Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams University

Face Recognition on Heterogeneous Architecture using Parallel Computing Paradigms

Thesis submitted as a partial fulfillment of the requirements for the degree of
Master of Science in Computer and Information Sciences

By

Dalia Shouman El-Shahat Ibrahim

Teaching Assistant in Computer Systems Department,
Faculty of Computer and Information Sciences,
Ain Shams University

Under Supervision of

Prof. Dr. Hossam Faheem
Professor in Computer Systems Department,
Faculty of Computer and Information Sciences,
Ain Shams University

Dr. Salma Hamdy
Assistant Professor in Computer Science Department,
Faculty of Computer and Information Sciences,
Ain Shams University

Cairo – 2017

Acknowledgment

I would like to thank my supervisor Dr. Salma Hamdy for the patience, guidance, and feedback given during my research work. Working under her supervision, I learned much in both academic and personal life.

My deepest gratitude to Prof. Dr. Hossam Faheem for facilitating my master research by providing a well-established environment to test my proposed approaches, and supporting me with great ideas to improve my work.

Thanks for Faculty of Computer and Information Science (FCIS) for providing us with the high-performance computing laboratory, which we used to conduct the experiments of our research.

I also thank my parents, sisters and my husband for motivating me to complete my work, make it easy to focus on my research, and for their insistence to see me advance my career. Thank you, again, for being in my life.

Abstract

Face recognition applications are widely used in different areas, specifically, in security and biometrics. The decision often should be highly accurate and fast. Principle Component Analysis is a feature extraction algorithm used in facial recognition applications by projecting images on a new face-space. It is mainly applied to reduce the dimensionality of the image. However, PCA consumes a lot of processing time due to its high intensive computation nature.

To overcome the single computing systems limitations, different parallel processing paradigms are used to accelerate the process. In this thesis, two face recognition scenarios are implemented using different parallel programming memory architectures.

First, we show how a cluster of supercomputers can be used to accelerate a face recognition system. The work focuses on speeding either the training or testing phase of PCA. In addition, the suggested environment is dynamic to different numbers of supercomputers. Experimental results show that the proposed architecture improves execution time up to 25X in training of the first scenario and 5X in recognition phase for both scenarios, reaching super-linear and linear speed-up, respectively. It also achieves system scalability on different data sizes from the Facial Recognition Technology (FERET) database.

Furthermore, a hybrid architecture is suggested to optimize face recognition by exploiting the benefits of multi-core and distributed systems combined. Hybrid MPI/OpenMP libraries are used to perform two-levels decomposition; one on the distributed systems, and the other using cores

inside each supercomputer. The proposed approach significantly reduces the algorithm complexity when implemented over a cluster with parallel computing architecture. The first scenario achieves 2975X and 102X faster than the sequential implementation in the training and recognition phases, respectively. However, the second scenario achieves 74X faster than the sequential implementation in the recognition phase.

In addition, a heterogeneous computing system architecture is proposed for enhancing the projection time of PCA algorithm. The speed-up reaches up to 290X compared to the sequential implementation of the projection step. This affects the total training time up to1.6X.

List of Publications

[1] D. Ibrahim and S. Hamdy, “Parallel Architecture for Face Recognition using MPI,” in International Journal of Advanced Computer Science and Applications (IJACSA), vol. 8, no. 1, 2017, pp. 425–430.

[2] D. Ibrahim and S. Hamdy, “Hybrid MPI/OpenMP Implementation of PCA,” in The 8th IEEE International Conference on Intelligent Computing and Information Systems (ICICIS), 2017, pp. 205-211.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

APIs	Application Program Interfaces
2DPCA	Two-Dimensional PCA
CUDA	Compute Device Unified Architecture
DBMS	Database Management Systems
EBGM	Elastic Bunch Graph Matching
EM	Expectation Maximization Algorithm
FERET Database	Face Recognition Technology Database
FPGA	Field Programmable Gate Arrays
GPGPUs	General Purpose Graphics Processing Units
GPUs	Graphics Processing Units
HMM	Hidden Markov Model
HPC	High Performance Computing
ICA	Independent Component Analysis
ILP	Instruction-Level Parallelism
KFLD	Kernel Fisher Linear Discriminant Analysis
KLT	Karhunen-Loeve Transform
KPCA	Kernel Principle Component Analysis
LAPACK	Linear Algebra Package Library
LBP	Local Binary Patterns
LDA	Linear Discriminant Analysis
LFA	Local Feature Analysis
MIMD	Multiple Instruction Multiple Data Stream
MIMPCA	Modular Image PCA

MISD	Multiple Instruction Single Data Stream
MKL	Math Kernel Library
MMP	Multimodal Face Recognition
MPCA	Modular Principal Component Analysis
MPI	Message Passing Interface
MSE	Mean Square Error
OpenCL	Open Computing Language
OpenMP	Open Multi-Processing
PC	Principle Component
PCA	Principle Component Analysis
PVM	Parallel Virtual Machine
SIMD	Single Instruction Multiple Data Stream
SIMT	Single Instruction Multiple Threads
SISD	Single Instruction Single Data Stream
SMP	Symmetric Multi-Processing
SVD	Singular Value Decomposition
wMIMPCA	Weighted Modular Image PCA

Chapter 1
Introduction

Over the last decades, many researchers tried to enhance the execution time of applications either by optimizing the algorithm to decrease the clock cycles, or by distributing their code and data over multiple processing units. Usually, the software is written for sequential computations, so the instructions are executed one at a time on a single processor [1]. Developers aim to run computational applications faster than the traditional sequential implementation. This can be done by producing more complex chips to enable faster execution. Alternatively, parallelizing the solution may require cheaper and less complex chips [2]. Because of the industrial revolution, beginning with transistors and multi-core processors, all the way to graphics processing units (GPUs), mainstream computers, desktops and supercomputing clusters; the direction shifts to parallel and heterogeneous systems to achieve less execution time for real-time application [3].

Parallel programming is used for maximum utilization of most High Performance Computing (HPC) systems composed of some clusters of shared memory nodes. Therefore, it combines the distributed memory parallelization over the interconnected nodes with the shared memory parallelization inside each node. This combination is called hybrid programming model [4].

Nowadays, cell phones and laptops are equipped with Graphics Processing Units (GPUs) for computer games, rendering, transformation, and projections. Researchers are using General Purpose Graphics Processing Units (GPGPUs) to increase the processing power and make development easier [5]. Modern GPGPUs are provided with hundreds of streaming cores and able to manage thousands of threads [6].

The leverage of parallel programming becomes evident in the computing world with the growing need of heavy processing in real time. This involves applications that demand fast system response with large amounts of data; one of which is face recognition.

Over the last decade, face recognition has become one of the most important issues in computer vision and machine learning. It is considered an easy task for humans as they can do it routinely every day. The challenge is making automated systems recognize faces at runtime with accurate results. This requires heavy processing and hence, will much benefit from parallelization. This can be used instead of traditional means like ID cards and passwords in applications like airport security [7], surveillance systems [8], automated student attendance [9], album organization [10], computer entertainment, virtual reality [11], online banking [12] and video indexing [13].

In addition, face recognition is a key component for smart environments. As smart homes wearable computers are being populated everywhere; cars, homes, clothes and malls, it can interpret human emotions, expressions, intentions, and behavior [14]. Life will be more comfortable for people with disabilities. Many interactions will not require any human intervention in the near future.

Besides, surveillance systems will be able to identify a particular person among large crowds with no physical interaction. The matching results will not require an expert to be interpreted, as the target person will be compared against images from a database. Such automated process could be used for finding known criminals and terrorists.