Ridge-Slope-Valley Feature for Fingerprint Liveness Detection

Feng Wang, Jian Cheng, and Yan Jiang
School of Electronic Engineering
University of Electronic Science and Technology of China
feng.wff@gmail.com

Abstract Attacking fingerprint-based biometric systems by presenting fake fingers is a serious threat for unattended devices. In this work, we introduce a novel algorithm, by extracting features along the fingerprint curves, to discriminate between fake fingers and real ones on static images. Pairs of mean value and standard deviation are sampled from the ridge, slope and valley of the curves. Then bag-of-words model is used to select cluster centers and form a 128-dimension feature of words' frequency. We test our method on a dataset collected by Chinese Academy of Science, which contains 960 live fingerprints and 960 fake ones made by silicon. Though the fake fingerprints is too verisimilar to be distinguished by naked eyes, we still get an accuracy of 98.85%. Because our method is based on single static fingerprint image, it can be freely embedded into existing fingerprint-based biometric systems.

Keywords fingerprint liveness detection, bag-of-words model, image texture analysis.

1 Introduction

In recent years, fingerprint verification systems for personal identity recognition reached a high degree of accuracy. Unfortunately, as the ancient Chinese proverb says, while the priest climbs a post, the devil climbs ten, various of fake fingerprint molds have been developed to attack the biometric authentication system [2]. Materials of Play-Doh, gelatin, latex, silicone, and even the printed paper can pass some authentication systems. From a security and accountability perspective, fingerprint authentication systems should have the ability to detect when fake finger samples are presented. This requirement gives huge motivation for researchers to find effective method to discriminate fake fingers from real ones. One of the solution is to analyse the difference of the texture between real and fake fingerprint images. It is cheap, convenient and easy to be embedded in to a present fingerprint verification system.

Up to now, no obviously physical feature in static images has been found to provide a clear standard of judging the liveness of fingerprint. Some researchers did preliminary analyse in using statistical methods, such as power spectrum [7], wavelet [8] [9] [10],curvelet [11], fusion of multiple static feature [12]. Perspiration pattern and other noise in valley of fingers are widely considered to be a useful information [13] [14] [15], however, whether perspiration exists or not is more determined by the users, not the machine. Pores on the ridge are also extensively used pattern [12] [15], while the abraded fake fingers would not contain the tiny details.

Due to the speciality of the classification of live and fake fingerprint, that the direction of fingerprint curves change everywhere, an isotropous statistical method cannot get a satisfied performance. The useful but subtle information will be diluted with the high response of the magnitude difference of ridges and valleys. Moreover, the width of ridges and valleys mainly depends on the pressure of the users, so statistical features like whether black or white gray levels cannot be used as the basis of judgement.

To bypass the difficulty caused by variation of direction and width, some researchers started to extract features along the fingerprint curves. Derakhshani et al. [13] and Choi et al. [12] use a 1-D long signal on the ridges, and find that the 11 to 33 FFT points have considerable ability of discrimination. [15] add signal samples from valleys, which denotes the perspiration and noise. Maybe because they intent to avoid the influence of the width, they only use the thinned skeletons of ridges and valleys, leaving the rest information from slopes apart. Furthermore, in a fingerprint of good quality, the valleys should be clean white, and the ridges are expected to be almost totally black except for pores. Whether the ridges and valleys have enough information are still in doubt.

In our approach, we use not only features extracted from ridges and valleys, but also from the slopes between them, which are ignored by most previous researchers. Unlike real human fingers, fake fingers made by silicon, gelatin and latex have s-moother edges located between the ridges and valleys. With the fraying of the material, the smooth degree of the edges will be more prominent. Mathematically, signal sampled alone the slope of live fingers should have a higher variance compared with that from fake fingers. Utilizing this property, we designed our proposed method.

2 Algorithm

As we aim at gathering the subtle difference between live and fake fingerprint, it is suggested that *no* pre-processing should be taken on the fingerprint image, to preserve all the details captured by the hardware. Some pre-processing method, e.g. histogram equalization, may enhance the fingerprint to a better visualization, but also simultaneously change the local distribution of pixel intensity.

Our algorithm contains the following procedures.

1. Partition the fingerprint image into 16×16 blocks. Calculate the directions of fingerprint curves in all blocks.
2. Generate sampling lines along the fingerprint curves. Each line collect three information: the pixel intensity distribution $\hat{X}$, mean value $E[X]$ and standard deviation $Std[X]$.
3. Calculate the conditional distribution $P(\hat{X}|E[X])$. Generate a filter that allows all $\hat{X}$ with $P(\hat{X}|E[X]) > 0.01$ pass the filter.
4. Use the filter to recalculate the $E[X]$ and $Std[X]$ of all sampling lines.
5. Execute k-means algorithm to get 128 cluster centers of all the 2D points $(E[X], Std[X])$. By aggregating the frequency of each word, a 128-dimension feature is extracted to describe a fingerprint's texture.
6. kNN classifier is used to discriminate fake fingerprints from real ones.

2.1 Feature extraction

The fingerprint is firstly partitioned into blocks of 16×16 pixels, in which the fingerprint ridge can be approximately treated as some straight lines of a certain direction. Ridge orientation is calculated using the method suggested in [16].

In each block, we define the ridge orientation as the y-direction, and the orthogonal orientation as the x-direction. Then a series of lines is generated, with width of 1 pixel and length of 16 pixels, parallel with the y-direction (Fig. 1(a)). Different from the method in [16], we introduce a over-sampling strategy: our lines are overlapped with the neighbors to get a more accurate ridge-valley envelope. The method to generate the lines can be Wu's antialiasing algorithm [17] or the overlapping area of the lines and the pixel grids (Fig. 1(b)). The method of overlapping area is strongly recommended because it is more similar to a sampling procedure.

Here, the intensity (0-1) of a pixel on the line is considered as the probability p_i, and the intensity of the pixel on the covered position of the fingerprint is considered as the magnitude x_i. The mean value and standard deviation on each sampling line can be calculated as:

$$E[X] = \frac{\sum_i p_i x_i}{\sum_i p_i} \tag{1}$$

$$Std[X] = \sqrt{\frac{\sum_i p_i x_i^2 - (\sum_i p_i x_i)^2}{\sum_i p_i}} \tag{2}$$

where p_i denotes each pixel in line image P and x_i denotes each pixel in image block X. The mean values is one-to-one correspondence to the standard deviations (Fig. 1(c)).

When press on the fingerprint scanner, people always use different pressure with different parts of finger, which leads to asymmetrical local contrast. Here we normalize the $E[X]$ signal with the maximum and minimum values in 5x5 neighbour

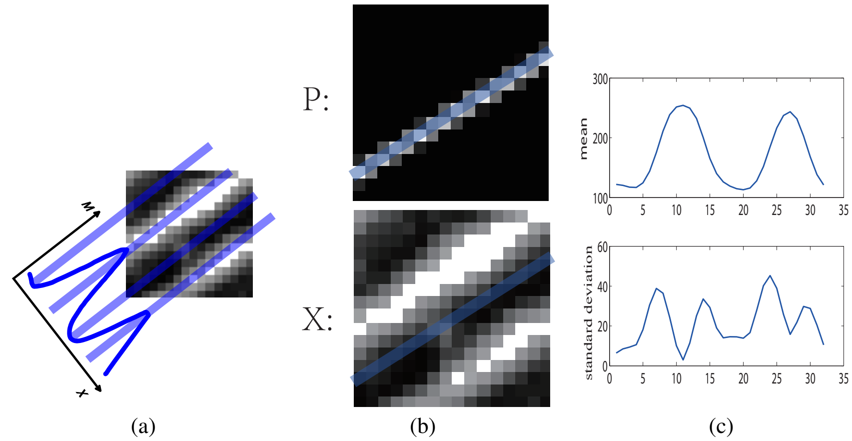

Fig. 1 (a) Four sampling lines and sampled mean value. (b) The method of overlapping area to generate a sampling line: pixels' intensity is defined by the overlapping area between pixels and the sampling line. $E[X]$ and $Std[X]$ can be calculated by simply multiplying the matrix P, X and X^2. (c) The sampled mean value and standard deviation from (b).

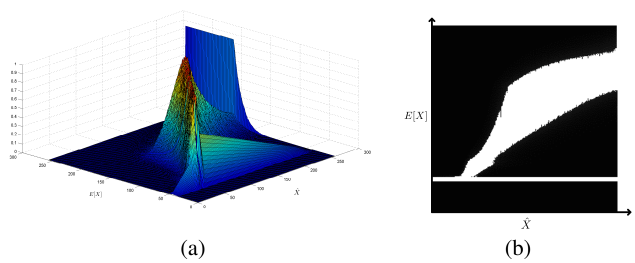

Fig. 2 (a) Normalized filter map $P(\hat{X}|E[X])$. (b) Manually fixed filter matrix. The white horizontal line is where the pore pattern lies in.

blocks[1] to eliminate this effect. Since $Std[X]$ is proportional to X, the same scale transform should be applied to $Std[X]$ in each block, too.

2.2 Noise Filter

In section 2.1, we assume that the orientation of fingerprint curve is straight in each block. The assumption is quite precise in most blocks. However, it is not always correct, which leads to a bad performance. For example, in the middle of the fingerprint the curves are always like a circle. Moreover, at the edge of the finger where the curves break, the sampling line will mistaken the blank pixels as useful information.

[1] The accuracies of different sizes of normalization window are illustrated in Fig. 4

To fix these mistakes, we introduce a two-stage noise filter method. In the first stage, we generate the joint distribution of $\hat{X}$ and $E[X]$ as $P(\hat{X}|E[X])$. Under an assumption that the noise is much weaker than the useful signals, we design a filter to allow the signal with $P(\hat{X}|E[X]) > 0.01$ pass only.

Notice that this filter can be modified arbitrarily. There are some obvious noises in the map. For example, in the middle levels of $E[X]$ (slope), there are too much X values of 255, which should be erased manually. Note that some of the areas with low $E[X]$ and high $\hat{X}$ have considerable probability. These are the pores on the edge, so it is necessary to let all of them pass the filter. Finally, the prior map (or filter) is shown in Fig. 2(b).

In the second stage, we calculate the $E[X]$ and $Std[X]$ again, using the filter matrix to filtering the noise out. This filter method is a big qualitative leap that increases the accuracy from about 89% to 98.85%, and makes this algorithm to be one of the state-of-the-art algorithms.

2.3 Bag-of-words model and feature representation

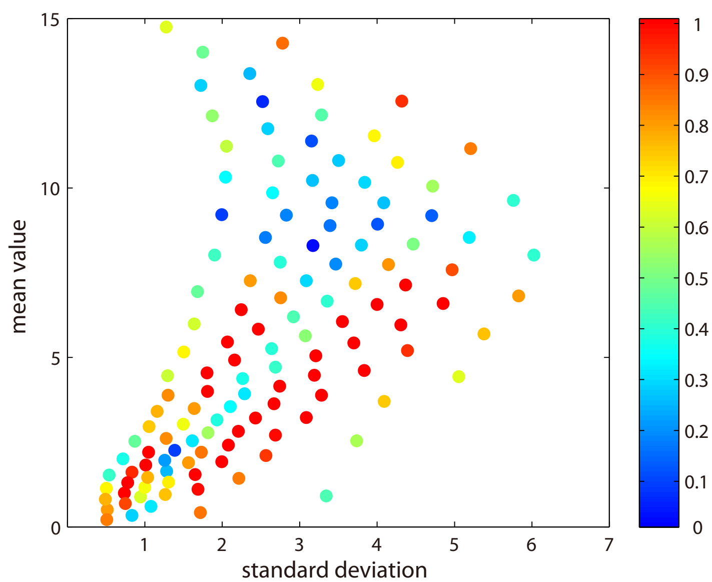

Fig. 3 Centers in bag-of-words model and the ability of discrimination of the words. All the words w_i with $J(w_i) > 1$ are colored by red, regarding that they have enough ability of discrimination.

One easiest method to represent the feature is to quantize the mean-standard deviation scatter to a grid, and use the frequency of each node as the feature[2]. This would cause a problem that some of the dimensions of the feature are always zero. On account of this phenomenon, we use bag-of-words model instead to avoid the waste of dimension.

After the filter step, we collect all the pairs of mean value and standard deviation and use a 2D *k*-means algorithm to select 128 cluster centers (*k*=128). We use the frequency of the appearance of the 128 words in one image as the feature.

To judge the ability to distinguish between the live and fake finger of each word, we modified the Fishers linear discriminant.

$$J(w_i) = \frac{\left|\mu_{live} - \mu_{fake}\right|}{\sqrt{\sigma_{live}^2 + \sigma_{fake}^2}} \tag{3}$$

Where $\mu_{live}(\mu_{fake})$ and $\sigma_{live}^2(\sigma_{fake}^2)$ represent the mean and variance of the word w_i in the training live(fake) class. The word is considered to be enough distinctive to classify if $J(w_i) > 1$.

All the words and their ability of discrimination is shown in Fig. 3. It can be concluded from the plot that the ridge and slope patterns have better performance. However, the slope patterns have been ignored by researchers for a long time [12] [15].

3 Experiment

We use *k*NN classifier to discriminate fake subjects from the live ones. It is almost the simplest classifier, with only one parameter k and one method to measure the distance from samples. One of the disadvantage of *k*NN classifier is the time and space complexity increase as the training set grows. We have also tested other classifier, such as SVM and random forest, they get about 4%-5% less classify rate than the *k*NN classifier. This may because we have not found the latent manifold of our feature.

Hellinger distance, *ell*2 distance, KL-divergence is utilized to determine the distance between the features. As the result illustrated in Fig. 4, there are no apparent differences between the three distances.

The dataset is provided by Chinese Academy of Sciences [1]. It contains 960 fake fingerprint made by silicon and 960 corresponding live fingerprint. The fingerprints have been divided into two subsets, one for training and another for testing. To make our results more persuasive, we have also implemented some of other algorithms such as LBP [18], Choi's method [12] and Tan's method [15]. The recognition rate are listed in Table 1.

[2] The result has also been illustrated in Fig. 4 labeled by "Grid Feature"

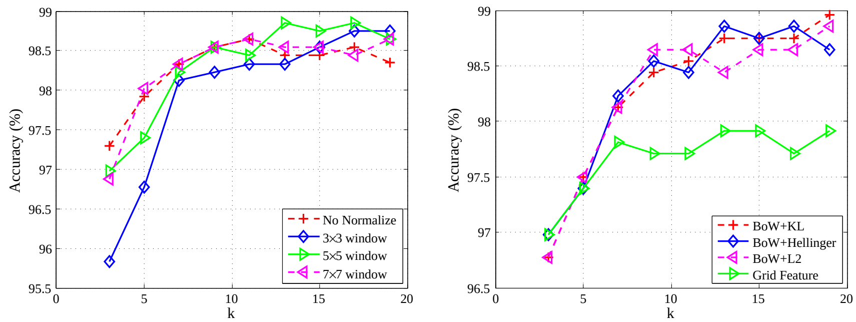

Fig. 4 (a) The affect of different sizes of the normalization window. (b) The recognition rate of different measurements of distance and the affect of various k in kNN. Note that the performance of "Grid Feature" mentioned in Section 2.3 is also displayed here. The results of more algorithms is listed in Table 1.

Table 1 Recognition rate of different algorithms

BoW+KL	**Grid Feature**	**No Noise Filter**	LBP [18]	Choi's method [12]	Tan's method [15]
98.85%	97.92%	89.17%	91.75%	79.167%	85.52%

4 Conclusion

In this paper, we proposed a fingerprint texture extraction algorithm for fingerprint liveness detection. Our method overcomes the shortage of previous ones, ignoring the information from the slope. Experiment results show that the slope patterns perform no less than the feature extracted from the ridge, and beyond the valley signals. By combining all the patterns together, our algorithm promote the recognition rate to more than 98% on the dataset from Chinese Academy of Sciences.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (61201271, 61301269), the Fundamental Research Funds for the Central Universities (ZYGX2013J019, ZYGX2013J017), Sichuan Science and Technology Support Program (cooperated with the Chinese Academy of Sciences) (2012JZ001), and Science and Technology Support Program of Sichuan Province, China (2014GZX0009).

References

1. Fake fingerprint database from Chinese Academy of Sciences: http://www.fingerpass.csdb.cn

2. Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems", *IBM systems journal*, vol. 40, no. 3, pp. 614–634, 2001.
3. Athos Antonelli, Raffaele Cappelli, Dario Maio, and Davide Maltoni, "Fake finger detection by skin distortion analysis", *Information Forensics and Security, IEEE Transactions on*, vol. 1, no. 3, pp. 360–373, 2006.
4. Jia Jia and Lianhong Cai, "Fake finger detection based on time-series fingerprint image analysis", in *Advanced Intelligent Computing Theories and Applications. With Aspects of Theoretical and Methodological Issues*, pp. 1140–1150. Springer, 2007.
5. Yangyang Zhang, Jie Tian, Xinjian Chen, Xin Yang, and Peng Shi, "Fake finger detection based on thin-plate spline distortion model", in *Advances in Biometrics*, pp. 742–749. Springer, 2007.
6. Jia Jia, Lianhong Cai, Kaifu Zhang, and Dawei Chen, "A new approach to fake finger detection based on skin elasticity analysis", in *Advances in Biometrics*, pp. 309–318. Springer, 2007.
7. Pietro Coli, Gian Luca Marcialis, and Fabio Roli, "Power spectrum-based fingerprint vitality detection", in *Automatic Identification Advanced Technologies, 2007 IEEE Workshop on*. IEEE, 2007, pp. 169–173.
8. Aditya S Abhyankar and Stephanie C Schuckers, "A wavelet-based approach to detecting liveness in fingerprint scanners", in *Defense and Security*. International Society for Optics and Photonics, 2004, pp. 278–286.
9. Yiu Sang Moon, JS Chen, KC Chan, K So, and KC Woo, "Wavelet based fingerprint liveness detection", *Electronics Letters*, vol. 41, no. 20, pp. 1112–1113, 2005.
10. Bozhao Tan and Stephanie Schuckers, "Liveness detection for fingerprint scanners based on the statistics of wavelet signal processing", in *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*. IEEE, 2006, pp. 26–26.
11. Shankar Bhausaheb Nikam and Suneeta Agarwal, "Fingerprint liveness detection using curvelet energy and co-occurrence signatures", in *Computer Graphics, Imaging and Visualisation, 2008. CGIV'08. Fifth International Conference on*. IEEE, 2008, pp. 217–222.
12. Heeseung Choi, Raechoong Kang, Kyoungtaek Choi, Andrew Teoh Beng Jin, and Jaihie Kim, "Fake-fingerprint detection using multiple static features", *Optical Engineering*, vol. 48, no. 4, pp. 047202–047202, 2009.
13. Reza Derakhshani, Stephanie AC Schuckers, Larry A Hornak, and Lawrence O'Gorman, "Determination of vitality from a non-invasive biomedical measurement for use in fingerprint scanners", *Pattern Recognition*, vol. 36, no. 2, pp. 383–396, 2003.
14. Sujan TV Parthasaradhi, Reza Derakhshani, Lawrence A Hornak, and Stephanie AC Schuckers, "Time-series detection of perspiration as a liveness test in fingerprint devices", *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 35, no. 3, pp. 335–343, 2005.
15. Bozhao Tan and Stephanie Schuckers, "Spoofing protection for fingerprint scanner by fusing ridge signal and valley noise", *Pattern Recognition*, vol. 43, no. 8, pp. 2845–2857, 2010.
16. Lin Hong, Yifei Wan, and Anil Jain, "Fingerprint image enhancement: algorithm and performance evaluation", *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 8, pp. 777–789, 1998.
17. Xiaolin Wu, "An efficient antialiasing technique", *ACM SIGGRAPH Computer Graphics*, vol. 25, no. 4, pp. 143–152, 1991.
18. Timo Ojala, Matti Pietikäinen, and David Harwood, "A comparative study of texture measures with classification based on featured distributions", *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.