A COST-EFFECTIVE MINUTIAE DISK CODE FOR FINGERPRINT RECOGNITION AND ITS IMPLEMENTATION

Tsai-Te Chu, and Ching-Te Chiu

Department of Computer Science, National Tsing-Hua University, Hsinchu, Taiwan
tsaite.chu@gmail.com, ctchiu@cs.nthu.edu.tw

ABSTRACT

Fingerprint is one of the unique biometric features for the application of identity security. Minutiae cylinder code (MCC) constructs a cylinder for each minutia to record the contribution of the neighbor minutiae, which has great performance on fingerprint recognition. However, the computation time of the MCC is high. Therefore, we proposed a new disk structure to encode the local structure for each minutia. The proposed minutiae disk code (MDC) clearly illustrates the distribution of the neighbor minutiae and encodes the neighbor minutiae more efficiently by having 280.08x speed faster than the MCC encoding part on Matlab platform. The proposed MDC approach has 96.81% recognition rate on FVC2000 and FVC2002 datasets. The hardware implementation can achieve the operating frequency at 111MHz, which can process 1234 fingerprint images per second with the image size of 255×255 and the maximum of 64 minutiae, under TSMC 90nm CMOS technology. The hardware implementation has 141.27x speed faster than the MCC method.

Index Terms— Fingerprint Recognition, Minutiae, Minutiae Cylinder Code, Minutiae Disk Code, CMOS

1. INTRODUCTION

Fingerprint is a reliable biometric features deployed on various commercial and forensic applications on identity security. Nowadays, many mobile devices are equipped with fingerprint sensors for access control. The accuracy and the response time are the two main issues for such applications to be effective.

The conventional minutiae-based method aim at finding the transformation between two fingerprints. The alignment process calculates the amount of translation and rotation referred to the reference minutiae and applies to other minutiae points in [1], [2].

The local minutiae structure is less invariant to the global transformation. The local structure has variety of forms, such as counting the minutiae number in a certain region [3], encoding the neighbor minutiae for each minutia by the invariant distances and angles [4], [5], minutiae triangles [6], [7], Delaunay quadrangle [8], and so on.

Minutiae cylinder code (MCC) [9] is one of the local minutiae descriptors created by a 3D cylinder structure to record the local relationship. The local similarity is estimated without global alignment. The local matching can rapidly determine the pairs of minutiae which match well locally. The final step is checking whether the local matching pairs can remain similar at the global level. The benefit of the MCC method is that MCC is a fixed length binary code, which has advantage for further applications on security (i.e. encryption, template protection and so on).

However, the computation time and the arithmetic computation complexity of the MCC is high. MCC is constructed by the cells which record the contribution of neighbor minutiae. The computation of the spatial and directional contribution is complex, taking the integration of Gaussian function in directional contribution calculation for instance. Therefore, we proposed a new disk structure, minutiae disk code (MDC), to record the spatial and directional contribution of the neighbor minutiae rather than using the cells as in the MCC method, which reduces the computation time by having up to 280.08x speed in MCC encoding part on Matlab platform. In addition, the proposed MDC can illustrate the distribution of the neighbor minutiae in a local region clearly and has 96.81% recognition rate on FVC datasets.

The proposed MDC method is further implemented on hardware to accelerate the fingerprint matching time. The hardware design sequentially constructs the MDC of each minutia of the query fingerprint and parallelly computes the relaxation process in the global matching procedure. The chip can achieve the operating frequency of 111MHz, and process 1234 fingerprint images per second with the image size of 255×255 and max 64 minutiae, under TSMC 90nm CMOS technology.

The rest of the paper is organized as follows. The proposed minutiae disk code (MDC) is illustrated in section 3. Section 4 shows the performance evaluation results. Section 5 introduces the hardware architecture of the MDC based fingerprint recognition. The conclusion is given in section 6.

This work was supported by Min-Hui Chu, Hunt Kao and Adam Hsu from the Novatek microelectronics corporation, Hsinchu, Taiwan.

2. MINUTIAE DISK CODE

2.1. MDC Structure

Minutiae disk code (MDC) is proposed to record the spatial and directional relationships between the center minutia and its fixed-radius neighbor minutiae by a multi-layer disk structure shown in figure 1(a). MDC consists of three major parts: sector, track, and layer. A MDC of 8 sectors, 16 tracks, and 4 layers is constructed in this work.

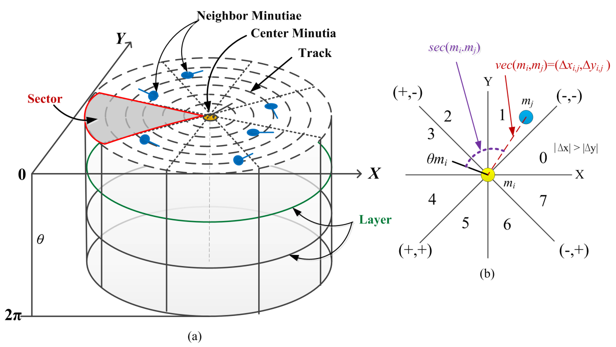

Fig. 1. (a)The proposed MDC structure. (b)The calculation of the sector value between m_i and m_j. In this case, the sector, $sec(m_i, m_j) = 3 - 1 = 2$.

Let $M = \{m_1, m_2, \ldots, m_n\}$ be the minutiae set which is obtained after the fingerprint image preprocessing. For a fingerprint, there are n minutiae in a fingerprint. The neighbor minutiae set, N_{m_i}, is the other surrounding minutiae of m_i in the disk range, R. For each minutia m_i, a MDC$_{m_i}$ is constructed by evaluating the relative relationships between m_i and the neighbor minutiae m_j in the neighbor minutiae set N_{m_i}, where $j = \{j \in \mathbb{N}, 1 \leq j \leq n, j \neq i\}$.

The sector records the difference of direction between the direction of $vec(m_i, m_j)$ and the orientation of m_i, where $vec(m_i, m_j)$ is the vector between the m_i and m_j. The computation of sector is the difference of the quantized values. The quantized direction of the vector can be estimated by the sign of $(\triangle x_{i,j}, \triangle y_{i,j})$ and magnitude of the vector, which is illustrated in figure 1(b). The sign of $(\triangle x_{i,j}, \triangle y_{i,j})$ classifies the vector into one of the four quadrants roughly, while the $\triangle x_{i,j}$ and $\triangle y_{i,j}$ magnitudes determine the vector is located in the upper or lower part of the corresponding quadrant.

The distance between the center minutia and the neighbor minutiae inside the disk range R is recorded by the track T. The approximate distance is estimated by the sum of absolute difference and quantizes the distance into the range of [0,15]. The layer L, records the orientation difference between m_i and m_j and quantizes the orientation difference $(\triangle\theta)$ into the range of $[0, 3]$. Different layer represents the relative orientation between m_i and m_j.

Next, a Gaussian-like weight filter, $[1, 2, 4, 2, 1]$, is applied on the track T and the layer L. The corresponding track and layer index have the highest weight. The neighbors of the corresponding track and layer get lower weight.

The spatial and direction contribution is the multiplication of the track and layer. Then, the contribution is accumulated to the corresponding sector. Figure 2 illustrates the MDC structure of a given minutia associated to an example where there is only one minutia surrounding. At last, the accumulated disk of a MDC is transferred into binary value by a predefined threshold, μ_{bin}.

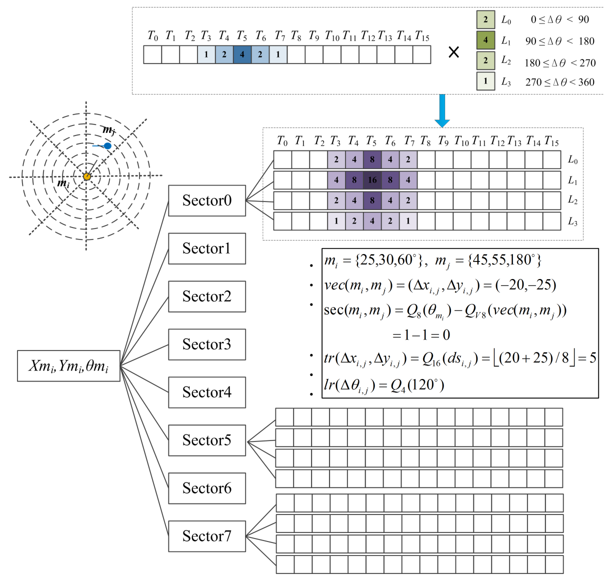

Fig. 2. A simple MDC case of m_i where only one neighbor minutiae m_j is surrounded. A Gaussian-like weight filter is applied on the track and layer. The multiplied values of tracks and layers are accumulated in the corresponding sector.

Figure 3 shows the constructing of MDC in real case. The distribution of neighbor minutiae is shown by the MDC structure. For instance, there are three neighbor minutiae in sector two. The three minutiae have different distances toward the origin minutia with a square box, while the relative distance can be easily illustrated by tracks. In addition, the layer shows the orientation difference between the origin and its neighbor minutiae.

2.2. Local Similarity Between Two MDCs

Suppose we would like to find the similarity between two fingerprints (fingerprint G and H). We apply the bitwise XOR operator to calculate the difference between two MDCs (D_{g_i} and D_{h_j}). The local similarity, $ls(D_{g_i}, D_{h_j})$, between two MDCs is defined as followed:

$$ls(D_{g_i}, D_{h_j}) = \alpha \times (1 - \frac{|D_{g_i} \oplus D_{h_j}|_1}{Length_{MDC}}) \tag{1}$$

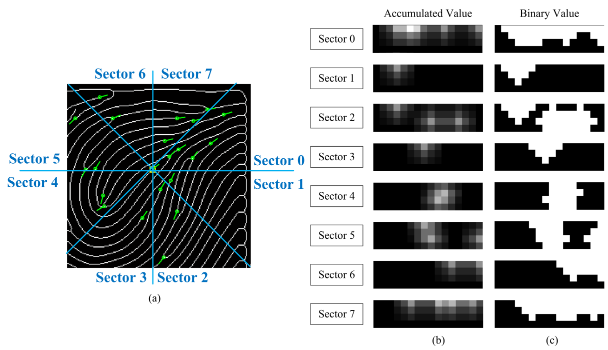

Fig. 3. The real case of constructing a MDC of the rectangle marked minutia in (a). The accumulated values from the neighbor minutiae are shown in (b), while the binary values are shown in (c).

where the $Length_{MDC} = 8 \times 16 \times 4 = 512$ in this work. The local similarity is multiplied with a constant, α, in order to enlarge the difference. The local similarity score is high when the difference between two MDCs is small. A local similarity matrix is generated after each MDC in the fingerprint G was compared with every MDC in the fingerprint H.

2.3. Global Score

The basic concept of the computation of global score is iteratively adjust the local similarity scores based on the global compatibility. Suppose fingerprint G has n_R minutiae, which has a less number of minutiae than the fingerprint H. We proposed a Local Greedy (LG) algorithm to select the highest local similarity score for each minutia in the fingerprint G. The advantage of LG algorithm is that the LG algorithm does not require to store the whole local similarity matrix which reduces the memory usage.

The relaxation process modifies the local similarity of a pair t at iteration i, λ_t^i, is defined as following equations:

$$\lambda_t^i = \omega_R \cdot \lambda_t^{i-1} + (1-\omega_R) \cdot \left(\sum_{k=1, k\neq t}^{n_R} \rho(t,k) \cdot \lambda_k^{i-1} \right) / (n_R - 1) \tag{2}$$

where ω_R is a weight factor and

$$\begin{aligned} \rho(t,k) &= f(d1) \times f_2(d2, thr_{d2}) \times f_2(d3, thr_{d3}), \\ d1 &= | \triangle \varphi_{tk}^g - \triangle \varphi_{tk}^h |, \\ d2 &= | ds_{tk}^g - ds_{tk}^h |, \\ d3 &= | \triangle \theta_{tk}^g - \triangle \theta_{tk}^h |, \end{aligned} \tag{3}$$

where $\rho(t,k)$ is a function to evaluate the global compatibility between the two pairs, and $\triangle\varphi_{tk}^g, ds_{tk}^g, \triangle\theta_{tk}^g$ are represented the sector, the difference of distance and orientation between the t and k minutiae in the fingerprint G respectively. There are three factors that affect the value of ρ, which are the amount of difference of the sector($d1$), distance($d2$), and orientation($d3$). The differences are transformed to the value of 0, 0.5, or 1 through a unit step function with a inserted middle threshold. The final global score is determined by the average of the relaxed scores.

3. PERFORMANCE EVALUATION

The simulation evaluation was performed on the PC with the CPU of Intel Pentium Dual Core (2.5GHz). The experiment is performed on the FVC2000[10], and FVC2002[11] datasets. Each dataset has 800 fingerprints consisting of 100 people with 8 different fingerprint expressions.

The FVC testing protocol was adopted for the equal error rate (EER) evaluation. The EER results of the conventional match based on alignment [2], MCC [9], and the proposed MDC are shown in Table 1, where MCC$_SDK$ is the software development kit provided by the authors of [9]. The proposed MDC method slightly improves the performance of the MCC$_SDK$ on average.

Table 1. Comparison of EER Results on FVC Datasets (%)

Method	2000DB1a	2000DB2a	2000DB4a	2002DB1a	2002DB2a	2002DB4a	Average
Alignment [2]	11.33	7.08	**7.03**	**4.94**	**5.09**	**7.75**	**7.20**
MCC$_SDK$ [9]	6.07	3.89	18.85	17.70	18.52	17.88	13.82
MDC	12.91	7.23	15.52	10.39	13.51	16.62	12.70

The recognition rate was computed by whether the query fingerprint has the right identification result. The query image was compared with the fingerprints in the template, which includes different n people with different m fingerprint expressions. The comparison scores of the same person were accumulated. The person with the highest score was the identification result. The experiment is performed on the FVC datasets which have 100 people with 8 different fingerprint expressions comparing with each other. The recognition rate result is shown in table 2. The proposed MDC method has the best recognition rate than other methods.

Table 2. The Comparison of Recognition Rate on FVC Datasets (%)(800×800, 100 people with 8 expressions)

Method	2000DB1a	2000DB2a	2000DB4a	2002DB1a	2002DB2a	2002DB4a	Average
Alignment [2]	89.00	94.38	95.38	**98.75**	**98.5**	97.38	95.56
MCC [9]	**98.88**	**99.50**	93.13	90.13	88.88	**98.38**	94.81
MDC	96.50	98.63	94.75	98.00	96.38	96.63	**96.81**

Table 3 shows the average computation time performed on the FVC datasets. The MCC$_SDK$ was executed on the C$\sharp$ platform, while the other implementations were executed on the MATLAB platform. The conventional match requires lots of rotation and translation computation in the alignment process. The proposed MDC applies the disk structure to encode the minutiae, which has 280.08x speed faster than the MCC encoding part. The proposed MDC method has the lowest computation time than the other methods on MATLAB platform.

Table 3. The Comparison of Average Computation Time on FVC Datasets

Method	Encoding	Matching	Total	Speed Up (Compare with MCC_$Impl.$)		
				Encode	Matching	Total
Alignment [2]	0	2.26s	2.26s	0	0.12	5.25
MCC_SDK (C♯) [9]	65.32ms	49.11ms	114.43ms	117.59	5.29	103.64
MCC_$Impl.$	11.6s	0.26s	11.86s	1	1	1
MDC	41.31ms	141.02ms	182.33ms	**280.80**	1.84	65.05

4. HARDWARE IMPLEMENTATION OF MDC BASED FINGERPRINT RECOGNITION

4.1. Overall Architecture

The hardware architecture of the MDC based fingerprint recognition is shown in figure 4. The hardware architecture computes the matching score between the a query fingerprint G and a template fingerprint H. There are four major units in this architecture: the read input data unit, the create MDC unit, the local similarity calculation unit, and the relaxation and global score unit. The read input data unit sequentially read the input data including the coordinates and angles of the minutiae of the query and the template fingerprint and system parameters. The create MDC unit constructs a MDC for a minutia from the query fingerprint. When a MDC is built completely, the *MDC-Out-Valid* signal is raised to start the calculation of local similarity can be started. The template MDCs from the external memory is sequentially transported into the hardware. Then, the local similarity calculation unit computes the local similarity and stores the local similarity score in the paring table based on the LG algorithm. After all the minutiae from the query image finish the computation of MDC and the local similarity with the template MDCs, the control signals *Global_Ind1* and *Global_Ind2* select the pairs from the pairing table for the relaxation process. The relaxation and global score unit adopts parallel computation on the relaxation process and calculates the final matching score.

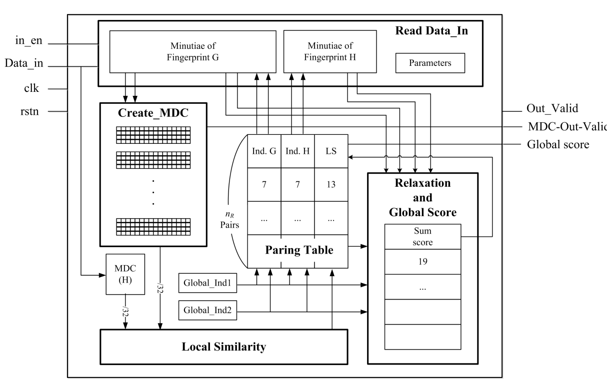

Fig. 4. The hardware implementation of the fingerprint recognition architecture.

4.2. Implementation Results

The hardware architecture has the following constraints: the max minutiae number is 64 for both query and template fingerprints, and the maximum image size is 255×255. The proposed hardware architecture is synthesized under TSMC 90nm CMOS technology. The core size is $0.693 \times 0.693 mm^2$, including the DFT circuit. The design operates at $111MHz$, and processes one comparison for $0.81ms$ which amounts to 1234 comparisons per second.

The performance comparison of the proposed method and the other works are shown in table 4. The conventional match [2] method spent the longest time on finding the best alignment result. The execution time of MCC [9] is around 114.43*ms* on the personal computer. Gutierrez *et al.* [12] applied GPU to accelerate the MCC algorithm by adopting multiple threads for parallel computation of MCC, which resulted in 20∼56.6x acceleration. The proposed MDC used 0.81*ms* on one comparison, which was faster than other works and has 141.27x speed faster of MCC method on an Intel Pentium Dual-Core 2.5GHz computer.

Table 4. The Performance Comparison with Other Works

Design Reference	Technology (nm)	Area (mm^2)	Frequency (MHz)	Avg. Comparison Time (ms)	Speed Up (Compare with MDC(SW))
Alignment[2]	(PC)Intel Pentium Dual-Core 2.5G+Matlab			2260	0.08
MCC[9]	(PC)Intel Pentium Dual-Core 2.5G+C♯			114.43	1.59
MCC+GPU [12]	(Intel Xeon E5-2630+NVIDIA Tesla M2090 with 512 CUDA cores) / (Commodity PC+ NVIDIA Geforc GTX680 with 1536 CUDA cores)			$\approx$ 2.02∼5.72	$\approx$ 31.88∼90.26
MDC(SW)	(PC)Intel Pentium Dual-Core 2.5G+Matlab			182.33	1
MDC(proposed)	TSMC90nm	1.5203	111	0.81	225.10
Other Fingerprint Recognition Algorithm					
[13]HW	Xilinx ML401 Virtex 4 FPGA		25	5.488	-
[13] SW	(Laptop) Intel i5 2.26G + Matlab			$\approx$580	-
[14] HW+SW	Altera Excalibur EPXA10F102C1 (MCU + FPGA)		50(MCU & FPGA)	362	-

5. CONCLUSIONS

In this paper, we proposed a new disk structure to efficiently encode the spatial and directional contribution from the local neighbor minutiae. The disk structure has clearer illustration of the distribution of the neighbor minutiae. We reduce the high computation time of the MCC method by the proposed MDC approach and renders a 96.81% recognition rate for fingerprint identification. The MDC method is also implemented on hardware to accelerate the fingerprint matching. The design can execute 1234 fingerprint images per second with the image size of 255×255 and maximum 64 minutiae at the operating frequency of 111MHz under the TSMC 90nm CMOS technology. The hardware implementation has 141.27x speed faster than the MCC method.

6. REFERENCES

[1] A. Jain and L. Hong, "On-line fingerprint verification," *Pattern Recog. (ICPR)*, pp. 596–600, August 1996.

[2] V. K. Alilou, "Fingerprint matching: A simple approach." [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/44369-fingerprint-matching–a-simple-approach

[3] A. Willis and L. Myers, "A cost-effective fingerprint recognition system for use with low-quality prints and damaged fingertips," *Pattern Recognition*, vol. 34, pp. 255–270, 2001.

[4] X. Jiang and W. Yau, "Fingerprint minutiae matching based on local and global structures," *Int. Conf. Pattern Recognition*, vol. 2, pp. 6038–6041, 2000.

[5] N. Ratha, V. D. Pandit, R. M. Bolle, and V. Vaish, "Robust fingerprint authentication using local structural similarity," *Proc. IEEE Workshop Applications of Computer Vision,*, pp. 29–34, 2000.

[6] G. Parziale and A. Niel, "A fingerprint matching using minutiae triangulation," *in Proc. Biom. Authenticat.*, pp. 241–2482, 2004.

[7] H. Deng and Q. Huo, "Minutiae matching based fingerprint verification using delaunay triangulation and aligned-edge-quided triangle matching," *in Proc. 5th internaltional AVBPA conference*, pp. 270–278, 2005.

[8] W. Yang, J. Hu, and S. Wang, "A delaunay quadrangle-based fingerprint authentication system with template protection using topology code for local registration and security enhancement," *IEEE Transactions on Information Forensics and Security*, pp. 1179–1192, July 2014.

[9] R. Cappelli, M. Ferrara, and D. Maltoni, "Minutiae cylinder-code: A new representation and matching technique for fingerprint recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, pp. 2128–2141, December 2010.

[10] BioLab, "The fvc2000 datasets," 2015. [Online]. Available: http://bias.csr.unibo.it/fvc2000

[11] BioLab., "The fvc2002 datasets," 2015. [Online]. Available: http://bias.csr.unibo.it/fvc2002

[12] P. D. Gutierrez, M. Lastra, F. Herrera, and J. M. Benotez, "A high performance fingerprint matching system for large databases based on gpu," *IEEE Trans. Information Forensics and Security*, pp. 62–71, January 2014.

[13] A. Alilla, M. Faccio, T. Vali, G. Marotta, and L. DeSantis, "A new low cost fingerprint recognition system on fpga," *IEEE International conf. on Industrial Technology(ICIT)*, pp. 988–993, 2013.

[14] M. Fons, F. Fons, and E. Canto, "Embedded vlsi accelerators for fingerprint signal processing," *IEEE International Symposium on Intelligent Signal Processing*, pp. 1–6, 2007.