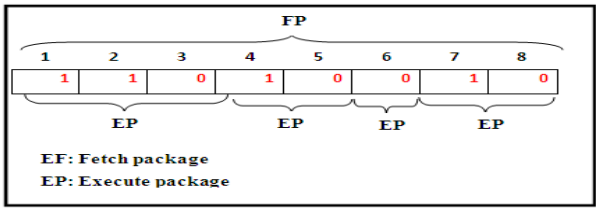

Figure 5. Fetch package and execute package

The DSP core contains a three-stage pipeline: "fetch", "decode" and "execute" (Figure 5). An execution cycle begins with the "fetch" or the search of the instruction packet from the program memory. Then, in the "decode" stage, the dispatch (DP) or the "fetch" packet is divided into "execute" packets, and for each instruction the appropriate functional unit is assigned. Each instruction is then decoded (DC). The following "fetch" packet is searched only when the "execute" packets are sent to the decoder. The implementation phase is divided into five phases. The first phase is used by any type of instruction for reading the operands (E1). The second phase is used to send addresses and data (E2). Subsequently, the data memory (E3) is accessed to be sent to the CPU (E5). Finally, the results are written in the registers (E5). These registers are 32 bits in number and 64 are divided into two parts to have two data paths A and B.

3.1.1. THE INTERNAL MEMORY

The internal memory, of size 1Mo, is an important advantage of this DSP. It is a fast on-chip memory, performance approaching the cache memories.

3.1.2. CACHE MEMORIES

The C64x uses a two-level cache-based architecture and has a powerful and diverse set of peripherals. The Level 1 program cache (L1P) is a 128-Kbit direct mapped cache and the Level 1 data cache (L1D) is a 128-Kbit 2-way set-associative cache. The Level 2 memory/cache (L2) consists of an 8-Mbit memory space that is shared between program and data space. L2 memory can be configured as mapped memory or combinations of cache (up to 256K bytes) and mapped memory.

3.1.3. EXTERNAL MEMORIES

The 6416DSK platform has two types of external memory:

- Synchronous DRAM: The DSK uses a pair of industry standard 64 megabit SDRAMs in CE0 of EMIFA. The two devices are used in parallel to create a 64-bit wide interface. Total available memory is 16 megabytes. The DSK uses an EMIFA clock of 100MHz. The integrated SDRAM controller is started by configuring the EMIF in software. When using the SDRAM, we note that one row of the memory array must be refreshed at least every 15.6 microseconds to maintain the integrity of its contents.

- The DSK uses a 512Kbyte external Flash as a boot option. It is connected to CE1 of EMIFB with an 8-bit interface. Flash is a type of memory which does not lose its

contents when the power is turned off. When read it looks like a simple asynchronous read-only memory (ROM). Flash can be erased in large blocks commonly referred to as sectors or pages. Once a block has been erased each word can be programmed once through a special command sequence. Then, after, the entire block must be erased again to change the contents. The Flash requires 70ns for both reads and writes. The general settings used with the DSK use 8 cycles for both read and write strobes (80ns) to leave a little extra margin.

3.1.4. IMPROVED DMA (EDMA)

A DMA (Direct Memory Access) is a component that allows the transfer of data blocks without the intervention of the processor, it only gives the transfer order and the addresses of the source and destination.

The EDMA (Enhanced DMA), is a peripheral that can be set up to copy data from one place to another without the CPU's intervention. The EDMA can be setup to copy data or program from a source (external/internal memory, or a serial port) to a destination (e.g. internal memory). After this transfer completes, the EDMA can "auto-initialize" itself and perform the same transfer again. It can also be reprogrammed.

3.2. PRACTICAL REALISATION

3.2.1. SUITABILITY OF THE ALGORITHM ON THE ARCHITECTURE

The properties of the TMS320C6416 DSK development platform presented above will be exploited in this part to implement the algorithm.

3.2.2. MEMORY ALLOCATION

Unlike the PC that has only one RAM, this platform contains two types of memory (internal and external RAM) with separate physical address ranges. On a PC, the memory management is done by the operating system, programmer C simply typing the commands of static allocation of variables (type variable_name or type table [] for the allocation of a vector) or dynamic (Malloc, calloc, realloc ...) and the operating system takes the responsibility of allocating memory areas for each variable.

For this development platform, you can use these memory allocation instructions, but you must add a command file (.cmd) and include it in the project. This file must fix the different memory sections. This operation is complex, and if you want to modify the project, you must also modify the partitioning memory.

A very practical solution is to use the Texas Instruments DSP / BIOS real-time kernel that is well suited to the architecture of the DSP used.
DSP/BIOS™ kernel is a scalable real-time multi-tasking kernel, designed specifically for the TMS320C6000™, TMS320C55x™, and TMS320C28x™ DSP platforms. Together with its associated networking, microprocessor-DSP communications, and driver modules, DSP/BIOS kernel provides a solid foundation for even the most sophisticated DSP applications. DSP/BIOS kernel is also one of the world's mostly widely used real-time operating systems.(Figure 6).

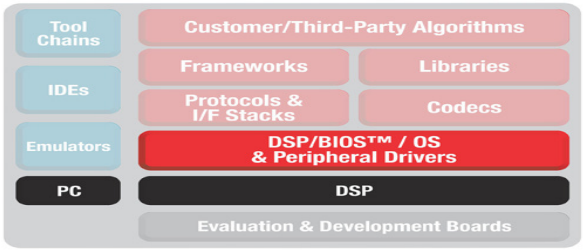

Figure 6. DSP BIOS kernel

DSP/BIOS kernel is available both standalone and as an integral part of the Code Composer Studio™ Interactive Development Environment (IDE) and includes graphical Kernel Object Viewer and Real-Time Analysis tools specifically focused on debugging and tuning multitasking applications.

The advantage of this kernel is that it provides standardized APIs across C6000™, C55x™ and C28x™ DSP platforms to support rapid application migration and is also optimized to run on the DSP cores.

For memory allocation, the MEM module was used. We must use a heap for memory allocation. A "heap" is a memory area reserved for dynamic allocations whose location in the memory range of the DSP is well defined. By using the instructions "MEM_alloc" and "MEM_calloc" which are APIs of the MEM module, the DSP / BIOS deals with the allocation. For each memory, it is possible to specify a heap and fix its size while giving it an identifier of the memory segment. Then, whenever an allocation instruction is used, the identifier of the heap must be specified to specify in which memory the variable (which is usually a vector) is to be allocated (Figure 7).

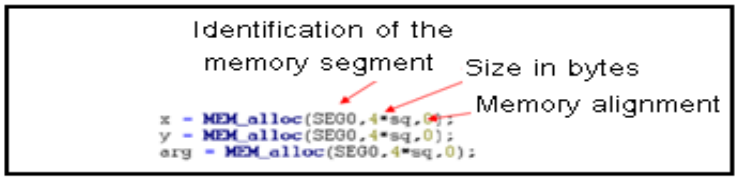

Figure 7. Memory allocation code

If a variable is statically allocated without specifying the memory segment, the DSP / BIOS will allocate these variables automatically in the internal RAM. The problem here is that the size of it is reduced (1 MB) and it contains the executable file, so we have to check that the overall size does not exceed the size of the internal RAM, otherwise the allocation would be made anywhere and the results will be wrong. Each allocated variable must be freed by the "MEM_free" instruction, otherwise it remains persistent in memory (Figure 8).

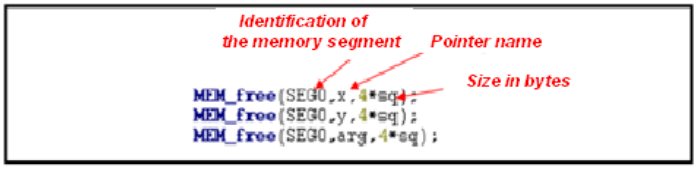

Figure 8. Memory release code

3.2.3. MEMORY ACCESS PROBLEM

An image processing application requires a large data size. Or, the fast internal memory can not contain all the data, so we must then use the external memory which is quite sufficient size. Similarly, for this type of application, there is also a large number of accesses to the external memory. However, with a CPU that can operate up to a frequency of 1 GHz and an external memory of 125 MHz, we note the big difference, the CPU remains in this case 7 clock cycles on hold at each memory access.

Therefore, to process on a 300 × 300 or 90 000 pixel image, a number of 7 × 90 000 = 630 000 lost clock cycles is lost, which gives 630 µsec, that represents a great loss and degrades performance considerably.

a. Use of the EDMA

A first idea is the use of EDMA (Enhanced Direct Memory Access) since it does not require the intervention of the CPU only to launch the transfer that takes place in the background.
We know that the internal memory is a fast memory. It was therefore thought to use the EDMA to bring the data to be used from the external memory to the internal memory, to process them and then to return those to their original place while using fixed size memory areas as a kind of buffer.

```
Void EDMA_2D_single(void*src,Void*dest,Uint32 num_elements)
{
EDMA_Handle hEdma;
Uint32 count;
Uint16 msb;
msb = num_elements / 4 = 1;
count = 4 + msb *0x00010000;

hEdma = EDMA_open(16, EDMA_OPEN_ENABLE);
/* EDMA_OPT Specifies the transfer action see EDMA document user guide
*Count specifies the number of frames in a block to the least significant word and the block number +1 in the
half-word
*The index is the penultimate parameter which gives the jump after each block
*The last parameter does not affect this type of option
*/
EDMA_configArgs(hEdma,EDMA_OPT,(Unit32)src,count,(Unit32)dest,0x00000000,0x00040000);
/*Set channel to start the transfer */
EDMA_setChannel (hEdma);
EDMA_close(hEdma);
}
```

Figure 9. Use function of EDMA

A function has been created which has as parameters a pointer to the source address, a pointer to the destination address and the number of elements to be transferred (Figure 9). This function allows the configuration of the EDMA register and start the transfer.

Example of using the EDMA: How do we setup the six EDMA parameters registers to transfer 4 byte-wide elements from loc_8 to myDest? (Figure 10)

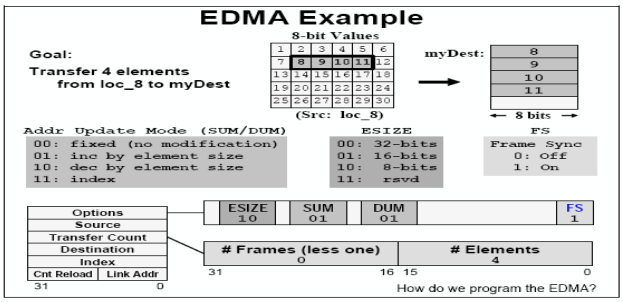

Figure 10.EDMA Example

- Looking at the EDMA parameters we register at a time.
- ESIZE should be self-explanatory based on our previous definitions.
- SUM and DUM fields indicates how the source and destination addresses are to be modified between elements and writes.
- Frame Sync (FS) indicates how much data should be moved whenever the EDMA is triggered to run.
- Source should have the source address of loc_8.
- Transfer counter will have the value 4. Actually it is 0x00000004.

It has been found that this solution is presents some problems such as the management of the order of transfer of the vectors (it is only possible to make one transfer at a time) and if there is a processing which requires a size of Which is greater than the size of the internal memory, then it is necessary to use the SDRAM memory. Moreover, on the performance side, we did not notice a big change concerning the execution time.

We then thought about the activation of the second cache level.

b. Using the cache memory

Why using the cache?
In a DSP application, a large amount of fast on-chip memory would be ideal. But in recent years, the performance of processors have increased significantly compared to those of the memories. Thus, there is now a gap between the speed of CPUs and memories. Ultra-fast memories exist but they consume much more memory size and are more and more expensive if their size increases in comparison with ordinary memories [15].

The flat architecture (figure 11 .a) shows that the processor and the fast on-chip memory have the same clock frequency (300 MHz), so there is no processor waiting problem. On the other hand, in order to access the external memory (of lower operating frequency), the processor must wait for that which degrades the performances. If the frequency of the CPU is further increased to 600 MHz, the difference would be greater and even the internal memory can no longer serve the processor than every two clock cycles.

In this case, the penalty would be important for the loops that make the memory access in each clock cycle. Unfortunately, the available memory manufacturing technology still can not keep up with the speed of the processors, and the ultra-fast on-chip memories are becoming more expensive by increasing the size of the memory.

The solution is to use a memory hierarchy (Figure 11 .b). A super fast and small memory is placed near the CPU so that it has access to this memory without waiting. This memory is the Cache Level 1 (L1 Cache) memory. We can insert another level of speed memory a little lower compared to that of the cache and larger size that will be an intermediary between the external memory and the Cache, it is called Cache level 2 (L2 Cache).

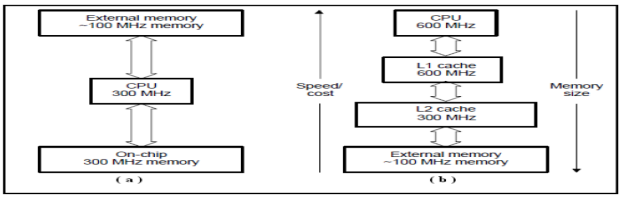

Figure 11.Architectures with and without cache

We therefore bring the data from the external memory to the cache L2, with a few waiting cycles. And when needed, the L1 cache brings the L2 data. Thus with this type of architecture, the average access time to the memory would be close to the access time of the fastest memory.

3.2.4. IMPLEMENTATION OF THE ALGORITHM

Writing to FLASH memory:

To test the functions implemented on the platform, we need test images. Since we do not have a sensor connected to the evaluation board, we had the idea to save some images in Flash memory. For this, we took advantage of the Flash Burn tool. This allows you to specify the location in the Flash memory where you want to write the data. However, this tool only writes specific file formats (.hex, .dat, or .txt). The type of .dat files has been chosen, which contains the values of the pixels of the images already transformed into gray levels in order to gain memory size.

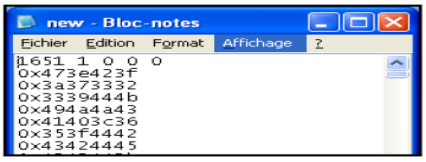

Figure 12. Format of DAT files

DAT files, in general, do not have a specific shape. They are raw, header-free data files. However, for this case, this type of file must have a well-defined header so that it can be read by the Flash Burn tool. This header must begin with a magic number "1651", then the format of the data (1 for hexadecimals, 2 for integers and 3 for real). Finally, the useful part can be written according to the format used. In this project, the hexadecimal format was used. During reading, the tool used reads in 4 bytes, that is to say 4 pixels at a time by storing them as they are in the memory.

Let's take the example of figure 12. The number 0x473e423f contains the value of 4 pixels in hexadecimal of respective values 0x47, 0x3E, 0x42 and 0x3F.

The hardware platform used makes good use of this concept. It has two caches level L1: one data (L1D) and one program (L1P). A portion of the internal RAM can be configured as an L2 cache memory.

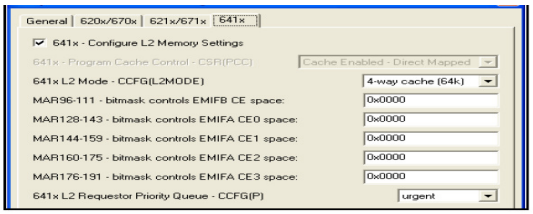

Figure 13. Configuring the L2 level of Cache

Activation of the L2 cache level can be done through the DSP / BIOS configuration interface. This architecture will be very interesting in this project, because the image processing requires a lot of loops and access to the external memory, and especially for the convolution operation.

4. RESULT AND DISCUSSION

In order to evaluate performance, we note the execution times of the different pre-processing) functions.

Table 1. Pre-processing Run Times

Function	Estimated execution time
Standardization & Segmentation	54.12 ms
Directional Map	314.22 ms
Frequency map	89.71 ms
Gabor Filtering	1215.8 ms

Total time: 1.6 Seconds.

We note that the function that took most of the execution time is the Gabor filtering. Indeed, the operation of this filter rests on a large number of convolutions of blocks of larger or smaller sizes. This explains the value obtained.

Also, in order to highlight the importance of image preprocessing we have considered a database based on a hundred images captured by different types of sensors, in the first place we tried to go directly to the stage filtering without making any changes to the images, we extracted the useful information from each fingerprint, and then we passed to the authentication phase. And then, second, we went through the preprocessing of images before filtering and the steps that follow; the different results are displayed in the following table:

Table 1. Result of pre-processing Application

	Image without pretreatment	Images with preprocessing
Total number of images	100	100
Rate of erroneous results	46	22

We therefore note that thanks to the various operations carried out to improve the quality of the images, the rate of the erroneous results is reduced by 24 %.

5. CONCLUSIONS

Biometrics is an emerging field where technology enhances our ability to identify a person. The advantage of biometric identification is that each individual has his or her own physical characteristics that can not be changed, lost or stolen.

Some are the type of sensor used or the algorithm principle applied; the image quality to be processed remains a very important factor that directly affects the overall system performance.
For this fact, we have proposed a set of operations to apply to the image in order to increase its quality to be able to facilitate the different phases of treatment that follow.

The results obtained are motivating in terms of total error rate reduction, however, the total processing time remains directly related to the type of hardware platform used.

We then applied a hardware implementation on a DSP processor, the results obtained show that the platform used presents good performances compared to an ordinary PC since such an algorithm presents several functions that rely on a large number of convolutions of blocks of larger or smaller sizes and that include several iterations and A large volume of data and mathematical functions, and this type of computation and processing generally requires powerful computers and well-dedicated processors, hence the importance of using the DSP especially at the level of functions with a specified processing of data. However the system is quit long in terms of execution time since this platform represents very limited memory resources.

REFERENCES

[1] M V Ganeswara Rao, P Rajesh Kumar and A Mallikarjuna Prasad, "Implementation of real time image processing system with FPGA and DSP"IEEE International Conference on Microelectronics, Computing and Communications (MicroCom), 2016

[2] Debnath Bhattacharyya, Rahul Ranjan1, Farkhod Alisherov A., and Minkyu Choi, Biometric Authentication: A Review, International Journal of u- and e- Service, Science and Technology, Vol. 2, No. 3, September, 2009

[3] Yun Zhang ; Zheng Ma and Yufeng He "A high-speed iris recognition system based on DM6446" IEEE 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), 2012

[4] Mariano Fons ; Francisco Fons and Enrique Canto "Embedded VLSI Accelerators for Fingerprint Signal Processing" IEEE International Symposium on Intelligent Signal Processing, 2007. WISP 2007.

[5] Maddu Kamaraju and Penta Anil Kumar "DSP based embedded fingerprint recognition system" IEEE 13th International Conference on Hybrid Intelligent Systems (HIS), 2013

[6] G. F. Zaki ; R. A. Girgis ; W. W. Moussa and W. R. Gabran "Using the hardware/software co-design methodology to implement an embedded face recognition/verification system on an FPGA" IEEE . Internatonal Conference on Microelectronics, 2007. ICM 2007

[7] http://www.agent-de-securite.be/biometrie/securite_biometrique.html

[8] Guodong Wang and Xiaojian Liu « A parallel image processing platform based on multi-core DSP" IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), 2017

[9] Ajay Rupani, Pawan Whig, Gajendra Sujediya and Piyush Vyas "A robust technique for image processing based on interfacing of Raspberry-Pi and FPGA using IoT" IEEE International Conference on Computer, Communications and Electronics (Comptelix), 2017

[10] L.Lam, S.W. Lee and C.Y. Suen, "Thinning Methodologies-A Comprehensive Survey", IEEE Transactionson Pattern Analysis and Machine Intelligence, Vol. 14, Issue 9, pp. 869-885, September 1992

[11] http://www.orcanthu s.com/main/product/product.php?idpr=4

[12] Duan Jinghon, Deng Yaling and Liang Kun "Development of Image Processing System Based on DSP and FPGA" IEEE 8th International Conference on Electronic Measurement and Instruments, 2007. ICEMI '07.

[13] http://www.ti.com/tool/tmdsdsk6416

[14] https://www.ti.com/seclit/wp/spry051/spry051.pdf

[15] R. Stefanelli, A. Rosenfeld, Some parallel thinning algorithms for digital pictures, Journal of the ACM, Vol.18, No2, pp.255-264, April 1971.

AUTHORS

Farah Dhib Tatar
Specialty: Electrical Engineering (Embedded Systems)

Education:

- 2010: Engineering Degree from sfax College of Engineering, ENIS, Tunisia
- 2012: Master degree from Sfax College of engineering, ENIS, Tunisia
- In progress: Ph.D. degree from Tunis College of engineering, ENIT, Tunisia
- Experience:
- 2010-2016: College of Higher Studies and Technologies ISET, Tunisia

Dr. Mohsen Machhout
Specialty: Electrical Engineering (System of comminication)

Experience:

- 1996-now: Faculty of Science of Monastir, Tunisia

Dr. AGUILI Taoufik
Head of Research Structure. The (Sys'Com)
Communication System Lab.
National school of the studies of engineer of Tunis,Tunisia.