Faster Fingerprint Matching Algorithm Using GPU

Sidra Riaz, Sang-Woong Lee
Computer Vision and Multimedia Laboratory, Chosun University, Gwangju, 501-759, South Korea

Abstract
This paper is based on embedding the biometrics techniques on GPU for better computational efficiency and fast matching process using the parallel nature of the GPU processors to compare thousands of images for fingerprint recognition in a fraction of a second. In this paper we worked on GPU (INVIDIA GeForce GTX 260 with compute capability 1.3 and dual core-2-dou processor) for fingerprint matching and found that the efficiency is better than the results with related work already done on CMOS, CPU, ARM9, MATLAB Neural Networks etc which shows the better performance of our system in terms of computational time. The features matching process proposed for fingerprint recognition and the verification procedure is done on 5,000 images which are available online in the databases FVC2000, 2002, 2004 [1].

1. Introduction
Biometrics consists of methods for uniquely recognizing humans based upon one or more intrinsic physical or behavioral traits. In computer science, in particular, biometrics is used as a form of identity access management and access control. It is also used to identify individuals in groups that are under surveillance. Biometric characteristics can be divided in two main classes:

Physiological: Physiological are related to the shape of the body. Examples include, but are not limited to fingerprint, face recognition, DNA, Palm print, hand geometry, iris recognition, which has largely replaced retina, and scent.

Behavioral: Behavioral are related to the behavior of a person. Examples include, but are not limited to typing rhythm, gait, and voice.

Fingerprint analysis is one of the most commonly-used technologies in biometrics. Extensive research has been done on fingerprints in humans. Two of the fundamentally important conclusions that have risen from research are:

1. A person’s fingerprint will not naturally change structure after about one year after birth.

2. The fingerprints of individuals are unique. Even the fingerprints in twins are not the same. In practice two humans with the same fingerprint have never been found.

A fingerprint is comprised of ridges and valleys. The ridges are the dark area of the fingerprint and the valleys are the white area that exists between the ridges. Many classifications are given to patterns that can arise in the ridges, for example crossover, core, bifurcation, ridge ending, island, delta, pore etc. These points are also known as the minutiae of the fingerprint. The most commonly used minutiae in current fingerprint recognition technologies are ridge endings and bifurcations because they can be easily detected by only looking at points that surround them.

Graphics Processing Unit’s (GPU’s) powerful compute engine is leveraged to enhance the stand-alone capability of a central processing unit (CPU). Fingerprint (FP) algorithm is analyzed and implemented to execute on a system equipped with a graphics card. Modern GPUs are very efficient and their highly parallel structure makes them more effective than general-purpose CPUs for algorithms where processing of large blocks of data is done in parallel. The GPU used in this system is GeForce GTX 260 with compute compatibility of 1.3.

2. Related Works
Image processing techniques has been implemented on GPU to achieve high performance and parallel
processing. As presented by Zang and Chen in [2] the sobel edge detector and homorphic filtering algorithms are embedded into GPU with input images of different resolutions to compare the computational efficiency of GPU and CPU. The approach used in [3] is the automated fingerprint recognition based on minutia, which are extracted directly from the finger prints by artificial neural networks (ANN) based system. Fingerprint recognition system in maritime condition [4] is the study of prolonged exposure of fingers to water on the performance of existing fingerprint recognition systems. A study on fingerprint matching system which is implemented on ARM7 and ARM9 processors with 60MHz to 200MHz frequency by using the minutiae points [5]. Same work on low quality fingerprint matching is done by Zin.et.al, on CPU by using Gabor filtering [6], and in [7] fingerprint recognition system is proposed by using minutiae and pores. A CPU implementation of GLCM-based fingerprint recognition algorithm is proposed by [8]. Another work on CMOS is also present which uses adaptable pixel networks and column parallel processors for image enhancement and recognition [9]. A neural network approach is presented for recognition of fingerprints in online examination system [10]. Furthermore the fingerprint matching algorithm to calculate the computational time which is compared in table 1 is given by [11], [12].

3. Implementation Design
An urgent need to develop accurate biometric recognition system is expressed by governmental agencies at the local, state, and federal levels, as well as by private commercial companies. Fingerprinting is the most practical and widely used biometric technique. The pattern of ridges and valleys of each fingerprint is unique. The minutiae based algorithm is widely used for fingerprint authentication. One of the major parts of this algorithm is the noise removal of fingerprints which allows minimizing significantly the noise from those fingerprints which are not properly taken by scanner. If the core of a fingerprint is not visible, then identification cannot proceed. In various application sector of the fingerprint matching there is chance that image of fingerprint will not be properly scanned due to dust, moisture and some other factors. After Pre-processing (Noise removal) the scanned image is subjected to binarization. After that we take the Fast Fourier Transform of that image. Then after FFT edge detection process is carried out. At the end minutiae based features i.e. minutia position, direction, and ridge information are extracted. Yet in some cases, partial fingerprints need to be identified, for that we will recommend novel Contactless Line Scanner for recognition of fingerprint pattern that converts a three dimensional object like a finger into a two dimensional image with minimal distortion. This novel imaging technique based on line by line scanned image required the development of a new recognition algorithm. In fingerprint matching commonly used algorithms are two. The first algorithm, called the Spaced Frequency Transformation Algorithm (SFTA), is based on taking the Fast Fourier Transform of the images. The second algorithm, called the Line Scan Algorithm (LSA), was developed to compare partial fingerprints and reduce the time taken to compare full fingerprints. Implementation of fingerprint matching is described in figure 1.

3.1 Matlab Section
Many of the algorithms require a linear scan of the fingerprint image. Scanning is achieved by moving a fixed size window across the picture in a grid-like pattern. However, it is possible that areas of interest do not lie squarely in one of the windows. To account for this the window is then shifted just vertically, just horizontally, and then vertically and horizontally by half the window size and the grid scan is completed again. Therefore, it takes four scans of the image to do the linear scan. This is not a problem because it is used for the preprocessing of a fingerprint image which occurs once and is done in a linear manner. By applying the noise removal functions, binarization, fast-fourier-transform and edge-detection we then extract the features which are used in the process of matching in CPU and GPU. All the features which are extracted are written in the.dat files in column wise manner. For example one feature is saved in one column for one fingerprint. Similarly we have the 2000
.dat files of extracted features in the features database. Matching process on CPU(matlab) and GPU(cudal) requires the access to this features database for matching of fingerprints.

3.2 CUDA C++ Section
Feature matching for fingerprint recognition is performed in cuda c++ using our proposed method and then efficiency of our system is checked by taking the computational time on GPU with and found it better than other related work present on CPU, ARM9, CMOS, Matlab-NN etc. GPU result showed the efficient matching due to the parallel processing. Cuda code for same GPU but with different cores show different matching times. For example, efficiency of ‘core i5’ was much higher than that with the ‘dual core-2-dou’.

4. Experimental Results and Conclusion
Elapsed time on CPU for 2000 files in the database shows a significant amount of time, which would increases as the number of files would increase in the database. We have seen the less computational time on GPU for 2000 images in the database. These result are shown in figure 2.

(Fig. 2) CPU & GPU result for 10~2000 files in the database

The comparison table 1 shows that the computation time on GPU is less than CPU for large number of database of fingerprint images. The obtained results are compared with the work of [11] and [12] for specific number of files in the database.

Table 1. Comparison table of computational time

<table>
<thead>
<tr>
<th>No.</th>
<th>Image in database</th>
<th>Computational Time</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>960 images</td>
<td>770 msec</td>
<td>CPU [11]</td>
</tr>
<tr>
<td>2.</td>
<td>800 images</td>
<td>440 msec</td>
<td>AMD Anthon [12]</td>
</tr>
<tr>
<td>3.</td>
<td>2000 images</td>
<td>325 msec</td>
<td>GPU (proposed)</td>
</tr>
</tbody>
</table>

Reference