A Fingerprint Indexing Scheme with Robustness against Sample Translation and Rotation

Guoqiang Li, Bian Yang, Christoph Busch

Norwegian Biometric Laboratory, Gjøvik University College, Norway
guoqiang.li@hig.no, bian.yang@hig.no, christoph.busch@hig.no

Abstract: Automatic fingerprint identification systems (AFIS) are getting prevalent around the world, and the size of fingerprint databases involved in AFIS is continuously growing. Thus, studying fingerprint indexing algorithms is desirable in order to facilitate the search process in a large-scale database. In this paper, we firstly propose a feature extraction method to generate a binary template based on minutia information. A fingerprint indexing is designed by combining this binary template and Locality Sensitive Hashing indexing algorithm developed in a state-of-the-art fingerprint indexing method (minutia cylinder-code based indexing method). Experiments have been conducted on several public databases with different settings. The results show that the proposed approach achieves competitive performance or even better performance when benchmarked to the state-of-the-art fingerprint indexing methods.

1 Introduction

Fingerprint recognition system has been increasingly gaining attention around the world. Many systems have been deployed such as FBI's Integrated Automated Fingerprint Identification System (IAFIS), the European Visa Information System (VIS) and many other systems are under construction. Depending on the application context, there are two types of fingerprint recognition systems [MMJP09]

Verification system: this system carries out a one-to-one comparison to verify whether the identity is the person who he/she claims. A typical scenario is to compare the fingerprint data stored in a Europen passport with the fingerprint data captured from the subject, who holds the passport.

Identification system: this system identifies a person by searching the whole database, which results in a one-to-N comparison process. A typical scenario is to check whether a criminal suspect has been recorded in FBI's IAFIS by using his/her probe fingerprint samples for the query.

According to the information published on FBI's website [FBI], the FBI IAFIS contains enrolled fingerprint from more than 100 million subjects. Thus it is almost impossible

This work is funded by the EU 7th Framework Program under grant agreement n° 284862 for the large-scale integrated project FIDELITY.

to conduct a one-by-N comparison in such a large-scale database. Therefore, studying fingerprint indexing techniques is desirable in order to reduce the number of candidate identities, which will be further considered by a verification algorithm [LBA07].

A variety of fingerprint indexing approaches have been presented in the literature. Extracting appropriate features for building indexing tables is the core of a fingerprint indexing approach. The approaches in the literature can be grouped into three categories based on their feature extraction methods: local feature based – primarily focusing on using minutia [RM07, CFM11] or local ridge information [LBA07, BRAC08]; global feature based – using orientation field and singular points as a global reference points [WHP07]; other feature based – such as using symmetric filters [LYW06] or scale invariant feature transformation (SIFT) [SZH08].

Instead of exploring global features, we are focusing on only using minutia location and direction to extract a compact feature vector, since minutia information has been recognized as most reliable and basic feature representing fingerprints, and a standardized definition of this feature vector is given by ISO/IEC 19794-2:2011 [ISOa]. In addition, the majority of existing fingerprint indexing approaches generate the features with real values, which might lead to more computational complexity comparing to binary features. Thus we explore to extract a binary feature vector in this paper, and further build the indexing tables by using Locality Sensitive Hashing (LSH) indexing method which was developed by Cappelli et al. [CFM11, CFM15] and has been proven to be suitable for binary feature vectors.

The remainder of this paper is structured as follows: Section 2 introduces the proposed feature extraction method; the details of creating indexing tables and candidates retrieval are described in Section 3; Section 4 reports the experimental results under different settings; the conclusion is drawn in Section 5.

2 Feature Extraction method for Fingerprint Indexing

The feature extraction method is the critical component in a fingerprint indexing scheme due to the fingerprint sample variations caused at acquisition stage. The proposed feature generation method generates a set of fixed-length binary vectors for a fingerprint template. The number of binary vectors for a fingerprint template depends on the minutiae' number in this template. The proposed feature extraction method consists of three stages: local alignment, training and binary vectors generation. The details will be discussed in the following subsections.

2.1 Local Alignment and Quantization

Instead of detecting a singular point or considering the ridge information surrounding the minutia, we focus on using a local alignment concept to extract a binary vector, which can represent this local area. The basic idea of the local alignment is considering each

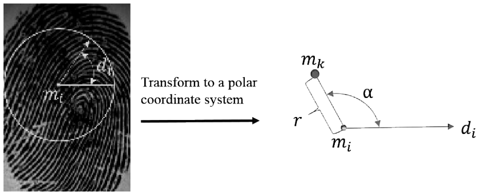


Figure 1: Local alignment: all minutiae included in the yellow circle are aligned with the central minutia in a polar coordinate system whose reference point is m_i and reference angle is the direction (denoted by d_i) of m_i. This central minutia and another minutiae included in the circle are named as a minutiae-disk.

minutia as a reference point, and then nearby minutiae are aligned with repect to this reference point (called central minutia). As illustrated in Figure 1, all minutiae included in the yellow circle are aligned with the central minutia in a polar coordinate system. This central minutia and another minutiae included in the yellow circle are defined as a minutiae-disk.

We assume a fingerprint template T including n minutiae $\{m_1, m_2, ..., m_n\}$, and each minutia comprises three properties: $m_i(x, y, d)$, where x and y are the minutia location and d is the minutia direction. A minutiae-disk (MD_i) can be formed for each minutia m_i. A polar coordinate system is defined by using m_i as reference point and the direction (denoted by d_i) of m_i as reference angle. Then each minutia m_k included in the minutiae-disk will have a new coordinate $m_k^{'}(r^{'}, \alpha^{'})$ denoted in Equation (1) and (2).

$$r^{'} = DIS(m_k, m_i) \tag{1}$$

where DIS is Euclidean distance between the two minutiae.

$$\alpha^{'} = \frac{(atan2(m_k(y) - m_i(y), m_k(x) - m_i(x)) + 2\pi) * 180}{\pi} \tag{2}$$

where function $atan2$ is 'Four-quadrant inverse tangent' defined in [ata].

In addition, the minutiae angle difference $\theta^{'}$ between m_i and m_k is denoted by the following equation:

$$\theta^{'} = |m_k(d) - m_i(d)| \tag{3}$$

In order to further tolerate the sample variation, three attributes $(r^{'}, \alpha^{'}, \theta^{'})$ are quantified by using Equation (4)$\sim$(6).

$$r = floor(r^{'}/5) \tag{4}$$

$$\alpha = floor(\alpha^{'}/5) \tag{5}$$

$$\theta = floor(\theta^{'}/5) \tag{6}$$

where, function $floor(X)$ returns the nearest integer less than the variable X.

Eventually, an aligned minutia $m_k^{'}$ with three attributes (r, α, θ) is created.Since the proposed feature generation method applies local alignment for each minutia, this indicates that each minutiae-disk will be associated with a minutia which is called central minutia. The radius of the minutiae-disk is denoted as R.

2.2 Training and Binary Vectors Generation

A training step is required in the proposed feature extraction method prior to the binary vector generation. The unsupervised learning scheme $K-means$ is chosen for this training step, since it has been proven to be appropriate for fingerprint indexing by other researchers [RM07, LYB14]. The input of $K-means$ is a set of (r, α, θ) vectors generated from the training samples. $K-means$ classifies these (r, a, θ) vectors into K clusters, and each cluster is represented by its centroid $\{C_1(r, \alpha, \theta), C_2(r, \alpha, \theta), ..., C_K(r, \alpha, \theta)\}$.

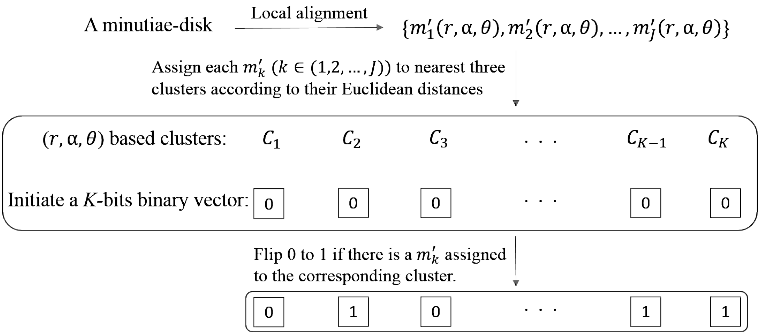


Figure 2: Procedures of generating the binary vector for a minutia-disk.

The proposed feature extraction method generates a fixed-length binary vector for each minutiae disk. Since one minutia will form one minutiae disk, the number of binary vectors for a template is equal to the number of minutiae in this template. Figure 2 illustrates the procedures of generating the binary vector for a minutiae-disk. There are four steps involved in this process:

Step 1: Apply local alignment on this minutiae-disk to generate alignment minutiae: $m_1^{'}(r, \alpha, \theta), m_2^{'}(r, \alpha, \theta), ..., m_J^{'}(r, \alpha, \theta)$.

Step 2: Initiate a K bits binary vector with all components set to 0.

Step 3: Assign each $m_k^{'}(r, \alpha, \theta), k \in (1, 2, ..., J)$ to nearest three clusters (closest cluster, second closest cluster and third closest cluster) according to their Euclidean distances.

Step 4: Flip 0 to 1 if there is a $m_k^{'}$ assigned to the corresponding cluster. Eventually, a binary vector is generated to represent this minutia-disk.

Note that only the first change will take effect even if multiple $m_k^{'}$ have been assigned to the same cluster. The reason of choosing nearest three clusters is to tolerate sample intra-class variations.

3 The Indexing Algorithm

Cappelli et al. [CFM11] have proofed that Locality Sensitive Hashing (LSH) is suitable to index the binary vector. We follow their techniques proposed in paper [CFM11] to build the indexing tables and retrieve the candidates by using our newly generated binary vectors. The following subsections give the details of indexing tables creation and candidates retrieval.

3.1 Creating Indexing Tables

Algorithm 1 . Indexing tables creation

```
Require: Minutiae templates of enrolled subjects: {T_1, T_2, ..., T_E};
    Hash functions:{f_{H_1}, f_{H_2}, ..., f_{H_Λ}} (Λ is the number of hash functions);
Ensure: Indexing tables: H_1, H_2, ..., H_Λ
for each template T_i(i ∈ 1, 2, ..., E) do
    Generate binary template from minutia template by using proposed feature extrac-
    tion method: {T(i,1), T(i,2), ..., T(i,J)}(J is the number of binary vector gener-
    ated from minutiae temple T_i)
    for each binary vector T(i,j)(j ∈ 1, 2, ..., J) do
        for each hash function f_{H_λ} do
            b = f_{H_λ}(T(i,j)))
            if CountOneBits(b) ≥ min_{OneBits} then
                record (i,j) in b − th bucket of indexing table H_λ.
            end if
        end for
    end for
end for
```

Before describing the algorithm of creating LSH-based indexing tables for fingerprint templates, it is necessary to introduce the techniques of LSH indexing method. Figure 4 gives an example of creating indexing tables by using a set of hash functions $\{f_{H_1}, f_{H_2}, ..., f_{H_\Lambda}\}$, where the number of hash functions is $\Lambda = 3$, and the number of bits selected by each hash function is $\eta = 3$. Let assume there is a binary (T_1, V_1) which denotes the first binary vector of the first fingerprint template. Each hash function will randomly select 3 bits from (T_1, V_1), then calculate the decimal value based on selected bits and store the pair (1,1) in a corresponding bucket. For instance, the decimal value calculated from f_{H_2} is 3, then (1,1) will be stored in the third bucket in hash table H_2. The number of hash tables is equal

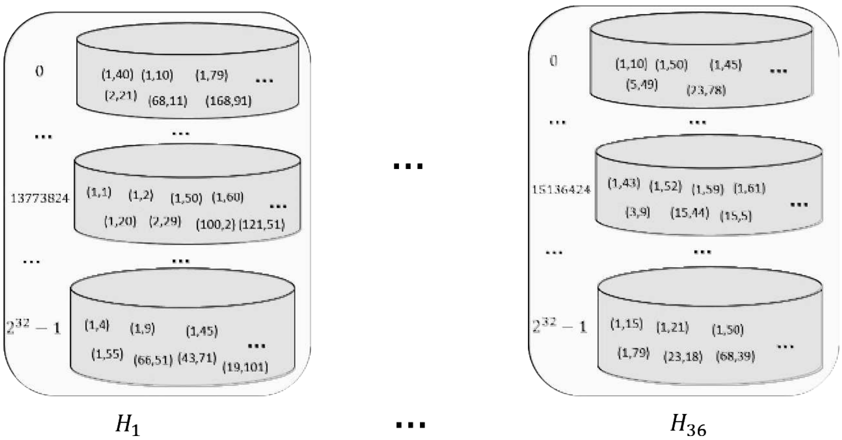


Figure 3: An example of created indexing tables, where the pair (i, j) indicates the $j-th$ binary vector of $i-th$ fingerprint template.

to the number of hash functions, and the number of buckets in each hash table is 2^{η}.

During indexing tables creation stage, we apply the similar procedures illustrated in Figure 4 to enrol fingerprint minutia template. **Algorithm 1** gives the details of creating indexing tables for a set of fingerprint minutiae templates: $\{T_1, T_2, ..., T_E\}$. The first step of enrolling these minutia template is to generate the binary template by using proposed feature extraction method. The function $CountOneBits(b)$ is to count the number of 1 bits in selected bits, for instance $CountOneBits(1010001) = 3$. The pair (i, j) will be recorded only when $CountOneBits(b)$ is not less than a parameter $min_{OneBits}$. Figure 3 gives an example of the indexing tables after completing enrolment. In addition, the original minutiae templates need to be stored somewhere else (minutiae template can be indexed by their template ID), since they will be used during candidate retrieval stage.

3.2 Candidates Retrieval

Algorithm 2 lists the procedures of retrieving candidates for a probe sample P. The same hash functions used in enrolment are used as input for candidates retrieval. Another inputs are: indexing tables $\{H_1, H_2, ..., H_\Lambda\}$, enrolled minutiae templates $\{T_1, T_2, ..., T_E\}$ as well as the minutia template of the probe sample P. The function '$Mated(m_\omega, m(i, j))'$ is to measure whether minutia m_ω from probe sample and minutia $m(i, j)$ from the reference sample meet a pre-defined geometric constraint. If they satisfy the geometric constraint, the similarity score between this probe sample and the reference sample will increases 1. '$Mated(m_\omega, m(i, j))'$ is defined in Equation (7). In order to reduce the computational complexity, we don't normalize the similarity score which is different to the candidates retrieval method used in paper [CFM11].

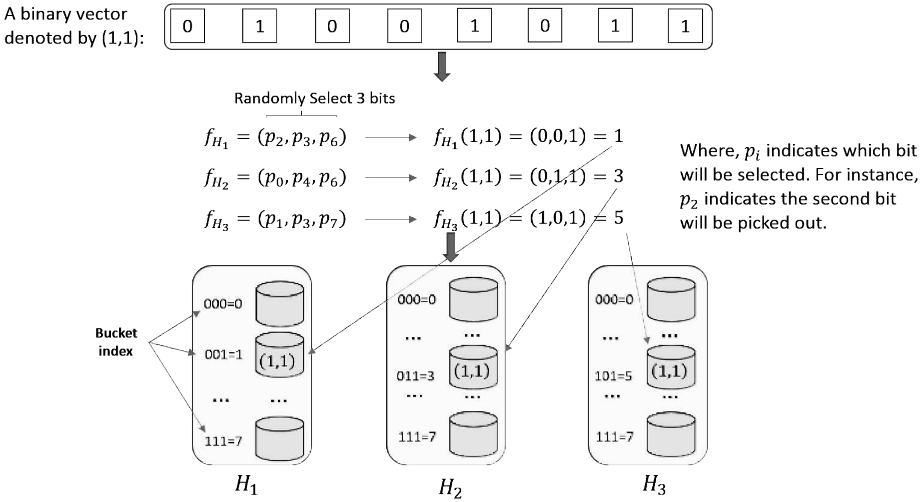


Figure 4: An example of Locality Sensitive Hashing (LSH) indexing algorithm.

$$Mated(m_\omega, m(i,j)) = \begin{cases} true & \text{if } DIS((m_\omega, m(i,j)) \leq \rho \text{ and } |m_\omega(d) - m_{(i,j)}(d)| \leq \sigma \\ false & \text{otherwise.} \end{cases} \tag{7}$$

where,$|m_\omega(d) - m_{(i,j)}(d)|$ is the direction difference between two minutiae.

4 Experimental Settings and Results

In order to evaluate the performance of proposed indexing approach, a couple of experiments have been conducted on several public databases. In accordance with ISO/IEC 19795-1 [ISOb], the performance of fingerprint indexing algorithm is reported by two criteria: penetration rate and pre-section error rate. Penetration rate is a proportion of enrolled references in a database where the identification system has to search. A pre-selection error occurs when the enrolled reference corresponding to the probe sample is not included in the pre-selected candidates. Generally speaking, the better fingerprint indexing approach will achieve lower pre-selection error rate at the same penetration rate comparing to other approaches. The minutia cylinder-code based indexing method (shortly called MCC-Index) [CFM11] was used as benchmark under same protocol in our experiments.

Algorithm 2 . Candidates retrieval

Require: Indexing tables: $H_1, H_2, ..., H_\Lambda$;
Hash functions:$\{f_{H_1}, f_{H_2}, ..., f_{H_\Lambda}\}$;
Minutiae template of enrolled subjects: $\{T_1, T_2, ..., T_E\}$;
Minutiae template of probe sample: P.
Ensure: Candidate entities.

Generate the binary template for probe sample: $V_1, V_2, ..., V_\Omega$ (Ω is the number of binary vectors);
Initiate an array to store similarity score: $S[E]$;
for each binary vector V_ω **do**
  Assume m_ω is the central minutia associated with binary vector V_ω
  for each hash function f_{H_λ} **do**
    $b = f_{H_\lambda}(V_\omega)$
    if $CountOneBits(b) \geq min_{OneBits}$ **then**
      for each pair (i, j) in $b - th$ bucket of indexing table H_λ **do**
        Assume $m(i, j)$ is the central minutia associated with the pair (i, j);
        if $Mated(m_\omega, m(i, j))$ == true **then**
          $S[i] = S[i] + 1$;
        end if
      end for
    end if
  end for
end for
Sort $S[E]$ by descending order, and select the top-N as candidate entities.

4.1 Databases Preparation and Common Settings for All Experiments

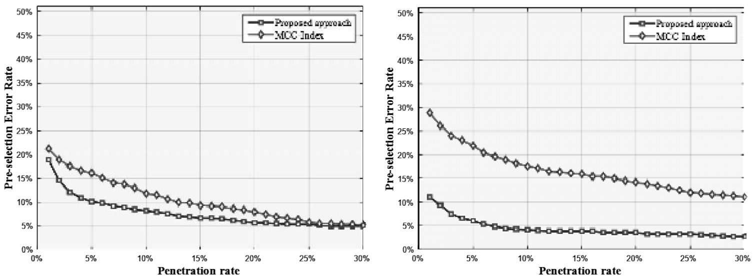
Several FVC databases are selected for the experiments: $FVC2002$ [MMJP09], $FVC2004$ and $FVC2006$ [CFFM07]. The details for the respective database will be described in the following sections. The minutia templates were extracted by a commercial product 'NeuroTechnology Verifinger extractor 6.0' [Neu]. The experimental results of MCC-Indexing method were generated by MCC sdk v1.4 [CFM10, CFM11, FMC12]. And Table 1 lists the settings of some parameters used for all experiments.

4.2 Experiments on $FVC2002$

We run the experiments on $FVC2002\_DB1$ and $FVC2002\_DB2$ respectively. There are two subsets in $FVC2002\_DB1$ as well as in $FVC2002\_DB2$. We used $FVC2002\_DB1\_B$ consisting of 80 samples as a training set for the test set $FVC2002\_DB1\_A$ which comprises 800 sample from 100 fingers. The first sample of each finger was enrolled in the indexing tables and the rest of samples were used for searching, since the quality of first sample is relatively better than the rest of samples in $FVC2002$. The similar settings

Parameter	value	Remark
R	300 pixels	the radius of the minutiae-disk
K	1024	the length of binary vector
Λ	48	the number of hash functions
η	32	the number of bits selected by hash function
ρ	256	minutia distance threshold
σ	45	minutia direction difference threshold
$min_{OneBits}$	2	the number of '1' bits in a binary vector

Table 1: Parameters setting for all experiments.



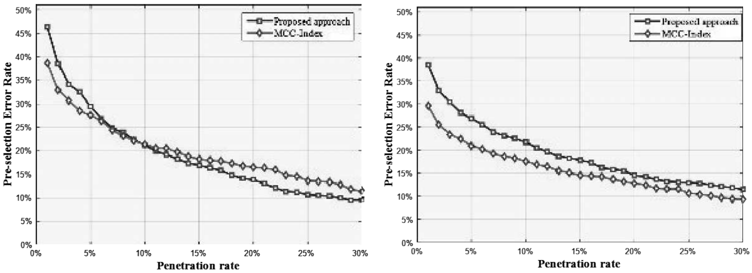
(a) Experiment on $FVC2002\_DB1\_A$ (b) Experiment on $FVC2002\_DB2\_A$

Figure 5: Performance evaluation on $FVC2002\_DB1\_A$ and $FVC2002\_DB2\_A$

were applied on $FVC2002\_DB2$: $FVC2002\_DB2\_B$ was used for the training set; $FVC2002\_DB2\_A$ was used for the test set; the first sample was used for enrolment, and the rest of samples were used for probe samples. Figure 5 demonstrates the performance running experiment on $FVC2002\_DB1$ a $FVC2002\_DB2$. The figures show the significant improvements of proposed approach on these databases.

4.3 Experiments on $FVC2004$

In order to establish similar settings as for the experiments on $FVC2002$, we used the $FVC2004\_DB1\_B$ as a training set for the test set $FVC2004\_DB1\_A$, and used the $FVC2004\_DB2\_B$ as a training set for the test set $FVC2004\_DB2\_A$. Again we enrolled the first sample of each finger to the indexing tables as we did for $FVC2002$. Figure 6 shows that the MCC-Index method outperformed our proposed approach. Then we observed the sample images of $FVC2004$. We found that the first sample of each finger doesn't have higher quality, and even it can be seem as partial fingerprint comparing other sample from the same finger as seen in Figure 7. This 'partial sample' trait might have more impact on the proposed approach than MCC-Index method, since the radius of



(a) Experiment on $FVC2004\_DB1\_A$ (b) Experiment on $FVC2004\_DB2\_A$

Figure 6: Performance evaluation on $FVC2004\_DB1\_A$ and $FVC2004\_DB2\_A$: the **first sample** of each subject was enrolled in indexing tables.

minutiae-disk is 300 pixels in proposed approach and MCC-Index method used 70 pixels. In order to investigate the impact of 'partial sample', we enrolled the forth sample of each finger and used the rest of sample as probes. Figure 8 depicts the results of using forth sample as enrolled template. The performance of proposed approach are both improved, especially on $FVC2004\_DB1\_A$.

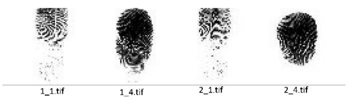


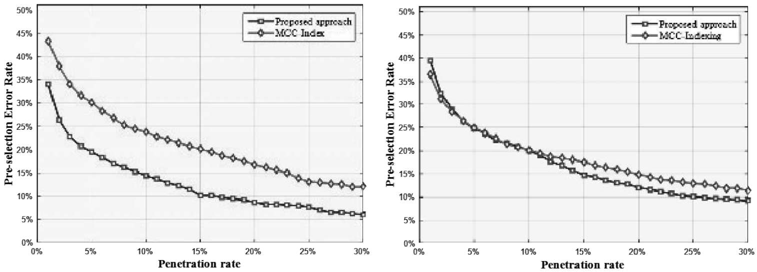
Figure 7: Fingerprint samples selected from $FVC2004\_DB1\_A$.

4.4 Experiments on $FVC2006$

FVC_2006_DB2 was selected to evaluate the performance. The training set is FVC_2006 $\_DB2\_B$ which consists of 120 samples, and the test set is $FVC\_2006\_DB2\_A$ consisting of 1680 samples which were captured from 140 fingers (12 sample per finger). The first sample of each finger was used for enrolment. Another 11 samples were chosen as probe samples for searching. In total, there are 1540 probe samples. Figure 9 shows the improvement of the proposed approach. The improvement is relatively low, since the performance of MCC-Index method is already a good baseline.

5 Conclusion

In this paper, a fingerprint indexing algorithm is designed by only using minutia location and direction information. It is invariant to sample translation and rotation, since the



(a) Experiment on $FVC2004\_DB1\_A$ (b) Experiment on $FVC2004\_DB2\_A$

Figure 8: Performance evaluation on $FVC2004\_DB1\_A$ and $FVC2004\_DB2\_A$: the **forth sample** of each subject was enrolled in indexing tables.

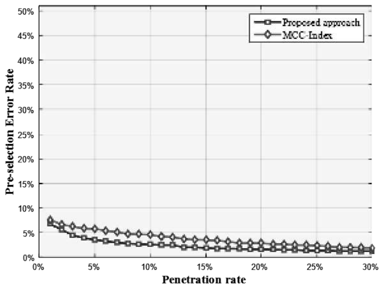


Figure 9: Performance evaluation on $FVC2006\_DB2\_A$.

proposed approach applies the local alignment on each minutia to generate a binary vector rather than using a global reference point. Based on the binary vectors for the template, an indexing approach is designed by combining LSH indexing algorithm developed in MCC-Index method. The experiments on several public database have demonstrated that the proposed approach achieved comparative performance or even better performance than the state-of-the-art fingerprint indexing method. Our future work will extend the experiment to larger-sized databases as well as investigate the impact of the radius of minutiae-disk in order to make the proposed approach more robust to a partial fingerprint sample.

References

[ata] Four-quadrant inverse tangent. `http://se.mathworks.com/help/matlab/ref/atan2.html#buct8h0-4`. Accessed: 2015-01-30.

[BRAC08] Soma Biswas, Nalini K Ratha, Gaurav Aggarwal, and Jonathan Connell. Exploring ridge curvature for fingerprint indexing. In *Biometrics: Theory, Applications and Sys-*

tems, 2008. BTAS 2008. 2nd IEEE International Conference on, pages 1–6. IEEE, 2008.

[CFFM07] Raffaele Cappelli, Matteo Ferrara, Annalisa Franco, and Davide Maltoni. Fingerprint verification competition 2006. *Biometric Technology Today*, 15(7):7–9, 2007.

[CFM10] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(12):2128–2141, 2010.

[CFM11] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. Fingerprint indexing based on minutia cylinder-code. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):1051–1057, 2011.

[CFM15] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. Large-scale fingerprint identification on GPU. *Information Sciences*, 306:1–20, 2015.

[FBI] FBI IAFIS. `http://www.fbi.gov/about-us/cjis/fingerprints_biometrics/iafis/iafis`. Accessed: 2015-01-30.

[FMC12] Matteo Ferrara, Davide Maltoni, and Raffaele Cappelli. Non-invertible Minutia Cylinder-Code Representation. 2012.

[ISOa] ISO/IEC 19794-2:2011. Information technology – Biometric data interchange Formats – Part 2: Finger minutiae data.

[ISOb] ISO/IEC 19795-1:2006. Information technology – Biometric performance testing and reporting – Part 1: Principles and framework.

[LBA07] Xuefeng Liang, Arijit Bishnu, and Tetsuo Asano. A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles. *Information Forensics and Security, IEEE Transactions on*, 2(4):721–733, 2007.

[LYB14] Guoqiang Li, Bian Yang, and Christoph Busch. A score-level fusion fingerprint indexing approach based on minutiae vicinity and minutia cylinder-code. In *2014 International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2014.

[LYW06] Jun Li, Wei-Yun Yau, and Han Wang. Fingerprint indexing based on symmetrical measurement. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 1, pages 1038–1041. IEEE, 2006.

[MMJP09] Davide Maltoni, Dario Maio, Anil K Jain, and Salil Prabhakar. *Handbook of fingerprint recognition*. springer, 2009.

[Neu] Verifinger. `http://www.neurotechnology.com/verifing-er.html`. Accessed: 2015-01-30.

[RM07] Arun Ross and Rajiv Mukherjee. Augmenting ridge curves with minutiae triplets for fingerprint indexing. In *Proceedings of SPIE Conference on Biometric Technology for Human Identification IV*, volume 6539, page 65390C, 2007.

[SZH08] Xin Shuai, Chao Zhang, and Pengwei Hao. Fingerprint indexing based on composite set of reduced SIFT features. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008.

[WHP07] Yi Wang, Jiankun Hu, and Damien Phillips. A fingerprint orientation model based on 2d fourier expansion (fomfe) and its application to singular-point detection and fingerprint indexing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(4):573–585, 2007.