JurnalSarjanaInstitutTeknologi Bandung bidangTeknikElektrodanInformatika
Vol. 1, No. 3 Oktober 2012

# Simple Hashing for Core Point and Minutiae Based Fingerprint Identification

S.H. Utami#1, Gede Indrawan#2, Benhard Sitohang#3

*#School of Electrical Engineering and Informatics, Institute Technology of Bandung,*
*10th Ganeca Street, Bandung, Indonesia.*
1SerriHUtami@gmail.com
2gede.indrawan@gmail.com
3benhard@stei.itb.ac.id

*Abstract*— **There are many existing direct access method for fingerprint identification. A simple direct accessing method design to observe reducing number of matching that can be performed. With this simple hashing, we expect not only number of matching will be reduced, but also reduce complexity of hash function. This method design based minutiae, as local level component, and core point, which is global level component as reference point. The result is, there will always be a collision in this method, but as long as stored fingerprint distribution good, performance, speed and accuracy, will be good either.**

*Keywords*— **Fingerprint identification, direct access, simple hashing, reduce number of matching, reduce number of complexity, minutiae, core point, distribution, performance.**

## I. Introduction

Fingerprint is one of biometric elements that are widely used. Fingerprint has been used for systems which provide community service, e.g. e-KTP (Indonesia), IAFIS (FBI, America), CCRTIS (Canada), etc. That systemwould need very large storage to store fingerprints data. In this case, sequential based identification method requires system to perform matching for every single fingerprint in database.

Fingerprint matching is a complex mechanism that involves a lot of components and uses some complex procedure in computation.Sequential method which complexity is O(n) is not reliable anymore for identification in large scale fingerprint storage.In this case, there is a need to design a direct access method to reduce the number of matching performed.In this research, we design a simple way for fingerprint mapping.

There are some steps in fingerprint recognition, which include image processing, file organization, and matching. Focus of this research is file organization, which included fingerprint distribution and some factor that affect fingerprint distribution. Fingerprint matching only used for support designed access method.

In the second section, there is some related work to this research, included fingerprint components, fingerprint matching, and previous fingerprint identification method. Design of hashing method is elaborated in third section. Testing steps, testing results, and results analysis is written in fourth section and fifth section. In the last section, there is the conclusion of this research.

## II. Related Work

Some correlated theory is needed to design hashing for fingerprint identification, included direct access method, fingerprint components, and existing direct access methods.

### *Direct Access Method*

Direct access method is an alternative for inefficient problem that perform in sequential method. In identification, sequential search all object to get some information which related to input, which is not efficient for system [1]. Direct access has some clue for location of retrieved information which means there is no need to search every single object in storage anymore. Direct access complexity in identification is up to O(x)[2].

There is some type of direct access method, but hashing, indexing and direct addressing are the popular one.Direct access can be grouped into unique value and the type which can have similar value. Indexing and direct addressing are the type that can be grouped into unique value. This type hasunique value for every single input. But, in hashing two in put or more can mapping into similar value (collision). This design tolerance the collision of data which make hashing decided to be designed access method. Hashing store data into some bucket based on hash value which calculate using a function called hash function.

### *Fingerprint Components*

Proposes hashing is using some component which occurred in every fingerprint. There are three levels of component in fingerprint, which is global level, local level, and very local level component[3]. Global level is the component that represents the whole fingerprint image. Singular area and core point is the two components in fingerprint. In a fingerprint, more than one singular area may be detected, but there is only (maximum)one core point detected, as shown in figure 1.

There is only one component that occurs in local level component, which called as minutiae. Definition of minutiae is an ending of a ridge pattern. Based on Galton’s classification, there are six types of minutiae, but ridge ending and bifurcation are the more often minutiae that using for fingerprint recognition.

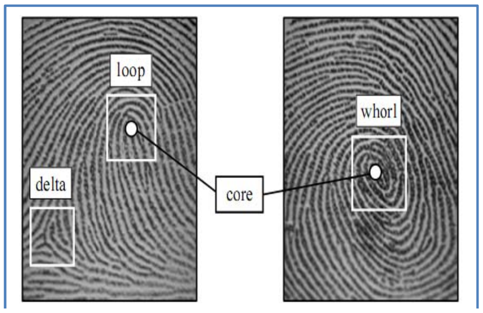

fig. 1 Global Component of Fingerprint

Very local fingerprint component can only be identifiedfrom image with quality minimum 1000 dpi.That fact makes identification method which is used this component needs large storage for fingerprint. That makes identification based on fingerprint components in this level only developed for experimental needs.

From the fact in the previous paragraph, we can conclude that we will not design hashing based on very local component, so the candidates only global level and local level component. But, hashing based global level do not has significant access time difference to sequential method because left loop, right loop, and whorl have 30% of fingerprint population each [4].In conclusion, we will used minutiae as the main components that is used in designed hashing.

*Existing Fingerprint Identification Method Using Direct Access*

There is some research which design a direct access method for fingerprint identification. One of them is triplet based indexing method proposed by BirBhanu and Xuejun Tan [5]. That method is using three minutiae for calculating index value for fingerprint. Feature used in indexing isangle, direction, minutiae type, and maximum side from triplet. The result shown that, proposed method has good performance as the effect of reducing number of matching candidate.

In the other hand, there is another indexing method proposed by JianjiangFeng and AnniCai[6]. Proposed Indexing based on minutiae and ridges around minutiae. There isa lot of information that contained in this method which makesaccuracy very well, but this effect bad access time for identification.

## III. Hashing for Core Point and Minutiae Based Fingerprint Identification

In this previous section, explained about existing direct access methods. There is a lot of method which is used complicated way for fingerprint mapping. In this research, we want to apply a simple approach for fingerprint mapping, so we can observe numbers of matching reduction using this approach. Fingerprint distribution will be the most important thing to observe for predicting reduction number.

As shown in figure 2, for storing fingerprint, system must receive information about used component from extraction process. After that, hash value calculating to decide destination bucket for storage. Hash value calculated using selected components, but every single components extracted is stored into destination bucket.

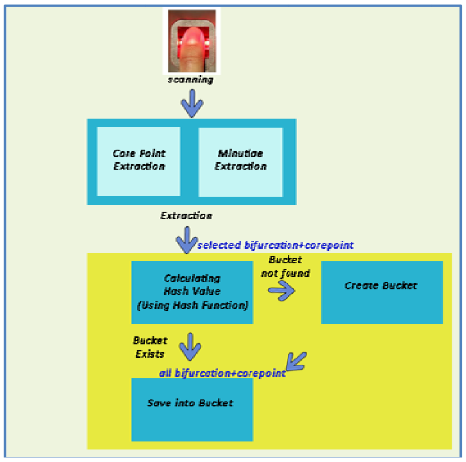

fig. 2 Fingerprint Storing Method

Figure 3 show that, there are two different mechanism in designed fingerprint identification method. First mechanism is hashing mechanism which using only selected components to calculate hash value for decide destination bucket. If destination bucket exists, second mechanism, which is matching mechanism, is used for searching every single fingerprint in destination bucket. Matching mechanism need more detail information, so this mechanism not only using selected components, but also every single component extracted. If fingerprint found in the bucket, then information about fingerprint and fingerprint owner is shown.

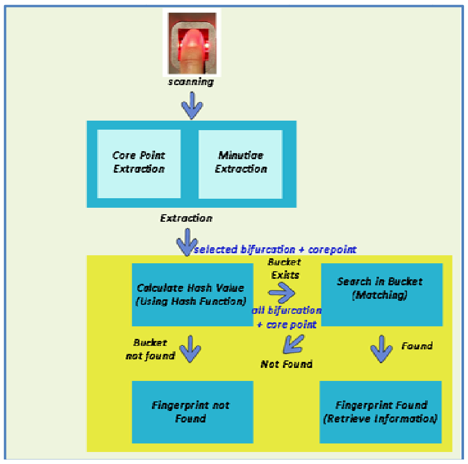
scanning
Core Point Extraction
Minutiae Extraction
Extraction
selected bifurcation + corepoint
Calculate Hash Value (Using Hash Function)
Bucket Exists
all bifurcation + core point
Search in Bucket (Matching)
Bucket not found
Not Found
Found
Fingerprint not Found
Fingerprint Found (Retrieve Information)

fig. 3Fingerprint Identification Method

In second section written that the main fingerprint component used for design hash function is minutiae. But, for tolerance translation and rotation a reference point needed. Core point decided to be reference point for fingerprint, which make design hashing based on two components, minutiae and core point. Figure 4 showillustration of designed hash function.

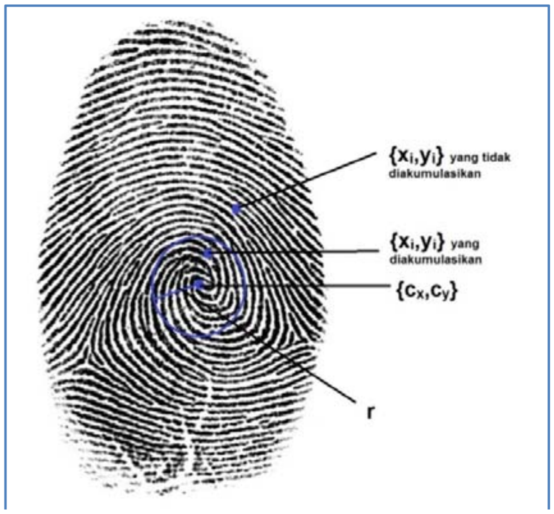
{xi,yi} yang tidak diakumulasikan
{xi,yi} yang diakumulasikan
{cx,cy}
r

fig. 4Ilustration for Simple Hash Function

For calculate hash function core point and minutiae position is needed. Area with radius r from core point is marked as function area f. Then, the distance from core point c to every single minutiae $m_i$ in area f accumulated. Designed hash function is very simple if compared to indexing methodin the second section. Hash function shown using equation 1.

$$hv = \sum_{i=1}^{m} d_i \ , untuksetiapd_i \leq r \qquad \text{(eq. 1)}$$

Matching mechanism use complete minutiae extraction result. Matching design using geometric approach, especially minutiae distance. In matching two fingerprint, fingerprint with more minutiae number become the reference F1 and the other called F2. N random minutiae selectedfa$_i$ from fb. Every fa$_i$ is compared with every minutiaefb$_i$ in F2 which is calculating using the number of simultaneous matched minutiae distance R.Find nearest minutiae from fa$_i$,cai, and nearest minutiae from fb$_i$, cb$_i$, then compare their types and distances.If pair of fa$_i$ and ca$_i$match with pair of fb$_i$ and cb$_i$, than find nearest minutiae from ca$_i$ and cb$_i$ for next comparison. Match rate R is add for every matched pair of minutiae which comparison start form fa$_i$ and fb$_i$. If match rate reach the minimum rate, R$_{min}$, than F1 and F2 match. Here is the pseudocode for matching mechanism.

```
Match(F1, F2)
if (F1.minutiaenumber>=F2.minutiaenumber)
{
  Reference = F1;
  Pair = F2
}
else
{
Reference = F2;
Pair = F1;
}

//Matching
fa = Reference[0];
While (not match and not end of Reference and
rate<minRate)
{
While (not match and not end of Pair and
rate<minRate)
{
        fb = Pair[0]
        ca = fa; cb = fb;
        if (matchtype(ra, rb) and matchtype(ca, cb)
and matchdistance(ra,rb,ca,cb))
        {
                match = true;
                ++rate;
                repeat
                {
                        ra = ca; rb= cb;
                        ca = nearestminutiae(ra);
                        cb = nearestminutiae(rb);
                        if (matchtype(ra,rb) and
matchtype(ca,cb) and matchdistance(ra,rb,ca,cb))
                        {
                                ++rate;
                                match=true;
                        }
                        else
                        {
                                match = false;
                                rate = 0;
                        }
                }
                until(notmatch and finished(fb) and
rate>=minRate)
        }
        ++fb;
}
fa = fa + addnumber;
}
```

## IV. Testing

Testing use fingerprint from FVC 2000 db1_a[7] and FVC 2002 db3_a[8] which resolution is 300x300 pixel. Minutiae extract (only bifurcation) from DatabaseAnalyzermodul in SourceAFIS[9].Core point extract from ortcorepoint[10]. For feature transformation, which used for identification data test, we make generator for transformation of an extraction result.

Designed method is tested using software which built for testing purpose, called FHIS (Fingerprint Hashing for Identification Simulator). Here is some observed object in testing:

a. Distribution of stored fingerprint and some factor that affect fingerprint distribution, included radius of hash function area and number of feature in hash value.

b. Method performance, included accuracy, which means tolerance factors, and number of matching process. Performance will be observed during identification process.

## V. Result and Analysis

### A. Distribution of Stored Fingerprint

In this subsection, analysed effect from some factors, radius value, number of variable used as hash value, and fingerprint's characteristic, to the distribution of stored fingerprint.

Table 1 shows the radius effect to fingerprint distribution. In choosing radius value, there is a need to take most minimum value possible because that will increase the tolerance for fingerprint modification (transformation) in identification input. For the first try, 10 pixel isused as radius value for hash function area. But there is a problem, a lot of fingerprints are mapped into 0 bucket, which means many fingerprints don't have any minutiae in hash value area. After the second try with radius 20 pixel do not give a good enough data distribution, an analysis is done for the test set. Data set analysis result shown that average of minimum distance minutiae to core point in fingerprint is from 30's to 40's pixel. To minimize collision in 0 bucket, then, radius 50 pixel is used and the result for fingerprint distribution is good enough. There are only 26.4% fingerprints that are saved into existing bucket.

TABLE 1
RADIUS EFFECT TO FINGERPRINT DISTRIBUTION

| **Comparison** | **Radius of Hash Function Area** | | |
|---|---|---|---|
| | **10 Pixel** | **20 Pixel** | **50 Pixel (After fingerprint position analysing)** |
| Percentage hash function area from whole fingerprint image | 0.17% | 0.7% | 4.36% |
| Fingerprint distribution | There aremanyfingerprints in bucket with hash value 0. | Less collision in 0 bucket, but the numbers are still many. | Distribution is good. Only 4 maximum fingerprint in a bucket (2% from data test), which is in 0 bucket and -1 bucket (no core point bucket) |

Hypothetically, more feature use as hash value, more buckets will be formed. In testing, we modified the hash value with added the number of minutiae in hash function area as the other feature. After do experiments for radius 10 pixel, 20 pixel, and 50 pixel, we conclude that with added number of selected minutiae as hash value feature, the number of fingerprints in 0 bucket cannot be reduced. But, with added feature, distribution in other bucket is better. The point is a long as distribution using less features is good, there is no need to add feature, because that will make hashing computation more complicated.

TABLE 2
NUMBER OF FEATURE USED IN HASH FUNCTION EFFECT TO FINGERPRINT DISTRIBUTION

| **Comparison** | **Feature in Hash Function** | |
|---|---|---|
| | **Σ Distance** | **Σ Distance and Number of Selected Minutiae** |
| Total fingerprints are saved into existing bucket. | 26.4% | 10.15% |
| Maximum number of fingerprints in a bucket | 4(2%) in 0 bucket and -1 bucket | 4(2%) in 0 bucket and -1 bucket |

In data set, distance from nearest minutiae to core point further than the distance between minutiae (bifurcation). So, if one of bifurcation is used as reference point, there is a chance that area hash function is smaller.

### B. Performance of Simple Hashing

From observing identification process, the result is number of matching in identification process depends on storedfingerprint distribution.For example, identification in hashingusing radius 50 pixel, which is much better than hashing using radius 10 pixel.

For better accuracy, after get the hash value from input fingerprint, there is a need to check ±n bucket around the exact bucket. Based on testing result, n value is 5 pixel which around 1.8% from hash value range of data test. This make number of matching increase into 3 to 5 times bigger. Tolerance n is given from estimation of average minutiae selected in calculation of hash value, because error occurs from the rounding off in distance calculation.

Scaling is not tolerance in this method. But, translation and rotation is accepted as long as hash value area is not damaged.

## VI. Conclusions

Simple hashing can be used to reduce number of matching in identification. But, some factors that affect distribution of stored fingerprint must be estimate carefully to get the maximum performance. In identification there is some error tolerance that must be given to select identification candidate for matching.

## References

[1] A. Silberschatz, H. F. Korth, and V. Govindarahu, *Database System Concept 6th Edition*. New York: McGraw-Hill, 2011.

[2] T.H. Cormen, C.E. Leiserson, and C. Stein, *Introduction to Algorithm 2nd Edition*. Massachusetts: MIT Press, 2002.

[3] D Maltoni, *Handbook of Fingerprint Recognition 2nd Edition*. London: Springer, 2000.

[4] C Wilson, G. Candela, and C. Watson, "Neural Network Fingerprint Identification," *Journal Artificial Neural Network*, vol. 1, no.2, pp. 203-208, 1993.

[5] B. Bhanu and X. Tan, *A Triplet Based Approach for Indexing of Fingerprint Database for Identification*.

Riverside: University of California, 2001.

[6] J. Feng and A. Cai, *Fingerprint Indexing Using Ridge Invariant*. Beijing: Beijing University of Posts and Telecomunications, 2006.

[7] D. Maio, D. Maltoni, and L. Capelli. (2000, August) FVC2000. [Online]. http://bias.csr.unibo.it/fvc2000

[8] D. Maio, D. Maltoni, and L. Capelli. (2002, April) FVC2002. [Online]. http://bias.csr.unibo.it/fvc2002

[9] R. Vazan. (2009, December) Sourceforge. [Online]. http://sourceforge.net/projects/sourceafis

[10] L. Rosa. (2012, April) Advanced Source Code. [Online]. http://wwwadvancedsourcecode.com/ortcorepoint.pdf