

Palm Print Identification using Centered Discretization Hashing Technique

Guided by Prof. S. A. Saoji
Department of Computer, University of Pune

Surabhi Mundada
Computer Engineering
PVG's COET
Pune, India

Pooja Wankhade
Computer Engineering
PVG's COET
Pune, India

Shital Kolte
Computer Engineering
PVG's COET
Pune, India

Shweta Konde
Computer Engineering
PVG's COET
Pune, India

Abstract- Palmprint is the unique identity of any person or human being. As it is unique identity it helps to prevent the deceptive crimes. We will be implementing hashing based palmprint identification i.e. we using hashing technique to search for the palmprint templates in the database, as hashing is one of the fastest technique used for searching in database. The present two hashing based technique for fast palmprint identification are principal orientation pattern (POP) and Orientation pattern (OP). We introduce Centered Discretization, a simpler discretization method; It allows for smaller tolerance regions without impacting the usability of the system. We then introduce our fast identification method based on Centred Discretization. Discretization is used in click-based graphical passwords but in our project we will be using this so that approximately correct control points (CP) can be accepted and stored generating a hash value from the palmprint template.

Keywords – Hashing, Palmprint Identification, Centered Discretization, CP.

1.INTRODUCTION

The rapid growth in the use of e-commerce applications requires reliable and automatic personal identification for effective security control. As an emerging biometric technology palmprint recognition is considerably interesting for research in recent years. Palmprint recognition system may operate in either of two mode verification and Identification. In verification mode user must provide an identifier (Eg I-Card) before the palm prints are compared with the stored templates whereas in identification mode, palmprints are automatically compared by the system with the stored templates. Thus identification is more appealing and is expected to have a broad range of applications. There is another approach, the knowledge-based approach, where the

user has to remember its password, here in this approach the password can be forgotten or can be guessed. Fingerprint-based personal identification has drawn considerable attention over the last few years. However, workers and elderly may not provide clear fingerprint because of their problematic skins and physical work. Recently, voice, face and iris-based verifications have been studied extensively [1]. As a result, many biometric systems for commercial applications have been successfully developed.

Nevertheless, limited work has been reported on palmprint identification and verification, despite the importance of palmprint features [1]. In our identification process you don't need to show any identity, it automatically identifies the person by comparing palmprint with all the existing template. There are many unique features in a palmprint image that can be used for personal identification.

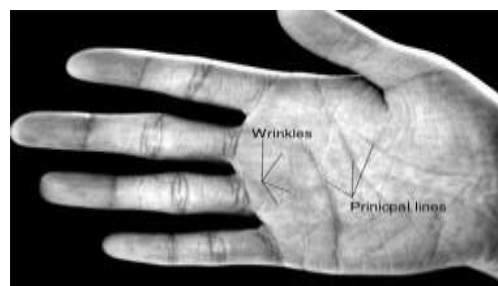


Fig. 1

Principal lines, wrinkles, ridges, minutiae points, singular points and texture are regarded as useful features for palmprint representation. Features like principal lines and wrinkles, which are defined in Fig. 1, can be obtained from a low-resolution palmprint image [6].

In general, high-resolution images are essential for some applications such as law enforcement, where ridges, singular points and minutiae points are extracted and matched in latent prints for identification and verification. But there

are few challenges in identification system like the average identification time is equal to size of the database and the other challenge is as the no of registered subject increases the identification accuracy decreases. Various features from the palm print images can be extracted by using image processing technique.

Early researches on fast palmprint identification can be roughly classified into two categories, hierarchical matching and palmprint classification. Hierarchical matching approaches typically involve first extracting multiple kinds of features and then searching in a layered fashion. Simpler features which can be quickly extracted and matched are used at higher layers because they allow a large number of candidates to be discarded.

The drawback is that the templates discarded at higher layers may contain the target. Classification strategies often make use of expert knowledge to design the classification rules. They proceed by dividing palmprints into several classes and matching the query only with the templates in its class. The drawback here is that the initial classification may have put the query and its target template into different classes, making a successful match impossible. Therefore, while both strategies speed up the identification process, they do so at the expense of accuracy.

To overcome this problem hashing based fast palmprint identification methods were introduced. First we will see the existing hashing based fast palm print identification system which is OP and POP[1]. We then introduce centered discretization which is another hashing based fast palmprint identification technique[6], this was used in graphical password identification but is also suitable for hashing based fast palm print identification[6]. Centered Discretization, offers usability and security improvements. To evaluate the effectiveness of our methods, we will carry out a number of experiments on several palmprint databases. The results will show that our centered discretization method will be faster and the identification accuracy is comparable or slightly better.

2. PREPROCESSING TECHNIQUES

Process 1: Gray scaling

Gray scaling is a process in which coloured image is converted into a black and white image. The average of the each pixels red, green, blue value is taken as the value of gray scaling i.e. $(R + G + B) / 3$.

Process 2: Histogram Equalization

Histogram equalization is a technique for adjusting image intensities to enhance contrast. Histogram is a graphical representation of the distribution of data.

Process 3: Thresholding

Thresholding is an image processing technique for converting a gray scale or colour image to a binary image based upon a threshold value. If a pixel in the image has intensity value less than the threshold value, the corresponding pixel in the resultant image is set to black, otherwise pixel is set to white. Thus, creating a binarized image, or an image with only 2 colours, black (0) and white (255). Image thresholding is very useful for keeping the

significant part of an image and getting rid of the unimportant part or noise. This holds true under the assumption that a reasonable threshold value is chosen.

Process 4: Thinning

Thinning is a morphological operation that is used to remove selected foreground pixels from binary images. This technique is used to create image thin by using thinning algorithm. In our system the common use of thinning is to reduce the threshold output of a principal line, to lines of a single pixel thickness, while preserving the full length of those lines (*i.e.* pixels at the extreme ends of lines should not be affected)

3. HASHING TECHNIQUE

Previous studies reveal that hashing is very effective for fast search of competitive-code-like high-dimensional fuzzy data. Hash Function practical use is a data structure called a hash table where the data is stored associatively. Searching linearly for a person's name in a list becomes cumbersome as the length of the list increases, but instead a hashed value can be used to store a reference to the original data and retrieve constant time. Hash functions are primarily used in hash tables, to quickly locate a data record (e.g., a dictionary definition) if its search key is given. Briefly, the hash function is used to map the search key to an index, then the index gives the place in the hash table where the corresponding record should be stored. Each slot of a hash table is associated with (implicitly or explicitly) a set of records, rather than a single record and thus each slot of a hash table is often called a bucket, and hash values are also called bucket indexes. Each bucket consists of fixed number of records.

Precisely, a hash function is utilized to partition a large amount of data into a number of small buckets, and indexes each buckets by a hash value (usually an integer). When a query is generated, the hash value is calculated first using the hash function, and then the answer are found only within the bucket indexed by the hash value. On one hand, it is obvious that if the size of each bucket is much smaller than that of the original dataset, the search efficiency would be greatly improved. On the other hand, if the bucket does not contain the target, the search would fail. Thus the design of hash function is very crucial for hashing based fast and accurate search algorithms. Also, the form of the hash function should be as simple as possible so that the hash value can be computed with low computational cost. A hash function may map different data to the same hash value. This is called a collision [1]. As for the palmprint identification task, the collisions of samples captured from different subjects (imposter collisions) should be minimized to improve search efficiency. At the same time, we should ensure the collisions of samples captured from the same subject (genuine collisions) to reduce accuracy loss. Thus a hash function is expected to be:

- 1) Discriminant between different subjects to minimize the occurrence of imposter collisions;
- 2) Consistent within each subject to ensure the occurrence of genuine collisions;

3) Random across all the subjects to make the partitions as even as possible.[1]

4. PREVIOUS TECHNIQUE USING HASHING

4.1. Orientation Pattern Hashing:

We define the orientation pattern (OP) as a collection of orientation features. D orientation features ($D=1024$ here), i.e., $OF = \{of_i \in \{1,2,3 \dots, 6\}\}$, $i=1,2, \dots, D$. If an OP is defined at a collection of K positions $P = \{p_j \in \{1,2, \dots, D\}\}$, $j=1,2, \dots, K$, then the value of OP is $V = \{of_{p_j}\}$. The parameter K is the size of an OP, and the parameter P determines its position. Two OPs are equal if and only if their positions and values are the same. The orientation features are discriminant across different subjects and consistent within one subject thus OP has the properties 1) and 2) of a hash function. However, orientation features have strong local correlation. The OP hashing method consists of two stages, pre-processing and searching. In pre-processing stage, the templates are indexed in the database according to their orientation features, and a hash table is built to facilitate fast searching. Also an offset table and a size table is built to record the offset and the size of each bucket in the hash table. The computational cost of OP construction is very low, which makes OP very appropriate for hashing based fast palmprint identification.[1]

4.2. Principal Orientation Pattern

Principal orientation pattern (POP) is defined as the OP whose orientation features are located in the regions corresponding to principal lines in the palmprint image. Thus POP are viewed as a special case of OP. The orientation features of POP are called principal orientation features. The principal lines are more evident and stable features in palmprint images, their orientation features are considered to be more consistent than those in other regions. Thus, the POPs are more likely to have genuine collisions. It is very difficult to detect the principal lines accurately and consistently across all the principal images. Since palm line is a kind of dark line, after the convolution of the palmprint image with Gabor filters, the minimum response is regarded as an indicator of principal lines. It is observed that most of them locate in the regions corresponding to principal lines. Therefore, they use filter response to find principal orientation features. The POP hashing method also consists of pre-processing and searching stages. The algorithms for pre-processing and checking collisions are the same as those of OP hashing. As for the searching algorithm, the only difference is that POP hashing requires extracting both orientation features and their responses.

5. PROPOSED SYSTEM

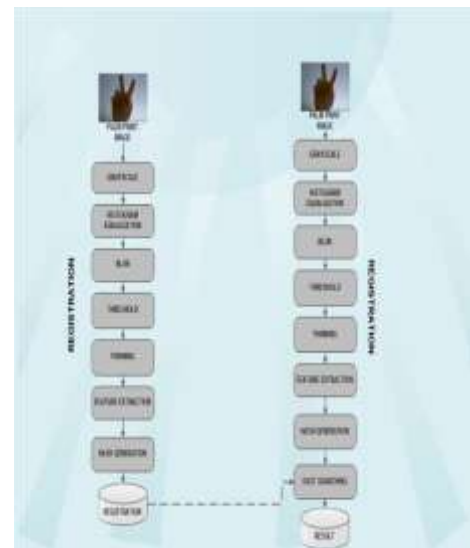


Fig. 2

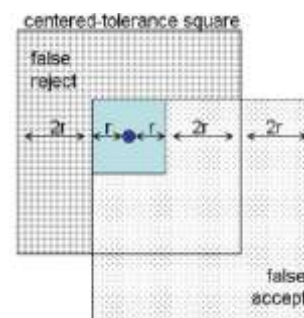


Fig. 3

Figure 3: The small circle is the original control-point. The centered-tolerance square is the evenly distributed tolerance likely expected for forming the buckets. The non-overlapping region of the centered tolerance square is the area where false rejects would occur in Discretization, while the non-overlapping region of the Discretization square indicates false accepts in Discretization.

In the Centered Discretization technique we will be considering the major and minor control points. The major control points will be the strong CP's i.e. the joints of the principal lines. Thus control points will be defined as the point whose neighbourhood has more than one point in its diagonal direction then they will form a join and thus it will also be considered as strong CP's. When the image undergoes through the entire image processing stages, its strongest control point is considered and with respect to this point the bucket is formed assuming r -radius from this CP, with this bucket the other buckets will be formed with same radius r . We not dividing the palmprint image into buckets but rather creating buckets with respect to the strongest control point on the principal axis of the palmprint, thus this is known as centered discretization. Here, it can be noted that if the value of r is assumed thus if r is increased the buckets formed will be less, but if r is less then the buckets formed will be more. The principal lines are shown in Fig 4 and Fig 5. [3]

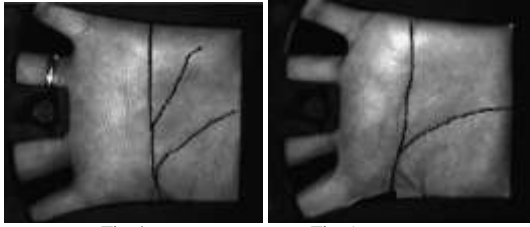


Fig 4

Fig 5

Consider a 1-dimensional line, L , with a continuous set of control points. A strong CP is considered on the principal axis and is represented by a real number χ . Our initial objective is to discretize this principal axis into equal buckets with respect to χ , where χ falls in the centre, and this ensures an even tolerance on both sides of χ . A tolerance r is selected based on system or user preferences. Each bucket is of length $2r$ as shown in Fig 6. χ is centred in its bucket. The offset is represented by parameter d . The number of points which shift from their original position are determined while comparing CP's of two templates and will be calculated using offset d .

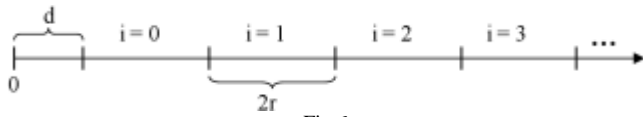


Fig 6

First assume that a principal line on the palm print image consists of a single control point χ having (x,y) co-ordinates. To store this image, we must discretize the point by calculating its offset d (where $0 \leq d < 2r$) and its corresponding segment identifier i (where $i \geq -1$, with $i = -1$ occurring if χ is within r of the origin). Offset d is stored in the clear, while i is stored in protected form as its hash value $h(i, d)$. The offset d is included in the hash to uniquely identify the bucket. The system must also be aware of tolerance r that specifies the acceptable inaccuracy during image re-entry. The bucket identifier i is computed for x co-ordinates as $i_x = [(x - r)/2r]$ and similarly for y co-ordinate as $i_y = [(y - r)/2r]$ identifying the bucket containing χ . The offset $d_x = (x - r) \bmod 2r$ and similarly $d_y = (y - r) \bmod 2r$ determines the distance between the origin and the left boundary of bucket.

To verify if a re-entered image's control point χ' is acceptable, the system computes $i'_x = [(x' - d)/2r]$ and $i'_y = [(y' - d)/2r]$. This calculates which bucket contains χ' using the same offset as the original point. Note that χ' is not necessarily centered within its bucket; we are simply calculating which bucket contains χ' based on χ 's pre-determined buckets. If χ' is within tolerance r of χ then $i' = i$ and hence $h(i', d)$ equals the stored value of $h(i, d)$ and system accepts the entry. If χ' is outside of the accepted tolerance r , it falls in a different bucket and $i' \neq i$, thus $h(i', d) \neq h(i, d)$ and the system rejects it.

For example, assume $(x,y)=(15,13)$ and $r = 5.5$. We compute $i_x = [(x - r)/2r] = [(15 - 5.5)/11] = 0$ and $d_x = (x - r) \bmod 2r = (15 - 5.5) \bmod 11 = 9.5$ similarly $i_y = [(y - r)/2r] = [(13 - 5.5)/11] = 0$ and $d_y = (y - r) \bmod 2r = (13 - 5.5) \bmod 11 = 7.5$. Offset $d = 7.5$ is stored in the clear along with protected

$h(i_x, d_x) = h(0, 9.5)$ and $h(i_y, d_y) = h(0, 7.5)$. If a user enters $(x', y') = (10, 14)$ during login, the system calculates $i'_x = [(x' - d)/2r] = [(10 - 9.5)/11] = 0$ and $i'_y = [(y' - d)/2r] = [(14 - 7.5)/11] = 0$. It then compares $h(i'_x, d_x)$ and $h(i_x, d_x)$ similarly $h(i'_y, d_y)$ and $h(i_y, d_y)$, and the CP is accepted since they match.

Thus if there are more than one control points, we will take them as $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$ then it is calculated and matched in the same way.

6. CONCLUSION

In this paper we propose the centered discretization method for the fast identification of the palmprint image on the large scale databases. Centered discretization is one of the hashing techniques. We have given the definition and detailed information about the centered discretization. Till now no work has been done using centered discretization hashing technique in palm print identification. The centered discretization reduces the number of comparisons of buckets, thus the time required will be less to identify the person. The computational cost is also low for the construction of centered discretization and also it offers usability and security improvements. By comparing centered discretization with Orientation pattern (OP) and Principal orientation pattern (POP) hashing technique the centered discretization is slightly more accurate. And comparing with other techniques like brute force search for the fast palmprint identification approaches demonstrate the speed advantage of our method. The qualities of our system are shown in Fig 7[3]. Hence by discussing all the above techniques we conclude that our method is very promising for large scale database in palmprint identification system.

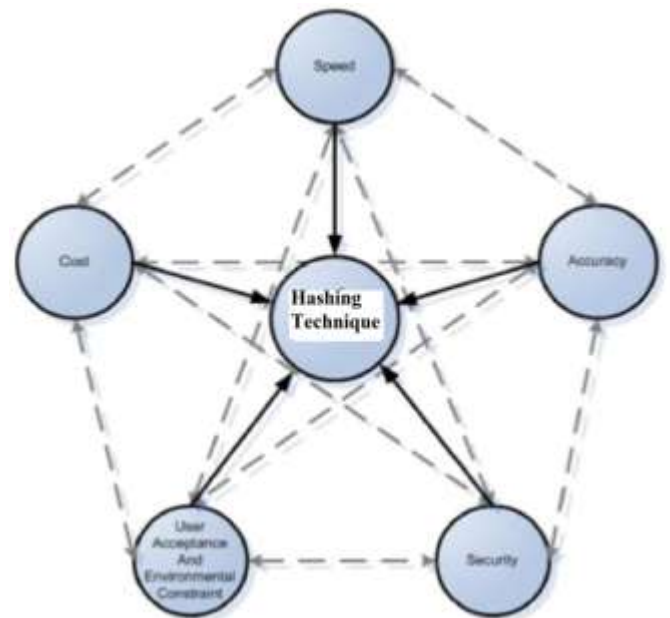


Fig 7

7. ACKNOWLEDGEMENT

We sincerely thank our guide Prof. Mr. S.A. Saoji for his support and valuable guidance. We even like to thank Prof. Mr. G.V. Garje (Head of Computer Department at PVG's COET) for all the facilities that were provided.

8. REFERENCES

1. Feng Yue, Bin Li, Ming Yu, and Jiaqiang Wang. "Hashing Based Fast Palmprint Identification for Large-scale Database" .
2. Hafiz Imtiaz & Shaikh Anwarul Fattah "A Spectral Domain Dominant Feature Extraction Algorithm for Palm-print Recognition" .
3. Adams Konga, DavidZhangb, MohamedKamelc "A survey of palmprin trecognition" .
4. Alberto Martin and Sabri Tosunoglu "Image processing Technique use for machine" .
5. Sonia Chiasson, Jayakumar Srinivasan, Robert Biddle, P. C. van, "Centered Discretization with Application to Graphical Passwords"
6. David Zhang, Wai-Kin Kong, Jane You, and Michael Wong "On-Line Palmprint Identification"

IJERT