FPGA-based Low-Cost Real-Time Face Recognition

Laszlo Schaffer
Dept. of Technical Informatics
University of Szeged, Szeged, H-6725
Email: schaffer@inf.u-szeged.hu

Zoltan Kincses
Dept. of Technical Informatics
University of Szeged, Szeged, H-6725
Email: kincsesz@inf.u-szeged.hu

Szilveszter Pletl
Collage of Applied Sciences
Subotica Tech, Subotica, Serbia
Email: pszilvi@vts.su.ac.rs

***Abstract*—Nowadays face recognition plays a central role in surveillance, biometrics and security. In this paper a Field-Programmable Gate Array (FPGA) based low-cost real-time architecture for face recognition is presented. The face recognition module receives the detected faces from a video stream and processes the data with the widely used Eigenfaces, also known as the Principal Component Analysis (PCA) algorithm. The architecture is implemented on a low-cost Zynq-Z7010 FPGA. The proposed architecture is a part of a system, that capable of finding faces in a crowd based on a preliminarily defined set of faces. In the future it can be integrated into real-time surveillance systems of frequently crowded places (e.g. airports, bus stations), to sort out the supposed sources of threat, and hereby reduce the risk of possible criminal actions.**

I. Introduction

In the recent years automated person identification became a widely studied area, because entry control and security are playing a central role in the industry. Furthermore, public places are exposed to criminal activity, therefore eliminating threat sources from frequently crowded places also requires the recognition of known criminals. Face recognition is a well applicable solution to person identification, and has many applications in surveillance, access management, law enforcement and biometrics. In general the number of persons to be identified are low, but there are places e.g. bus stations, airports where a high number of faces need to be identified in a very short amount of time. In the identification process it is required to compare the face of the person to the faces in the database. Therefore the recognition time depends on the number and size of the stored faces. In order to identify a person in real-time the usage of a specialized hardware is necessary.

A widely used approach for face recognition is the Eigenfaces approach, which means the usage Principal Component Analysis (PCA) on a database containing human faces. Due to the dimension reduction of the features the calculation can be faster and the implementation is easier than other face recognition approaches. It is concluded, that the Eigenfaces is tolerant to small rotation, translation and scale changes [1].

In the literature various software-based approaches can be found for face recognition. In [2] the applications of PCA are discussed in vision based computing. The results showed, that PCA based dimension reduction and image classification has a potential for industrialization, but it requires large amount of computation. A new two-dimensional technique for PCA is shown in [3], where PCA is based on 2D image matrices, rather than 1D. This 2DPCA is more efficient, than simple PCA, but it requires more memory. A face recognition algorithm based on a modular PCA approach is presented in [4]. The modular approach performs better in various illumination and expression conditions, but the computational cost is increased. Although the software-based PCA face recognition implementations are capable of accurate classification, but their real-time usage is not possible.

Field-Programmable Gate Arrays (FPGA) are applied to improve computational efficiency with hardware parallelism, therefore a high level of acceleration is possible. In the literature several FPGA-based face recognition approaches can be found. In [5] a real-time PCA-based portable emotion detection system is proposed, which is capable of helping autistic children understand facial emotions of other people. The proposed architecture uses an optimized eigen calculation with power-deflation iteration method, and implemented on a Virtex-7 FPGA. The results showed a 82.3% detection accuracy. An FPGA-based neural network, that capable of PCA computation is presented in [6]. The algorithm was implemented on a Virtex-2 FPGA, and this architecture can recognize 1400 faces in an image frame, which makes it suitable for real-time face recognition. In [7] an FPGA-based face recognition architecture is proposed, which performs PCA on wavelet transformed images. The architecture is implemented on a Virtex-2 Pro with 46.79 MHz and capable of 100% classification accuracy with the usage of 40 eigenvectors/face.

In this paper a low-cost FPGA-based architecture for face recognition is presented, which is capable of real-time classification of human faces using Eigenfaces, also known as the PCA algorithm. The proposed face recognition architecture is the part of an automated person identification system, which enables real-time surveillance and threat monitoring.

The paper is organized as follows. In Section II the mathematics and the operation of the Eigenfaces (PCA) is described in detail. The automated person identifier system is presented in Section III. The FPGA implementation of the Eigenfaces can be read in Section IV. In Section V the results is presented, while Section VI concludes the paper.

II. The Eigenfaces method

The main idea behind the Eigenfaces method is to use only the most relevant information possible from the images of human faces and make a comparison based on those features between the faces. Using the variation of face images for the comparison base is a simple, but efficient approach. Mathematically that means the principal component extraction of the face distribution, which is identical to calculate the covariance matrix of the set of human face images and then compute the eigenvectors of the covariance. The eigenvectors are the features, that together describe the face images and can be displayed as a ghostly face image, called the eigenface. The eigenfaces characterize the variation between the face images, therefore each human face can be reconstructed by a linear combination of the eigenfaces. The eigenvectors with the largest eigenvalues belongs to the highest variance between the face images, so a subspace from the eigenfaces can be spanned, that called the face space. Therefore each individual human face can be characterized by a small set of eigenvectors, which means a compact and efficient representation of the original image.

The EigenFaces method or PCA can be separated into the learning and the recognizing phase, as can be seen in Fig. 1. In the learning phase firstly a set of human face images are needed, which will be the training set. Then calculate the eigenfaces of the training set and define the face space. Finally the projection of the face images onto the face space is required, by computing the distribution in the face space of each face image. These steps could be performed again whenever a new face has to be learned. After the learning phase ended, the recognizing phase can begin by computing the eigenvectors of the face to be recognized. Finally the projection to each eigenface is required, which will show the degree of similarity between the input face image and the set of training faces.

Furthermore it can be determined, that the input is a face or another object by the distance between the projection result and the elements of the face space. Also there is the potential to learn the unknown faces if that face is already recognized as unknown face several times.

A. Eigenface Calculation

A face image I is a two-dimensional NxN array containing grayscale $0-255$ values. In the calculation of the eigenfaces firstly the creation of column vectors is required from the original face images. Consequently the length of the vectors will be N^2. The next step is to compute the mean face, which can be done as follows:

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n, \tag{1}$$

where Ψ is the mean face, Γ_n is the n_{th} face image from the training set and M is the number of faces. Then the difference from the average can be calculated:

$$\Phi_i = \Gamma_i - \Psi, \tag{2}$$

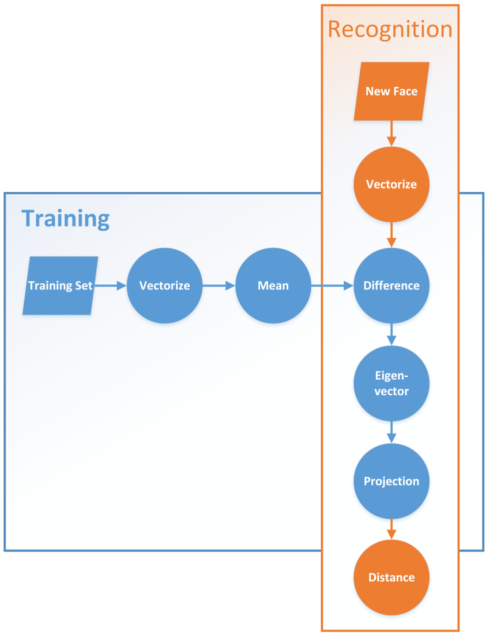

Fig. 1. Schematic diagram of the EigenFaces method

where Φ_i is the i_{th} difference image. Based on the difference images the distribution of the data has to be defined by finding the orthonormal vectors u, also knowns as the eigenvectors. The k_{th} eigenvector can be chosen as:

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M} {u_k^T\Phi_n}^2, \tag{3}$$

where λ_k is the k_{th} eigenvalue of the C covariance matrix:

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T = AA^T, \tag{4}$$

where the A matrix consists of the difference images $[\Phi_1\Phi_2...\Phi_M]$. However the covariance matrix has an N^2xN^2 dimension, and the computation of the eigenvectors could be time and resource consuming even for small images. Therefore a computationally feasible approach is required for the eigenvector computation.

There is a practical solution for the problem. Based on the number of images there will be only $M-1$ usable eigenvectors with the highest eigenvalues, while the other eigenvectors have eigenvalues close to zero. This scenario can be solved easily by calculating the eigenvectors of a MxM dimension matrix, rather than a $4096x4096$ matrix. Taking the eigenvectors v_i of A^TA as:

$$A^TAv_i = \mu_i v_i, \tag{5}$$

and multiplying (5) with the A matrix on both sides:

$$AA^TAv_i = \mu_i Av_i, \tag{6}$$

where μ_i is the i_{th} corresponding eigenvalue, it can be seen, that Av_i is the eigenvector of the original covariance matrix

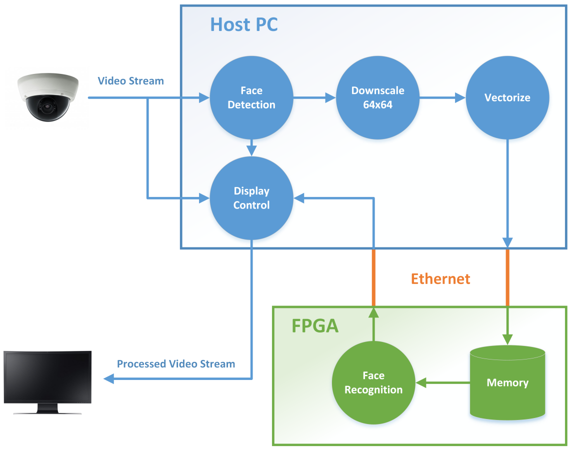

Fig. 2. Schematic diagram of the automatic person identification system

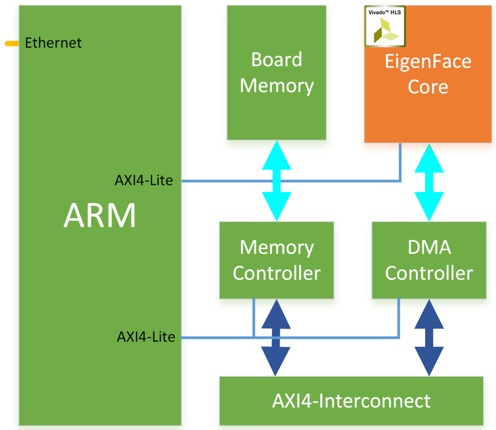

Fig. 3. Architecture of the face recognition system

C. Using this method the eigenfaces of the M training face images can be determined as follows:

$$u_i = \sum_{k=1}^{M} v_{ik}\Phi_k, \quad i = 1 \dots M \tag{7}$$

Using this method the order of computations is only M, which is the number of images in the training set, rather than N^2, which is the number of pixels in one image.

B. Classification

In the classification process a new face is projected onto the face space as follows:

$$\omega_i = u_i^T(\Gamma_{new} - \Psi), \quad i = 1 \dots M, \tag{8}$$

where ω_i is the i_{th} projection value based on the i_{th} eigenface and the new face image Γ_{new}. The projection values represent the contribution to the corresponding eigenfaces to reconstruct the input face. A vector Ω can be formed of the projection values as follows:

$$\Omega = [\omega_1 \; \omega_2 \dots \omega_M] \tag{9}$$

Based on the Ω projection vector the most similar face can easily determined by a distance metric as follows:

$$d_i = ||\Omega_T - \Omega_i||, \quad i = 1 \dots M, \tag{10}$$

where Ω_T is the projection matrix based on the training images. The minimum of the distances will be the most similar face to the new face [8]-[9].

III. Overall System

The automatic person identification system which capable of detecting and recognizing human faces can be seen in Fig. 2. The system consists of a video camera, a host PC, an FPGA, and a display. In this research only the face recognition has been implemented and tested on the FPGA hardware, the other software parts of the system are running in MATLAB on the PC. The configuration of the PC consists of a Core i7-4770 running on 3.4 GHz and 8 GB DDR3 RAM.

The automatic identification system works as follows. The video stream provided by the video camera is processed on the host PC to detect faces using the Viola-Jones algorithm with Haar-like features [10]. The detected faces are downscaled to $64x64$ and vectorized, which is the preparation for the face recognition with the Eigenfaces method. The FPGA can be configured through the Gigabit Ethernet to learn or recognize the transferred face, therefore it can be used to learn faces automatically or to define the training database. The vectorized faces are transferred via Gigabit Ethernet to the FPGA, which classifies the faces in real-time based on the training database. The classes of the identified faces then transferred back to the PC, which visualizes the results on the monitor based on the original video frame, the detected faces and the resulting face classes.

IV. FPGA Implementation

The face recognition architecture was implemented on a Zybo development board using Vivado 2016.4 and Vivado High Level Synthesis (HLS). Vivado HLS enables the usage of C/C++ or SystemC with some FPGA specific restriction. The Zybo development board contains a a Zynq-Z7010 FPGA with 650 MHz dual-core Cortex-A9 microprocessor, an external 512 MB DDR3 memory with 1050 Mbps bandwidth, a high-bandwith 1 Gigabit Ethernet, and other peripherals.

A. The architecture

The architecture of the proposed system is built-up from six main parts as can be seen in Fig. 3. These are the *ARM Processor*, the *DMA Controller*, the *AXI-4 Interconnect*, the *Memory Controller*, the *Board Memory* and the *EigenFace Core*.

The *ARM Processor* communicates with the host computer via Gigabit Ethernet and controls the data-flow on the AXI4-Lite and AXI4 buses. Furthermore it pre-calculates the required matrices for the *EigenFace Core*. The *Memory Controller* and the *Board Memory* are responsible to store the human faces extracted from the video stream and also the sub and final results of the classification process. The data

TABLE I
RESOURCE REQUIREMENTS OF THE EIGENFACES CORE

#	BRAM(18K)	DSP48E	FF	LUT
Available	120	80	35200	17600
Required	80	11	2802	4493
Utilization	66%	13%	7%	25%

transfer between the *EigenFace Core* and the *Board Memory* is handled by the *DMA Controller*. In the memory the data is stored sequentially, but the *EigenFace Core* requires it in a mixed manner, therefore scatter-gather (SG) DMA instructions are used. The *EigenFace Core* computes the algorithmic steps of the EigenFaces method. In the first step it calculates the mean face. In the second step difference from the mean is computed for each face vector. As the third step the eigenvectors are computed. In the fourth step the projection to the face space is computed. In recognition mode, the distances between the projections of the training set and the projection of the input face should be calculated. In the *EigenFaces Core* only the data for the actual step is stored, to minimize the BRAM memory usage. The *AXI-4 Interconnect* provides the connection between the parts of the system.

B. Implementation

In mean face calculation only addition and a multiplication with the reciprocal of M is required at the end of each row, when the faces are column vectors. This calculation can be done parallel with a factor of 64, which is one dimension of the faces. The eigenvector computation consists of three matrix multiplication, which unfortunately can not be done parallel due to data dependency. However the multiplication of two matrices can use parallelism in the resulting matrix element calculation, therefore one element of the new matrix can be computed in a clock cycle. After one column of the resulting matrix is computed it can be used immediately for the next matrix multiplication. The projection step includes the multiplication of each face vector with the eigenvector matrix, which can be parallel with a factor of 64. Finally the euclidean distance calculation is required, which is done serially due to the high DSP requirements of the square roots.

V. RESULTS

The resource requirement of the EigenFace Core can be seen in Table I. To maintain a low Block RAM memory usage, the required matrices are loaded from the DDR3 memory in every step and used immediately.

The FPGA implementation of the EigenFaces method has been tested with the Faces94 database, which contains 20 face images of 153 people. The facial expression on the images are slightly varied. From the 153 people 8 were chosen for testing. From the 20 images of the 8 people only 1 was used for training and the other 19 for validation. Using these face images 95% recognition efficiency can be achieved, with a 13026 faces/second throughput on 100 MHz clock frequency.

TABLE II
COMPARISON OF SYSTEM COSTS

Paper	FPGA	Cost (EUR)	Face/Sec	Cost / Face / Sec (EUR)
[5]	Virtex-7	3613	-	-
[6]	Virtex-2 PRO	1045	-	-
[7]	Virtex-2 PRO	1045	43049	0.02
Proposed	Zynq-Z7010	169	13026	0.013

A comparison with the system costs of previous works can be seen in Table II, which shows that the proposed system is more cost-efficient, than the similar real-time FPGA-based systems.

VI. CONCLUSION

In this paper a low-cost real-time FPGA-based implementation of the EigenFaces method for human face recognition is proposed.

The test results show that the architecture is capable of 95% recognition accuracy, and can process 13026 faces in a second. Furthermore, the architecture can be implemented on a low-cost Zynq-z7010 FPGA.

Our future work is to implement a face detection algorithm on FPGA, and use it together with the proposed face recognition architecture.

REFERENCES

[1] A. Lemieus, M. Parizeau, "Experiments on Eigenfaces Robustness", Proceedings of the IEEE International Conference on Pattern Recognition, August 2002, pp. 421-424

[2] C. Z. Liu, M. Kavakli, "Extension of principle component analysis with applications on vision based computing", Springer, Multimedia Tools and Applications, Vol. 75, Issue 17, September 2016, pp. 10113-10151

[3] Jian Yang, D. Zhang, A. F. Frangi, Jing-yu Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition", IEEE Transactions on Pattern Recognition, Vol. 26, Issue 1, June 2004, pp. 131-137

[4] R. Gottumukkal, V. K. Asari, "An improved face recognition technique based on modular PCA approach", Elsevier, Pattern Recognition Letters, Vol. 25, Issue 4, March 2004, pp. 429-436

[5] K. G. Smitha, A. p. Vinod, "Low Complexity FPGA Implementation of Emotion Detection for Autistic Children", 7th International Symposium on Medical Information and Communication Technology, March 2013

[6] A. R. Mohan, N. Sudha, P. K. Meher, "An Embedded Face Recognition System on A VLSI Array Architecture and its FPGA Implementation", 34th Annual Conference of IEEE on Industrial Electronics, November 2008

[7] A. Y. Jammoussi, S. F. Ghribi, D. S. Masmoudi, "Implementation of face recognition system in Virtex II Pro platform", IEEE, 3rd International Conference on Signals, Circuits and Systems, November 2009

[8] M. Turk, A. Pentland, "Eigenfaces for Recognition", MIT Press, Journal of Cognitive Neuroscience, Vol. 3, Issue 1, January 1991, pp. 71-86

[9] H. Abdi, L. J. Williams, "Principal component analysis", Wiley Interdisciplinary Reviews: Computational Statistics, Vol. 2, Issue 4, June 2010, pp. 433-459

[10] P. Viola, M. J. Jones, "Robust Real-Time Face Detection", Springer, International Journal of Computer Vision, Vol. 57, Issue 2, May 2004, pp. 137-154