Fingerprint Matching on GPU for a Large Scale Identification System

Hong Hai Le, K18 Information Systems

Introduction

Objective

Although state-of-the-art algorithms are very accurate, but the need for processing speed for databases with millions fingerprints are very demanding (Minutia Cylinder-Code, a state-of-the-art matching algorithm, takes 3 milliseconds to perform a matching on CPU. So it takes 3 seconds to identify a fingerprint in a database of 1000 fingerprints).

GPU Processing

Graphic Processing Unit (GPU) has been proven to be a very useful tool to accelerate the processing speed of computationally intensive algorithms. GPU consists of a set of Streaming Multiprocessors (SM). Figure 1 shows the computing architecture of GPU. Each SM schedules and executes threads in groups of 32 parallel threads. If threads take different paths (due to flow control instructions), they have to wait for each other.

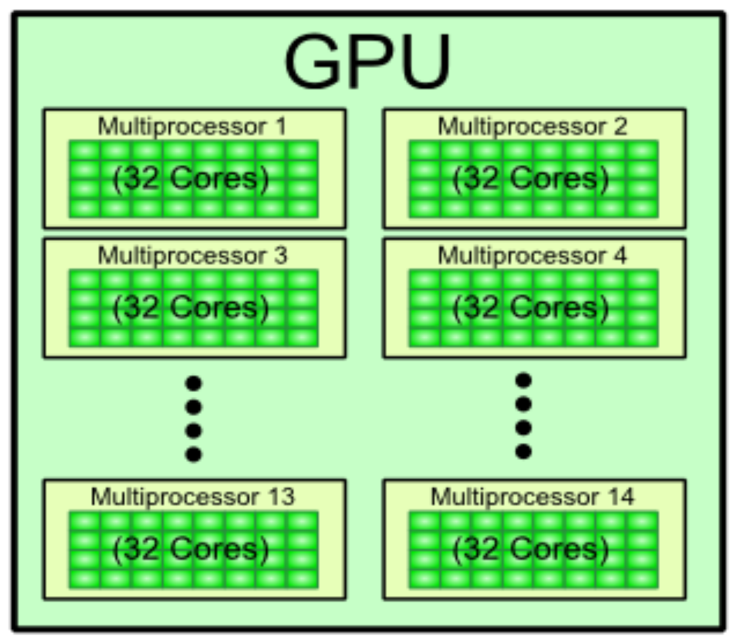

Figure 1: GPU computing architecture

Most approaches make use of GPU for the filtering process. After that, more accurate matching algorithms on CPU for remaining fingerprint candidates are used.

Methods

Idea

Our approach is based on the fact that using 32 minutiae for each fingerprint is enough for the matching process. From statistics of FVC 2002 fingerprint databases, the average number of minutiae of each fingerprint is 30. The average number of matches for a genuine matching is 6.

Implementation Detail

We use one block for matching a fingerprint pair, each block has 32 threads. Each thread of the block is used to calculate a column in the similarity matrix of MCC algorithm and to find the maximum value in that column. Figure 2 presents steps of our fingerprint process on GPU. The proposed fingerprint matching process fits well with the architecture of GPU.

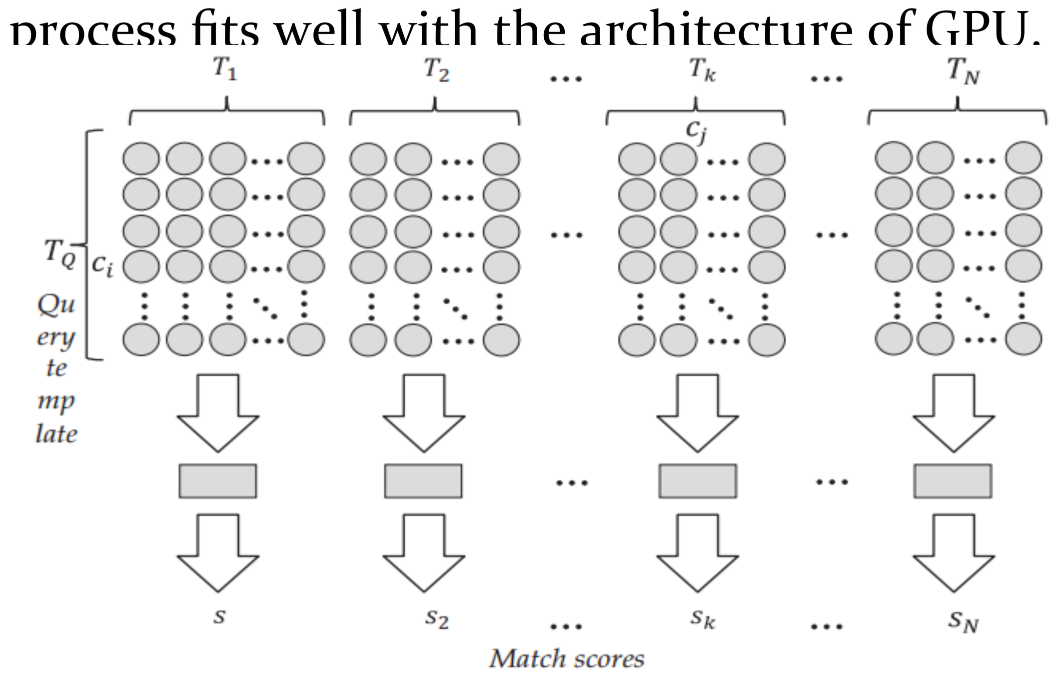

Figure 2: Processing steps on GPU

Results

Experiment

For evaluating the speed of the proposed algorithm, we carried out all the experiments on an NVIDIA GeForce GTX 680 with 1536 CUDA cores, Kepler Architecture and 2GB of memory.

FVC 2002 DB was scaled to different database sizes (ranging from 10000 to 200 000) to study how the GPU based algorithm scaled with the database size.

Speed result

Some experimental results are presented in Table 1.

DB size	Time (ms)	Throughput
50000	284	1760
100,000	567	1763
200,000	1105	1809

Table 1: Execution time of the first ten queries with different database sizes

At larger DB sizes, throughput of the proposed algorithm is stable at 1.8 millions matches per second

It is higher than the result reported for previously published GPU algorithm, which gains 55.7 KMPS on a single GPU device, which is the same as our device

Accuracy result

We achieved an EER of 1.34% against a 1.26% of MCC-baseline

These are certainly minor differences and can be accepted in real world applications

Conclusions

We propose a simple approach of adapting MCC to GPU. Using all minutiae for calculating cylinders, then choosing 32 minutiae for matching, the approach actually fits well with the GPU computing architecture and can be easily implemented.

The proposed algorithm is stable at 1.8 millions matches per second and does not affect to the accuracy of the original algorithm. The proposed approach can be easily scaled-up. Thus, it is possible to implement a large-scale fingerprint identification system on inexpensive hardware.

Bibliography

[1] H-H. Le, N-H. Nguyen, T-T. Nguyen. Exploiting GPU for Large Scale Fingerprint Identification, 8th Asian Conference on Intelligent Information and Database Systems International, 2016.

[2] H-H. Le, N-H. Nguyen, T-T. Nguyen. A complete fingerprint matching algorithm on GPU for a large scale identification system, 7th International Conference on Information Science and Application, 2016.