GPU Accelerated Face Recognition With Eigenfaces

Baruffaldi Juan Manuel
baruffaldi.jm@gmail.com

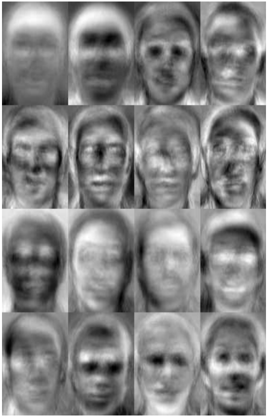
Pellejero Nicolás
pellejero.nicolas@gmail.com

Facultad de Ciencias Exactas e Ingeniería
Universidad Nacional de Rosario

18 de Septiembre 2013

Overview

1. Introduction
2. PCA
   - Average Image
   - Covariance Matrix
   - Calculating Eigen Pairs
   - Projection process
3. Experimental Setup
4. Results
5. Conclusions and Future Work

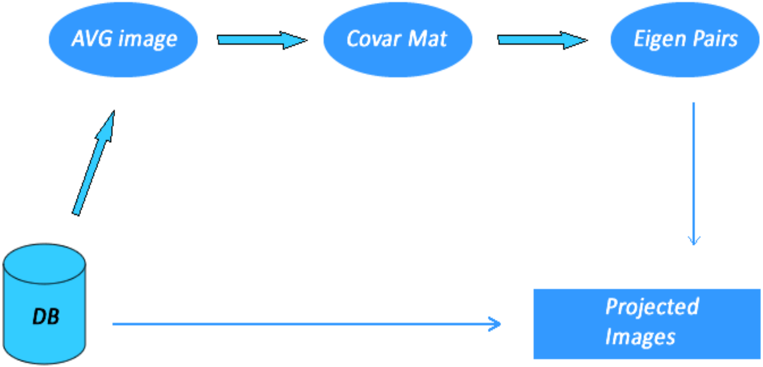


- We present a GPU implementation of the Eigenfaces approach to the problem of Face Recognition.
- This method uses PCA as a process to extract the most relevant information about Faces Images.
- The goal is to optimize the algorithm to run on the SIMD architecture of the GPU.
- Several Tests were made, varying the amount of training pictures, to analyze the scalability of the implementation.

Eigenface Approach

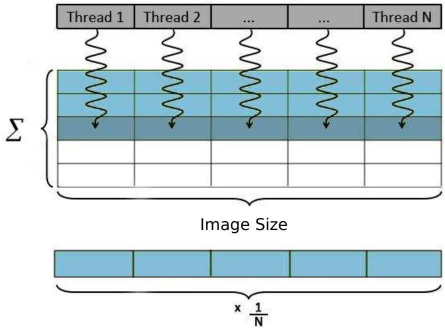
- Training Process
  - Find Eigenspace.
  - Project training samples.
- Recognition Process
  - Project test image into the Eigenspace.
  - Find the distance between this projected image and all the others.
  - Compute the minimum of the distances.

In this work, we focused on the most computationally expensive step of the approach: The Training Process.


AVG image
Covar Mat
Eigen Pairs
DB
Projected Images

Average Image

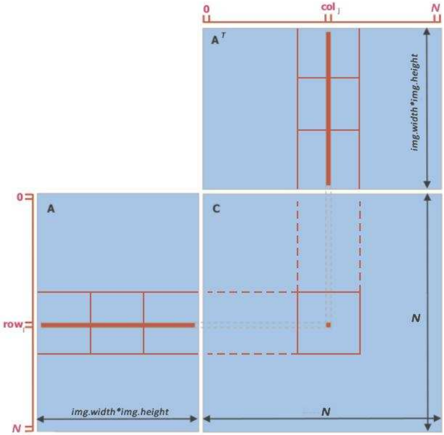
- Thread n-th computes the average of the n-th column.



- Under this scheme, the calculation scales on the size of the images and not on the quantity.

Covariance Matrix

- The Covar Matrix is computed by $C = AA^T$ where each row of A is $\theta_i = IMG_i - AVG$
- NxN threads are launched where thread (i,j) computes the (i,j) th element of the covariance matrix.
- The Matrix-Matrix operation is done by blocks, and each block is stored in shared memory.
- To load data from global,to shared memory, the accesses are made in a transposed way, avoiding bank conflicts.



Still working ...

- Actually we are using the Rotation Jacobi method to calculate Eigenvalues and Eigenvectors of the covar matrix.
- The implementation is totally secuential.

Input

- Orthonormal Matrix of Eigenvectors, the basis of the Eigenspace.
- Matrix of Training Images.

Output

- A Matrix containing the projected images into the Eigenspace.

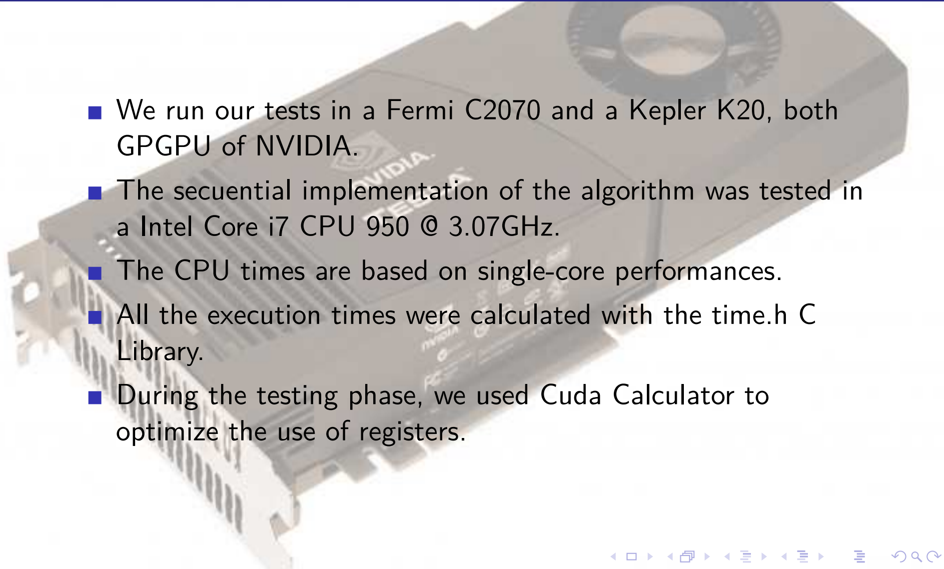
The same optimizations of the covar-matrix step were made:

- Matrix operations per block.
- Block stored in shared memory.
- Memory acceses made in a traspose way.

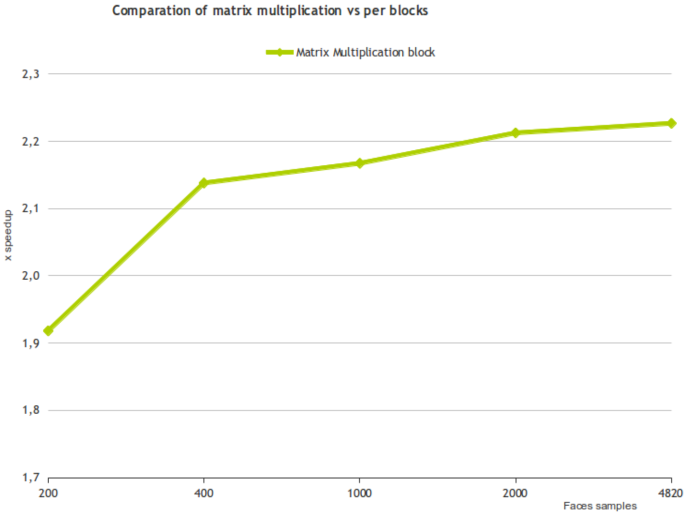
Projection process

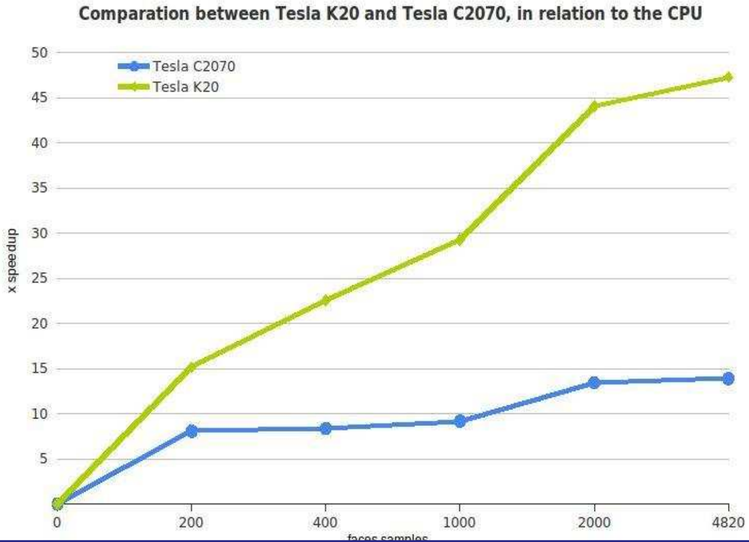
Number of Images	Subspace Size	% Reduced
200	139	30,5
400	286	28,5
1000	698	30,2
2000	1357	32,15
4820	2997	37,82

- The table shows the dimensionality reduction achieved with PCA.



- We run our tests in a Fermi C2070 and a Kepler K20, both GPGPU of NVIDIA.
- The secuential implementation of the algorithm was tested in a Intel Core i7 CPU 950 @ 3.07GHz.
- The CPU times are based on single-core performances.
- All the execution times were calculated with the time.h C Library.
- During the testing phase, we used Cuda Calculator to optimize the use of registers.

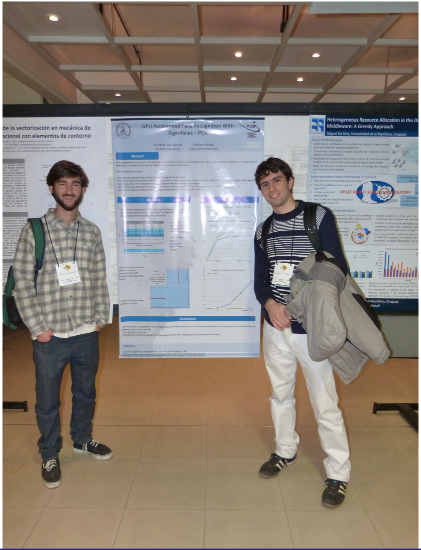

Comparation of matrix multiplication vs per blocks
Matrix Multiplication block
x speedup
2,3
2,2
2,1
2,0
1,9
1,8
1,7
200
400
1000
2000
4820
Faces samples


Comparation between Tesla K20 and Tesla C2070, in relation to the CPU
Tesla C2070
Tesla K20
x speedup
50
45
40
35
30
25
20
15
10
5
0
200
400
1000
2000
4820
faces samples

Analyzing the nature of the problem, the operations required to solve it, and the results, we found that...

- The approach is very suitable to run on a many-core SIMD architecture.
    - There are few dependencies between data.
    - In general, there are no need to use Atomic Operations.
    - Good scalability.

- Finishing the implementation of the Eigen Pairs Calculation Process is the main future work.
- The plan is to do it through Householder Reflections, doing the Matrix-Matrix Operations per blocks.


de la vectorización en mecánica de
acional con elementos de contorno
GPU Accelerated Face Recognition With Eigenfaces – PCA
Heterogeneous Resource Allocation in the Ou
Middleware: A Greedy Approach
WHAT ABOUT SCHEDULING POLICIES?

Thank you very much for your atention!

... Questions?

baruffaldi.jm@gmail.com
pellejero.nicolas@gmail.com