Simple Verification of Low-resolution Fingerprint Using Non-Minutiae Feature

Sri Suwarno\textsuperscript{1} and P. Insap Santosa\textsuperscript{2}
\textsuperscript{1}Information Technology Faculty of Duta Wacana Christian University, Yogyakarta, Indonesia.
\textsuperscript{2}Dept. of Electrical Engineering and Information Technology Gadjah Mada University, Yogyakarta, Indonesia.

Email: sswn@staff.ukdw.ac.id, insap@ugm.ac.id

Abstract. It is desirable to have a simple, fast and reliable fingerprint verification application. The application can be build by avoiding long and complicated computation. However, most of the verification methods need some complicated preprocessing. Moreover, they need good quality fingerprints. In this paper, we propose a simple method to verify low-resolution fingerprint using Haar-like transformation feature. Fingerprints were captured by a commercial scanner. They were transformed using Haar-like transformation to generate feature vectors. The vector then verified to the reference vectors by using Hamming Distance. Our experiments results show that the proposed method has an accuracy of up to 80%.

1. Introduction
Fingerprint has been successfully implemented in some areas of real-time applications, such as employee attendance, personal verification in police or immigration offices. It is common that fingerprints produced by the low-cost scanner are in low-resolution. Moreover, the scanner and included SDK (Software Development Kits) are not flexible to be customized. The default setting of the scanner is customized to their own application products. As a result, the developers dealing with a low-resolution fingerprint in their applications.

Most fingerprint researches rely on features such as ridges, delta, and core, named minutiae, that can be extracted from good images. Although these features can be determined easily by a human, however, it needs complicated computation when extracted using a computer. For low-resolution fingerprint images such as seen in Figure 1, the problem is more complicated. In this figure, some ridges are broken so that they can be considered wrongly as a core.

![Example of low-resolution fingerprints](image-url)

Fig. 1 Example of low-resolution fingerprints
2. Related Work

It is common in fingerprint research that minutiae are used as a feature. The results are very reliable. However, they need a lot of complicated computation. Some researchers have conducted some experiments using the non-minutiae feature, such as shown in [1]. In this paper, researchers generated the feature from the spectrum and direction domains of fingerprints. The spectrum is generated by two dimensional Fourier Transform. Then the spectrum is divided into sectors with an equal number of pixels. Two feature vectors are generated by calculating its average as the first vector and standard deviation as the second vector. By using these two vectors, they measured the similarity of two fingerprints. They showed that their method gave a better result compared with the methods proposed in the literature mentioned in their paper. However, in their method, they still using direction domain that needs minutiae estimation.

Center of the mass of fingerprint can also be used as a feature, such as described by Luna-Ortega et al. [2]. First, they detected the location of the core using Radon Transform, then based on this location, a 70×70 pixels image area is isolated. After that, the center of the mass is calculated in that area. By using this feature, fingerprints are verified using LVQ. They reported that the proposed method gave the result of more than 90% recognition on the well-defined core. It is clear that the proposed method also need core detection.

As mentioned previously, detecting minutiae are complicated. As a replacement, the texture of a fingerprint can be used as a feature. Khalil, et al. [3] implemented GLCM (Gray Level Co-occurrence Matrix) to generate a feature. First, they enhanced the image of the fingerprint by some preprocessing. Then they isolated 129×129 pixels block around the center point, that determined using Poincare index method. Based on this block, four features were generated by GLCM, namely Correlation, Contrast, Energy and Homogeneity. After experimenting with 600 patterns as training set and 200 patterns as a testing set, they reported that the proposed method was resulting False Acceptance Rate (FAR) of 0.62%, the False Rejection Rate (FRR) of 0.08% and Equal Error Rate (ERR) of 0.5%. Although the method using GLCM feature, however, it needed core detection to locate the center of reference.

Another method to generate a fingerprint feature is the Gabor filter. Patil et al. [4] implemented a Gabor filter on 128×128 pixels of the grayscale fingerprint image. First, the divide input image into non-overlapping blocks of size 8×8 pixels. By using the Sobel operator, the gradient of blocks is calculated to estimate their local orientation. After that, Poincare index is used to determine the center point of the chosen area. Finally, the Gabor filter is implemented on each of the 8×8 blocks to generate a finger code. Based on this finger code, fingerprints are verified using Euclidean distance. They reported that the proposed method resulted up to 93% accuracy or verification. However, they claimed that the method only suitable for the small-scale verification system. Besides that, the method still needs some preprocessing to locate the core.

One of the methods that are purely avoiding minutiae detection is using wavelet transform. Some researchers, such as Wein et al. [5], Ting Tang [6], Omidiyeganeh et al. [7] and Sanjekar and Dhabe [8] used a wavelet transform to generate fingerprint features. Wein et al. firstly used the Prewitt operator to detect ridge edge. After that, the image is decomposed using wavelet resulting in four components, namely one low-frequency component and three high-frequency components. The norm of each component is then calculated to make a feature. The feature is generated by summing the norms differences for each component. Finally, the feature is compared to two thresholds to determine the similarity of fingerprints. They reported that the proposed method is effective and low computation complexity. Unfortunately, the method is only implemented into 21 fingerprints.

Ting Tang [6] using a wavelet domain feature to recognize a fingerprint. To do that, the researcher conducted some preprocessing to determine the region of interest (ROI). After the ROI is found, the area of 128×128 pixels around the ROI is isolated. The ROI is then divided into 16×16 of non-overlapping blocks. These blocks are decomposed into 2 levels using nine wavelets basis. The feature vector is composed of three elements, namely mean energy, standard deviation, and Shannon entropy. By using Euclidean distance, this vector is used to determine the similarity of fingerprints. The paper reported that the method achieved good recognition accuracy. Furthermore, it reported that Spline Bior
5.5 wavelet gave the best result with EER up to 3.16%. Meanwhile, Morlet wavelet resulted relatively poorer than another wavelet basis.

A similar method is found in Omidyeganeh et al. [7] using three wavelet basis, namely Haar, Daubechies, and Symmlet. Like some other researchers, they determine the point of reference to choose the area to be decomposed. The area is then divided into sub-area of 160×160 and 80×80 pixels and decomposed using wavelets. The Generalized Gaussian Distribution (GGD) is then generated from wavelet coefficients. The Kullback-Leibler Distance is implemented to determine the similarity of fingerprints. They reported that the Daubechies and Symmlet gave a better result compared to Haar wavelet. Meanwhile, the recognition rate of the method can achieve up to 100% by using Daubechies basis.

Haar wavelet was also used by Sanjekar and Dhabe [8] to generate feature. Initially, they decomposed the fingerprint images using Haar wavelet up to three levels. Then mean, and standard deviation of each level components are computed. Finally, these mean and standard deviation values are arranged to form a feature vector. They reported that the proposed method resulted up to 82.08 % of success.

3. Proposed Method
The main reason to propose this method is the fact that fingerprint images produced by the common scanners are low resolution. Moreover, the proposed algorithm is intended to be simple, fast yet reliable, so we use a non-minutiae feature approach. The flowchart of the method can be seen in Figure 2. In our experiment, we use fingerprints of the grayscale format of size 80×100 pixels.

\begin{align}
    a_m &= \frac{f_{2m-1} + f_{2m}}{2} \times \sqrt{2} \\
    d_m &= \frac{f_{2m-1} - f_{2m}}{2} \times \sqrt{2}
\end{align}

where \(a_m\) is average component, \(d_m\) is detail component, \(f_{2m-1}\) is the pixel value in the odd position, and \(f_{2m}\) is the pixel value in even position.
\[ a_p = f_{2m-1} + f_{2m} \]
\[ d_p = f_{2m-1} - f_{2m} \]

where \( a_p \) is addition component, \( d_p \) is the difference component, \( f_{2m-1} \) is the pixel value in the odd position, and \( f_{2m} \) is the pixel value in even position. The feature vector \( V \) is formed by combining or concatenating \( a_p \) and \( d_p \) as seen in equation 5. When \( V \) of every row is combined, we get feature matrix \( M \) as seen in equation 6.

\[ V = \begin{pmatrix} a_p & d_p \end{pmatrix} \]

\[ M = \begin{pmatrix} a_{p1} & d_{p1} \\ \vdots & \vdots \\ a_{pn} & d_{pn} \end{pmatrix} \]

In equation 6, \( n \) is the height of the fingerprint image in pixels.

The last step of the method is to verify a tested fingerprint with fingerprint reference. In our experiment, we collected four fingerprints of the same fingers of each person. We used three fingerprints for reference and one finger for a test data. There are some choices of verification methods. However, most of them need complicated computation that is not in accordance with the purpose of this paper. In this paper, we used the modified Hamming Distance to measure the similarity of two fingerprints. Initially, Hamming Distance is used in binary data. The similarity of two vectors of equal length is determined based on the number of elements of the same position that are different. It means that the smaller the distance, the higher the degree of similarity. In our experiment, we used decimal number, so we modified standard Hamming Distance by using a range to determine the equality of two elements. We calculate the degree of similarity by comparing the modified Hamming Distance value to the number of elements.

4. Experimental Result

We collected fingerprints from 50 respondents; each is taken 4 times. We used 3 fingerprints as a reference and 1 fingerprint as a test sample. Due to the limited space, Table 1 shows only 20 samples of our experiment result. The elements of Table 1 are distance values of samples to references. It can be seen from the table that for columns 1 and 2, the minimum values are found in row 1. It means that for sample number 1 and 2 the best matching is with reference 1. For sample number 4 and 5, the minimum values are in row 2. It means that for sample number 4 and 5 the best matching is with reference 2.

The result of our experiment can also be presented as a chart such as Figure 3. The horizontal axis contains a reference number, while the vertical axis is their corresponding distance. Finding the best matching is done by finding the smallest value of each sample. It can be seen in Figure 3 that sample 1 and sample 2 has minimum distance values in reference 1 and 2. By using the same procedure, the similarity degree of each sample can be related to the corresponding reference.

5. Conclusion

We have proposed a simple method to verify low-resolution fingerprints by using the non-minutiae feature. Our method does not need any preprocessing so that the process is simple. Our method operates in three stages, namely Haar-like transformation, feature generation, and verification. To verify the fingerprint, modified Hamming Distance was implemented. From the experiment result, it can be concluded that the proposed method has verification accuracy of up to 80%. To increase the verification accuracy, we will apply preprocessing, such as image enhancement, in our future research.
Table 1 Part of the experiment result

| Sample | Ref1 | Ref2 | Ref3 | Ref4 | Ref5 | Ref6 | Ref7 | Ref8 | Ref9 |
|--------|------|------|------|------|------|------|------|------|------|
| Sample1| 0.818| 0.826| 0.831| 0.834| 0.830| 0.833| 0.828| 0.827| 0.835|
| Sample2| 0.819| 0.828| 0.831| 0.831| 0.827| 0.828| 0.821| 0.826| 0.831|
| Sample3| 0.828| 0.813| 0.814| 0.825| 0.827| 0.820| 0.831| 0.819| 0.831|
| Sample4| 0.832| 0.812| 0.815| 0.822| 0.824| 0.820| 0.831| 0.826| 0.834|
| Sample5| 0.830| 0.812| 0.827| 0.822| 0.824| 0.820| 0.831| 0.826| 0.834|
| Sample6| 0.830| 0.814| 0.816| 0.820| 0.831| 0.820| 0.831| 0.817| 0.826|
| Sample7| 0.830| 0.820| 0.820| 0.820| 0.820| 0.820| 0.820| 0.820| 0.820|
| Sample8| 0.830| 0.820| 0.820| 0.820| 0.820| 0.820| 0.820| 0.820| 0.820|
| Sample9| 0.830| 0.820| 0.820| 0.820| 0.820| 0.820| 0.820| 0.820| 0.820|

Figure 3 Modified Hamming Distance Test

Acknowledgment
This work was supported by The Ministry of Research, Technology and High Education of Indonesia.
References

[1] M. S. Helfroush and M. Mohammadpour, “Fingerprint verification system: A non-minutiae based approach,” in 2009 2nd Int. Conf. Comput. Control Commun., Vol. 2, No. 1, pp. 1–4, Feb. 2009.

[2] C. A. De Luna-Ortega, J. A. Ramírez-Márquez, M. Mora-González, J. César Martínez-Romo, and C. A. López-Luévano, “Fingerprint Verification Using the Center of Mass and Learning Vector Quantization,” in The 12th Mexican International Conference on Artificial Intelligent, pp. 123-127, Nov. 2013.

[3] M. S. Khalil, D. Muhammad, and Q. AL-Nuzaili, “Fingerprint Verification Using the Texture of Fingerprint Image,” in 2009 Second International Conference on Machine Vision, 2009.

[4] P. M. Patil, “Fingerprint Verification Based on Fixed Length Square Finger Code,” in The 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’05), pp. 656-662, Nov. 2005.

[5] Z. Weina, H. Jun, Z. Xiaoyang, and Y. Weiwei, “Fingerprint verification based on wavelet and edge detection,” 2009 9th Int. Conf. Electron. Meas. Instruments, Vol. 1, pp. 3-1001-3–1004, Aug. 2009.

[6] T. Tang, “Fingerprint recognition using wavelet domain features,” 2012 8th Int. Conf. Nat. Comput., pp. 531–534, May 2012.

[7] M. Omidyeganeh, A. Javadtalab, S. Ghaemmaghami, and S. Shirmohammadi, “A Robust Wavelet-based Approach to Fingerprint Identification,” 2012 IEEE International Conference Multimedia Expo Work., pp. 413–417, Jul. 2012.

[8] P. S. O. Miss. Priti S. Sanjekar, “Fingerprint Verification Using Haar Wavelet Miss. Priti S. Sanjekar,” in International Conference on Computer Engineering and Technology, 2010, pp. 361–365.

[9] J. S. Walker, A Primer on Wavelets and Their Scientific Applications, Second Edition (Studies in Advanced Mathematics). 2008.