FPGA Implementation of Face Recognition Algorithm

Tijana Šušteršič(✉), Aleksandra Vulović, Nenad Filipović,
and Aleksandar Peulić

Faculty of Engineering, University of Kragujevac,
Sestre Janjić 6, 34 000 Kragujevac, Serbia
{tijanas, aleksandra.vulovic, fica,
aleksandar.peulic}@kg.ac.rs

Abstract. Field of face recognition has been developing in the past several decades. Although percentage of successful recognition algorithms is constantly getting higher, there is room for improvement. Field Programmable Gate Array (FPGA) is technology that can be used for speed and accuracy improvement. The main goal of this paper was to load photos from files to FPGA and display, as well as describe implementation of the Eigenface algorithm on DE2 Altera board. We showed that DE2 Altera board can be used for reading databases and photos from crime scenes and discussed how Eigenface algorithm can be implemented on this board in order to speed up the process of face recognition. Speed of recognition process is an area where improvement is necessary, especially considering the need for instant face recognition in places like airports, or public meetings.

Keywords: FPGA · Face recognition · Eigenface algorithm

1 Introduction

In recent years, researches in face recognition techniques have gained significant momentum [1]. Though it is much easier to install face recognition system in a large setting, the actual implementation is very challenging as it needs to account for all possible appearance variation caused by change in illumination, facial features, variations in pose, image resolution, sensor noise, viewing distance, occlusions, etc. [2].

In the past decade, an increased number of requests was addressed to forensic expert witnesses in Serbia with the focus on verification of the identity of individuals presented on video or images available from crime scenes. Mainly due to budgetary constraints, Serbia did not develop any automatic or semiautomatic system for face recognition. Practically, when the police and court obtain image-based evidence from video surveillance cameras, forensic experts are only left with manual analyses. Moreover, the process is limited to few experienced forensic experts and their traditional anthropological approach which is not based on objective measurements. The method lacks consistent methodology and there is a great need to standardize the process. Moreover, the numerous studies show that using human perception alone (e.g. eye witnessing) is not always a reliable source to confirm identification, and is

N. Oliver et al. (Eds.): MindCare 2016/Fabulous 2016/IIoT 2015, LNICST 207, pp. 93–99, 2018.
https://doi.org/10.1007/978-3-319-74935-8_13

significantly affected by differences in lighting, familiarity, expression and viewpoint or pose [3]. The application of face recognition techniques can be categorized into two main parts: law enforcement application and commercial application. Law enforcement applications would include identification of wanted individuals, while the commercial applications range from static matching of photographs on credit cards, ATM cards, passports, driver's licenses, and photo ID to real-time matching with still images or video image sequences for access control. Each application presents different constraints in terms of processing [4]. Most viable application in the area of law enforcement is at the airports and in transportation (e.g. airports, train stations, border crossings). There is also an application in public security systems (e.g. criminal identification, digital driver license) [5]. Commercial use of face recognition can be found in identification systems (e.g. automatic banking, computer log-in, etc.) [5] and gaming (e.g. keeping out the compulsive gamblers).

Among different types of algorithms for face recognition, those that are simple, and at the same time efficient, stand out. In high-dimensional data, the Principal Component Analysis (PCA) is designed to model linear variation. Its goal is to find a set of mutually orthogonal basis functions that capture the directions of maximum variance in the data and for which the coefficients are pairwise decorrelated [6]. PCA was used to describe face images in terms of a set of basic functions, or "eigenfaces". Eigenfaces was introduced early on [7] as a powerful use of PCA to solve problems in face recognition and detection. Principal component analysis for face recognition is based on the information theory approach in which the relevant information in a face image is extracted as efficiently as possible [8].

Most of the available algorithms are implemented in software. As a result, the recognition speed is not as expected [9]. On the other hand, hardware implementation such as Field Programmable Gate Array (FPGA) has many promises. In recent years, people have made effort to apply it to biology and neuroscience due to its favorable performance [9]. Compared with the software simulation, FPGA shows more advantages over PC-solution. First, parallel processing of FPGA significantly improves computational efficiency, which effectively solves the time-consuming problem in a general-purpose system. Second, because of its re-configurable nature, FPGA implementation allows for development of a module repertoire which includes a variety of neuron models for different purposes [10].

The DE2 board has many features that allow a user to implement a wide range of designed circuits, from simple circuits to various multimedia projects. DE2 board used in this paper has Altera Cyclone® II 2C35 FPGA device (Fig. 1).

It has Altera Serial Configuration device – EPCS16, USB Blaster, 512-KB SRAM, 8-MB SDRAM, 4-MB Flash memory, SD Card socket, 4 pushbutton switches, 18 toggle switches, 9 green user LEDs, 50-MHz oscillator and 27-MHz oscillator for clock sources, VGA DAC with VGA out connector, RS-232 transceiver, PS/2 mouse/keyboard connector, two 40-pin Expansion Headers with diode protection [11].

Therefore, our motivation was to load face photos from AT&T database and to discuss hardware implementation of the Eigenface algorithm in a Cyclone II Field Programmable Gate Array (FPGA) chip from Altera Inc.

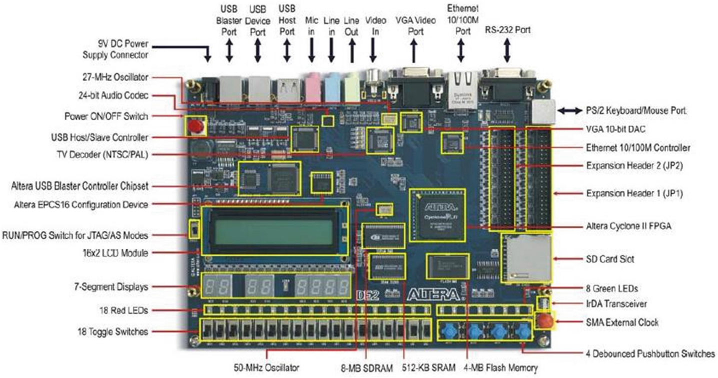

Fig. 1. The DE2 board

2 Materials and Methods

Material used for this paper was AT&T database which consisted of grayscale photos of 40 people, where each person was photographed 10 times (400 photos on the whole). These 10 photos of each person were made at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). Each image had the size of 112 × 92 pixels with 256 levels of grey.

FPGA can be used to implement any logical function that an application-specific integrated circuit (ASIC) could perform. Unlike previous generation FPGAs using I/Os with programmable logic and interconnects, today's FPGAs consist of various mixes of configurable embedded SRAM, high-speed transceivers, high-speed I/Os, logic blocks, and routing. We used that property of writing from a file to SRAM to load any photo from AT&T database into the FPGA's SRAM memory and display it on a computer monitor connected to an FPGA via VGA video port.

In order to implement an Eigenface algorithm, we have to load the database and the photo from crime scene. This is done by storing the photos in chip memory and processing the images with Eigenface algorithm using Nios II after which the results are displayed on a computer monitor (Fig. 2).

Furthermore, we present Eigenface algorithm for face recognition image processing implemented on a FPGA.

Implementation of the Eigenface algorithm

Step 1. Identification by eigen faces

a. Compute the average face
b. Subtract the average face from the training faces
c. Compute covariance matrix
d. Determine eigenvectors and eigenvalues
e. Choose a certain number of eigenvectors with highest eigenvalues

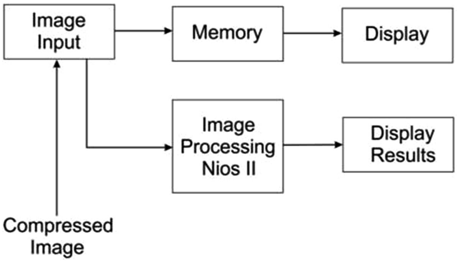

Fig. 2. Block diagram of image processing on a FPGA chip

f. Project training faces into Facespace by multiplying the eigenvectors with the matrix obtained in the step 1-b.

Step 2. Recognition by matching eigen faces

a. Subtract the average face from the face
b. Compute the projection into Facespace by multiplying the eigenvectors with the matrix obtained in the step 2-a
c. Compute the threshold based on the distance in the face space between the face and all known faces by using Euclidian distance.

Step 3. Categorization of the result by grouping

a. Distinguish between the cases based on the threshold obtained in step 2-c
    I. It is an unknown face (person is not in database) - update the database with this face and inform the user that it is an unknown face
    II. It is a known face (person is already in database) - inform the user what are the possibilities and probabilities for correct recognition (pick three photo matches with shortest Euclidian distance).

It can be seen that the operations in Eigenface algorithm include simple mathematical operations with huge matrices. In cases of even greater databases, the speed of recognition is relatively small. An FPGA can provide us with necessary resources to achieve such improvements in face recognition process speedup. The resources include built in blocks, various communication interfaces, millions of logic gates, scopes to run C codes into the digital hardware circuitry, high level design tools, performance, long term maintenance, reliability, etc. The cyclone chip is relatively cheaper and includes ROM. DE0 board has been chosen as a tool for debugging process. We have only discussed implementation of Eigenface algorithm.

3 Results and Discussion

One of the prime concerns of our research was to start with the simpler algorithm, to confirm that it was possible to implement other, more complicated, algorithms using FPGA, so that we can work on it in future.

To achieve implementation of Eigenface algorithm on Altera Cyclotrone II FPGA, a processor using Qsys needs to be created. Various components such as CPU, SDRAM, PLL, Tri state bridge, Onchip memory, etc. need to be added. Connection by connecting master to slave, source to sink, assigned base address and connected clock through PLL needs to be made.

The result of writing a file to an SRAM displaying it on a computer monitor connected to an FPGA via VGA video port is given in Fig. 3. The visualization was achieved using Quartus II 13.0 which was used in order to run the DE2 Control Panel throughout which the image was loaded and stored in SRAM memory as well as for the SRAM controller the User Port 1 (Asynchronous 1) is chosen.

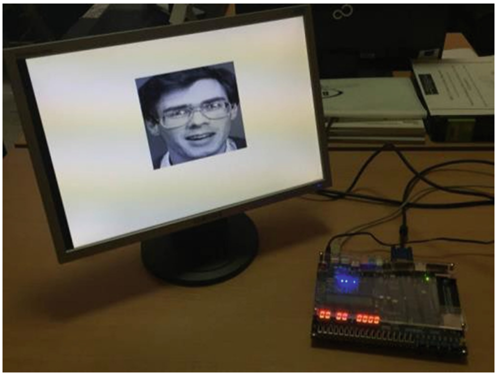

Fig. 3. Displaying loaded photo on a computer monitor connected to an FPGA

Next step would be to write Verilog/VHDL code to interface in our FPGA through pin assignment. Then we should include our SOPC code in Verilog/VHDL code and interface with our board's pin which generates the .SOF file. That would complete the hardware configuration. Another possibility is to write our Matlab code for Eigenface algorithm and convert it to HDL code. We showed that it was possible to load any photo to an FPGA which can be further used in the process of recognition.

The time constraints of our system are bounded by the time constraints required by real time face recognition. That is, in applications where the recognition of multiple faces is required, the process must not take more than 2 s for each person. However, this process includes the overhead of obtaining the image, performing face recognition, and displaying the results. Normally, using CPU, the face recognition process is in the order of milliseconds when working with small databases. Logical parallelism within an image processing operation is well suited to FPGA implementation when working with large databases, and it is here where many image processing algorithms may be accelerated significantly. This is accomplished by unrolling the inner loops, so that rather than performing the operations sequentially, parallel hardware is used.

Prior to obtaining any measurements, we developed a hypothesis that the projection phase would consume the longest portion of time. This assumption stemmed from the fact that this phase involves high computational demand in the form of a matrix multiplication operation. In [9] it took approximately 10% less clock cycles to execute face recognition algorithm using hardware FPGA than the software implementation.

4 Conclusion

FPGA is a piece of hardware that implements thousands of gates of logic that are preferably used when the speed up in the design process is desired. Using hardware programming languages such as VHDL and Verilog someone can create complex logic structures. Speed is the biggest advantage of FPGA. It is reprogrammable so that more than one project can be implemented using same FPGA board. FPGAs exceed the computing power of digital signal processors by taking the advantage of hardware parallelism.

Therefore, our goal was to outline the possibilities of implementation of a simple algorithm for face recognition – Eigenface on a FPGA. Since the algorithm itself includes simple mathematical operations, total benefits of the FPGA can be obtained. We loaded photos from the AT&T database and displayed them on a computer monitor. Furthermore, we gave the pseudo code for Eigenface algorithm which will be used in order to perform face recognition image processing by Nios II.

However, FPGAs are more expensive than microcontrollers. If the designed problem needs greater integration density, then FPGAs are appropriate. For smaller projects, microcontrollers would be a better solution. For the purpose of creating the Face recognition system in Serbia, FPGAs would greatly help. Future work with FPGA will include real time face recognition on FPGA Altera board.

References

1. Li, Z.S.: Encyclopedia of Biometrics. Springer, New York (2009)
2. Surti, R.N., Rodrigues, A.: Performance evaluation of statistical approaches of face recognition techniques. Int. J. Statistika Mathematika **10**, 08–11 (2014)
3. Smeets, D., Claes, P., Vandermeulen, D., Clement, J.G.: Objective 3D face recognition: evolution, approaches and challenges. Forensic Sci. Int. **201**, 125–132 (2010)
4. Tolba, A.S., El-Baz, A.H., El-Harby, A.A.: Face recognition: a literature review. Int. J. Sig. Process. **2**, 88–103 (2006)
5. Mou, D.: Machine-Based Intelligent Face Recognition. Springer, Heidelberg (2010)
6. Ragab, A., Albert, C., Senousy, M.B.: A comparative study of face recognition techniques. Int. J. Electron. Commun. Comput. Eng. **6**, 762–770 (2015)
7. Turk, M., Pentland, A.: Eigenfaces for recognition. J. Cogn. Neurosci. **3**, 71–86 (1991)
8. Agarwal, M., Agrawal, H., Jain, N., Kumar, M.: Face recognition using principle component analysis, eigenface and neural network. In: International Conference on Signal Acquisition and Processing (ICSAP 2010), pp. 310–314. IEEE Computer Society (2010)
9. Dewan, P.D., Shamma, T.H., Abbas, A., Mondol, R.K.: Design and VLSI implementation of high performance face recognition system. A report submitted to department of Electrical & Electronic Engineering, BRAC University in partial fulfillment of the requirements for thesis work (2013)

10. Eric Deng, V.P.: Accelerator for POWER, revolutionizing the datacenter. In: Open Power Summit, San Jose, CA, 5–8 April 2016
11. Terasic Technologies: Development and Education Board DE2, User Manual (2006). https://www.terasic.com.tw