

THIẾT KẾ SOPC CHO ỨNG DỤNG NHẬN DẠNG MẶT NGƯỜI DÙNG THUẬT TOÁN WMPCA

Trương Thanh Như, Trần Thị Điểm

Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM

(Bài nhận ngày 21 tháng 03 năm 2011, hoàn chỉnh sửa chữa ngày 22 tháng 11 năm 2011)

TÓM TẮT: Phần cứng gia tốc cho kỹ thuật lượng tử hóa vector (Vector Quantization-VQ) đã được phát triển thành một thành phần nhúng (system on a programmable chip) trong các ứng dụng nén ảnh và nhận dạng ảnh thời gian thực. Ngày nay, với kỹ thuật FPGA (Field Programmable Gate Array) và công cụ SoPC (system on a programmable chip) cho thấy được sự hiệu quả cao trong việc thiết kế các ứng dụng phần cứng gia tốc. Bên cạnh đó, một trong những phương pháp xác suất thống kê, phân tích thành phần chính điều biến trọng số, cho ta thấy sự hiệu quả cao trong các ứng dụng nhận dạng ảnh. Bài báo này sẽ giới thiệu một kiến trúc song song dựa trên thuật toán WMPCA và kiến trúc SoPC cho hệ thống nhận dạng mặt người online.

Từ khóa: SoPC, nhận dạng mặt người, thuật toán WMPCA.

MỞ ĐẦU

Nhận dạng nói chung, nhận dạng mặt người nói riêng là một bài toán đã và đang được nghiên cứu rất rộng rãi. Các nghiên cứu đi từ bài toán đơn giản, mỗi ảnh chỉ có một khuôn mặt người nhìn thẳng vào thiết bị thu hình và đầu ở tư thế thẳng đứng trong ảnh trắng đen. Cho đến bài toán mở rộng với nhiều trường hợp phức tạp hơn như ảnh màu, nhiều khuôn mặt trong cùng một ảnh, nhiều tư thế thay đổi trong ảnh... Không những thế bài toán còn mở rộng cả phạm vi nghiên cứu: từ môi trường xung quanh khá đơn giản (trong phòng thí nghiệm) cho đến môi trường xung quanh rất phức tạp (như trong thực tế). Có nhiều hướng nghiên cứu cho bài toán nhận dạng mặt người nhưng nhìn chung có hai hướng chính như sau [5]: Nhận diện một khuôn mặt (face

recognition); Xác định vị trí những khuôn mặt trong một bức ảnh (face detection).

Khi quan sát một khuôn mặt, ta thường tập trung chú ý vào những bộ phận đặc biệt như: mắt, mũi, miệng, cằm,... và cũng chính những bộ phận này cung cấp nhiều thông tin nhận dạng xác thực nhất. Các bộ phận khác như vùng trán, tóc,... thường không cung cấp nhiều thông tin hữu ích cho quá trình nhận dạng. Chính vì vậy, phương pháp WMPCA được phát triển nhằm loại bỏ những vùng ảnh chứa ít thông tin nhận dạng và tập trung vào những vùng ảnh quan trọng hơn trên ảnh khuôn mặt. Một quy trình nhận dạng dựa theo phương pháp WMPCA phải trải qua các bước sau:

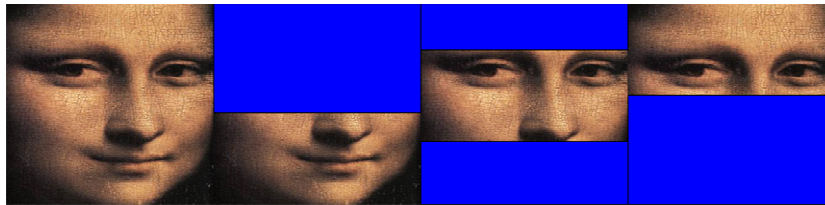
Thu thập dữ liệu

Để phù hợp với cấu trúc của cơ sở dữ liệu, mỗi ảnh khuôn mặt có kích thước $M' \times N'$

được biểu diễn như một vector M chiều ($M = M' * N'$) theo phương thẳng đứng, mỗi phần tử trong vector có giá trị bằng độ xám của một điểm ảnh tương ứng (ảnh gray), sau khi xếp các vector này liên tục nhau, ta được một ma trận có kích thước $M \times N$ với N bằng số ảnh. Ma trận này chính là biểu diễn số học của cơ sở dữ liệu. Dữ liệu có thể được lấy từ camera hoặc cắt dữ sẵn trong bộ nhớ.

Phân vùng ảnh

Ngay sau khi kết thúc quá trình thu thập dữ liệu, ma trận biểu diễn số học của cơ sở dữ liệu được chia theo phương ngang thành các ma trận con có kích thước $M' \times N$ với N bằng số ảnh, mỗi ma trận con này xem như một cơ sở dữ liệu riêng biệt và được tiến hành phân tích độc lập dựa theo phương pháp PCA.



Hình 1. Ảnh khuôn mặt trước và sau khi phân vùng.

Phân tích dữ liệu dựa theo phương pháp PCA [6][7]

Giả sử x_1, x_2, \dots, x_n là các vector $N \times 1$

Bước 1:

$$\bar{x} = \frac{1}{M} \sum_{i=1}^M x_i$$

Bước 2: trừ hai giá trị trung bình.

$$\Phi_i = x_i - \bar{x}$$

Bước 3: tạo thành ma trận $N \times M$

$$A = [\Phi_1 \ \Phi_2 \ \dots \ \Phi_M]$$

Sau đó tính

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = A A^T$$

Bước 4: tính toán các trị riêng của C

$$\lambda_1 > \lambda_2 > \dots > \lambda_N$$

Bước 5: tính toán các vector riêng của C

$$u_1, u_2, \dots, u_N$$

Vì C đối xứng ($N \times N$) nên u_1, u_2, \dots, u_N hợp thành một cơ sở (bất kể một vector x nào hay thậm chí (\hat{x}, x) cũng đều có thể viết dưới dạng một tổ hợp tuyến tính của các vector riêng)

$$x - \bar{x} = b_1 u_1 + b_2 u_2 + \dots + b_N u_N = \sum_{i=1}^N b_i u_i$$

Bước 6: bước giảm số chiều

Chỉ giữ lại những thuộc tính tương ứng với các trị riêng lớn nhất.

$$\hat{x} - \bar{x} = \sum_{i=1}^K b_i u_i \quad \text{trong đó } K \ll N$$

Do đó, sự diễn biến của $\hat{x} - \bar{x}$ trong u_1, u_2, \dots, u_k là :

$$\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_k \end{bmatrix}$$

Phép biến đổi tuyến tính $R^N \rightarrow R^K$ nhằm làm giảm số chiều sẽ là:

$$\begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_k \end{bmatrix} = \begin{bmatrix} u_1^T \\ u_2^T \\ \dots \\ u_k^T \end{bmatrix} (\underline{x} - \bar{\underline{x}}) = U^T (\underline{x} - \bar{\underline{x}})$$

PHƯƠNG PHÁP

Hệ thống nhận dạng mặt người dùng thuật toán WMPKA được mô tả theo sơ đồ khối sau:

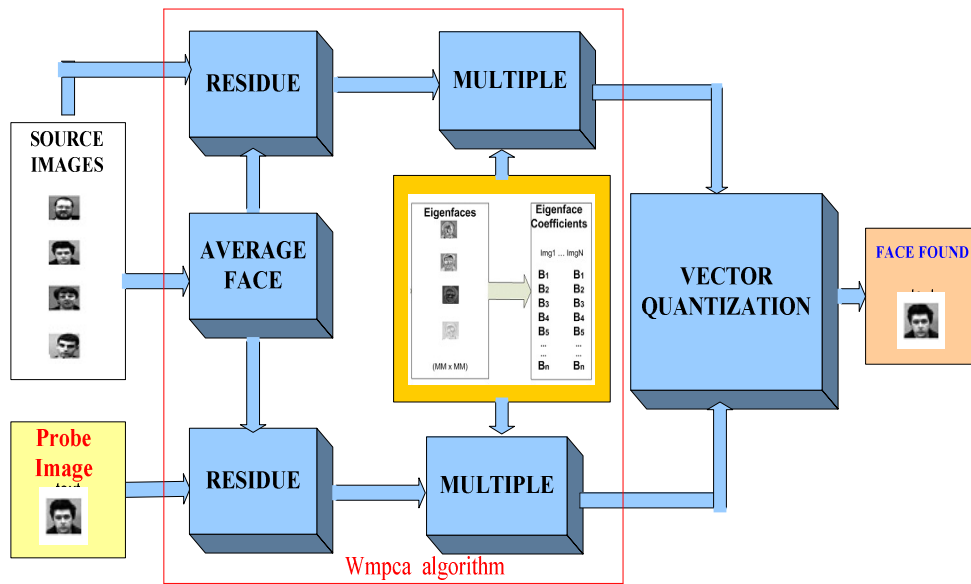
Nguyên tắc hoạt động

Bước 1: Từ tập ảnh đầu vào (lấy từ camera hoặc tập dữ liệu có sẵn) ta tìm ảnh trung bình cho toàn bộ tập dữ liệu.

Bước 2: Thực hiện việc chuẩn hóa tập ảnh đầu vào và ảnh cần nhận dạng bằng cách trừ với ảnh trung bình để tìm ra sự sai lệch.

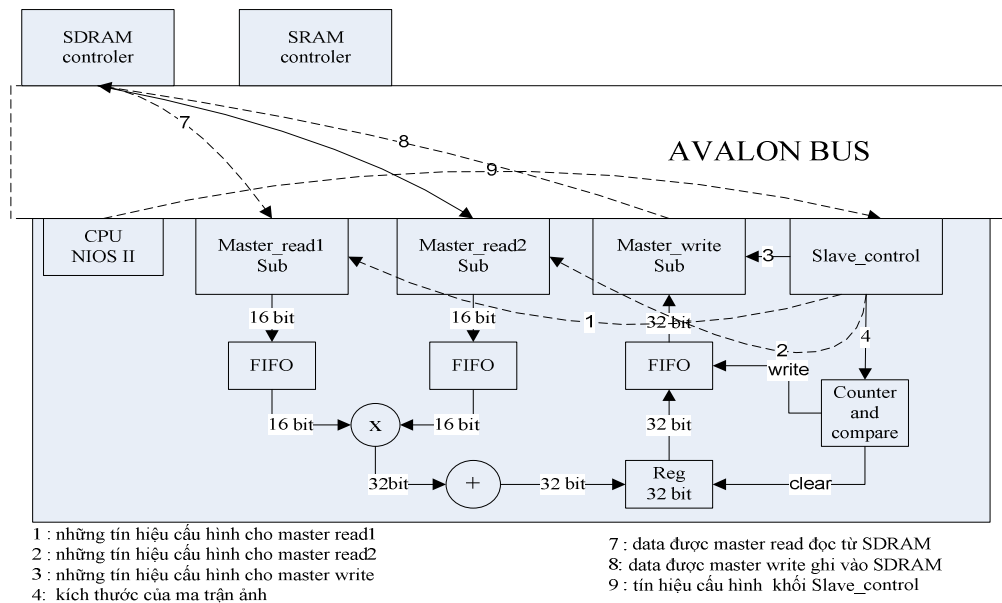
Bước 3: Thực hiện phép chiếu để tìm ra các giá trị tương ứng của từng mặt người. Bước này được thực hiện bằng cách nhân sự sai lệch với những hệ số eigenfaces tương ứng.

Bước 4: Để tìm ra đối tượng cần nhận dạng ta sử dụng khoảng cách Euclide kết hợp với thuật toán Winner Take All thông qua lưu đồ thuật giải Vector Quantization [1].



Hình 2. Sơ đồ tổng quát phần cứng WMPKA

Khối nhân ma trận



Hình 4. Sơ đồ khối tìm hình chiếu

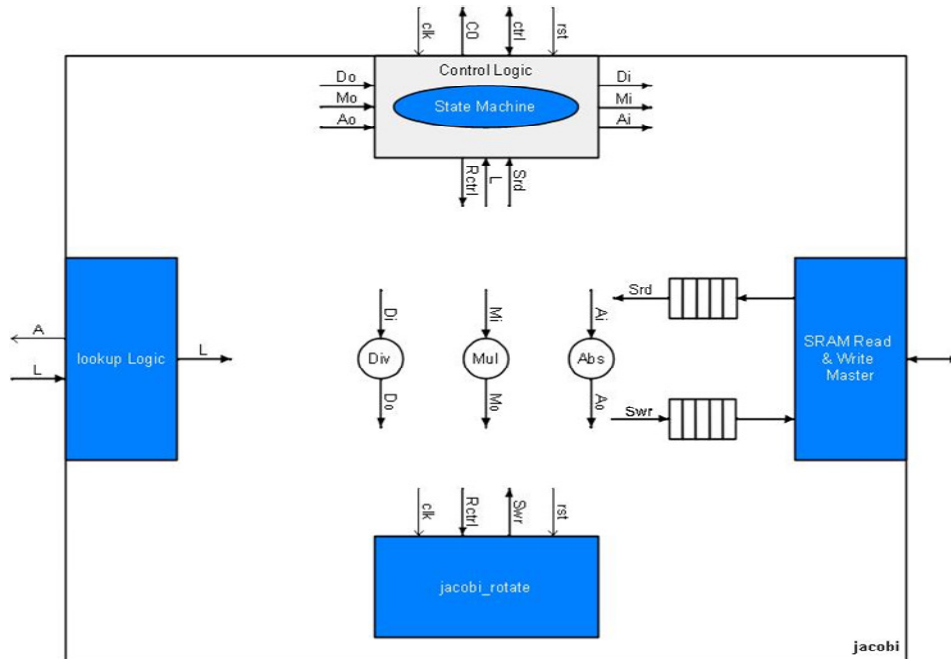
Khối thiết kế bao gồm hai master read được sử dụng để đọc dữ liệu từ memory off chip (SDRAM) sau đó những dữ liệu này được đưa qua hai bộ FIFO. Dữ liệu được lấy ra từ FIFO sau đó được đưa qua bộ nhân và tiếp tục đưa vào bộ cộng tích lũy trước khi đưa vào bộ FIFO mới. Master write sẽ lấy dữ liệu từ bộ FIFO này lưu vào memory off chip.

Khối jacobi_arbitrator

Hình 5 trình bày cấu trúc khối jacobi. Khối gồm 2 cổng Master, 1 khối Control Logic điều khiển hoạt động của toàn khối và phát tín hiệu C yêu cầu quyền truy cập SRAM cho SRAM Read & Write Master, 1 khối jacobi_rotate thực hiện phép quay jacobi, 2 khối FIFO làm bộ nhớ đệm cho quá trình đọc ghi dữ liệu; 1 mạch lấy

trị tuyệt đối (Abs), 1 mạch nhân (Mul) và 1 mạch chia (Div) thực hiện quá trình tính ma trận đặc trưng của ma trận hiệp phương sai trên SRAM.

Thuật toán jacobi bao gồm một chuỗi những phép biến đổi trực giao. Mỗi phép biến đổi là một phép quay để hủy một trong những phần tử ngoài đường chéo chính, hay đưa những phần tử ngoài đường chéo về giá trị zero. Kết quả cuối cùng chỉ còn lại những phần tử trên đường chéo chính. Phép biến đổi jacobi sẽ cho kết quả gồm những vector riêng là sự tích lũy của những phép biến đổi và những trị riêng là những phần tử trên đường chéo chính còn lại của ma trận đầu vào. Phương pháp jacobi áp dụng cho tất cả ma trận đối xứng thực.

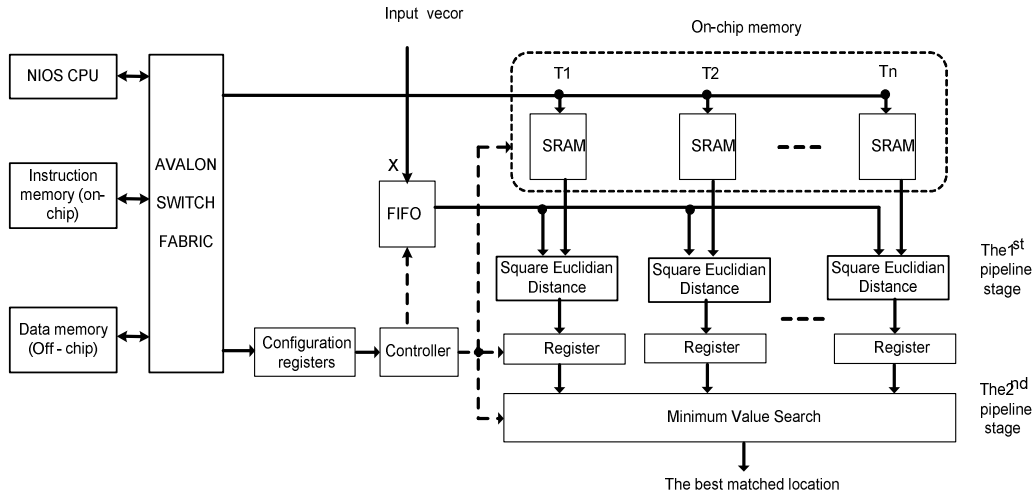


Hình 5. Mô hình khối Jacobi

Khối lượng tử hóa vector

Hình 7 trình bày nguyên tắc hoạt động của khối lượng tử hóa Vector. Khi khởi động hệ thống CPU nhúng sẽ nạp codeword (giá trị sau khi qua mạch nhân), đồng thời nạp các tham số cấu hình cho bộ điều khiển để xác định số phần tử của các codeword và số lượng codeword trong hệ thống. Khi phần tử x_j của vector X được gửi vào hệ thống thì tất cả các phần tử t_j của vector mẫu T trong bộ nhớ đồng loạt được

đưa vào các bộ trừ. Kết quả của phép trừ này được gửi tới mạch tính khoảng cách Euclide. Để tìm ra vector trùng khớp nhất ta sử dụng mạch Minimum Value Search. Nhằm tăng tốc độ thực hiện của hệ thống, tác giả sử dụng một số phương pháp như dùng các bảng tra cứu (LUT), bộ nhân được đơn giản hóa, kỹ thuật một lệnh thao tác trên nhiều dữ liệu (Single Instruction Multiple Data-SIMD), kỹ thuật đường ống (Pipeline) hai chặn [3][4].



Hình 6. Sơ đồ khối lượng tử hóa vector

KẾT QUẢ

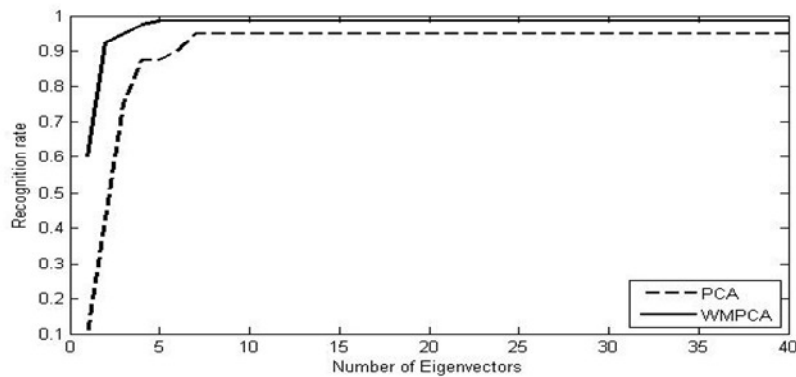
Kết quả kiểm chứng bằng phần mềm

Kết quả thực nghiệm cho thuật toán nhận dạng bằng phần mềm được ghi nhận lại trong bảng sau:

Bảng 1. Kết quả nhận dạng qua các tập dữ liệu khác nhau.

Tập dữ liệu	Số người khảo sát	Số trạng thái cho 1 người	Số thành phần chính giữ lại	Kết quả nhận dạng
AT&T (ORL)	40	400	5 đến 15	85% đến 5%
Yale	10	90	5	98,8%
Grimace, Face 96, hoặc Face 95	20	20	5	100%

Kết quả kiểm nghiệm với nhiều tập dữ liệu khác nhau:

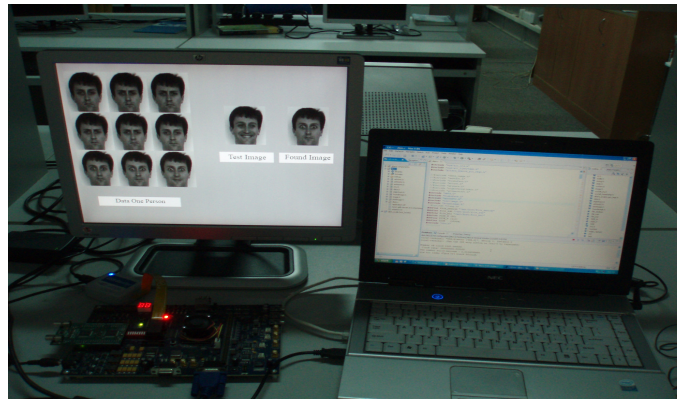


Hình 7. Kết quả so sánh giữa PCA và WMPCA

Kết quả trên phần cứng

Để kiểm tra hệ thống nhận dạng mặt người dùng thuật toán đề xuất như ở trên, trong

nguyên cứu này tác giả sử dụng board DSP Development Kit, stratix II professional Edition FPGA FP2S180.



Hình 8. Kết quả demo trên phần cứng

Thời gian thực hiện trên phần mềm

```

C:\ "D:\code\codeeigenface\Debug\codeeigenface.exe"

Nhập số ảnh cần test?8

*****
Thời gian tìm kiếm mặt người trung khớp nhất
Bắt đầu thời gian tính cho các khối PCA
Đơn vị tính: 7502977.054346 microgiay
*****
Press any key to continue_
    
```

Hình 9. Kết quả thời gian thực hiện trên phần mềm của các bộ gia tốc PCA

Thời gian thực hiện trên phần cứng

```

Problems Console Properties Debug
diem Nios II HW configuration [Nios II Hardware] Nios II Terminal Window (10/14/09 1:10 PM)
nios2-terminal: connected to hardware target using JTAG UART on cable
nios2-terminal: "USB-Blaster [USB-0]", device 1, instance 0
nios2-terminal: (Use the IDE stop button or Ctrl-C to terminate)

Number of clock 165026752.000000
Clock rate: 100000000.000000
The number mills-seconds : 1650.2674560547
gia tri index 78gia tri block 9finish
    
```

Hình 10. Kết quả thời gian thực hiện trên phần cứng của các bộ gia tốc PCA

Qua những kết quả thực nghiệm ghi nhận, ta thấy được hiệu quả khi thực hiện trên phần cứng luôn luôn nhanh hơn trên phần mềm. Nếu tính toàn bộ hệ thống thì phần mềm thực hiện mất hơn 7.5 mills giây trên một laptop Intel core 2, trong khi phần cứng thực hiện trong khoảng 1.6 mills giây trên kit FPGA hoạt động ở 100Mhz. Ngoài ra, toàn bộ tài nguyên sử dụng cho hệ thống với tập dữ liệu là 100 trường hợp cho 10 đối tượng mất hết gần 23% tài nguyên của chip FPGA. Do đó, ta hoàn toàn có thể mở rộng hệ thống này cho tập dữ liệu lớn hơn (khoảng 400 mặt người). Với khả năng như vậy, ta thấy hệ thống hoàn toàn có tính áp

dụng thực tế cho công nghiệp cũng như trong nghiên cứu.

KẾT LUẬN

Hệ thống phần cứng được thiết kế với những bộ gia tốc như là những components cho CPU Nios hoạt động ở tần số cao. Phần cứng lượng tử hóa vector được thiết kế linh hoạt cho nhiều ứng dụng khác nhau. Với cấu trúc sử dụng bảng tra, SIMD (Single Instruction Multiple Data) và pipeline hai tầng, đặc tính hệ thống phù hợp cho những ứng dụng thời gian thực. Hệ thống được thiết kế tối ưu, cho kết quả so sánh giữa phần cứng và phần mềm gần như trùng khớp nhau.

DESIGNING A SOPC FOR FACE RECOGNITION USING WMPCA ALGORITHM

Truong Thanh Nhu, Tran Thi Diem

University of Science, VNU-HCM

ABSTRACT: *A flexible accelerator hardware for full-search vector quantization (VQ) has been developed as a component for a system on a programmable chip (SoPC) to use in real-time image compression and recognition applications. Nowadays, FPGA and its SoPC (System on Programmable Chip) tools are powerful enough to efficiently develop a flexible hardware accelerator for VQ application. In addition, one of statistical analysis methods, weighted modular principal component analysis, has showed efficiencies in recognition applications. In this paper, a parallel architecture for online face recognition using weighted modular principal component analysis (WMPCA) and its system-on-programmable-chip (SoPC) implementation are discussed.*

Key words:

TÀI LIỆU THAM KHẢO

- [1]. Thuan Huynh, Thuong Cao, Diem Tran, Phuong Nguyen, Anh Dinh, *Designing a Hardware Accelerator for Vector*

Quantization as a Component of a SoPC, CCECE/CCGEI, 479-484, May (2008).

- [2]. Diem Tran, Thi To, Phu Bui, Phuong Nguyen, *Designing a Hardware*

- Accelerator for Vector Quantization and Principal Component Analysis as a Component of SoPC*, Proceedings of the 5th Vietnamese Japanese Scientific Exchange Conference, pp.159-162, October (2009).
- [3]. A. Nakada, et al., A fully Parallel Vector Quantization Processor for Real Time Motion Picture Compression, *IEEE J. Solid State Circuit*, 34, 822-829, (1999).
- [4]. M. Ishikawa, K. Ogawa, T. Komoro, i. Ishii, *A CMOS Vision Chip with SIMD Processing Element Array for Ims Image Processing*, IEEE International Solid-State Circuits Conference, 206-207, (1999).
- [5]. K. Delac, M. Grgic, S. Grgic, Independent Comparative Study of PCA, ICA, and LDA on the FERET Data Set, *International Journal of Imaging Systems and Technology*, 15, 252 – 260, (2006).
- [6]. B. M. Pentland, T. Starner, *View-Based and Modular Eigenspaces for Face Recognition*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, Washington, USA, 84-91, (1994).
- [7]. M. A. Turk, A. P. Pentland, *Face Recognition Using Eigenfaces*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, USA, 586-591, 3-6 June (1991).
- [8]. W. Zhao, R. Chellappa, A. Krishnaswamy, *Discriminant Analysis of Principal Components for Face Recognition*, Proc. of the 3rd IEEE International Conference on Face and Gesture Recognition, FG'98, Nara, Japan, 336-341, (1998).
- [9]. I. T. Jolliffe, *Principal Component Analysis*, 2nd edition, Springer, chapter 6, 111-147, October 1, (2002).
- [12]. Lindsay Ismith, *A Tutorial on Principal Components Analysis* February 26, (2002).
- [13]. N. Kambhatla, T. K. Leen, *Dimension Reduction by Local PCA*, Chapter 7, 1493-1516, (1997).
- [14]. Altera Corporation, *SoPC Builder* User Guide, ver 1.0, Dec (2010).
- [15]. URT:<http://www.altera.com/literature/lit-sop.jsp>