Direct Feature Point Correspondence Discovery for Multiview Images: An Alternative Solution When SIFT-Based Matching Fails

Jinwei Xu[1](✉) and Jiankun Hu[2]

[1] Marketing Analytics and Insight, Data2Decisions, Sydney, NSW 2000, Australia
link.xu.job@gmail.com
[2] School of Engineering and Information Technology,
The University of New South Wales, Canberra, ACT 2600, Australia
j.hu@adfa.edu.au

Abstract. 3D fingerprint identification is an emerging biometric authentication method, which is powerful against spoofing attacks. For actualizing 3D fingerprint identification, this paper develops a 3D fingerprint minutiae cloud reconstruction technique based on 2D multiview touchless fingerprint images. This technique provides a practical solution for 2D minutiae matching for multiview fingerprint images, when traditional feature correspondence finding based on 2D SIFT (Scale Invariant Feature Transformation) feature points fails. In this case, developing a new 2D feature point correspondence establishment algorithm is necessary to cover the deficiency of the SIFT-based technique. In this paper, minutiae, a type of detailed ridge-valley features in fingerprint images, are utilized for the correspondence discovery. Furthermore, differential evolution, an efficient evolutionary computing framework is employed to directly infer the possible correspondence of minutiae sets from the different posed fingerprint images. Our experiments demonstrate that the proposed direct 2D feature point correspondence discovery strategy is able to handle the cases when the SIFT-based matching fails. To further illustrate the advantages of the proposed algorithm, 3D fingerprint minutiae cloud construction is conducted based on the feature correspondence discovered by the proposed algorithm. The experiments on 2D different posed fingerprint image matching and 3D fingerprint minutiae cloud construction show that the proposed algorithm can be used as an alternation when SIFT-based matching fails.

Keywords: 3D fingerprint reconstruction · Feature matching · Minutiae cloud · Differential evolution

1 Introduction

3D fingerprint identification system is an emerging biometric authentication method, which is powerful for anti-spoofing attacks. Beyond 2D image domain, finger object built on point cloud and curved surface can be displayed within 3D

S. Guo et al. (Eds.): TridentCom 2016, LNICST 177, pp. 137–147, 2017.
DOI: 10.1007/978-3-319-49580-4_13

space, which provides more stereo sense [5]. Being different from traditional 2D fingerprint identification systems, 3D fingerprint identification systems hold the following advantages.

- **Contactless Fingerprint Image Acquisition:** Traditional 2D fingerprint images have to be captured by placing, pressing or rolling fingers against optical scanning devices or finger trace cards. Such operations usually result in distorted, incomplete and low resolution fingerprint images. Instead, 3D fingerprint imaging systems provide contactless scanning setting, which directly prevent finger from touching imaging devices. The resultant fingerprint images are non-distorted, complete, multi-posed and resolution guaranteed. The 2D multi-posed fingerprint images captured by 3D fingerprint imaging system are shown in Fig. 1.
- **Man-made Finger Model Prevention:** Traditional 2D fingerprint identification systems are being challenged by man-made finger models in recent years. That is, the imposter finger models, which are made by printing finger ridge-valley patterns onto rubber, plastic, and other soft materials, are intended to fool the 2D fingerprint identification systems. Consequently, the 2D fingerprint identification systems are unable to distinguish whether the fingers under investigation are genuine human fingers. On the contrary, 3D fingerprint identification systems can easily validate whether the investigated fingers are artificially made ones.

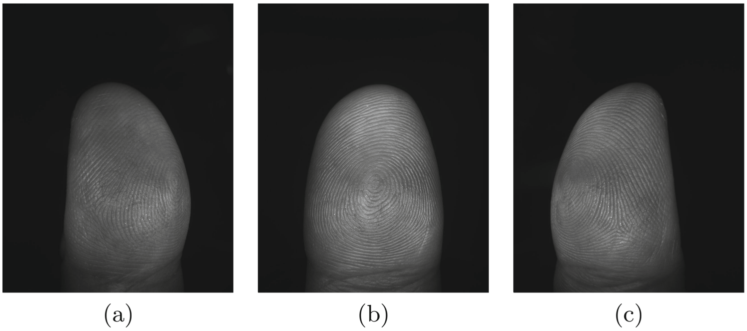


Fig. 1. 2D Multi-posed fingerprint images (left thumb): (a) left lateral view; (b) frontal view; and (c) right lateral view.

3D fingerprint construction process plays an essential role for 3D fingerprint identification system. Within 3D fingerprint identification system, the target finger can be captured by multiple built-in cameras. Since these built-in cameras are differently posed, the resultant pictures are differently posed. To generate 3D finger model, 3D construction process works on the obtained multi-posed 2D images [2,4]. To be explicit, as an initial step, correspondence between different posed images should be discovered. In most cases, the correspondence discovery depends on feature points obtained by specific feature point detection method such as SIFT. Afterward, based on the discovered correspondence in 2D domain, the spatial coordinates for the detected feature points in 3D space are calculated by 3D triangle reconstruction. Furthermore, other points' 2D correspondence can be inferred based on the prior correspondence establishment. Then their 3D coordinates are also calculated via 3D triangle reconstruction. At the end, the point cloud can be generated and it is further used to form the curved surface of the 3D finger model [1,3].

For discovering feature level correspondence between two differently posed images, SIFT is one of the most popular methods. After SIFT point detection process, the image points, whose multi-scale Gaussian filtering response are salient, are labeled as feature points. For each individual feature point, it holds a high-dimension feature vector. Such feature vector is a summary of local gradient information in terms of gradient magnitude and direction. Furthermore, two points whose feature vectors have higher similarity are matched between different images, while two points with lower feature vector similarity are not matched. The correspondence between different images is discovered via the matched feature points [5]. For fingerprint images, however, the feature vectors are generally similar, as the local regions centered at every feature points have very similar ridge-valley structures (the local summarized gradient information are also very similar). Such local structure similarity significantly reduces the discrimination of the feature vector-based matching. Therefore, massive mismatches happen between the different posed fingerprint images. In this case, the SIFT-based feature point matching approach is unable to work. Figure 2 shows an example when SIFT-based feature matching method is applied to the left and frontal images in Fig. 1.

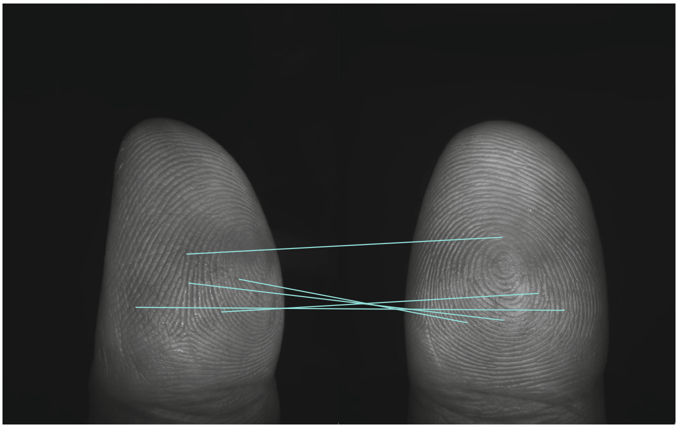


Fig. 2. An example: SIFT-based feature matching method fails to find the correspondence between the left and frontal views in Fig. 1.

For covering the SIFT deficiency, we propose a direct feature point correspondence discovery algorithm based on evolutionary computing in this paper. As an initial step, the minutiae in fingerprint images are utilized instead of using the SIFT feature points. Furthermore, the proposed technique treats the feature point matching problem as the optimization problem. To maximize the number of the matched feature points between two images, differential evolution, a fast searching evolutionary computing algorithm is adopted. The rest of this paper is organized as follows: in Sect. 2, the differential evolution-based minutiae matching method is introduced; in Sect. 3, the experiments for 2D different posed minutiae set matching and 3D minutiae point cloud construction are conducted; in Sect. 4, the conclusion and future research directions are given.

2 Proposed Method

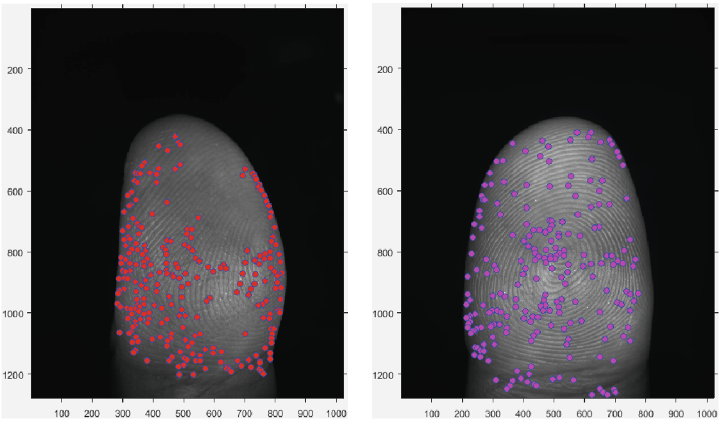
The minutiae matching process can be regarded as the process to search for more and more mated minutiae points from differently posed fingerprint images. To actualize this process, we utilize differential evolution algorithm [6,7]. A summary for the proposed differential evolution-based minutiae matching program is provided as follows.

- **Step 1:** Input two differently imposed 2D fingerprint images.
- **Step 2:** Employ Verifinger SDK to extract minutiae points for input images. The outputs obtained from Verifinger SDK are minutiae points' coordinates in 2D image domain.
- **Step 3:** Generate fundamental matrix and initially set up its elements. These elements belonging to the fundamental matrix are treated as the parameters which need to be optimized by the differential evolution algorithm. Also, the generated fundamental matrix must be a rank-two matrix.
- **Step 4:** Define objective function for optimization. Such objective function is required to represent the underlying relationship between the fundamental matrix's elements and the number of the matched minutiae points for both images.
- **Step 5:** Define epipolar constraint and spatial constraint for point-to-point matching.
- **Step 6:** Iteratively run the differential evolution algorithm until the termination conditions are satisfied.
- **Step 7:** Export the optimized fundamental matrix's parameters. Based on these optimized parameters, the minutiae point correspondence relationship can be obtained.

To be explicit, the technical details involved in each individual step are described as follows.

2.1 Minutiae Extraction

Minutiae extraction is the first step for the subsequent procedures. To ensure the reliability of the extracted minutiae points, Verifinger SDK, a widely used



(a) (b)

Fig. 3. The extracted minutiae by Verifinger SDK for the left and frontal views in Fig. 1: (a) left view's minutiae points; and (b) frontal view's minutiae points.

commercial software development kit for biometric applications, is employed in this paper. We believe that VeriFinger SDK is the most reliable minutiae extractor in public domain as far as we know. For the multi-posed fingerprint images captured by the 3D imaging system, we cannot find any other softwares or computer programs for fingerprint minutiae detection and extraction, which can be competitive or even better than VeriFinger SDK. Figure 3 shows the extracted minutiae for the left and frontal views in Fig. 1 respectively.

2.2 Fundamental Matrix

Fundamental matrix is a core concept in computer vision system, which explains projective geometric relationship for associated 2D objects. Also, the fundamental matrix is a low-rank matrix whose dimension is 3 × 3 and rank should be equal to two. This strict equality imposed on the matrix rank to regulate the epipolar lines in one view can be converged into a point (such point is called epipole). To guarantee the fundamental matrix's rank equals to two, the following matrix decomposition strategy is applied.

$$F = A \cdot B = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \\ A_5 & A_6 \end{bmatrix} \cdot \begin{bmatrix} B_1 & B_2 & B_3 \\ B_4 & B_5 & B_6 \end{bmatrix} \tag{1}$$

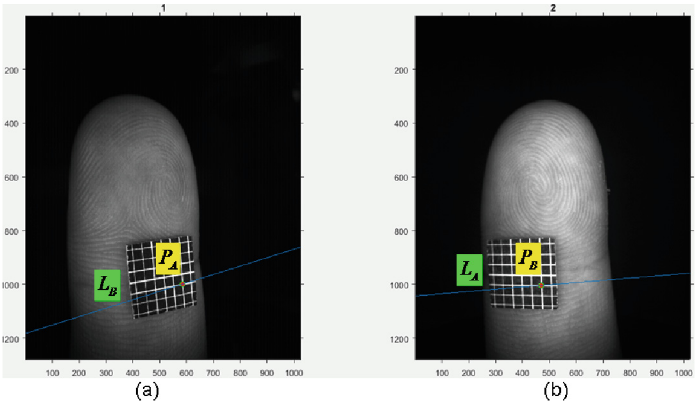
where the elements in matrixes A (dimension 3×2) and B (dimension 2×3) can be assigned by arbitrary numeric values (all zeros are exceptional) and $rank\,(F) = 2$. Due to the multiplication between A and B, the number of the parameters which need to be optimized for F is twelve.

2.3 Objective Function

The objective function pushes the differential evolution algorithm approaching to the optimal parameters and solutions. By evaluating individual fitness in whole population, the algorithm is not only to preserve the individuals with higher fitness but also to weed out the ones with lower fitness. To better drives the differential evolution algorithm to work towards the optimum, an objective function which can map the twelve parameters for fundamental matrix F onto the number of the matched minutiae points is created. In this sense, the increment of the matched minutiae points can positively feedback the evolution of the parameter's values. When the maximal number of the matched minutiae is achieved, it means the differential evolution algorithm has already converged to the optimal solutions for the target parameters.

2.4 Epipolar Constraint and Spatial Constraint

Both epipolar and spatial constraints are required to be considered simultaneously to avoid invalid solutions for target parameters during differential evolution searching process. Figure 4 illustrates how the epipolar constraint works.



(a)

(b)

Fig. 4. An example to show how the epipolar constraint works.

In Fig. 4, the images are captured from left and frontal views respectively. A point, located at the same position of the chessboard, appears in the two different views. We denote it as P_A in left view image and P_B in frontal view image. Under the projective transformation of the fundamental matrix, P_A is projected into an epipolar line L_A in the frontal view image and P_B is mapped to an epipolar line L_B in the left view image. In the left view image, the epipolar line L_B passes through P_A. Also, in the frontal view image, the epipolar line L_A passes through P_B. This observation evidences a very important characteristic for the fundamental matrix. It demonstrates that P_A can be matched against

P_B when P_A's epipolar line passing through P_B. This important characteristic directs the differential evolution algorithm to seek for the points, whose epipolar lines should pass through another view's points. Mathematically, a valid point-to-point match should be counted when the following condition is satisfied.

$$[P_B, 1] \cdot F \cdot [P_A, 1]^T = [P_B, 1] \cdot L_A = 0 \tag{2}$$

where $P_A = \left[P_A^{(x)}, P_A^{(y)}\right]$ and $P_B = \left[P_B^{(x)}, P_B^{(y)}\right]$ (dimension 1×2).

Being different from the epipolar constraint, spatial constraint only focuses on the minutiae points' relative locations in both horizontal and vertical directions. Figure 5 shows how the horizontal spatial constraint works. For the horizontal-axis constraint, the matched minutiae points' relative shifts along horizontal-axis should be the same. The point relative shift is calculated according to the following formula. In Fig. 5, the yellow and pink reference lines are determined according to the fingertip.

$$\tilde{x}_{point} = \frac{x_{point} - x_{yellow}}{x_{pink} - x_{yellow}} \tag{3}$$

where $\tilde{x}_{point}$ is the horizontal relative shift of the minutiae point. x_{point}, x_{yellow} and x_{pink} stand for the x coordinates for the minutiae point, the yellow reference line and the pink reference line respectively.

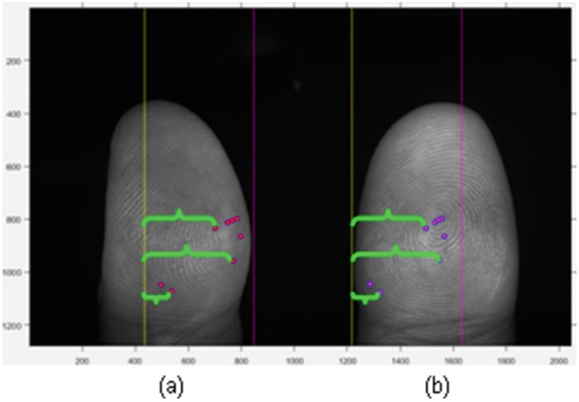


Fig. 5. An example to show how the horizontal spatial constraint works. (Color figure online)

Besides, Fig. 6 exhibits how the vertical spatial constraint works. For the vertical-axis constraint, the matched minutiae points' y coordinates should be the same. That is, the matched minutiae points' heights are at the same level in image domain.

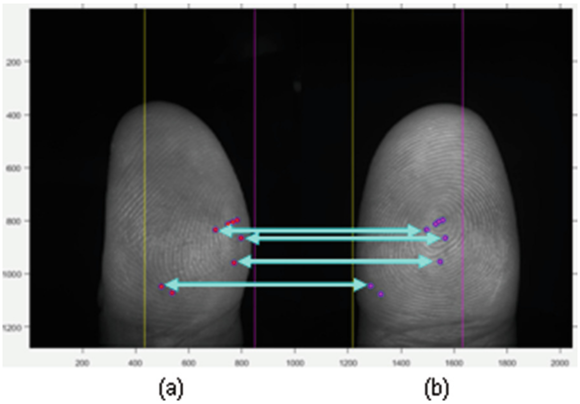


Fig. 6. An example to show how the vertical spatial constraint works.

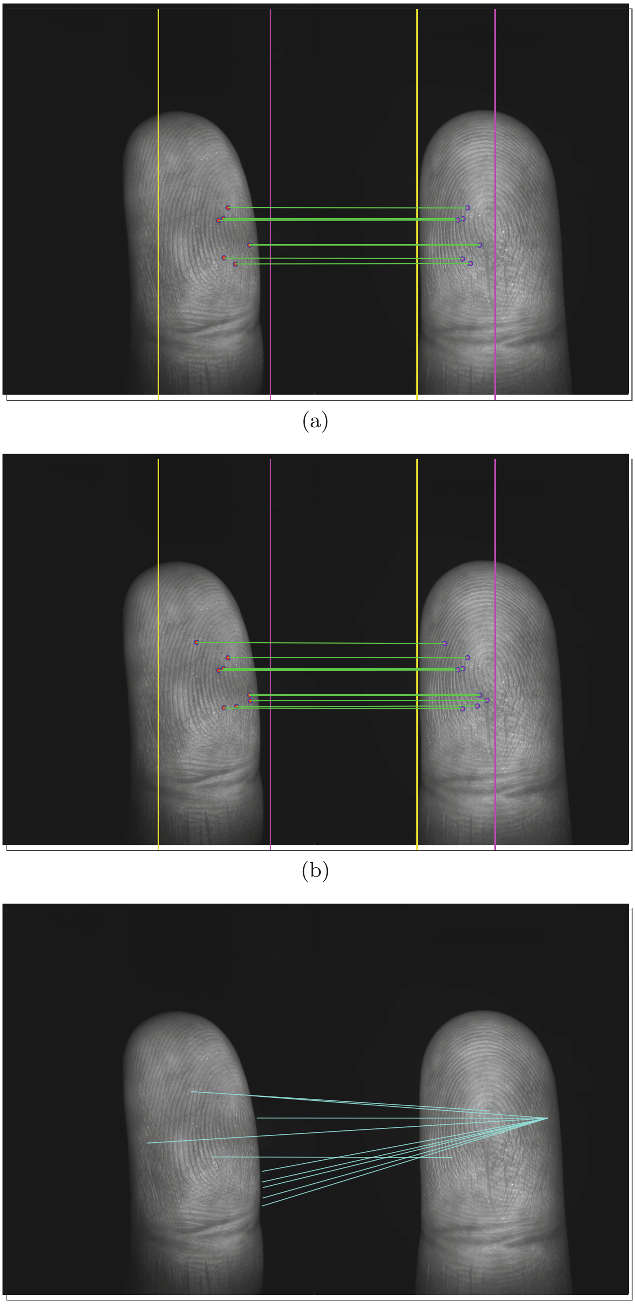
3 Experiments

To validate the feasibility and effectiveness of the proposed direct minutiae matching algorithm, we conduct the following two experiments: (i) the minutiae point level's correspondence discovery between two differently posed 2D fingerprint images; and (ii) minutiae point cloud construction within 3D space.

3.1 Minutiae Point Correspondence Discovery

The data set used for this study is collected from the internal volunteers from the University of New South Wales. This data set consists of 150 volunteers' 3D fingerprint imaging records. Each volunteer case includes 10 fingers from left palm to right palm. For each single finger, we capture its left, frontal and right view images by 3 built-in cameras in 3D fingerprint imaging system. For ensuring robustness and reliability, we scan every single finger twice. In total, we collect $60 = 10 \times 3 \times 2$ 2D fingerprint images for each volunteer. Afterwards, we extract the minutiae points by using VeriFinger SDK. Therefore, we obtain 60 minutiae point files for each volunteer case.

As an example, we demonstrate the minutiae point matching results in Fig. 7. The result shown in Fig. 7(a) is obtained based on a single run of the proposed algorithm. To search for more matched minutiae points, Monte Carlo simulation based on multiple runs is conducted (the result is shown in Fig. 7(b)). For each independent trial, the proposed differential evolution-based algorithm starts with a group of randomly initialized solutions. The randomly initialized solutions could be different for each independent run, since the random number generator built in the differential evolution approach works on the different random number seed every time. The Monte Carlo simulation repeatedly runs the proposed algorithm to make sure the randomly initialized solutions could be uniformly distributed. After multiple runs, we aggregate the final result by selecting the



(a)

(b)

(c)

Fig. 7. An example to show the results after running the proposed direct minutiae point matching algorithm for a right index finger. Also the SIFT matching result is compared: (a) a single run result; (b) an aggregated result after multiple runs; and (c) SIFT matching result.

repeatedly matched minutiae pairs. Compared with the results achieved by the proposed algorithm, the feature point correspondence discovered by SIFT is unreliable (the result is shown in Fig. 7(c)).

3.2 Minutiae Point Cloud Construction

The minutiae point cloud construction is based on the minutiae point correspondence discovered in Sect. 3.1. To calculate minutiae points' spatial coordinates in 3D space, camera parameter matrix for the built-in camera in the 3D fingerprint imaging system is firstly necessary. Mathematically, the camera parameter matrix can be denoted as follows.

$$C = K \cdot M = K \cdot [R, T] \tag{4}$$

where C stands for the camera parameter matrix. K is a 3×3 camera internal parameter matrix, which can be obtained via camera calibration based on chessboard testing. M ia a 3×3 camera external parameter matrix, which contains two components: (i) camera pose matrix R (dimension 3×3); and (ii) camera shift vector T (dimension 3×1). The camera pose matrix R is determined by the camera's rotations $(\theta_x, \theta_y, \theta_z)$ along x, y and z axis in 3D coordinate system. The camera shift vector $T = [T_x, T_y, T_z]$ is the bias apart from the origin in 3D coordinate system. The values for $(\theta_x, \theta_y, \theta_z, T_x, T_y, T_z)$ are given by the manufacturer of the used 3D fingerprint imaging system, therefore the camera external parameter matrix M can be calculated.

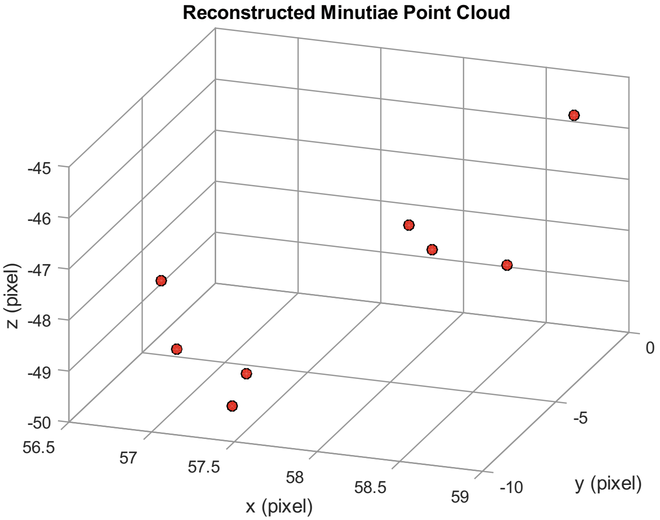


Fig. 8. An example to show the constructed 3D minutiae point cloud based on the calculated camera parameter matrixes and the discovered minutiae point correspondence in Fig. 7(b).

Furthermore, given the camera parameter matrixes and the minutiae points correspondence in Fig. 7(b), the minutiae points' 3D coordinates can be computed via triangulation reconstruction method. As an example, we demonstrate the resultant 3D minutiae point cloud in Fig. 8.

4 Conclusion and Future Work

This paper proposes a novel minutiae point correspondence discovery strategy based on differential evolution for 3D fingerprint imaging system. Such algorithm can be adopted to align the minutiae point set in one view against the point set in another one. As an alternative solution for tackling the feature point level matching, this algorithm is able to produce reliable matching result, when the SIFT-based matching approach cannot work at all. Furthermore, this algorithm can solidly support the subsequent 3D minutiae cloud construction. The effectiveness and superiority of the proposed algorithm are evidenced by the experiments.

However, the proposed algorithm still needs to be further improved in the following aspects: (i) the number of the matched minutiae pairs are still limited after taking Monte Carlo simulation; and (ii) the minutiae point matching method in 3D space is in demand.

References

1. Liu, F., Zhang, D.: 3D fingerprint reconstruction system using feature correspondences and prior estimated finger model. Pattern Recogn. **47**, 178–193 (2014)
2. Choi, H., Choi, K., Kim, J.: Mosaicing touchless and mirror-reflected fingerprint images. IEEE Trans. Inf. Forensics Secur. **5**, 52–61 (2010)
3. Liu, F., Zhang, D., Song, C., Lu, G.: Touchless multiview fingerprint acquisition and mosaicking. IEEE Trans. Instrum. Meas. **62**, 2492–2502 (2013)
4. Kumar, A., Kwong, C.: Towards contactless, low-cost and accurate 3D fingerprint identification. IEEE Trans. Pattern Anal. Mach. Intell. **37**, 681–696 (2015)
5. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2004)
6. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**, 341–359 (1997)
7. Qin, A., Huang, V., Suganthan, P.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evol. Comput. **13**, 398–417 (2009)