Fingerprints Recognition System-Based on Mobile Device Identification Using Circular String Pattern Matching Techniques

Miznah H. Alshammary(✉), Costas S. Iliopoulos, and Mujibur R. Khan

Department of Informatics, King's College London, London, UK
{miznah.alshammary, c.iliopoulos, md.khan}@kcl.ac.uk

Abstract. As fingerprint recognition systems have become increasingly adopted within a range of technology applications over the last decade, so too has their attention within emerging research. However, although this increased attention has led to an enhancement of the software and algorithms behind this recognition process, the majority of research has still not addressed the issues of incorrect rotation or proximity between the finger and the device. Current systems assume that the direction of the imprinted finger will align with that of the target fingerprint image; this decreases the accuracy of fingerprint recognition across a variety of finger orientations and scenarios.

In response to this use-case dilemma, this paper proposes a new technique of pattern matching that can account for this natural range of fingerprint orientations. This is achieved first through a preliminary stage of orientation identification, whereby the fingerprint image can be stored under multiple permutations by using approximate circular string-matching algorithms.

This enables the database of images for each approximate permutation of orientation to be stored in advance. It can then be matched against the strong information of the fingerprint at its exact relative rotation of input. The improved accuracy of recognition demonstrated through the results of this study may enable the functionality of fingerprint recognition to adapt to challenging device form-factors and provide the accuracy needed for military and medical applications.

Keywords: Mobile device fingerprinting · Fingerprint recognition · Rotation · Circular string matching

1 Introduction

With modern devices becoming increasingly configured for the high-speed transmission and sharing of data across mobile networks [6], this in turn has led the volume of information shared with unknown entities to rise. This has made the sharing of personal data through recognition systems a fundamental concern and is consequently pressuring manufacturers to increase recognition accuracy.

With precursors to fingerprint recognition dating back to the 19th century, the conceptual functionality of this technology has long since become attractive for

Published by Springer Nature Switzerland AG 2020
I. Maglogiannis et al. (Eds.): AIAI 2020 Workshops, IFIP AICT 585, pp. 225–231, 2020.
https://doi.org/10.1007/978-3-030-49190-1_20

tracking and managing criminal records. However, as demonstrated by the case of criminal records in Argentina [5], it also improves forensics and the effectiveness of local police. Furthermore, the unique profile of minutiae and ridge patterns that vary in fingerprints from one individual to the other have long made fingerprint recognition a highly accurate method of confirming and tracking identity. Moreover, past analogue methods – and even contemporary methods in the algorithm format – have refined the capacity to target this static structure, and then compare the imprint against past image records [4].

However, as this functionality has emerged as an authentication method for a range of computing devices whereby the rotation and direction of imprints has varied dramatically to match the mobile form-factor, this removes the traditional security advantage of fixed devices. In turn, this compromises the accuracy of fingerprint recognition in devices that require flexible finger orientation, many of which are used today for crucial functions such as communication and financial transactions.

Building on this dilemma, this study will explore past algorithms that have in part addressed this issue of relative proximity. It will then highlight contributions from the results which may constitute a recognition system that is prepared for these modern scenarios.

2 Literature Review

With traditional systems primarily focused on comparing the topographical similarity between an imprint and past stored images from the same user, past research has developed increasingly accurate systems of modelling the static structure of fingerprint minutiae. More recent models adopt a 3D structure that can account for pressure and depth [7–10].

However, although the contemporary recognition systems for minutiae-based matching enable more flexible user scenarios by predicting how the minutiae structure may morph in response to varying levels of pressure [11], this approach to matching still assumes that the finger is placed vertically. It is thus less accurate for devices held in a variety of orientations. Furthermore, in addition to lacking this flexibility for fingerprints other than in the vertical orientation, past methods also appear to experience reduced accuracy in a number of applications. These include comparing biological systems; the recovery of information and correlation of data from different files; and signal processing. The most significant algorithms explored in this study are the Landauand-Vishkin and the Needleman-Wunsch algorithms [3], both of which are incorporated into our proposed solution.

3 Our Contribution

Building on the lack of methods for accounting for orientation in previous studies, this paper proposes a pattern-matching process for mobile fingerprint authentication that uses classification to match fingerprint profiles. This is achieved by deriving the information of the minutiae by intercepting the fingerprint with a series of scan circles. This information is then translated into a string.

Following this initial stage, the string information of the fingerprint is then matched in a local image database. This matching process is conducted by a pattern-matching algorithm that is tolerant to topographical error and it thus enables the identification process of the minutiae to be done in linear time, according to the combined length of all the searched strings.

Furthermore, as this finger identification method leverages the efficiency of circular string matching, the accuracy of our identification method could be separated from the topographical similarity of the placed fingerprint relative to the vertical orientation. In addition, as our method utilised approximation rather than exact matching of the minutiae features, the distortion of the input image was significantly decreased.

However, in addition to the advantage of targeting circular strings rather than static features of the minutiae bound to a vertical orientation, our process also incorporated minutiae extraction, whereby the speed of the overall recognition process would be significantly impacted by this extraction speed. Moreover, in order to maximise the speed of extraction, we have aimed to improve the performance of minutiae extraction so that the overall execution time of the process is reduced.

Our developed alternative of Fast Minutiae Extraction saw a significant speed increase over Novel Minutiae Extraction-2 presented in [2] and Novel Minutiae Extraction presented in [1].

In addition, the series of scan circles, precise location and rotation of the input fingerprint captured by our method can be translated into a string. This information can then be stored as images within the fingerprint database for later matching. Moreover, this was achieved by using rapid circular string-matching algorithms, which enabled the total lengths of the strings to be searched in linear time, accelerating and enhancing the accuracy of the recognition process.

In this paper, our proposed algorithm not only aims to enhance minutiae extraction to improve the accuracy and flexibility of the fingerprint recognition process. It also reduces the overall time of execution. In order to outline our process, the following sub-sections detail the proposed algorithm in its two underlying components: Fast Novel Minutiae Extraction and our algorithm outline.

3.1 Fast Novel Minutiae Extraction

As our process repeatedly introduced minutiae extractions, this meant that the speed of this extraction process would significantly impact the overall time required for successful fingerprint recognition. Consequently, this project developed an alternative extraction method we refer to as Fast Novel Minutiae Extraction.

This alternative saw a significant speed increase over Novel Minutiae Extraction-2 presented in [2] and Novel Minutiae Extraction presented in [1].

The outline of Fast Novel Minutiae Extraction is presented below:

```
ALGORITHM Fast_Minutae_Extraction(char[][] img, int r, int cx, int cy)
// INPUT: "img", a 2d char array representing a fingerprint
// INPUT: "r", the radius of the circular string to be extracted
// INPUT: (cx, cy), the centre of the circular string to be extracted
// OUTPUT: "pattern", a circular binary string
{
        string topLeft;
        string topRight;
        string bottomRight;
        string bottomLeft;

        for(int i = (cx -r); i < cx; i++)
        {
                double dJ = Math.Sqrt(Math.Pow(r,2)  - Math.Pow((i-cx), 2));
                if(dJ == (int)dJ)
                {
                        int xOffset = cx -i;
                        int yOffset = (int)dJ;

                        if(pixel at img[cx - xOffset][cy + yOffset] < 125) {
                                Append "0" to the end of topLeft;
                        } else {
                                Append "1" to the end of topLeft;
                        }

                        if(pixel at img[cx + yOffset][cy + xOffset] < 125) {
                                Append "0" to the end of topRight;
                        } else {
                                Append "1" to the end of topRight;
                        }

                        if(pixel at img[cx + xOffset][cy - yOffset] < 125) {
                                Append "0" to the end of bottomRight;
                        } else {
                                Append "1" to the end of bottomRight;
                        }

                        if(pixel at img[cx - yOffset][cy - xOffset] < 125) {
                                Append "0" to the end of bottomLeft;
                        } else {
                                Append "1" to the end of bottomLeft;
                        }
                }
        }
        return topLeft + topRigt + bottomRight + bottomLeft;
}
```

3.2 Outline of the Algorithm

Our algorithm was structured to identify the circular strings from the saved fingerprint and input at the longest length: asmf/acdm was used to check that the relative rotation of the circular string of the input matched that of the saved fingerprint image. Furthermore, the Needleman-Wunsch approximation algorithm was used to check the exactness of the alignment.

The main component that dominates the speed of execution is the extraction of minutiae from the image and constructing the circular string, which is repeatedly invoked throughout the whole process.

We have outlined the complete process of Fingerprint matching as below:

1. **Database Formation:** A database of circular strings was constructed, with the centre of the image classed as the effective centre.
2. **Concentric Circle-to-Strings Formation:** Concentric circles were drawn, and then (using Fast Novel Minutiae Extraction) converted into strings comprising of 0's and 1's. For example, assuming that Cir is the circular string of the i^{th} image with a radius of r, all rotations of each of the circular strings are saved to the database. By letting DbRir be the collection of rotated strings from Cir; a Cir = "0111" would mean that DRir would hold values of "0111", "1011", "1101", "1110". Conversely, by letting DbRir be the collection of rotated strings from Cir; Cir = "0111" would mean that DRir would hold values of be "0111", "1011", "1101", "1110".
3. **Binary String Formation:** After the previous phase, Fast Novel Minutiae Extraction was repeated in order to construct circular binary strings for each circle.
4. **CsR List Formation:** After binary string formation, the process then proceeded to create a sorted CsR list of radius pairs and circular strings (<CStr, Radius>) in a descending order of the CStr length.
5. **Rotation Comparison:** Following list formation, the asmf/acdmf method was then applied along with the Needleman-Wunsch algorithm. This enabled the rotations to be compared, whereby a percentage of matched strings recorded as higher than a certain threshold (tS) would result in the sum of the match percentage being saved. The i^{th} image was marked as the 'candidate image' and then the (i + 1) th image was selected until all images within the databased had been utilised. Following this process, this stage could be repeated after choosing a different centre point.
6. **Image Matching:** After the previous stages, the process finally returns to the match images for verification.

4 Experimental Results

We coded the program in C# using an Intel Pentium processor (CPU 4415U) with 2.30 GHz, the random-access memory was 4.00 GB, the Operating System was 64-bit Windows 10, and the platform was MS.Net Framework 4.5.

We aimed to improve the performance of minutiae extraction so that the overall execution time of the process is reduced.

Comparing the time required in milliseconds for each method, it is clear that our solution algorithm provides better results for the overall execution time of extracting minutiae from the image and constructing the circular string, which is repeatedly invoked throughout the whole process. This is presented in Table 1 and Fig. 1.

Table 1. Summary of findings

Database size	Time required for novel methods (Milliseconds)	Time required for fast methods (Milliseconds)
50	881300	67765
100	3276930	1290428

Table 1 shows that there are two main factors that ensure the accuracy and performance of the proposed algorithm. These are the size of the database and the time required for extracting minute details from an image and creating a circular chain that is invoked repeatedly throughout the entire process.

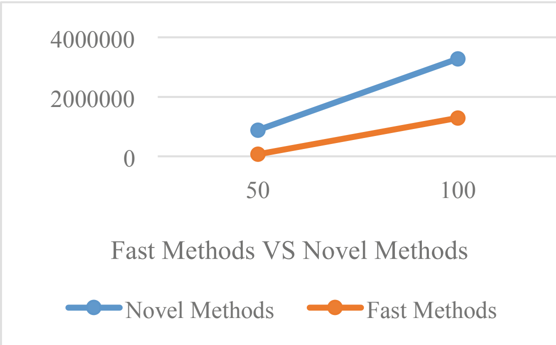

Fig. 1. Fast methods vs novel methods

Table 1 shows the results for two different database sizes. For all inputs, we captured the time required for each method and the results are compared in Fig. 1.

It is clear that our solution algorithm provides better results in terms of overall execution time and as the database size increases, it decreases the time required for extracting minutiae from the image and constructing the circular string, which is repeatedly invoked throughout the whole process.

5 Conclusion

With an increasing volume of devices adopting fingerprint recognition, this will pressurise manufacturers to re-evaluate existing minutiae-matching techniques and to explore enhanced models that are better adapted to the non-vertical orientations of input that are becoming prevalent among mobile devices.

Consequently, in response to the recent increase in algorithms and string-matching techniques, this paper explored this barrier of orientation flexibility, and how our proposed solution may lead to faster mobile user scenarios. As demonstrated by the results, our proposed solution offers an enhanced version of previous algorithms used for extracting topographical features of minutiae from the input image. This increases the overall efficiency of the identification process.

Overall, this study stands as an example of how mathematical algorithms can be used to enhance the functionality of a computing-based function in order to more accurately model a device's surrounding environment. Furthermore, in addition to accounting for a more realistic variety of user orientations in fingerprint recognition, this proposed system may also be able to enhance models beyond bioinformatics, which face a similar rise in device mobility and range of handheld orientation.

References

1. Ajala, O., Aljamea, M., Alzamel, M., Iliopoulos, C.S., Strigini, Y.: Fast fingerprint recognition using circular string pattern matching techniques. In: PATTERNS 2016: The Eighth International Conferences on Pervasive Patterns and Applications (2016)
2. Ajala, O.I., Iliopoulos, C.S., Khan, M.R.: Identification of fingerprints using circular string approximation for mobile devices. In: PATTERNS 2016, Computer Science and Information Systems. ACSIS, vol. 13, pp. 193–197 (2016). https://doi.org/10.15439/2017f3. ISSN 2300-5963
3. Landau, G.M., Vishkin, U.: Fast string matching with k differences*. J. Comput. Syst. Sci. **37**, 63–78 (1998)
4. Sebastian, S.: Literature survey on automated person identification techniques. Inter. J. Comput. Sci. Mobile Comput. **2**(5), 232–237 (2013)
5. Unar, J., Seng, W.C., Abbasi, A.: A review of biometric technology along with trends and prospects. Pattern Recogn. **47**(8), 2673–2688 (2014)
6. Alshammary, M.H., Alanezi, F.A.: A review of recent developments in NOMA and SCMA schemes for 5G technology. In: IEEE Communications, October 2017
7. Tan, X., Bhanu, B.: Fingerprint matching by genetic algorithms. Pattern Recogn. **39**(3), 465–477 (2006)
8. Jain, A.K., Hong, L., Pankanti, S., Bolle, R.: An identity-authentication system using fingerprints. Proc. IEEE **85**(9), 1365–1388 (1997)
9. Kovacs-Vajna, Z.M.: A fingerprint verification system based on triangular matching and dynamic time warping. IEEE Trans. Pattern Anal. Mach. Intel. **22**(11), 1266–1276 (2000)
10. Tan, X., Bhanu, B.: Robust fingerprint identification. In: International Conference on Image Processing, vol. 1, pp. 1–277. IEEE (2002)
11. Kai, C., Xin, Y., Xinjian, C., Yali, Z., Jimin, L., Jie, T.: A novel ant colony optimization algorithm for large-distorted fingerprint matching. Pattern Recogn. **45**(1), 151–161 (2012)