International Conference on "Emerging Trends in Computer Engineering, Science and Information Technology"- 2015
Special Issue of International Journal of Electronics, Communication & Soft Computing Science and Engineering, ISSN: 2277-9477

Performance of Fingerprint Matching Operations on GPGPU

Ms. Shital T. Tupe Prof. Amol D. Potgantwar

***Abstract* — Biometric features like Fingerprint matching typically used for recognition and identification. These can be characterized through some testing essentials which are called as minutiae. The identification of a given fingerprint requires the identical of its minutiae against the minutiae of other fingerprints. Biometric is an authentication and verification technique that relies on measurable corporal characteristics. An experienced GPU based hybrid model i.e. fingerprint method using the Hough transform-based algorithm which improve accuracy for matching fingerprint is used. Parallelization of fingerprint matching algorithms on GPU architecture is appropriate for single program multiple data parallelism is developed. Proposed technique reduce time consuming memory transfers.**

***Key Words* — GPU, Fingerprints, Biometric, Parallel Computing, CUDA.**

I. Introduction

Fingerprint recognition is still the most widely accepted mean for a person identification and authentication [35]. Two recent trends can be observed in automated fingerprint identification systems (AFIS): a shift towards mobile security systems and a querying the very large fingerprint databases (of order 10^6-10^7 and higher). Consequently two separate challenges arose in the field of fingerprinting i.e. how to identify a person with limited computational resources and how to speed up the database processing? The scalability problem is a long known issue in biometrics. With the number of records increasing so does the response time of the biometric system, and for a subset of AFIS, the real-time AFIS, the time is one of the quality characteristics.

In a generic AFIS there are following processing steps:
1. Image acquisition.
2. Image transmission to the processing unit.
3. Image enhancement (IE), can be omitted.
4. Feature extraction and template building.
5. Matching the template against database or part of it.
6. Decision making whether to grant or deny the access given the matching results.

In order to improve the overall system performance one should improve the performance of every step, which is usually obtained through the introduction of parallelism. A note should be taken that steps 1 and 2 usually don't require speed up due to the hardware implementation of those steps, and their addition to a total processing time tends to be relatively small. As most systems implement step 6 in a form of a rule-based module with a single rule, the parallel implementation in this case might also be redundant.

With the hyper threading and multi-core processors being available for some time now the simple way to improve the system performance was implemented in commercially available systems. This approach is also scaled to the computing clusters, yet the price of such solutions is high. There exists a cheaper alternative, as General Purpose Graphical Processing Unit (GPGPU) computations is now a proven way to enhance a system performance. Among the many GPGPU technologies (e.g. OpenCL, Direct Compute) NVIDIA CUDA is the most advanced and readily available.

In this paper we demonstrate our results of implementing various steps of AFIS processing using NVIDIA CUDA technology. The paper is organized as follows. In section II presents literature review of fingerprint matching algorithms. Section III describes a small overview of CUDA technology and system model. Performance Evaluation is shown in sections IV with Conclusion.

II. Literature Review

Gowrishankar in [3] describes a scheme of a feature extraction algorithm for a binary skeletonized image based on the neighbor colored pixels count and proposes the usage of massively parallel architecture of a computing device due to the information being extracted from a small neighborhood around each pixel.

In [13] 2006, Xinjian Chen, Jie Tian, Senior Member and Xin Yang, A New Algorithm for Distorted Fingerprints Matching Based on Normalized Fuzzy Similarity Measure, proposed A novel algorithm, normalized fuzzy similarity measure (NFSM), to deal with the nonlinear distortions. False acceptance on a pair of fingerprints captured from Cross Match sensor.

Investigate the capabilities of a NVIDIA graphic boards supporting CUDA for the parallel implementation of a well-known Finger Code algorithm [7]. Jiang , Manhua Liu and Alex C. Kot, Fingerprint recovery for recognition [15], Propose a new distance measure that better quantities the similarity evaluation between two orientation fields than the conventional Euclidean and Manhattan distance measures method used Orientation Field Computation and Segmentation Orientation Feature Vector Construction, Dominant Ridge Distance Computation.

In [18] , Weiguo Sheng , Gareth Howells, Michael Fairhurst and Farzin Deravi, A Mimetic Fingerprint Matching Algorithm, proposed An efficient matching operation to produce an initial population of local alignment configurations by examining local features of minutiae. Limits to the accuracy for rigid transformations.

In 2007, [19] Xiaoguang He, Jie Tian, Liang Li, Yuliang He and Xin Yang, Modeling and Analysis of Local Comprehensive Minutia Relation for Fingerprint Matching, presented A robust fingerprint matching scheme based on the comprehensive minutia and the binary relation between minutiae. A new fingerprint matching method to deal with non-linear distortion problems efficiently by clustering locally matched minutiae and warping the fingerprint surface using minutiae clusters [16] was proposed.

In [7], Sung Bum Pan, Daesung Moon, Youn- hee Gil, Dosung Ahn and Yongwha Chung, An Ultra-Low Memory Fingerprint Matching Algorithm and Its Implementation on a 32-bit Smart Card.

M. Lastra , J. M. Mantas, C. Urena, M. J. Castro and J.A Garcia-Rodriguez, found The potential data parallelism has been identified and an efficient implementation of this scheme to solve one-layer shallow-water systems has been derived for GPUs in [22].Prabhakar and Kameswara in [2] propose a scheme of a coarse minutia-based matcher for filtering out stored fingerprint templates that are unlikely to match the query.

A similar approach is proposed by Ratha and Jain in [4] for FPGA which are well known for their parallel capabilities. Additional steps are done to prevent the same feature from being matched more than once. Matching speed of $1.1\cdot10^5$ matches per second is reported. In [6], Introduce a framework to combine results of multiple classifiers by Wenwei Wang, Gerhard Rigoll. Sung Bum Pan, Daesung Moon,Younhee Gil, Dosung Ahn,Yongwha Chung [7] found a memory-efficient algorithm for the most memory consuming step (alignment) by doing more computations.

Marius Tico, Pauli Kuosmanen [8] proposed A novel fingerprint representation scheme that relies on describing the orientation field of the fingerprint pattern with respect to each minutia detail.

In [9], the theory and experimental results for automatic extraction of SPs including their spatial orientation from the global structure was found by Kenneth Nilsson, Josef Bigun.

In [10], [11], efficient surface representation method for surface matching and localized secondary features derived from relative minutiae information was proposed respectively. Proposing decomposing the input fingerprint image into decimation free directional images, it is easy to remove the noise directionally by means of adaptive mean filtering in [12].In [14], A novel method, a fuzzy feature match (FFM) based on a local triangle feature set to match the deformed fingerprints.

MUMmerGPU algorithm used as an open-source high-throughput parallel pair wise local sequence alignment program that runs on commodity Graphics Processing Units (GPUs) in common workstations [17].In [20][21], A Method to represent a minutiae set as a fixed-length feature vector, which is invariant to translation is proposed.

Mosaic method , Local Similarity Sort (LSS) method was proposed in [23] [24] respectively.In [26], Artificially expanding the set of training samples of spatial modelling technique was found. Indexing approaches: OF-Idx and MCC-Idx. Method is used in [27].

Anil K. Jain, Jianjiang Feng [28], proposed a system for matching latent fingerprints found at crime scenes to rolled fingerprints enrolled in law enforcement databases.

In [29], Fingerprint Preprocessing and Ridge Feature Extraction and Sensor Fingerprint Matching After Binarization in [30] is proposed.

Bai et al. in [31] investigate the capabilities of a NVIDIA graphic boards supporting CUDA for the parallel implementation of a well-known FingerCode algorithm [5], In a later press-release [32] authors state the increase of performance from 2.5 million matches per second on a CPU to 4 million on a GPU. Sunpreet S. Arora , Eryun Liu, Kai Cao, Anil K. Jain used Feedback Paradigm For Latent Matching algorithm [33].

III. OVERVIEW OF SYSTEM MODEL

A. NVIDIA CUDA Overview

The NVIDIA CUDA technology is used in a GeForce family of graphic boards [25]. Its programming model is shown in Figure 1. According to this model all computations are divided to CPU-based and GPU-based with CPU aliased as host and GPU as device. The host sends a program kernel – a compiled GPU shader to the device.

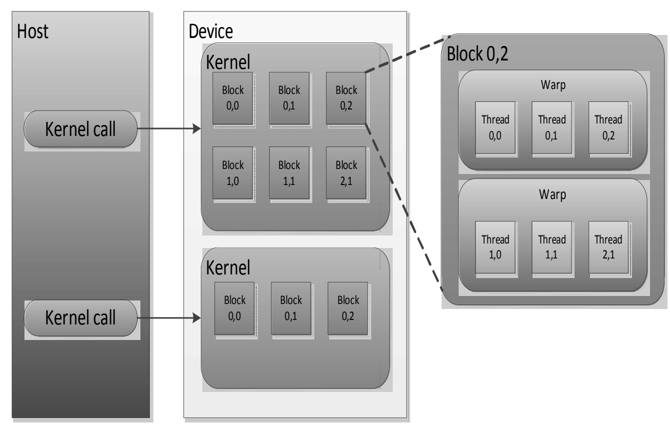

Fig.1. CUDA programming model

The set of blocks is organized as a grid. Currently the number of threads in a warp is 32, block and grid can't index threads more than in 3 dimensions simultaneously with a limitation of a 1024 threads maximum per a single block. Up to 16 blocks with a total number of threads less than or equal to 2048 can be executed simultaneously on a single multiprocessor. There are some limitations:

1. The block size should be chosen with register memory consumption per thread in mind as its amount is limited in a multiprocessor.
2. Global memory I/O is recommended to be optimized in such a way that, simply put, threads with a neighboring indexes are accessing neighboring 4-bytes words. This will allow accessing the data in batches instead of single word access for each thread.
3. It's forbidden to use recursive functions.
4. Branching to different paths is executed slower than branching to a single path by all threads.

Therefore implementing some algorithms in CUDA might be challenging.

B. Image Enhancement and Minutiae Extraction

Image enhancement is a crucial step for real-time AFIS, as the impression quality is generally lower during system usage than during enrollment. Different methods were proposed to improve the quality of the acquired fingerprint. It is shown that contextual enhancement is better than context-fewer enhancements and the superiority of the pixel-wise enhancement versus block enhancement.

The enhancement method involving Laplacian-like pyramid image decomposition and Linear Symmetry (LS) estimation is proposed by Fronthaler, Kollreider and Bigun in [34]. The main steps of this algorithm include image scaling, matrix subtractions and additions, convolutions of the source image with 2 kernels based on a Gaussian partial derivative, representing the result as a field of complex number and squaring it, and a final scalar multiplication with another kernel. This approach is considered to be efficient if implemented on massively parallel architectures due to the following reasons:

1. It may be viewed as a sequence of transformations where each

σ_1	0.6	T_{PS}	0.5	Neighborhood radius for max	9
σ_2	3.2	T_{LS}	0.9	Ring radii	Inner 4, outer 6

transformation uses only the previous results to obtain the next

Database (resolution)	NVIDIA GeForce 840M
FVC2004 DB1 (640 x 480)	0.259213 Seconds

one – every transformation is non-iterative. 2. Every transformation operates on 2D matrices in such a way that each element's computation is independent from the computation of other elements other than by the source data, which is immutable. 3. Furthermore, most of the transformations include 2D convolutions, either ordinary or complex, that can be efficiently parallelized [25], and most kernels are Gaussians or its derivatives, so their separability property may as well be used. In our experiments we have implemented convolutions in a straightforward way without using separability due to the relatively small size of the kernels and the coalescing memory access pattern. While a parallel processing was done on a matrix element level, so our results can further be improved. The size of Gaussian kernels was determined using a "3 sigma" rule by a formula $S = 1 + 2\lceil 3\sigma \rceil$. It could also be observed that amount of operations per pixel is constant. Parameters used are summarized in Table I, and processing time are shown in Table II.

TABLE I
Parameters for image enhancement routine

TABLE II
Image enhancement processing time

Database (resolution)	NVIDIA GeForce 840M
FVC2004 DB1 (640 x 480)	0.156450 seconds

Minutiae extraction, also described in [34] is performed afterwards. LS and Parabolic Symmetry (PS) are estimated in order to produce the final measure $PSi(x,y) = PS(x,y) \cdot (1 - |LS(x,y)|)$. The area of each local maximum (Xm, Ym) is checked for a consistent highLS, and the list of maximums is then sorted by a descending $|PS(Xm, Ym)|$ as a certainty measure resulting in a minutiae list. For the reasons described below we are storing only top 32 minutiae as a template. The same considerations as for image enhancement are valid in this case as well. Convolutions and local maximum estimations are implemented in parallel while the minutiae list generation and sorting is done sequentially. Generally, those operations are lightweight compared to LS and PS estimation. Table III contains the parameters of the process being used. Those parameters should be chosen carefully in order to minimize the amount of spurious minutiae. However we have observed a consistency in their detection among different impressions of the same finger. The performance results are shown in Table IV.

TABLE III
Parameters for minutiae extraction routine

TABLE IV
Minutiae extraction time

C. Minutiae Based Fingerprint Matching

Minutiae-based fingerprint matching is a cornerstone of a modern AFIS. As noted in [4], database matching can be done in parallel on 2 levels: micro-level where different minutiae are

σ_1	0.6	f_0	1.7	f_2,f_3	1.3	τ_2	0.3	$\sigma_{directions}$	2.0
σ_2	3.2	f_1	1.21	τ_1	0.1	Annulus radii	Inner 4, outer 6	Number of directions	20

matched in parallel with a given one and macro-level where different fingerprints are matched in parallel. However we need to point out that methods proposed in [2] and [4] are operating on an already aligned set of fingerprints, and Hough transform-based alignment is noted as a plausible for a selected FPGA without quality estimations. Fingerprint alignment is a challenging task that can be formulated in a following way: given two fingerprint templates t_1 and $t_2$what are the best parameters for translation and rotation to maximize the matching score.

In our approach the fingerprint database is preloaded to the GPU global RAM. This step adds up to the system security, as GPU RAM is separated from CPU RAM and is available only through driver operations. We are assigning one CUDA block to each of the database fingerprints so it scales naturally on different and more advanced graphic cards. Each block consists of 32x32 threads, and currently it is the technological maximum for CUDA. Thus we are limiting the template size to 32 minutiae. Both query template $Q\{q_i\}$ and database template $D\{d_j\}$are loaded to the cache memory of a multiprocessor. After that for every minutia q_n and d_k in these templates a template copy (Q_n $\{q_{ni}\}$and D_k $\{d_{kj}\}$ accordingly) is made and translation is done to make the minutia q_{nn} and d_{kk}coordinates equal to (0,0), so indices n and k can be used to indicate translation parameters. The thread T_{ij} then uses templates Q_i and D_j to calculate the possible rotational alignment.

In order to do this we are assuming that fingerprints are already coarsely rotationally aligned by a scanner that is, its acquisition window or similar sensor is designed in a way that makes it difficult or inconvenient to acquire a rotated fingerprint. It is usually the case for modern scanners. For every minutiae pair in $(q_{im} \in Q_i \backslash q_{ii};\ d_{jn} \in D_j \backslash d_{jj})$ the difference between Euclidean distances $(\overline{q_{ii} q_{im}})$ and $(\overline{d_{jj} d_{jn}})$ is computed. After that the slope angles of $\overrightarrow{q_{ii} q_{im}}$ and $\overrightarrow{d_{jj} d_{jn}}$ and their differenceθ_{imjn} are calculated. If distance difference and angle difference are below certain thresholds (in our experiments 3 and $\frac{\pi}{8}$ accordingly), (i, j, θ_{imjn}) is considered to be a plausible alignment parameters and is stored in a cache memory.

The second step is matching two fingerprints given a set of plausible alignment parameters. The set is distributed equally among 1024 working threads. Matching procedure is similar to the one described by Wegstein in [1], but it uses Euclidean distance differences instead of coordinate differences (6 in our experiments). Due to the constant amount of the minutiae in the templates the thread output is a number of the matched minutiae. Its maximum is searched among all thread outputs (and consequently among all the alignment parameters) and is considered to be the final similarity score.

FVC 2004 DB1 was used in conjunction with the image enhancement and minutiae extraction techniques to obtain the quality estimations for the matches. We report an average time of 0.000823 seconds for a matching two fingerprints GeForce 840M as shown in Table V.

TABLE V
Minutiae Matching Time

Database (resolution)	NVIDIA GeForce 840M
FVC2004 DB1 (640 x 480)	0.000823 seconds

IV. Performance Evaluation

By combining hough transform performance is increased significantly than individual implementations of algorithms as shown in Fig. 2. Parallelization of minutiae extraction and matching will be optimized for memory transfers using shared memory to store query and other fingerprint minutiae image sets. Optimized kernel thread configuration will be computed before launching of kernel. As compared with MCC, Hough Transform algorithm requires less time as shown in Figure 2. Our results demonstrate that convolution-based and/or scalar multiplication-based pixel-wise image enhancement and direct grayscale minutiae extraction can be implemented efficiently on modern GPUs. To evaluate the performance of proposed system FAR and FRR of proposed

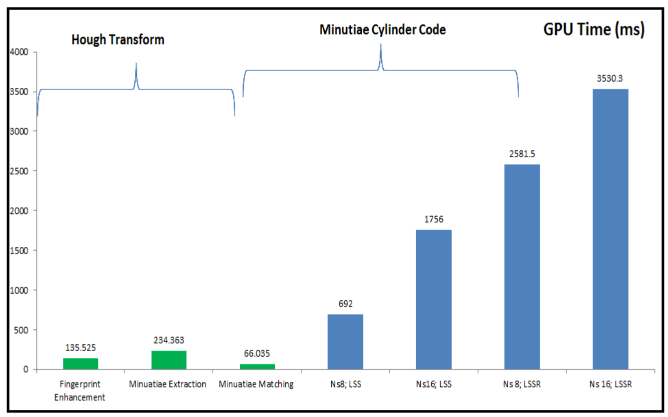

Fig 2: Comparison of Hough Transform and MCC Operations

system are calculated. Figure 3 shows the FAR and FRR of proposed fingerprint identification system. 32 minutiae are on average only 50 to 75% of its total amount in a given fingerprint. While some fingerprints in a database contain unrecoverable regions, we consider the spurious minutiae and operating on a subset of a template to be the main reasons of the high FAR. The alignment step of our approach is essentially a “brute force” search which can produce several thousand of possible alignments. With a pre-alignment step being done prior to the matching routine the implementation of a micro-level parallelism for minutiae matching on CUDA is a straightforward decision. Due to this, accuracy and quality of recognition is comparatively high.

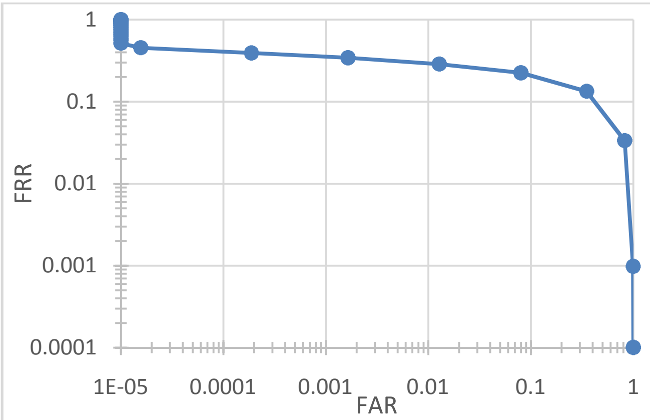

Fig 3: ROC curve

CONCLUSION

Biometric is an authentication and verification technique that relies on measurable physical characteristics. It focuses on measuring and analyzing biological data. A rigorous empirical inference fingerprint databases shows the efficiency by taking advantage of the GPUs. System is able to perform identification in a reasonable amount of time or a matching two fingerprints on GeForce 840M maintaining the accuracy of the CPU implementation. Thus accuracy and quality of recognition is comparatively high.

ACKNOWLEDGMENT

We appreciatively acknowledge the support of NVIDIA Corporation with the donation of the architecture (GeForce GT 840M) GPU used for this research. The authors would like to acknowledge Computer Engineering department, SITRC and all the people who provided with the facilities being required and conductive conditions for completion of the paper.

REFERENCES

[1] Wegstein J.H., " The M40 Fingerprint Matcher." U.S. Government Publication, National Bureau of Standards, Technical Note 878, U.S Government Printing Office, Washington, DC, 1972.

[2] Prabhakar, R.V.S.N and Rao, C.V.K. " A parallel algorithm for fingerprint matching." In Fourth IEEE Region 10 International Conference TENCON, 373 – 376, 1989.

[3] Gowrishankar, T.R., "Fingerprint Identification on a Massively Parallel Architecture in Proceedings of the 2nd Symposium on Frontiers of Massively Parallel Computation , 331–334, 1989.

[4] Ratha N.K., Jain, A.K., Rover, D.T., Cantoni, V, Lombardi, L., Mosconi, M., Savini, M. and Setti, A., " An FPGA-based point pattern matching processor with application to fingerprint matching." in*Proceedings of the Computer Architectures for Machine Perception,* IEEE Computer Society, 394, 1995.

[5] Jain, A.K., Prabhakar, S., Hong, L. and Pankanti, S.Finger Code: " A Filter bank for Fingerprint Representation and Matching." In Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, vol. 2, 2187, 1999.

[6] Wenwei Wang and Gerhard Rigoll, " Combination of Multiple Classifiers for Handwritten Word Recognition," IEEE Computer Society, New York, 2002.

[7] Sung Bum Pan, Daesung Moon, Younhee Gil, Dosung Ahn and Yongwha Chung," An Ultra-Low Memory Fingerprint Matching Algorithm and Its Implementation on a 32-bit Smart Card," IEEE Transactions On Image Processing, 2003.

[8] Marius Tico and Pauli Kuosmanen," Fingerprint Matching Using an Orientation-Based Minutia Descriptor," IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 25, No. 8, 2003.

[9] Kenneth Nilsson and Josef Bigun,"Localization of corresponding points in fingerprints by complex Filtering," 2003.

[10] Yiyong Sun , Joonki Paik, Andreas Koschan, David L. Page and Mongi A. Abidi," Point Fingerprint: A New 3-D Object Representation Scheme", IEEE Transactions On Systems, Man, And CyberneticsPart B: Cybernetics, Vol. 33, No. 4, 2003.

[11] Tsai-Yang Jea and Venu Govindaraju ,"A minutia-based partial fingerprint recognition system", Published by Elsevier Ltd, 2005.

[12] Muhammad Talal Ibrahim, Imtiaz A. Taj, M. Khalid Khan and M. Aurangzeb Khan,"Fingerprint Image Enhancement Using Decimation Free Directional Adaptive Mean Filtering", Springer-Verlag Berlin Heidelberg ,2006.

[13] Xinjian Chen, Jie Tian, Senior Member and Xin Yang, "A New Algorithm for Distorted Fingerprints Matching Based on Normalized Fuzzy Similarity Measure," IEEE Transactions On Image Processing, Vol. 15, No. 3, 2006.

[14] Xinjian Chen , Jie Tian, Xin Yang and Yangyang Zhang,"An Algorithm for Distorted Fingerprint Matching Based on Local Triangle Feature Set," IEEE Transactions On Information Forensics And Security, Vol. 1, No. 2, 2006.

[15] Xudong Jiang , Manhua Liu and Alex C. Kot, "Fingerprint Retrieval for Identi_cation," IEEE Transactions On Information Forensics And Security, Vol. 1, No. 4, 2006.

[16] Dongjin Kwon ,Dong Yun , Duck Hoon Kim and Sang Uk Lee,"Fingerprint Matching Method Using Minutiae Clustering and Warping," IEEE Transactions, 2006.

[17] Michael C Schatz ,Cole Trapnell,Arthur L Delcher and Amitabh Varshney," High-throughput sequence alignment using Graphics Processing Units," BMC Bioinformatics, 2007.

[18] Weiguo Sheng , Gareth Howells, Michael Fairhurst and Farzin Deravi, "A Memetic Fingerprint Matching Algorithm," IEEE Transactions On Information Forensics And Security, Vol. 2, No. 3,2007.

[19] Xiaoguang He, Jie Tian, Liang Li, Yuliang He and Xin Yang," Modeling and Analysis of Local Comprehensive Minutia Relation for Fingerprint Matching," IEEE Transactions On Systems, Man, And CyberneticsPart B: Cybernetics, Vol. 37, No. 5, 2007.

[20] Haiyun Xu , Raymond N. J. Veldhuis ,Asker M. Bazen, Tom A. M. Kevenaar, Ton A. H. M. Akkermans and Berk Gokberk," Fingerprint Veri_cation Using Spectral Minutiae Representations," IEEE Transactions On Information Forensics And Security, Vol. 4, No. 3, 2009.

[21] Qijun Zhao, Lei Zhang, David Zhang and Nan Luo," Direct Pore Matching for Fingerprint Recognition,"Springer-Verlag Berlin Heidelberg, 2009.

[22] M. Lastra , J. M. Mantas, C. Urena, M. J. Castro and J.A Garcia-Rodriguez,"Simulation of ShallowWater systems using GPUs", 2009.

[23] Heeseung Choi ,Kyoungtaek Choi and Jaihie Kim,"Mosaicing Touchless and Mirror-Reected Fingerprint Images", IEEE Transactions On Information Forensics And Security, Vol. 5, No. 1, 2010.

[24] Ra_aele Cappelli , Matteo Ferrara and Davide Maltoni," Minutia Cylinder-Code: A New Representation and Matching Technique for Fingerprint Recognition", IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 32, No. 12, 2010.

[25] Sanders, J.,Kandrot, E. CUDA by Example: An Introduction to General-purpose GPU Programming, Addison Wesley Professional, 2011.

[26] K. C. Leung, C. H. Leung,"Improvement of Fingerprint Retrieval by a Statistical Classi_er", IEEE Transactions On Information Forensics And Security, Vol. 6, No. 1,2011.

[27] Raaele Cappelli, Matteo Ferrara and Dario Maio," Candidate List Reduction Based on the Analysis of Fingerprint Indexing Scores", IEEE Transactions On Information Forensics And Security, Vol. 6, No. 3, 2011.

[28] Anil K. Jain and Jianjiang Feng," Latent Fingerprint Matching",IEEE Transactions On Pattern Analysis And Machine Intelligence,Vol. 33, No. 1,2011.

[29] Heeseung Choi, Kyoungtaek Choi and Jaihie Kims," Fingerprint Matching Incorporating Ridge Features With Minutiae", IEEE Transactions On Information Forensics And Security, Vol. 6, No.2,2011.

[30] "Real-time 2D skeletonization using CUDA", from University of Groningen, 2011. Retrieved May 16, 2013.

[31] Bai, S., Marques, J.P, McMahon, M.T., Barry, S.H.," GPU-Accelerated Fingerprint Matching," from GPU Technology Conference 2012, 2011. Retrieved May 16, 2013,

[32] "Minutia Cylinder-Code", 2011. from Biometric System Laboratory, University of Bologna. Retrieved May 16, 2013.

[33] Sunpreet S. Arora, Eryun Liu, Kai Cao and Anil K. Jain ," Latent Fingerprint Matching: Performance Gain via Feedback from Exemplar Prints", IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 6, No. 1,2014.

[34] Fronthaler, H., Kollreider, K. and Bigun, "Local features for enhancement and minutiae extraction in fingerprints",IEEE Transactions on Image Processing, 17 (3), 354–363,2014.

[35] Pablo David Gutirrez, Miguel Lastra,Francisco Herrera, and Jos Manuel Bentez," A High Performance Fingerprint Matching System for Large Databases Based on GPU", IEEE Transactions On Infor mation Forensics And Security , VOL. 9, NO. 1, January 2014.

AUTHOR'S PROFILE

Ms. Shital T. Tupe

is a P.G. student of Computer Engineering at SITRC College of Engineering , Nasik under Savitribai Phule Pune University . She has completed her undergraduate Course of Engineering from Savitribai Phule Pune University. Her areas of interest include Parallel Computing, Software Testing.
Email Id: sheetal_t_tupe@yahoo.com

Prof. Amol D. Potgantwar

Working as Asst. Professor of the Department of Computer Engineering, Sandip Institute of Technology and Research Centre, Nasik, Maharashtra, India. The focus of his research in the last decade has been to explore problems at Near Field Communication. In particular, he is interested in applications of Mobile computing, wireless technology, Image Processing and Parallel Computing. He has register patents like Indoor Localization System for Mobile Device Using RFID & Wireless Technology; He has been an active scientific collaborator with ESDS, Carrot Technology, Techno vision and Research Lab including NVIDIA CUDA, USA. He is a member of CSI, ISTE, IACSIT .
Email Id: amol.potgantwar@sitrc.org