Latent Fingerprint Matching in Large Databases Using High Performance Computing

Yenumula B Reddy
Department of Computer Science
Grambling State University, Grambling, LA 71245, USA
ybreddy@gram.edu

***Abstract*—Most of the fingerprint matching systems use the minutiae-based algorithms with matching of ridge patterns. These systems consider ridge activity in the vicinity of minutiae points, which has poorly recorded/captured exemplar prints (information). In this research, we recommend the MapReduce technique to identify a required fingerprint from the reference fingerprint database. In the proposed MapReduce process, minutiae of the latent fingerprint used as keys. The latent fingerprints are analyzed using Bezier ridge descriptors to enhance the matching of partial latent against reference prints. Since the retrieval of reference print is same as retrieval of the required document, we suggested the MapReduce model for detection of required fingerprint.**

Keywords: fingerprint, latent prints, minutiae points, MapReduce, Bezier ridge;

I. INTRODUCTION

Fingerprints help to identify individuals from the unique pattern of whorls and lines. The research concludes that no two people (even the twins) have the same fingerprint. The fingerprints do not change one year after the birth of an individual. Therefore, the fingerprints are part of the biological data acquisition to identify an individual. The issue raises a new challenge in storing, handling and analysis of data at the rate it generates. Common usage like analysis, comparing, transferring files, and processing is slow and computationally expensive due to the size of the database at any individual workstation. The analysis of fingerprints requires comparison of several features of the print patterns that include characteristics of ridges and minutia points. Since the amount of fingerprint data is unstructured or semi-structured and increasing exponentially, we need to look for a different solution. The new solution is Hadoop distributed file systems.

Big data is a popular term used for structured, unstructured, and semi-structured large volume of data. Analysis of such data is helpful in business, government and semi-government in operational efficiencies, decision making, reduced risk, and cost reductions. Big data measure in volume, velocity, and variety. The volume includes the unstructured streaming of social media data, internet data, ecommerce data, and Government data. Dealing the unprecedented speed (velocity) of this voluminous data is a challenging job. These data sets have a variety of formats includes text documents, emails, video, audio, stock data, and financial transactions.

Working with big data does not mean the acquiring of a large amount of data. It is the work you plan to design the unstructured or semi-structured data. The plan is to analyze the data to minimize the cost, real-time response or speed of returning the results, quick decision making, and optimization techniques. The examples of improving the performances are:

- use the new technology to return the real-time response and save the dollar amount
- optimize the routes to deliver the goods, analyze the stocks and maximize the profit
- increase the sales based on customer's past purchases
- calculate the risks and protect the business, identify the important customers in related business and
- use the artificial intelligence techniques or use the data mining techniques to improve the company

The big data analytics considers various types of data to uncover the hidden patterns, unknown correlations, customer preferences, market trends, revenue opportunities and advantageous to respective organizations. Big data strategy can help to pull all related data into a single system to identify the patterns among the customers and identify which type of clients buy a particular product. Big data analysis can be technical, Government, or business related. The companies prefer to use the Big Data management to receive a quick response. They identified the quick response is possible through high-performance computing. New algorithms and program techniques are on the way to produce real-time response using parallel processing techniques.

The rate of fingerprint data generated today is tough to store and analyze using traditional methods. Therefore, the fingerprint database is one of the largest databases, considered as big data. Further, fingerprint analysis has been used to identify the suspects and solve the crimes for more than 100 years. It is a precious tool for law enforcement. Crime scene data (latent) is unstructured data needed to be analyzed and processed before its use. Every fingerprint has a unique pattern made by friction ridges and furrows that appear on the pads of the fingers and thumbs.

When a crime occurs, law enforcement enables to obtain the fingerprints (latent) from crime area and analyze the patterns and match the potential matches. Currently, the system scans the prints and analyzes all ridges, swirls, loops, and other related patterns to uncover the possible matches. The current matching algorithms do not allow their use in large fingerprint databases due to their computational limitations. The GPU (graphics processing unit) based fingerprint matching methods can overcome these limitations [1 - 2]. Their study shows that GPU-based computation speed up the process and minimizes the cost. Further, the authors suggested that GPU-based processing opens the new field of possibilities to obtain real-time fingerprint identification in large databases.

The remaining document discusses the review of the literature, fingerprint identification method, technical approach, MapReduce method, MapReduce testing with documents, and conclusions.

II. REVIEW OF WORK

Many workplaces use personal identification, passports, cellular phones, credit cards, automatic teller machines, and driver licenses are using the personal identification in an encrypted form. Cryptography plays a significant role to encode and decode the identification for storing and retrieving. In spite of encrypted form, fraud is in credit cards alone reach billions each year in worldwide. Therefore, biometric identification helps to identify an individual in a unique way. Biometric system information verifies and identifies an individual. The biometric information may include voice, face, fingerprints, eyes, retina, signature, keystroke dynamics and much similar identity information. Fingerprints are becoming more common to identify a person and identification machines are becoming cheaper.

The fingerprint matching process helps to determine two sets of ridge details come from the same finger. The work in [3-9] studied fingerprint matching, authentication, comparative study of fingerprint data, the performance of fingerprint quality measures, and the statistical study. The proposed algorithms use matching minutiae points or similarities between two finger images. Ackerman [10] discussed a method of for fingerprint matching based on minutiae matching. The method uses the preprocessed region analysis to eliminate the processing delay. Bhuyan et al. [11] discussed the fingerprint classification using data mining approach. The author used linear k-means for the cluster formation. The authors claim the proposed approach has higher accuracy and eliminates the misclassification errors. In [12] the authors presented fingerprint classification algorithm and tested on NIST-4 fingerprint database. The classification algorithm classifies the input fingerprints into five categories according to the number of singular points, their relative position, and the presence of recurring ridges called type-1 and type-2. The proposed algorithms achieve better performance compared to previous algorithms.

Processing fingerprinting for big data and fingerprint matching problem using NVIDIA GPU are current research in Hadoop distributed file systems using GPU-based implementation. Tretyakov et al. [13] discussed the probabilistic fingerprinting to reduce the use of computational resources and increase in proceeding speed. The authors in [14] discussed the Gabor filter bank based algorithm that uses the fingerprint images. They claimed that GPU-based implementation was 11 times faster than CPU-based implementation. Awan [15] discussed the local invariant feature extraction using GPU and CPU (central processing unit) implementation. The GPU takes the shorter processing time compared to CPU. The authors used the feature extractors called scale invariant feature transform (SIFT) and speeded-up robust feature (SURF). They concluded that SURF consumes longer matching time on the GPU.

Gutierrez et al. [16] used minutia cylinder-code (MCC) based algorithm in GPU fingerprint-based system to enhance the performance. The tables show that the GPU-based MCC implementation is 35 times faster than the CPU (single thread) on a single GPU and approximately 100x faster on multiple GPUs. They conducted the experiments on 55k size database. All the above experiments were carried out with known fingerprint search. The papers do not show that the search was carried out with latent prints.

The automated matching of partial latent prints is difficult to current systems. The conventional methods require a sufficient number of minutiae (ridge bifurcation and termination) to support the search. To develop an automated system is a significant challenge to detect a matched fingerprint from the available latent print. This method should not rely on traditional minutiae matching methods. Walch and Reddy [2] used the GPU technology to solve the latent fingerprints matching problem. They proposed Bezier curves as ridge descriptors, to produce accurate overlays of the latent onto a reference print. The GPU-based method used by the authors performs near real-time results. The comparisons vary from 20 million to 971 billion depending upon the reference Beziers. They claimed that the processing of 8 days on CPU reduced to one hour on GPU cluster.

Contribution

The research presents the current state of fingerprint algorithms using various techniques that include traditional, pattern recognition, and hybrid methodologies. The latent prints are analyzed using Bezier ridge descriptors to enhance the matching of partial latent against reference prints. We implemented the Hadoop package using a single node and multiple nodes in our research laboratory to analyze the stream of documents and retrieve the required document. We then proposed MapReduce model to identify and retrieve the matched fingerprint from the database with the help of latent print.

III. FINGERPRINT IDENTIFICATION

Fingerprints are the impressions or mark made on a surface of a person's fingertip. These are unique patterns of recognition of an individual using the pattern of whorls and lines on the finger impression. The combination of ridge bifurcation, trifurcation, bridge, ridge crossing, hook, ridge ending, island (short ridge), dot and similar characteristics determines the uniqueness of the fingerprint. The identification points consist of bifurcations, ending ridges, dots, ridges, and islands. Most of the prints (quality fingerprint) contain 60 to 80 minutiae. A single rolled fingerprint may have as many as 100 or more identification points and used for fingerprint identification purposes. There is no exact size requirement for fingerprint match since the number of points found on a fingerprint impression depends upon the location of the print.

There are many algorithms to match the fingerprints of an individual stored in the database. Some of the algorithms for fingerprint detection include the nearest neighbor, fixed radius, phase-based image matching, feature-based matching and combination of phased-based and feature-based. In the nearest neighbor algorithm, the K adjoining minutiae is considered for matching the neighborhood of known minutiae. The fixed radius algorithm uses a circle of radius centered on minutiae and the neighborhood patterns to match with the database samples. The phase-based image matching algorithm uses the phase components in a two-dimensional (2D) discrete Fourier Transform of the given images. The feature-based matching algorithm uses pairwise corresponding image features of an edge, a corner, a line, or a curve. The combination of phase-based image matching and feature-based matching fingerprint recognition algorithm helps for weakly impressed and low-quality fingerprint images.

Fingerprint authentication models include the extraction of raw data, matching the extracted features, get match score and authentication decision. The authentication does not provide complete protection of data. The authentication also depends upon the operating conditions and among individuals. The operating conditions include dirt across fingerprint sensor, a sensor may malfunction, and static electricity may cause a sensor malfunction. Clancy et al. [17] discussed the fundamental insecurities that hamper the biometric authentication and cryptosystem capable of using the fingerprint data with cryptography key. Hong et al. [18] presented the automatics authentication system with fingerprints as an individual identity. Thai and Tam [19] discussed the standardized fingerprint model that is used to synthesize the template of fingerprints. The steps include preprocessing, adjusting parameters, synthesizing fingerprint and post-processing. Jain et al. [4] used the filter-based fingerprint matching algorithm that uses Gabor filters to capture both local and global details. The matching depends on the Euclidean distance between the two corresponding codes.

Conventional hashing algorism used for fingerprint matching is an expensive process as the data is increasing in terabytes. The conventional data collectors have different filenames with different data or same content with different file names. If we download the files for processing, most of the times we have duplicate data. Therefore, Tretyakov et al. [13] used the algorithm based on probability theory. The algorithm computes the fingerprint file by using few data samples. The model is useful for network architecture to obtain the samples stored remotely.

The probability model requires the organized data since hashing is the primary tool to compare and extract. The reference fingerprint data particularly latent prints are unorganized data and need to be analyzed and processed before use. In the analysis, we use particular characteristic called minutiae. Gutierrez et al. used Minutiae cylinder-code for accurate results [1]. The model was slow due to computational requirement with current CPU speed. The algorithm was modified to use the NVIDIA GPU processors. The performance was many times faster than single CPU. Further, the problem was to move the data into GPU memory, process, and return the results back to host memory to display or return the results. Currently, this process cannot be changed, but in future NVIDIA technology GPUs can access the host memory and process the data at GPUs.

The GPU-based model was also discussed by Walch and Reddy [2] to solve the latent fingerprint problem. Their model is a unique attempt to identify the criminals and terrorists. The authors used handwriting recognition that offers a means of generating ridge-specific markers. The tool uses Bezier descriptors. It described through four points that include 2 end points and 2 control points. The model involves finding a subset of corresponding ridge sections common to both latent and reference points. Then it uses the step-by-step curve matching process. The technique also used by Rahman et al. [20] to identify an original image from reference shape. They used the technique called parametric curve generation. The authors compared the results that the curve generated by composite Bezier curve.

For reliable processing, the fingerprint algorithms need to eliminate noise, extract minutiae and rotation and translation-tolerant fingerprint matching. We need to create an algorithm to be suitable for current hardware technology. The hardware may be cloud-based or cloud-based technology using GPUs. The algorithm must complete a large number of comparisons in microseconds. Since current technology with available CPUs meets their limitations, we need to use GPU-based processing to get real-time or near real-time response.

The research focus is on big data analysis and MapReduce process. We suggested using fingerprint data to store and retrieve. The latent fingerprint data as the search key for the analysis of fingerprint match in the database. Currently, the fingerprint data is outweighing for current

computing facilities. Performing the computational analysis of such data into usable information and provide the results in close real-time is the goal. Due to limitations of current CPU, we recommend the GPU technology for fast response to process such data. Special algorithms are needed to use GPU technology to meet customer satisfaction.

IV. TECHNICAL APPROACH

Latent fingerprints are difficult to analyze due their poor image quality and limitations of a number of minutiae available for identification. A latent fingerprint image is normally 20 to 30 percent of the original fingerprint image. Identifying the original fingerprint with 20% of its information and the minimum number of minutiae points required to match a fingerprint is an unsolved problem. Search becomes a difficult part without enough ridge bifurcations and terminations (minutiae). The latent prints may come to any location (part) of the fingerprint. The latent may contain meaningful information if they come from the core and less information from other regions of the print. The ridge specific markers help potential information and ridge geometry creates new features to supplement the missing bifurcations and ridge endings. If we create proper ridge specific markers, they should be functionally equal to traditional minutiae.

Bezier curve of the third order is commonly used to approximate the path of a curved object such as a ridge. The Bezier curve is a smooth mathematical curve that can be used to approximate the path of a curved object such as ridges. Bezier curves precisely fit into the curvature of ridges. Bezier descriptors can be used to 'mark' positions on the ridges creating 'minutiae' where traditional minutiae are scarce or even non-existent. The third order Bezier curve consists of two end points and two control points. Figure 1 describes the four related points called end points (P_0, $P_{3)}$ and control points (P_1, $P_{2)}$. The control points are positioned outside to the curve and define the shape of the curve. The Bezier curve does not pass through the control points. The control points show the direction of the curve.

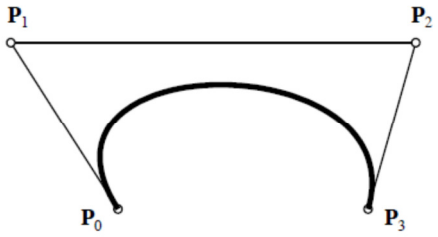

Figure 1: Cubic Bezier Curve

Bezier curves are polynomials of t that varies 0 to 1. The equation of cubic form of Bezier curve is of the Binomial form with end point n is given below.

$$Bezier(n,t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-i} t^{i} \qquad (1)$$

In the equation (1)

$\binom{n}{i}$	Binomial term
$(1-t)^{n-i}t^{i}$	Polynomial term
Σ	series of additions

Bezier curves can be drawn in many ways. Let us consider two ways of Bezier curves. First, use the de Casteljau's algorithm [21]. Consider various points between beginning and end of the curve (t=0 as starting point and t = 1 as an end point) with increment Δt. The smooth curve will be obtained with more points between beginning and ending (smaller value of Δt. The second method is sampling the curve at certain points and joining those points up with straight lines and smoothen the curve along those lines. The disadvantage is that we lose the precession of working with a real curve. Further, we can create a tool with ridge specific markers (that generates from minutiae) in the lines of handwritten character recognition. It will become a unique tool in fingerprint recognition with latent prints.

The sampling model can be used for latent Bezier curve creation. Sample each edge in the latent fingerprint skeleton to form the Bezier sample to compare with reference print. The process continues till the minimum acceptance (threshold) point reached between latent and reference prints. The matching does not include complete curve matching. The comparison is the similarity between the latent and reference prints. Exact length never happens with a latent print comparison. If the latent matches the part of the reference curve, we will continue to match next warp to match and continue to all warps of latent. The threshold is set the certain percent of matching with reference print.

V. MAPREDUCE METHOD

The e-commerce data, Facebook data, Twitter data, or any similar data are examples of Big Data. Currently, Big Data analysis uses the MapReduce technology. Recent analysis shows that the unstructured data increased exponentially and piled up in petabytes. Fingerprint data is semi-structured. The keywords based upon the latent fingerprints to search the reference database.

In this paper, the first approach uses the MapReduce techniques. In MapReduce approach, we use latent fingerprint data as keys and retrieve the required fingerprint from the fingerprint database. The proposed method helps to find the repetition of each key in reference database and retrieve the closest match. The method is similar to the number of times each keyword repeats in a text file or any stream of data to select particular document is important. Figure2 shows the MapReduce process.

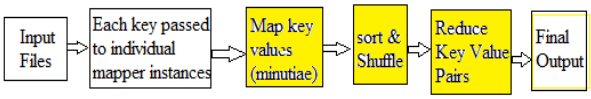

Figure 2: MapReduce Process

VI. MapReduce - Testing with Documents

To select a required document Using MapReduce, we provided the keywords and their importance that varies between 0 and 1. We then take the important factor multiplied by the number of times keyword and sum the result of all keyword importance. If the sum is ≥ threshold we conclude that the document is required. The algorithm was coded in two steps. During the first step, the reputation of the words and in the second step the importance factor and selection of the document were coded in Python. We processed six text files to compare the results of the current experiment. The keywords and impact factor provided are: medicine (0.02), profession (0.025), disease (0.02), surgery (0.02), mythology (0.02), and cure (0.05). The size of each file in words, times taken in seconds to process and impact factors respectively are: (128,729; 0.1539; 5.39), (128,805; 0.1496; 0.62), (266,017; 0.13887, 0), (277,478; 0.1692; 6.02), (330,582; 0.1725; 7.93), and (409,113; 0.2032; 18.87). The threshold set was 10.0. Therefore, the file with impact factor 18.87 is selected as required file. If we lower the threshold to 5.0 another two files with impact factors 6.02 and 7.93 would become our required files.

In the proposed fingerprint identification model with MapReduce process, we provide Minutiae data evolved from Bezier curves as keys to search the reference fingerprint database.

VII. Conclusions and Future work

The paper discusses the currently available fingerprint identification algorithms and models. We tested the MapReduce method for a stream of documents and recommended to retrieve the reference fingerprints using latent information as keys. Currently, we are working on the latent fingerprint data as keys to MapReduce process to retrieve the required reference fingerprint.

References

1. P. D. Gutierrez, M. Lastram F. Herrera, and J.M. Benitez., "A high Performance Fingerprint Matching System for Large Databases Based on GPU", Information Forensics and security, IEEE Transaction on Biometrics Compendium, Vol. 9, Issue 1, 2014, pp. 62-71.
2. "M. A. Walch and Y. S. Reddy., "Using GPU Technology to Solve the Latent Fingerprint Matching Problem," GTC Express Webinar, July 11, 2012.
3. A. Jain, S. Prabhakar, and A. Ross., "Fingerprint Matching Using Minutiae and Texture Features", Proceedings of the International Conference on Image Processing (ICIP), Thessaloniki, Greece, 2001, pp. 282–285.
4. A. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," IEEE Transactions on Image Processing, vol. 9, no. 5, pp. 846–859, May 2000.
5. A. Jain, L. Hong, S. Pankanti and R. Bolle., "An identity authentication system using fingerprints", Proc. IEEE 85(9), 1365–1388 (1997).
6. F. Alonso-Fernandez, etc., "A comparative study of fingerprint image quality estimation methods", IEEE Trans. on Information Forensics and Security 2(4), 734–743 (2007).
7. F. Alonso-Fernandez, etc., "Performance of fingerprint quality measures depending on sensor technology", Journal of Electronic Imaging, Special Section on Biometrics: Advances in Security, Usability and Interoperability (to appear) (2008)
8. A. Bazen and S. Gerez., "Systematic methods for the computation of the directional fields and singular points of fingerprints", IEEE Trans. on Pattern Analysis and Machine Intelligence 24, 905–919 (2002).
9. E. Bigun, J. Bigun, B. Duc, and S. Fischer., "Expert conciliation for multi modal person authentication systems by Bayesian statistics", Proc. International Conference on Audio- and Video-Based Biometric Person Authentication, AVBPA LNCS-1206, 291–300 (1997).
10. A. Ackerman and R. Ostrovsky. "Fingerprint recognition." UCLA Computer Science Department (2012).
11. M. Bhuyan, S. Saharia and D. Bhattacharyya., "An Effective Method for Fingerprint Classification", International Arab Journal of e-Technology, vol.1, no.3, Jan 2010.
12. L. Hong and A. Jain., "Classification of Fingerprint Images", 11th Scandinavian conf. Image Analysis, Kangerlussuag, Greenland, June 7-11, 1999.
13. K. Tretyakov, etc., "Fast Probabilistic file fingerprinting for big data", BMC Genomics. 2013, 14 (Suppl 2):S8; ISSN 1471-2164 - p. 8.
14. R. Lehtihet, W. Oraiby, and M. Benmohammed., "Fingerprint grid enhancement on GPU", International conference on Image Processing Computer Vision, and Pattern Recognition (IPCV 2013), 2013, pp 1-4.
15. A. Awad., "Fingerprint Local Invariant Feature Extraction on GPU with CUDA", Informatica, vol. 37, 2013, pp. 279-284.
16. P. D. Gutierrez, M. Lastra, F. Herrera, and J. M. Benitez., "A high Performance Fingerprint Matching System for Large Databases Based on GPU", IEEE Transactions on Information Forensics and Security, vol. 9, no. 1, Jan 2014, pp: 62-71.
17. T. Clancy, N. Kiyavash and D. Lin., "Secure Smartcard-based Fingerprint Authentication", ACM SIGMM workshop on Biometrics methods and applications, 2003, pp: 45-52.
18. L. Hong, A. Jain, S. Pankanti and R. Bolle., "Identity Authentication Using Fingerprints", First International Conference on Audio- and Video-Based Biometric Person Authentication, (AVBPA) 1997, pp. 103-110.
19. L. Thai and N. Tam., "Fingerprint Recognition using Standardized Fingerprint model", IJCSI International Journal of Computer Science Issues, vol. 7, No.7, 2010, pp. 11- 17.
20. M. Rahman, M. Ali and G. Sorwar., "Finding Significant points for Parametric Curve Generation Technique", Journal of Advanced Computations, 2008, vol. 2, no. 2, pp. 107-116.
21. F. Gerald & H. Dianne (2000). The Essentials of CAGD. Natic, MA: A K Peters, Ltd. ISBN 1-56881-123-3.