Study of a Full-View 3D Finger Vein Verification Technique

Wenxiong Kang, *Member, IEEE*, Hongda Liu, Wei Luo, and Feiqi Deng

***Abstract*— Finger vein modality has unique advantages, allowing it to play an important role in biometrics. However, the approach to vein imaging and information acquisition typically adopted in current vein verification systems employs a monocular camera to acquire a single-view 2D vein image from only one side of the finger, which causes two problems: it acquires limited vein pattern information for verification, and it causes clear differences among samples of the same subject captured from different finger positions in contact-free mode. Both of these problems have adverse effects on system performance. In general, existing systems are more sensitive to positional variations of the finger, particularly those caused by pitch and roll movements. This concern remains a challenge despite considerable efforts to address it in recent years. To provide a fundamental solution to the above issues, we propose an entirely new system, which includes a software and hardware platform that collects a full-view of the vein pattern information from whole fingers with three cameras, a novel 3D reconstruction method to build the full-view 3D finger vein image, and a corresponding 3D finger vein feature extraction and matching strategy based on a lightweight convolutional neural network (CNN) with depthwise separable convolution. Experimental results demonstrate the potential of our proposed system and show that compared to the traditional single-view 2D mode of finger vein recognition, the new system both efficiently improves the recognition performance and simultaneously takes full advantage of additional valid information provided by the finger vein biometrics.**

***Index Terms*— Finger vein authentication, 3D reconstruction, texture mapping, feature extraction and matching.**

I. Introduction

With the development of computer science and information technology, the importance of user authentication has increased, and traditional authentication methods (such as keys, passwords, personal identification numbers, and smart cards) have struggled to meet convenience, anti-spoofing and high-security requirements. Biometric approaches have been comprehensively studied and widely used to meet these needs; the most representative approaches involve fingerprint and face recognition. Biometrics uses one or more human physiological characteristics (such as fingerprints, faces, iris, and veins) or behavioral characteristics (such as gait and signature) for identity authentication. Among these, finger vein recognition technology, which uses the vein structure pattern under the finger surface as evidence, has begun to play an important role in the field of biometrics because 1) it is difficult to forge, as it uses the vascular structure pattern beneath the skin; 2) it can function under noncontact or weak contact measurement conditions, which are user-friendly features; and 3) it is resistant to interference because it is less vulnerable to scars, oil or sweat on the finger's surface [1]. Recent studies on the development and application of finger vein recognition have contributed to the rapid and promising growth of this technology and have demonstrated its great market potential.

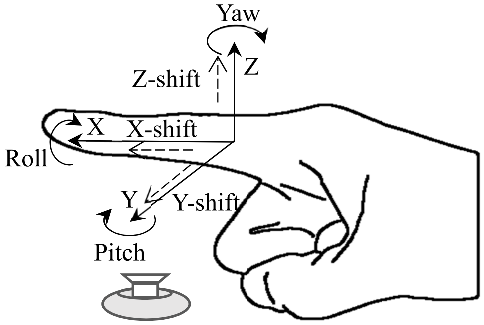


Fig. 1. Schematic diagram of finger poses.

A. Motivation and Related Work

At the present stage, finger vein recognition technology typically employs a monocular camera to capture a finger vein image on one side of the finger upon exposure to near infrared light. In this mode, the acquired vein information is limited because the finger resembles an elliptical cylinder, allowing approximately only one-third of its vein pattern to be acquired by the monocular camera. More importantly, locating a stable and reliable region of interest (ROI) for recognition is challenging due to variations in finger positioning (shown in Fig. 1, similar to the 6 degrees of freedom (DoF) movements of aircraft [2]) at different acquisition times, which is called the multipose problem in this paper. Existing traditional 2D image feature extraction methods are generally

Manuscript received December 28, 2018; revised April 30, 2019 and June 19, 2019; accepted June 20, 2019. Date of publication July 15, 2019; date of current version December 11, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61573151 and Grant 61976095, in part by the Guangdong Natural Science Foundation under Grant 2016A030313468, in part by the Science and Technology Planning Project of Guangdong Province under Grant 2018B030323026, and in part by the Fundamental Research Funds for the Central Universities under Grant 2018PY24. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Andrew Beng Jin Teoh. *(Wenxiong Kang and Hongda Liu contributed equally to this work.) (Corresponding author: Wenxiong Kang.)*

W. Kang and W. Luo are with the College of Automation Science, South China University of Technology & Engineering, Guangzhou 510641, China (e-mail: auwxkang@scut.edu.cn).

H. Liu and F. Deng are with the South China University of Technology & Engineering, Guangzhou 510641, China.

Digital Object Identifier 10.1109/TIFS.2019.2928507

sensitive to rotation and displacement deviations caused by this problem [3], which ultimately degrades system performance. To solve the multipose problem, we divide pose variations into two categories: shift-based pose variations, such as the *X/Y/Z-shift*, which represent displacement variations along the X/Y/Z axis, and rotation-based pose variations, such as pitch, roll, yaw which represent rotation around the finger axis. Among these varied poses, the Y-shift and X-shift cause a positional offset of the finger on a 2D image, the Z-shift causes a scale change of the finger on the 2D image, yaw causes a finger rotation on the 2D image, pitch causes a perspective distortion, and roll causes deformation of and regional changes in the vein pattern. To the best of our knowledge, no perfect solution exists for all types of pose variations, especially those caused by pitch and roll.

Existing research on the multipose problem can be divided into five categories:

1) ROI Image Alignment: As the earliest and most studied method, ROI image alignment addresses the multipose problem via two approaches. One approach is to employ a specially designed position constraint to limit the position of the fingers to ensure that the same region of the finger vein is captured each time [4]– [7]. The second approach is to design effective ROI interception algorithms. With regard to the Y-shift, the key step is finger region segmentation [8] or finger edge detection, such as Sobel [9], Laplacian [10], and other self-designed operators [11], [12]. For a Z-shift, a simple image rescaling operation can remove the scale variation. For X-shift, typical methods include fingertip detection [13] and knuckle positioning [14]. For yaw, a common method is to rectify the image according to the midline of the fitted finger [13]. All of the above methods can partially solve the problems caused by one or several pose variations of the finger, but they are unable to address all of the pose variation problems, particularly those caused by pitch and roll.

2) Image Correction With Prior Knowledge: The ROI image alignment methods described above overlook the vein pattern distortion on the finger sides caused by the mechanisms used to image finger veins. To address this problem, an elliptical model has been used to expand the 2D finger vein image to correct the vein pattern distortion, followed by cropping of the valid region from the corrected image for matching, which resulted in a promising improvement [3], [15]. Similarly, Matsuda *et al.* [16] expanded the image by using a circular model instead of an elliptical model. Furthermore, Huang *et al.* [17] proposed the use of a 3D model to normalize the single finger vein image; however, the core technology still involved adopting the elliptical model with a fixed eccentricity and fixed rotation angle to eliminate vein distortion. These image correction-based methods were designed to address the roll problem, and they made definitive improvements to vein texture deformation. However, regardless of the adopted type of physical model, these methods had inevitable defects. First, they decreased the common area with a valid texture. Second, the quality of the marginal area of the finger image was usually poor, stemming primarily from noise generated by the image correction for the marginal area. Third, the 2D expansion strategy based on the ellipse or circle could not exactly reflect the reverse operation of the 3D projection; consequently, it could not completely eliminate vein distortion.

3) Extraction of Invariant Features: These methods focused on taking full advantage of certain feature descriptors that are robust to shift, rotation, scaling variation and slight deformations for finger vein feature extraction and matching. Examples are the minutiae [18], [19], scale-invariant feature transform (SIFT) [20], and deformation-tolerant feature-point (DTFP) methods [16]. These methods could address multipose problems to some extent. However, in some cases, the number of feature points was insufficient to obtain good discrimination; thus, many false pairs could be obtained. Moreover, these algorithms were sensitive to image blurring and could not cope with finger pitch and roll variations at larger angles.

4) 3D Finger Reconstruction: Researchers adopted a multi-view geometric technique to reconstruct a 3D point cloud for finger vein matching [21], [22], but when using a feature-based stereo matching technique, it is difficult to find a sufficient number of matched points between two images to obtain all of the deep vein information, while area-based stereo matching techniques make it difficult to extract vein lines robustly in a practical application. Furthermore, the approach required two adjacent cameras that faced a single side of the finger; thus, they were unable to acquire vein pattern information from the entire finger.

5) Adoption of a Deep Learning Method: With the rapid development and wide application of deep learning methods, deep learning was also introduced for finger vein image quality assessment [25] and recognition, which partially overcome the multipose problem [24]– [27]. However, to completely solve the multipose problem, it is necessary to collect sufficient data with different postures. Existing public finger vein databases are usually small and seldom take the multipose problem into account, which increases the difficulty of operability in this scheme.

B. Our Work and Contributions

As discussed above, finger vein verification systems based on a monocular camera have received in-depth study and widespread attention, but this approach is flawed from two perspectives: it acquires limited vein pattern information, and there are evident differences between samples within a class. To address the above issues, we developed a full-view 3D finger vein representation and matching model. First, we designed a hardware system with three cameras mounted on the three vertexes of an equilateral triangle facing the center. Then, we utilized the three cameras to simultaneously capture full-view images that contain all of the finger vein patterns distributed under one finger surface. Next, we applied a novel 3D reconstruction algorithm and a corresponding texture mapping algorithm which use the three finger vein images to generate a full-view 3D finger vein panorama. Finally, we propose a 3D feature extraction and matching strategy to perform 3D finger vein verification. The key contributions of this paper can be summarized as follows:

1) We present a full-view 3D finger vein verification system, which, to the best of our knowledge, is the first system proposed and implemented in the field of biometric identification.

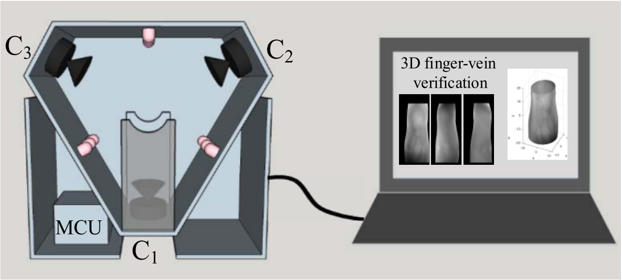


Fig. 2. Prototype of the proposed system.

2) We design a 3-camera acquisition method that can simultaneously obtain all finger vein patterns of one finger.

3) We propose a new 3D finger vein reconstruction technique and a corresponding texture mapping algorithm to generate 3D finger vein structure for the first time.

4) We develop a targeted feature extraction and matching method for full-view 3D finger vein verification.

II. System Overview and Block Diagram

Fig .2 presents the prototype of the full-view 3D finger vein acquisition and recognition system, showing both the hardware and software along with the core algorithms. In the following section, we describe the hardware and software parts and then introduce the core algorithms in subsequent sections.

A. Hardware Structure

The hardware of the system consists of two main components: a host computer and a slave machine. The slave machine consists of the light controlling unit (LCU) and image acquisition unit; the LCU uses a microcontroller unit (MCU) to control the light intensity of the near infrared LEDs, which have a wavelength of 850 nm and a bandwidth of 50 nm, and the image acquisition unit consists of three USB cameras equipped with an 850 nm NIR filter. The host computer communicates with the MCU via a serial port and with the cameras via a USB protocol. As shown in Fig.2, the three cameras (C$_1$, C$_2$ and C$_3$) are mounted on the three vertexes of an equilateral triangle. The cameras all face the triangle center, and the central axes of their visual fields are coplanar. Additionally, three LED groups are Installed Facing Their Corresponding Cameras.

B. Software Design

The software framework of the proposed system is shown in Fig. 3. In the slave machine, the MCU controls the brightness of the LEDs through pulse width modulation (PWM) and keeps three LED groups working when receiving the acquisition instructions from the host computer to enable full preparation for image collection. The host computer acts as the main controller: it controls the image acquisition and adjusts the light brightness. As the main processor, the host computer completes the image preprocessing, 3D reconstruction, and feature extraction and matching tasks. In addition, a targeted image-rectifying algorithm was developed to correct the camera distortion caused by the camera misalignments and calculation deviations. Because our proposed 3D finger vein reconstruction algorithm is very sensitive to the edge detection precision, we adopt the algorithm proposed in [12], which has been proven to be effective. Given the known edges, we propose the ellipse approximation-based algorithm to reconstruct a 3D finger model; then, we map the 2D vein texture onto the 3D finger model to generate the 3D finger vein structure. Finally, we perform 3D finger vein verification with the proposed targeted feature extraction and matching strategy.

III. 3D Finger Vein Reconstruction

As a result of the imaging mechanism of the finger vein acquisition device, only those veins distributed in the superficial subcutaneous layer of the fingers can be clearly imaged and captured; thus, it is reasonable to map the 2D vein texture captured by the three cameras to our proposed 3D finger model and finally reconstruct the full-view 3D finger vein model. Each axial vertical profile of one finger can be treated approximately as an ellipse; therefore, we propose building a 3D finger model by sequentially stacking multiple ellipses whose centers are not collinear.

A. 3D Finger Model (3DFM)

1) Mathematical Model Under an Ideal State: First, the imaging model of the cameras with small distortions can be represented by the pinhole model [28], as shown in Fig. 4. Furthermore, we can adopt the model shown in Fig. 5 to approximate the pinhole model based on the following considerations: 1) the specifications and models of the three cameras are the same, 2) the axis of the finger is approximately perpendicular to the plane of the three cameras' axes when the finger is placed on the device, and 3) the results are rendered despite the unbalanced sampling in the axial direction. In this way, we can use one ellipse to approximate each profile of the finger, as shown in Fig.6, where C_1, C_2 and C_3 are the cameras' positions, which can be considered to be three vertices of an equilateral triangle (although not perfect). Here, b_i and u_i represent the bottom and upper edge points, respectively, of the finger vein image in one column captured by camera i ($i = 1, 2, 3$). We call these six edge points *Set-EP*, while we call the 2D coordinate system*2D-CS*; this coordinate system is built from a 3D coordinate system (*3D-CS*), which will be introduced in Section III-A-3 by removing the x-axis. In addition, B_i and U_i represent the points corresponding to b_i and u_i in *2D-CS,* respectively. We finally build the 3D finger model by stacking each profile ellipse.

Overall, the emphasis is on building a mathematical model for each profile ellipse. When the finger profile is a real ellipse, then the six lines, C_iU_i, and C_iB_i (where $i = 1, 2, 3$; we call these lines constrained lines) must be tangent to the ellipse; in other words, the ellipse is an inscribed ellipse in the hexagon formed by the constrained lines. However, in reality, the finger profiles are not perfect ellipses, and in addition, the camera calibration and edge detection processes cause errors; therefore, the actual calculation process has difficulty achieving an ideal state. In other words, not every hexagon illustrated by the bold line segment in Fig. 6 has an inscribed

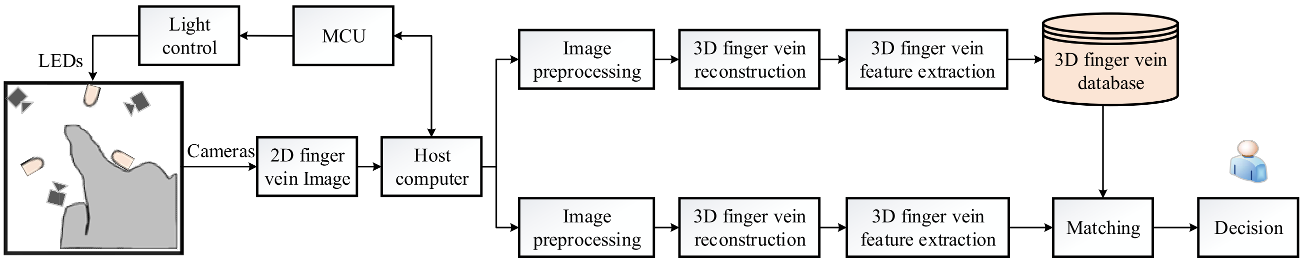


Fig. 3. Framework of the proposed system.

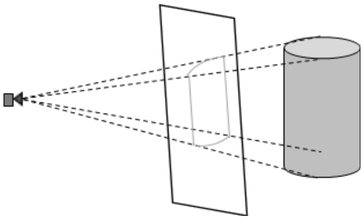


Fig. 4. Pinhole model.

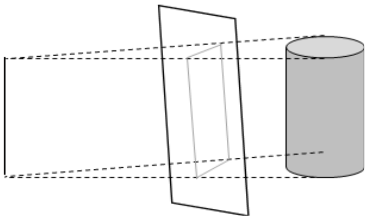


Fig. 5. Approximate model.

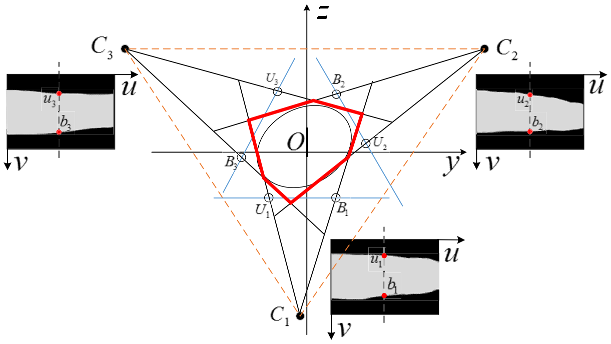


Fig. 6. Elliptical approximation in 2D-CS.

ellipse [29]. The more general situation is to find an optimal ellipse that is close to all of the constrained lines of the hexagon. Therefore, we propose a new algorithm to compute an approximate intrinsic ellipse of the hexagon, which operates as follows:

a) Initialize the Elliptical Equation and Line Equation: Suppose that the elliptical equation is

$$\begin{bmatrix} y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & c & 0 \\ c & b & 0 \\ d & e & f \end{bmatrix} \begin{bmatrix} y \\ z \\ 1 \end{bmatrix} = 0. \tag{1}$$

The projection center of each camera is recorded as $C_i(y_i, z_i)$. In this case, the equations of lines $C_iU_i(L_{ui})$ and $C_iB_i(L_{bi})$ can be calculated, as shown in Formulas (2) and (3), respectively. Here, we consider only the condition in which the slope of every line exists:

$$L_{ui} \ \ z = k_{ui}y + b_{ui} \tag{2}$$

$$L_{bi} \ \ z = k_{bi}y + b_{bi} \tag{3}$$

where $i = 1, 2, 3$; k_{ui} and b_{ui} represent the slope and intercept of the line L_{ui}, respectively; and k_{bi} and b_{bi} represent the slope and intercept of the line L_{bi}, respectively.

b) Constraint Condition: We find the lines parallel to the constrained lines, similar to the dashed lines in Fig 7, that are tangent to the ellipse. Suppose that the equations of these lines are as follows:

$$\tilde{L}_{ui} : z = k_{ui}y + b_{ui} + \xi_{ui} \tag{4}$$

$$\tilde{L}_{bi} : z = k_{bi}y + b_{bi} + \xi_{bi}. \tag{5}$$

Based on the condition that lines $\tilde{L}_{ui}$ and $\tilde{L}_{bi}$ are tangent to the ellipse, we obtain the following set of constraint equations (6):

$$\begin{cases} B_{ui}^2 - 4A_{ui}C_{ui} = 0 \\ B_{bi}^2 - 4A_{bi}C_{bi} = 0 \\ -0.1 \leq \xi_{ui}, \xi_{bi} \leq 0.1 \\ i = 1, 2, 3, \end{cases} \tag{6}$$

where

$$\begin{cases} A_{ui} = a + bk_{ui}^2 + 2ck_{ui} \\ A_{bi} = a + bk_{bi}^2 + 2ck_{bi} \\ B_{ui} = (2k_{ui}b + 2c)(b_{ui} + \xi_{ui}) + ek_{ui} + d \\ B_{bi} = (2k_{bi}b + 2c)(b_{bi} + \xi_{bi}) + ek_{bi} + d \\ C_{ui} = b(b_{ui} + \xi_{ui})^2 + e(b_{ui} + \xi_{ui}) + f \\ C_{bi} = b(b_{bi} + \xi_{bi})^2 + e(b_{bi} + \xi_{bi}) + f. \end{cases}$$

c) Construction of the Optimized Objective Function: As described above, the ellipse must be close to all of the constrained lines; thus, our goal is to minimize the sum of the distances between the ellipse and all of the constrained lines:

$$\min J = \sum_{i=1}^{3} \left(\frac{\xi_{ui}^2}{1+k_{ui}^2} + \frac{\xi_{bi}^2}{1+k_{bi}^2} \right). \tag{7}$$

2) Solution Algorithm for 3DFM:

a) For Each Profile Ellipse: As Fig. 6 shows, we must translate the edges' coordinates, i.e., (u_0, v_0), in the rectified coordinate image into the *2D-CS*. The transform relation is shown in Equation (8):

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sin\theta_i & \cos\theta_i \\ -\cos\theta_i & \sin\theta_i \end{bmatrix} \begin{bmatrix} v_0 - v_m \\ L_0 \end{bmatrix}, \tag{8}$$

where θ_i denotes the intersection angle between $\overrightarrow{OC_i}$ and the y-axis direction ($i = 1, 2, 3$); υ_m denotes the υ-value of the optical center (the principal point) in the pixels, which refers to the calibrated camera intrinsic parameter (refer to $\overline{K}$ in Section III-A-3)); and $L_0 = \left|\overrightarrow{OC_i}\right|$.

For each ellipse, we use the sequential quadratic programming algorithm to compute the optimal ellipse based on the constraint equation (6) and the destination equation (7). The main problem is how to set the initial values for those parameters when searching for the optimal result because setting the proper initial values is sufficient for improving searching speed and obtaining a globally optimal result with a greater probability. Inspired by the study in [29] and based on a large number of experiments, we propose the following method to propose the initial iteration values.

There are five independent parameters to determine an ellipse: the abscissa and ordinate values of the ellipse center, the length of the major axis, the eccentricity and the rotation angle. As noted in Section III-A-1, our problem can be translated into computing an approximate intrinsic ellipse of a hexagon. From Brianchon's theorem, a convex hexagon *ABCDEF* can have an inscribed ellipse if and only if the major diagonals *AD*, *BE* and *CF* are concurrent (as shown in Equation (9)):

$$[(\overrightarrow{C_0A} \times \overrightarrow{C_0D}), (\overrightarrow{C_0B} \times \overrightarrow{C_0E}), (\overrightarrow{C_0C} \times \overrightarrow{C_0F})] = 0 \tag{9}$$

However, while these major diagonals of the hexagon intersect, they generally do not intersect at a single point. Consequently, we set the initial center point of the ellipse (C_0) to be the midpoint of $\Delta P_1P_2P_3$ (as shown by the inset in Fig 7), which is determined by every intersection of two major diagonals, as shown in Equation (10), and set the initial length of the major axis as the minimal value of the distances between C_0 and every vertex of the hexagon, as shown in Equation (11):

$$C_0 = \frac{1}{3}(AD \times BE + AD \times CF + BE \times CF) \tag{10}$$

Where × is operator to get the location of intersection.

$$R_0 = \min\{\overline{C_0A}, \overline{C_0B}, \overline{C_0C}, \overline{C_0D}, \overline{C_0E}, \overline{C_0F}\}. \tag{11}$$

In addition, we set the eccentricity and rotation angle to constant ($e_0 = 1.4$ and $\alpha = 0$, respectively), as determined by the experiments.

b) For the Entire 3DFM: There is a clear trade-off, as using a larger number of profiles in the reconstruction to achieve a 3D finger model with higher accuracy; however, the time consumption simultaneously increases. For practical reasons, in terms of reconstruction time, a large cost to obtain the 3DFM is not realistic. Fortunately, we observed that only small variations exist among the finger profiles in the axial direction; therefore, we can first pick sparse profiles and calculate the corresponding approximate ellipses and then extend the profile data by interpolation. However, another problem is that if there is a large error in certain extracted edge points due to poor image quality, the reconstructed finger model will have large flaws in certain places. To reduce the impact of edge noise and find the optimal trade-off between precision and time consumption, we propose a more robust algorithm for 3D finger reconstruction:

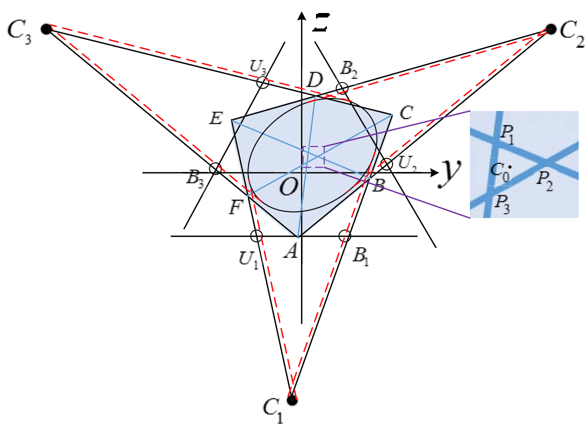


Fig. 7. Abridged general view of the process for solving the elliptical equation in 2D-CS.

Suppose that in each rectified image, a valid region exists from $u = 91$ to $u = 490$. We first divide the valid region into N parts according to the u-axis. In each subregion, we use the variance in the intersections of the hexagon's three diagonals as a reference to select the *Set-EP* which is most likely to estimate an optimal ellipse; then, we compute the approximate ellipse based on the selected *Set-EP*. After acquiring the ellipses of all of the subregions, we use a linear interpolation algorithm to expand the ellipse data. Finally, we transform the ellipse in *2D-CS* into *3D-CS* using Equation (12).

$$\begin{cases} x = \dfrac{u_k - \overline{K}(2,3)}{\overline{K}(2,2)} \times \left|\overrightarrow{C_iO}\right| \\ \begin{bmatrix} y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & c & 0 \\ c & b & 0 \\ d & e & f \end{bmatrix} \begin{bmatrix} y \\ z \\ 1 \end{bmatrix} = 0 \end{cases}. \tag{12}$$

where $\overline{K}$ is the intrinsic parameter of the calibrated camera (refer to Section III-A-3) and u_k is the abscissa value of the image. In other words, we set the same x value for one ellipse without changing the ellipse equation.

In short, our proposed 3DFM reconstruction algorithm is as follows:

3) Image-Rectifying Algorithm: Ideally, the three cameras are mounted strictly toward the origin of the coordinate system so that the central axes of their visual fields are coplanar, and the intrinsic parameters of the three cameras are identical. However, this configuration is not the case in the actual system; therefore, we propose an image-rectifying algorithm to address the problem, which consists of the following three steps:

a) Camera Calibration: First, we use the MATLAB calibration toolbox [30] to calibrate each of the three cameras and obtain their intrinsic parameters. These values are denoted as K_1, K_2, K_3. Then, we calibrate each pair of cameras and obtain their relative parameters R_{ij}, t_{ij}, where $i = 1, 2$; $j = 2, 3$; $i \neq j$;R_{ij} represents the rotation matrix in which camera i refers to camera j, with 3 DoF; and t_{ij} represents the translation matrix in which camera i refers to camera j.

Algorithm 1 3DFM Reconstruction

```
Input: the array of edges of the rectified images:$Edges \rightarrow \{(U_1^i, B_1^i, U_2^i, B_2^i, U_3^i, B_3^i)\,|i = 91, 92, \cdots, 490\}$,
       where $i$ denotes the value of the $v$-axis in the image that corresponds to the edges and the rectified cameras' parameters.
Output: a list of ellipses in three dimensions $\{(x_j, E_j)\,|j = 1, 2, \cdots, 800\}$, where $x_j$ is the $x$-axis of the ellipse in 3D-CS (refer to Section III-A-3) and $E_j$ is the ellipse equation in 2D-CS (refer to Section III-A-1).
Step 1: average separating of the array of edges into N parts, with each part denoted as an array $Sub\_Edges[20]$, which consists of 20 Set-EPs.
Step 2: for each $Sub\_Edges[s]$ in $Sub\_Edges[20]$ do for each Set-Eps[p] in Set-Eps [20] do
        · $i\_col = (s-1) \cdot 20 + \text{p}$;
        · transform the image coordinates $((i\_col, U_j^{i\_col}),(i\_col, B_j^{i\_col}), j = 1, 2, 3)$ of the edge points into 3D-CS using Equation (8);
        · calculate the six vertexes of the hexagon, i.e., the coordinates of $A$, $B$, $C$, $D$, $E$, $F$ in Fig. 7;
        · calculate the diagonal intersections of the hexagon, i.e., $P_1$, $P_2$, $P_3$, in Fig 7;
        · calculate the variance in the three diagonal intersections $\{i\_col, Var\}$.
    end for
        · select the Set-EP with the minimum variance: $\{(k, U_1^k, B_1^k, U_2^k, B_2^k, U_3^k, B_3^k)\}, k \in [(s-1) \cdot 20 + 1, s \cdot 20]$, where $k$ is an integer, and set $y_s = k$;
        · estimate the approximate ellipse via the single profile ellipse algorithm and obtain the $x$-axis and elliptic pair $\{(k, E_s)\}$;
    end for
        · obtain a list of sparse $v$-axis and elliptic pairs: $\{(u_1, E_1), (u_2, E_2), \cdots, (u_{20}, E_{20})\}$;
Step 3: use a linear algorithm to interpolate the missing ellipses, where $u$ ranges from 91 to 490 with an interval of 0.5, and acquire 800 ellipses;
Step 4: transform the ellipse from a 2D into a 3D coordinate system using Equation (12).
```

b) Establishment of the Coordinate System (3D-CS): The vectors t_{12}, t_{13}, and t_{23} are coplanar in theory but are likely not to be coplanar after camera calibration; thus, we propose the following method to acquire the three cameras' locations.

First, we define the coordinate system $C_1x'y'z'$ of camera 1 as a reference and use t_{12} and t_{13} to determine A' and B', respectively, to obtain their midpoint D. Then, we translate t_{23}, extend it and divide it equally by point D. Next, we define the start and end points of the translated t_{23} as A'' and B'', respectively, and set the midpoints of $A'A''$ and $B'B''$ to be C_2 and C_3, respectively. Finally, we establish the 3D-CS $Oxyz$ by setting $C_1C_2C_3$ as the yz plane and set the intersection of the three mid-perpendiculars of triangle $C_1C_2C_3$ as the coordinate origin O. Then, the mid-perpendicular of C_2C_3 is the z-axis, as shown in Fig 8. From the process of creating the system coordinates, we can obtain the transformation matrix t_s and the rotation matrix R_s from $C_1x'y'z'$ to $Oxyz$. In addition, we can obtain the three cameras' external parameters, $t_i = \overrightarrow{OC_i}$, $R_1 = R_s^T$, $R_2 = R_1R_{12}$, and $R_3 = R_1R_{13}$.

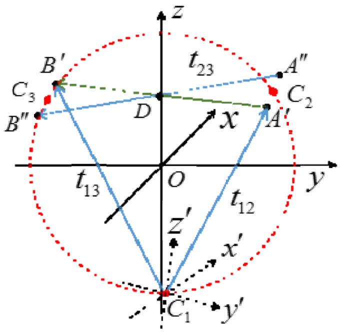


Fig. 8. Abridged general view of generating the (3D-CS) coordinate system.

c) Image Rectification: Suppose that the rectification equation is

$$p_i' = H_i p_i, \tag{13}$$

where $i = 1, 2, 3$ and H_i represents the 3×3 transformation matrix from the source image coordinate (in a homogeneous coordinate system) p_i to the rectified image coordinate p_i'.

Now, when P is a point in world coordinates, we can apply a projection formula specific to the source image:

$$p_i = K_i \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix} P, \tag{14}$$

After this transformation, the location of the camera remains unchanged; in other words, the transformation matrix t_i is unchanged, but the rotation matrix changes to $\overline{R_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_i) & -\sin(\theta_i) \\ 0 & \sin(\theta_i) & \cos(\theta_i) \end{bmatrix}$, where θ_i is the angle between $\overrightarrow{C_iO}$ and y axis. As a result, the projection formula changes to

$$p_i' = \overline{K} \begin{bmatrix} \overline{R_i} & t_i \\ 0 & 1 \end{bmatrix} P, \tag{15}$$

where $\overline{K}$ denotes the intrinsic parameter of the three cameras, which we set manually. The solution for H_i is

$$H_i = \overline{K}\,\overline{R_i} R_i^T K_i^{-1}. \tag{16}$$

B. Texture Mapping

At this point, we have completed the reconstruction of 3DFM. In this section, we describe how to map the texture from the 2D finger vein image to 3DFM. As shown in Fig 6, each pair of 2D finger images consists of overlapped and nonoverlapped areas. Thus, our primary task is to find accurate boundaries between the images and then determine which pixel in 3DFM mapped from one or two 2D images and (if mapped from two 2D images) their computational weights. Here, we provide an explanation of the texture mapping strategy on a single ellipse because the same strategy applies to all ellipses.

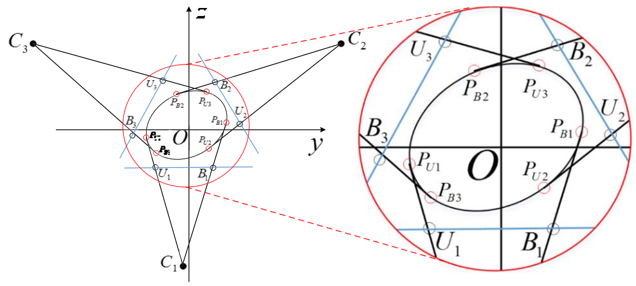


Fig. 9. Boundaries of each camera's view.

1) *Overlap and Nonoverlap Discrimination:*

a) Discrete Sampling: We must determine a method for discrete sampling on the ellipse. We first transform every ellipse into a standard form whose center point is the origin of coordinate system and whose major and minor axes are located on the coordinate system's axis. That is, we calculate the rotation derivation and shift derivation between the ellipse and its standard form using a specific procedure [31]. We can transform the ellipse from standard form to the polar coordinate system and then conduct N uniform sampling with an angle interval of $\frac{2\pi}{N}$. Finally, we transform those discrete points into the original ellipse based on the calculated rotation and shift derivation.

b) Boundary Determination: By using the projection equation (17), we can map the points in the 3D coordinate system into a 2D plane and acquire a 2D image. However, as illustrated in Fig. 9, only one part of the whole finger vein can be imaged by the corresponding camera on that side. Thus, we must determine the boundary that belongs to each camera:

$$\begin{cases} Z_b \begin{bmatrix} \overline{u} \\ \overline{v} \\ 1 \end{bmatrix} = K_i \begin{bmatrix} \overline{R_i} & t_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y \\ z \\ 1 \end{bmatrix} & (a) \\ [u, v] = round([\overline{u}, \overline{v}]) & (b) \end{cases} . \quad (17)$$

Identifying the boundary in the 3D coordinate system is not easy; however, we can acquire the boundaries by reverting to the 2D system and ignoring certain errors.

For each ellipse, as shown in Fig. 9, on each pair of tangents from the camera's position to the ellipse, there are two tangent points: P_{Ui} and $P_{Bi}(i = 1, 2, 3)$. In addition, (minor arc) is the imaging range of camera i, called $Area_i$, which also determines the imaging boundary of each camera. On this basis, we can easily distinguish the overlap and nonoverlap of each pair of 2D finger vein images. For example, is an overlapping region: a pixel value in this region will be estimated from both images 1 and 3. In contrast, is a nonoverlapping region in which a pixel value is determined solely from image 1.

2) *Texture Mapping:*

a) Gray Normalization: To provide the finger vein images captured by three cameras with the same grayscale distribution, the light intensity of the three groups of LEDs should be kept same during image acquisition. However, the finger usually appears as an elliptical cylinder shape, causing it to exhibit different thicknesses from three cameras with different viewing angles; thus, the grayscale distributions of the images captured by the different cameras are not the same. Here, we adopt the minimum-maximum normalization strategy to normalize the grayscale distributions of the images.

b) Texture Mapping Strategy: As in the analysis above, for nonoverlapping regions, mapping the texture from a single corresponding image is sufficient, while for overlapping regions, all of the pixel values in the region are estimated by a weighted fusion of the two images. In detail, suppose that (x_k, y, z) is a point on; we then use intrinsic and external parameters of camera 1 to calculate the image coordinates (u, v) according to Equation (17). Finally, we set the pixel value of the point in *3D-CS* to be $I(x_k, y, z) = I_1(u, v)$. Now, suppose that (x_k, y, z) is a point on. Then, we obtain the image coordinates (u_1, v_1) and (u_3, v_3) under camera 1 and camera 3, respectively, and finally, we set the pixel value of the 3D point to

$$I(x, y, z_k) = \omega_1 I_1(u\ _1, v_1) + (1 - \omega_1) I_3(u_3, v_3). \quad (18)$$

where ω_1 is a weight,

$$\omega_1 = \frac{i}{N}, \quad (19)$$

N is equal to the number of discretization points in the overlapping region, and i denotes the point's index of the overlapping region. We set the indexes of those discretization points from point P_{B3} to P_{U2} in the nonoverlapping area to $P_{B3}P_{U2}$, which are integer values from 0 to N. For the other overlapping and nonoverlapping areas, we can calculate the pixel values similarly to the method used for $P_{U1}P_{B3}$ and $P_{B3}P_{U2}$.

IV. 3D Finger Vein Feature Extraction and Matching

Studies on face and fingerprint recognition have recently begun to focus on 3D frameworks [32]– [37]. The 3D image feature extraction can be divided into two categories. One category extracts feature points from the 3D coordinates directly and then generates the feature vector by using specific methods, such as a bag of words or the matched pixel ratio, to measure the similarity between the samples directly. For instance, Kumar and Kwong [37] expanded the 2D minutiae extraction and matching algorithm into the 3D domain. The other approach is to transform the 3D images into a 2D plane according to certain rules and adopt the 2D feature extraction algorithm to generate feature vectors. For example, Wang *et al.* [36] proposed mapping the 3D fingerprint image into a cylinder first and then further expanding it into the 2D plane. The latter strategy is adopted in the present paper; in addition, it must be noted that the 3D images inferred above are mesh structures. For our constructed 3D finger vein images, rich information, including texture features and geometric features, will be used to enhance the recognition performance of our system.

Fig. 10 shows the flowchart of our proposed feature extraction and matching algorithm for 3D finger veins.

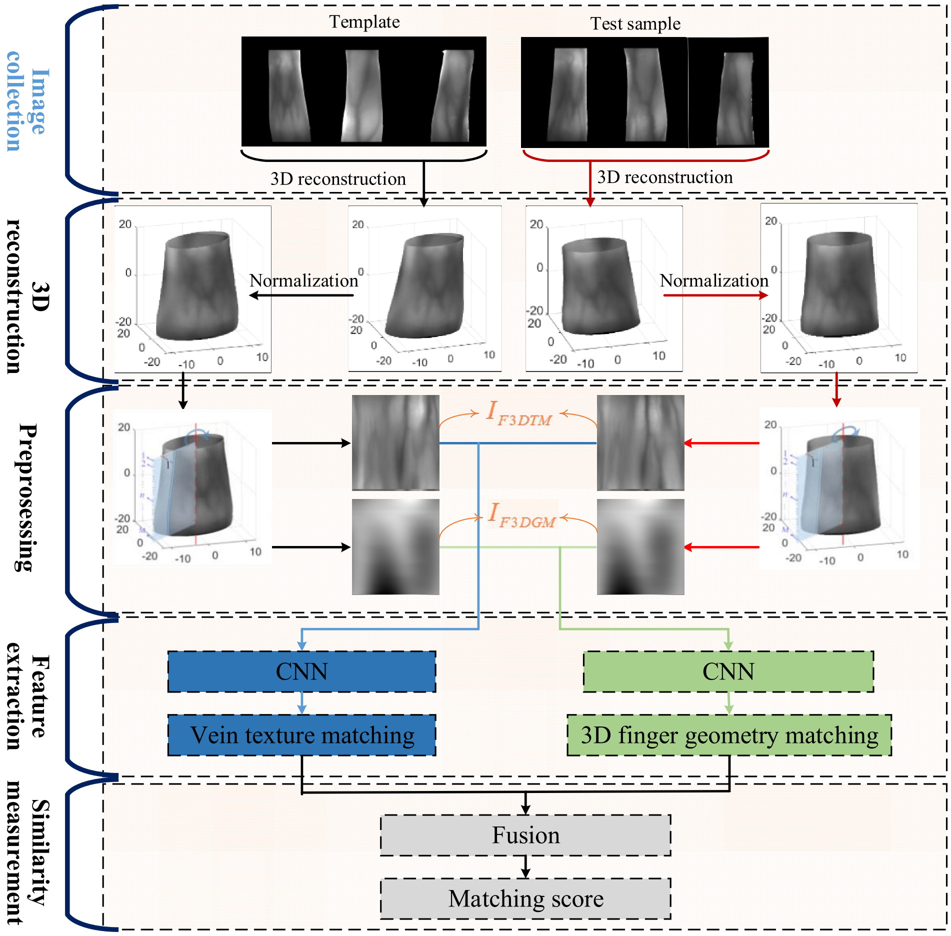


Fig. 10. Flow chart of 3D finger vein texture feature extraction and matching.

A. 3D Finger Normalization

Several strategies exist for addressing the problems described in Section I-A. After the center of each ellipse has been obtained during the 3D finger reconstruction procedure for discrete sampling points (as discussed in Section III-B-1), we can fit these center points into a straight line by the least-squares algorithm [38]. Suppose that the straight line equation we obtain is as follows:

$$\frac{x - x_m}{S} = \frac{y - y_m}{W} = \frac{z - z_m}{G}, \tag{20}$$

where (x_m, y_m, z_m) is regarded as the middle point of the ellipse and the vector (S, W, G) represents the direction of the central axis. To avoid discussion on the back angle, we stipulate that $G > 0$. The purpose of normalization is to make the central axis coincide with the x-axis of the *3D-CS* coordinate system and to allow the central point of the 3D finger model to coincide with the original point of the system coordinates. Thus, an appropriate translation and rotation operation should be performed. When performing a translation on the rotation sequence, the translation vector is $(-x_m, -y_m, -z_m)$, and the rotation axis and angle are defined as follows:

$$V = [S, W, G] \times [0, 0, 1] \tag{21}$$

$$\varphi = \arccos\left(\frac{[S, W, G] \cdot [0, 0, 1]}{|[S, W, G]| \cdot |[0, 0, 1]|}\right). \tag{22}$$

After normalization, variations caused by Y-shift or Z-shift movements and pitch or yaw movements can be eliminated, and the variations caused by rotation along the X-axis (roll movement) and X-shift movement are the only two problems that remain to be handled, which are addressed in the following feature extraction and matching step.

B. Feature Extraction and Matching

1) Flattened 3D Texture Map and Geometric Map Generation: First, we define a sectorial cylinder block as *SC-Block(i)*, where i is an index ranging from 1 to N ($N = 360/\Delta\alpha$, where $\Delta\alpha$ is the sampling angle interval). The sectorial cylinder block is a small block cut by the cylinder along the axis, as shown in the preprocessing panel in Fig. 10. We set the arc length of *SC-Block(i)* to $((i - 1) \cdot \Delta\alpha, i \cdot \Delta\alpha]$ and set the range of Z values as $[z_{\min}, z_{\max}]$, where $z_{\min}$ and $z_{\max}$ represent the minimum and maximum z values of the 3D points, respectively. Next, we map the 3D points in the *SC-Block(i)* to one column of the texture map (I_{F3DTM}) and

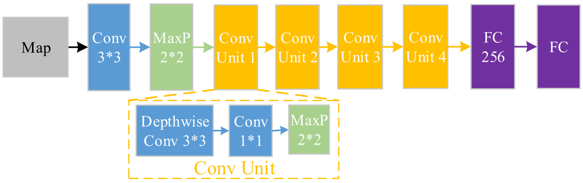


Fig. 11. CNN structure.

geometric map (I_{F3DGM}) according to the functions Γ_t and Γ_g, respectively.

$$I_{F3DTM}.col(i) = \Gamma_t(SC - Block(i)) \tag{23}$$
$$I_{F3DGM}.col(i) = \Gamma_g(SC - Block(i)), \tag{24}$$

where *F3DTM* and *F3DGM* are abbreviations for flattened 3D texture and geometric mapping, respectively, and $.col(i)$ indicates column i of the map. To implement the functions Γ_t and Γ_g, we divide *SC-Block(i)* into M parts in the z direction at a regular interval. Thus, every part of *SC-Block(i)* is mapped to one pixel in the maps. Specifically, a pixel of I_{F3DTM} is generated by the mean value of the 3D point cloud pixels in the part of *SC-Block(i)*, whereas a pixel of I_{F3DGM} is generated by the mean value of the distances from the 3D point clouds to the middle line. In our experiments, we set $\Delta\alpha = 1$ and $M = 360$.

2) Data Augmentation: In this paper, we adopt a convolutional neural network (CNN) to extract the representative features of I_{F3DTM} and I_{F3DGM} because of the acknowledged ability of CNNs in image recognition. Due to the lack of data in our constructed 3D finger vein database, we adopt data augmentation strategies to alleviate the issue of overfitting when training the CNN. Specifically, in addition to the augmentation methods commonly used in image recognition, such as translation, cropping, and rotation, we also adopt several targeted approaches, such as adding random noise and blurred sections to images, in order to simulate the possible noise and blurring in real finger vein images. Additionally, since illumination changes may result in brightness variations in the texture map, we introduce random brightness changes in the images to further augment the training data. All of the augmentation methods are initialized with Gaussian random parameters, and each map is expanded 24 times in this paper.

3) Network Construction: To consider the characteristics of finger vein texture, we construct a targeted CNN for feature extraction. Because the constructed I_{F3DTM} contains all of the vein structures distributed under the epidermis of the entire finger, the amount of feature information I_{F3DTM} is increased considerably compared with the traditional 2D model. To extract representational features from I_{F3DTM}, we build the effective and shallow network shown in Fig. 11. First, considering the efficiency of such networks on embedded platforms, we use a depth-wise convolutional layer [39] concatenated with a 1×1 convolutional layer as the basic structure of our network. This architecture reduces the time complexity while ensuring the feature extraction performance. After the convolutional layer, a fully connected layer with a 256-output unit is added as the feature layer, and we use the cosine distance to calculate the matching score for the output in subsequent feature matching. In addition, we insert several dropout layers to avoid overfitting. The entire network architecture is designed in the simplest way to verify the validity of the 3D vein texture. We believe that if sufficient data were available, using a deeper network model would provide better results.

4) Network Training: To extract effective features for finger vein recognition, we employ center loss [40] as the loss function. This approach effectively reduces the intraclass distance while increasing the interclass distance; thus, it achieves a lower equal error rate. In addition, we use stochastic gradient descent (SGD) with momentum for gradient reduction and set the batch size to 64 due to the large sizes of the input images. The training epoch is set to 30 based on the convergence process.

V. Experiments and Results

A. Database

To the best of our knowledge, no publicly available database exists for 3D finger vein images; consequently, we built a 3D finger vein database to evaluate the effectiveness of the proposed system. A total of 203 fingers, including the index finger and middle finger from the left and right hands, were utilized to collect the finger vein images. For each finger, we performed the image grabbing process (with three cameras) 14 times to generate 42 finger vein images. Thus, the new database contains a total of 8,526 finger vein images, samples of which are shown in Fig. 10 and Fig. 12. To better simulate the actual conditions, our database contains data of fingers in various poses: for the first six of 14 acquisitions, the data are collected with each finger in a normal pose; for the next six of 14 acquisitions, the data are collected by rotating around the finger axis; and for the remaining two acquisitions, the data are collected by rotating in the other two directions. The data with the normal finger pose collected during the first six acquisitions are denoted as non-multipose data, and all of the data collected in all 14 acquisitions are denoted as multipose data. In addition, the data captured by cameras 1, 2 and 3 are denoted as FV-C1, FV-C2 and FV-C3, respectively.

B. 3D Reconstruction Experiment

In this section, we present the reconstructed 3D finger vein images and then illustrate the reconstruction error analysis.

Fig. 12 presents four 2D finger vein image samples acquired from different fingers with their corresponding intermediate results and the final 3D finger vein after reconstruction. For each sample, we show the source images captured by the three cameras in the upper left panel and their corresponding intermediate results after filtering, segmenting and rectifying in the upper right panel. Finally, we show three different views of the reconstructed 3D finger vein images in the bottom panel. As shown in Fig. 12, our 3D finger vein reconstruction algorithm restores the 3D finger structure accurately, including the shape and curvature, and is consistent with the real finger. Most importantly, the vein textures are also accurately mapped onto the 3D finger model.

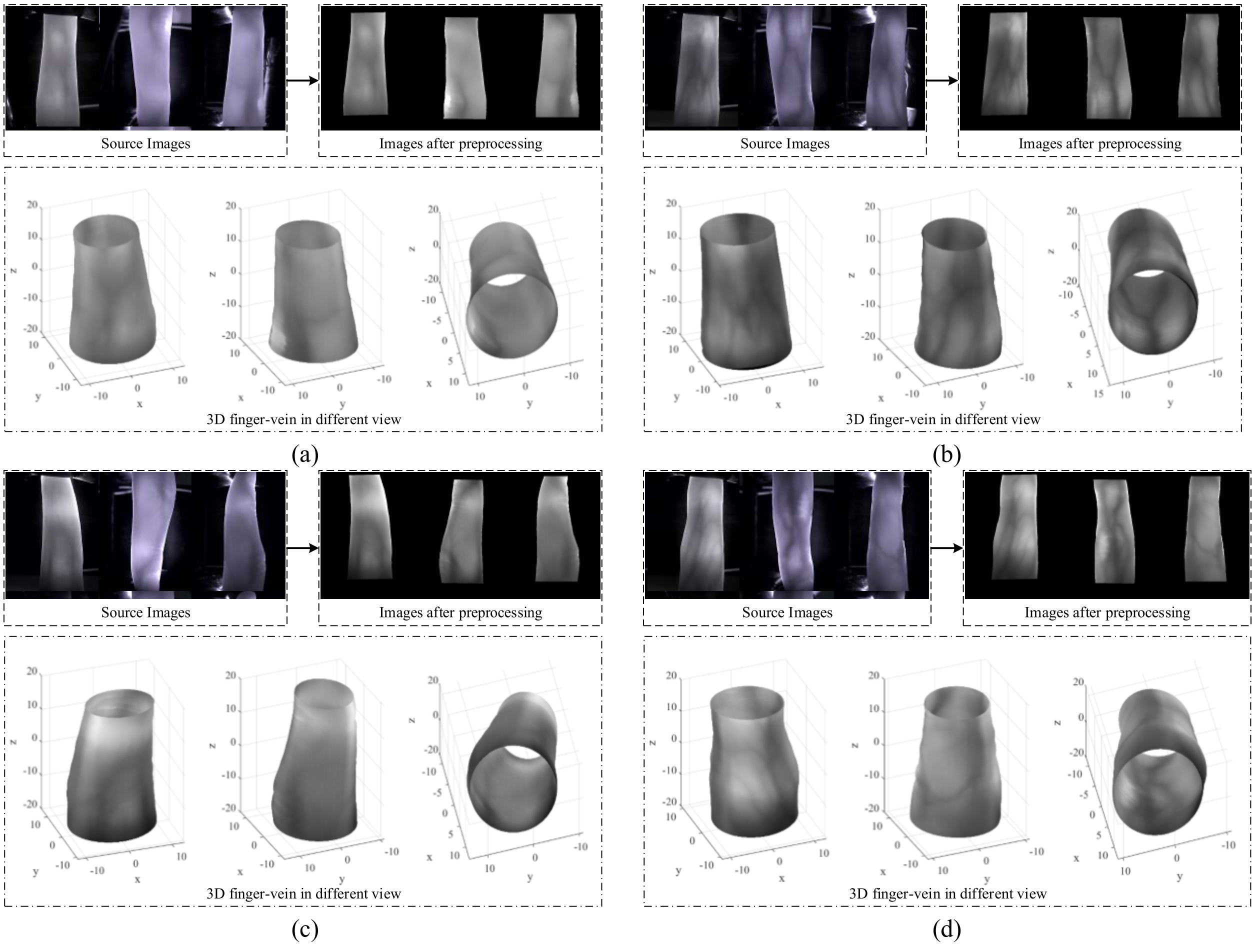

Source Images
Images after preprocessing
3D finger-vein in different view
(a)
Source Images
Images after preprocessing
3D finger-vein in different view
(b)
Source Images
Images after preprocessing
3D finger-vein in different view
(c)
Source Images
Images after preprocessing
3D finger-vein in different view
(d)
x
y
z

Fig. 12. 3D finger vein reconstruction results.

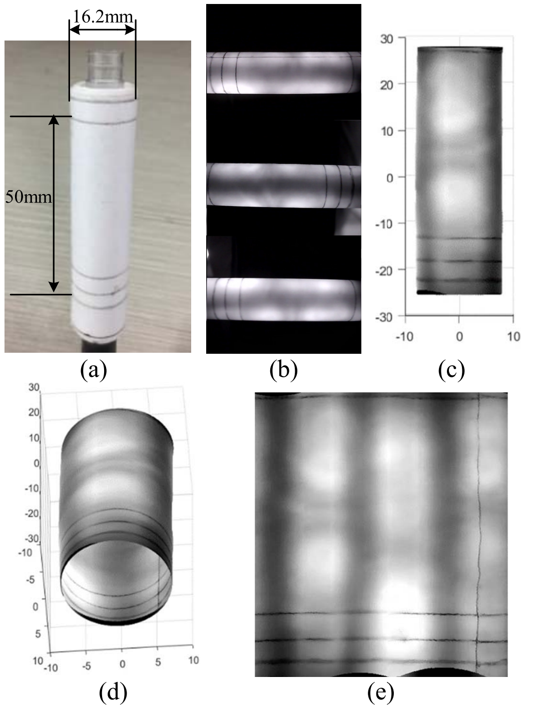
Although our algorithm can successfully reconstruct a 3D structure of the finger, some reconstruction and system errors are inevitable. Because it is difficult to accurately obtain the ground truth of a 3D finger vein structure, we utilize a textured cylindrical object with a fixed size (as shown in Fig. 13(a)) as a pseudo finger to evaluate these errors. Similar to the 3D reconstruction of a real finger, the textured cylindrical object is first imaged through our system and then preprocessed and rectified; finally, it is reconstructed to a 3D structure using our proposed 3D reconstruction algorithm, as shown in Fig. 13.

In this paper, we adopt three values—the difference between the fitted circle size and the real circle size (D_r), the difference between the fitted center and true center (D_c), and the difference between the height of the cylinder in the reconstruction model and the real object (D_h)—as indicators for the evaluation of the 3D reconstruction error. According to our experiments, the three error values (deviation ratios) are 2.05 mm^2(0.99%), 0.1359 mm (0.84%), and 0.5 mm (1%), respectively. In general, these errors are relatively small and appropriate for practical application. In addition, for texture mapping, our proposed 3D reconstruction method can effectively restore the texture in the 2D image to the surface of the 3D model. This is demonstrated by the fact that a circle on the real object remains a circle in the reconstructed 3D model after 2D to 3D conversion is performed (as shown in Fig. 12(c) and (d)), and the circle on the 3D model is almost a straight line after unfolding, as shown in Fig. 12(e). The above results show that our proposed 3D mapping strategy effectively addresses the problem of vein deformation not only on the v-axis but also on the u-axis of the image, demonstrating its reasonability and validity.

For real applications, the 3D reconstruction error might be greater for the following reasons: first, some errors in edge detection appear to be inevitable; second, a real finger profile is not a perfect ellipse; and third, the veins are not completely attached to the epidermis. However, it must be emphasized that the errors caused by the above factors do not have much impact on the recognition system; in other words, these errors are within an acceptable range.

C. Verification Experiment

To verify the fact that our proposed algorithms, including the 3D reconstruction strategy and the feature extraction and matching method, can achieve substantial improvements when solving the issues of limited texture information and the multipose problem in real-world situations, we conduct a suite of experiments. **The first experiment** tests our constructed network on different 2D finger vein databases and compares it with other state-of-the-art methods to evaluate its effectiveness



(a) (b) (c)

(d) (e)

Fig. 13. Illustration of estimating the 3D reconstruction error (a) A textured cylindrical object, (b) three images from different perspectives, (c) and (d) the reconstructed images from different views, and (e) the unfolded image of (d).

for finger vein identification. In particular, this serves to demonstrate that our newly built dataset is challenging for verification tasks. **The second experiment** was conducted to demonstrate that our system capitalizes on the full-view finger vein image to acquire more texture information and is more robust on the multipose problem. In this experiment, we compare the verification results using the three images captured by the three cameras and using the *Flattened 3D Texture Map* described in Section IV-B based on the feature extraction and matching algorithms we proposed. **The third experiment** is designed to further demonstrate the significance of the full-view 3D finger vein verification strategy. In this experiment, we first demonstrate the effectiveness of the 3D finger geometry as the sole biometric mode and then fuse it with the texture feature to further emphasize the significance of our 3D finger reconstruction strategy and show our system's potential for improving the verification performance. During the verification experiments, we randomly divided the different databases into training sets, validation sets and test sets in proportions of 0.6, 0.1 and 0.3, respectively. During training, we selected the model with the best performance on the validation sets as the final output model; then, in the testing process, we applied that model to the test set to calculate the equal error rate. All of these experiments were executed on the TensorFlow platform.

1) Validation of the Proposed Network: Because this paper concentrates on the effectiveness of the 3D finger vein verification strategy, we adopted the simple CNN described in Section IV to demonstrate the validity of our designed network. In this experiment, we execute the proposed network on three published 2D finger vein databases and the self-constructed non-multipose database captured by camera 1 (denoted as FV-C1-6) and compare the results with those of other existing methods, including three classic CNN architectures. The equal error rates are given in Table I. We also provide a comparison diagram of the proposed network on three published 2D finger vein databases and the FV-C1-6 database in Fig. 14.

TABLE I

EERs for Different Databases

Database / Method	*FV-USM*	*SDUMLA-HMT*	*PolyU*	*FV-C1-6*
Wang [41]	4.75	2.36	-	-
Qiu [3]	2.32	1.59	-	-
Qin [26]	1.42	-	2.70	-
Qin [42]	-	-	2.86	
AlexNet [43]	1.01	3.40	5.22	4.60
VGG-16 [44]	2.01	5.50	5.10	5.30
ResNet50 [45]	0.61	2.20	2.72	3.49
Our method	0.94	1.69	2.40	2.84

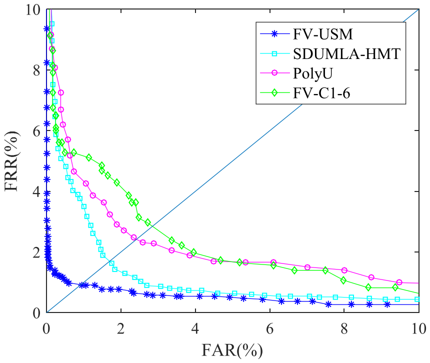


Fig. 14. EER curves for different databases.

As shown in the first three columns of Table I, compared to the other methods, our proposed network achieves promising performances (always best or second best) on the FV-USM [13], SDUMLA-HMT [45] and PolyU [46] databases. The effectiveness of our designed network is apparent despite its simple structure. Note that the classic CNNs, such as AlexNet, VGG-16 and ResNet50, are deeper than our designed network; thus, they should have more powerful feature expression capabilities, but they perform poorly in our experiment because the amount of data in the publicly available 2D finger vein databases and our custom-built database is limited. A deeper network requires more training data; otherwise, it can easily cause overfitting, which is why we designed a shallow network. Rather than training from scratch, these deep networks were fine-tuned on the pretrained weight with ImageNet to mitigate the overfitting issue, which ensures that the results achieved are more comparable to the results of our proposed network. In our experiment, AlexNet has fewer layers than VGG-16; therefore, its overfitting degree is relatively small. In addition, the block structure of the CNN also has a notable effect on learning vein patterns. From our experimental results, we find that the VGG-16 network, which is constructed by many 3×3 convolution kernels, has a weaker capacity to express vein patterns, while ResNet50, which uses

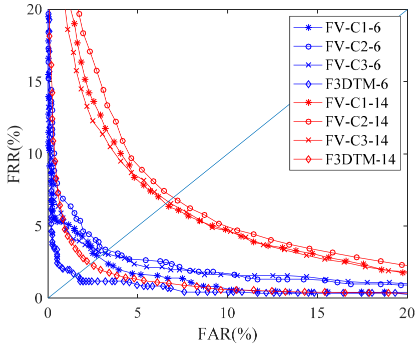


Fig. 15. EER curves for texture matching. FV-C1-6 and FV-C1-14 denote the non-multipose and multipose data captured by camera 1, respectively. A similar convention is used for FV-C2-6, FV-C2-14, FV-C3-6, and FV-C3-14.

TABLE II

EERs FOR TEXTURE MATCHING

Data	*FV-C1*	*FV-C2*	*FV-C3*	I_{F3DTM}
Non-multipose	2.84	3.29	3.17	1.60
Multipose	6.53	7.00	6.70	2.37

residual blocks, has a better ability to express vein patterns. The separable convolution block has the best effect on vein pattern expression when the amount of training data is small.

In addition, as shown in the last row of Table I and Fig. 14, our designed network achieves an equal error rate (EER) of 2.84 with the FV-C1-6 database, which is slightly higher than the values achieved with the other three published databases. This high EER result occurs because our current research focuses on implementing a model machine to verify the effectiveness of our proposed 3D finger vein reconstruction and identification methods while largely neglecting imaging quality. However, these results also demonstrate that our newly built dataset is more challenging for verification tasks.

2) Validation of Texture Reconstruction: To highlight the advantages of our proposed 3D finger vein reconstruction and recognition algorithm in providing sufficient vein information and coping with the multipose problem, we used the same network structure for the images captured by cameras 1, 2, and 3 and the I_{F3DTM} generated by our algorithm for both the non-multipose data and multipose data. The experimental results are shown in Fig. 15, and the EERs are shown in Table II.

As noted in Section V-A, 57% of the finger vein images in the self-built database were collected with various finger positioning, including 43% with rotation around the finger axis. Thus, our database is more challenging for verification tasks than the existing publicly accessible finger vein databases because the latter gives little or no consideration to the multipose problem. We compare the results when using the 2D finger vein images (FV-C1, FV-C2 and FV-C3) captured by three cameras and the I_{F3DTM} generated by our proposed algorithm and arrive at the following three conclusions. **First**, the effectiveness of the vein information on the back of the finger indicates that the vein images captured by the three cameras all contain information useful for recognition. **Second**, the I_{F3DTM} generated by our system is effective because when using I_{F3DTM}, the EER is much lower than the time using any single 2D image captured by the three cameras. Thus, the 3D finger vein reconstruction method and the flattened texture map generation method proposed in this study yield more discriminative features than traditional 2D finger vein verification systems. **Third**, our algorithm has clear positive effects on the multipose problem. As the experimental results show, our algorithm reduces the EERs in the non-multipose data by 1.24 and in multipose data by 4.16. These results emphasize the effectiveness of our algorithm for solving the multipose problem.

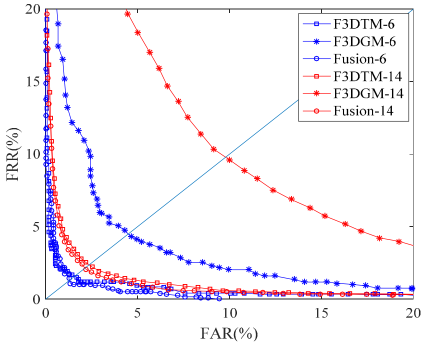


Fig. 16. EER curves for the different features. F3DTM-6 and F3DTM-14 denote using the flattened 3D texture mapping strategy on the non-multipose data and multipose data, respectively. A similar convention is used for the F3DGM-6, F3DGM-14. Fusion-6, and Fusion-14. where 'Fusion' represents the fusion of texture and geometric feature matching at the score layer.

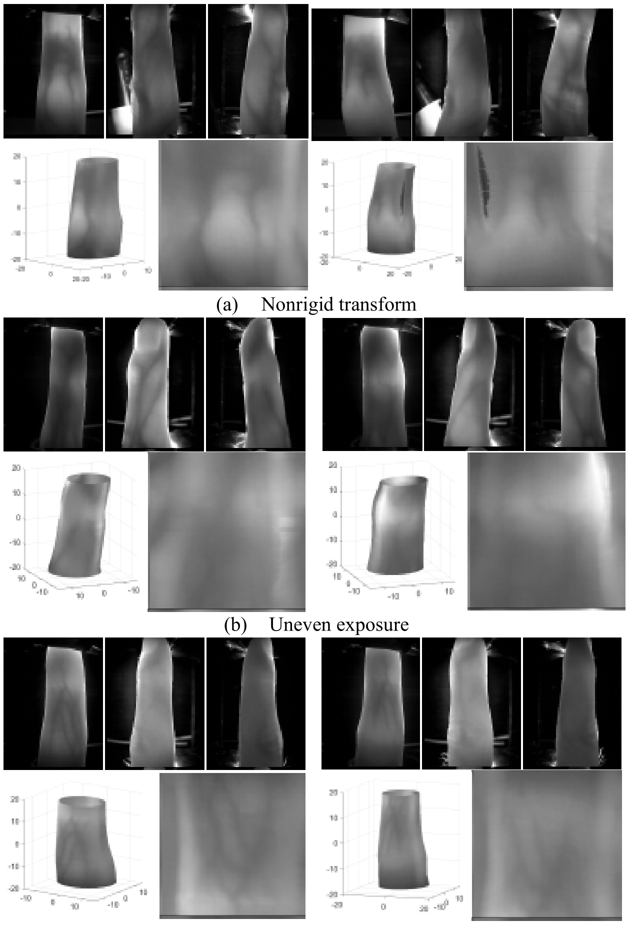
TABLE III

EERs FOR THE DIFFERENT FEATURES

Methods / *Data*	*F3DTM*	*F3DGM*	*Fusion*
Non-multipose	1.60	4.47	1.35
Multipose	2.37	9.80	2.13

3) Validation of the 3D Information: In addition to the improvements brought by the abovementioned flattened 3D texture map, another advantage of our 3D finger vein system is its ability to take full advantage of the more comprehensive information from the 3D geometric shape of one finger. Thus, in this section, we conduct two experiments: one evaluates the effectiveness of the 3D finger geometry as a sole biometric mode, and the other shows our system's potential for improving recognition performance by fusing the 3D geometric shape with the vein features. All of the results are illustrated in Fig. 16, and the EERs are shown in Table III.

Based on the experimental results shown in Table III, we can arrive at the following conclusions. First, the 3D finger geometric feature is useful for discrimination. The EER of F3DGM is closer to that using a single vein image captured by any camera with both non-multipose data and multipose data. Second, combining more biometrics for a finger improves performance. As seen from previous sections that benefit from the proposed 3D finger vein imaging mode, our approach obtains 3D finger geometric shapes simultaneously; then, when we combine the full-view finger vein features with the 3D geometric features, the performance can be further improved. Future studies can focus more on taking full advantage of 3D geometric features and searching for a more effective



(a) Nonrigid transform

(b) Uneven exposure

(c) Shift transform

Fig. 17. Examples of error cases.

extraction and matching method for 3D geometric features to further improve the effectiveness of our 3D finger vein verification strategy. Most importantly, the approach that combines the I_{F3DTM} and I_{F3DGM} features reaches an EER of 1.35% for non-multipose data, which emphasizes the effectiveness of our proposed 3D finger vein verification system. In addition, the same approach reaches an EER of 2.13% for multipose data, which is a significant improvement compared with the performance of traditional 2D finger vein biometrics with multipose problems.

D. Analysis of Error Cases

The verification experiments above show that our algorithm achieves a significant improvement when solving multipose problems in the field of finger vein verification; however, it also introduces some errors during 3D reconstruction. The following three cases are the main factors that affect the verification result, as Fig. 17 shows. Each case in Fig. 17 shows a pair of samples. For each sample, we show the sources of the three viewed images, the constructed 3D finger vein images and the flattened texture map.

First, regarding the nonrigid deformation, our proposed algorithm is based on the assumption that only rigid deformation or slight nonrigid deformation exists during finger vein data collection; thus, it is not suitable for solving large nonrigid deformation problems. Taking the condition shown in Fig. 17(a) as an example, the finger bending degree is large; consequently, the error of the approximation ellipse calculated by our proposed algorithm is relatively large, which results in a large difference between the reconstructed 3D model and the actual 3D finger causeing the wrong texture to be mapped onto the 3D finger vein image. This affects both texture and geometric feature extraction. Second, during finger vein image collection, the brightness at the edge of the finger changes substantially due to uneven local exposure and various finger thicknesses, which leads to differences in the reconstructed 3D vein textures. As Fig. 17(b) shows, the light intensity of the first knuckle is higher with respect to camera 1, causing the corresponding region of the constructed 3D finger vein image to be brighter than surrounding positions, which seriously affects the verification result. Third, we imposed certain constraints on the depth of the finger when collecting data, but for reasons beyond our control, displacement deviations of the fingers in the finger axis direction still exist. Additionally, during the flattened 3D texture map generation, because the angle of the starting line is fixed, a certain displacement deviation must occur in the flattened texture map. Although the CNN is robust to displacement deviation to some extent, it is inevitable that incorrect matching results still exist.

VI. Conclusion and Future Perspective

With the goal of addressing two challenging issues in finger vein verification systems (i.e., limited vein pattern information and sensitivity to positional variations of the finger), in this paper, we propose a new full-view 3D finger vein verification system. Inspired by 3D biometrics, such as 3D face recognition and 3D fingerprint recognition, we utilize three cameras mounted on the three vertexes of an equilateral triangle to simultaneously capture vein images around the entire range of the finger. In this way, we can capitalize on most feature information from one finger, including all finger vein patterns distributed under one finger surface and the 3D geometric features of the finger. Then, we propose a new 3D finger vein reconstruction and a corresponding texture mapping algorithm to generate the 3D finger vein structure which, to the best of our knowledge, is the first instance of generating full-view 3D finger vein structures from three 2D images. Next, we develop a targeted feature extraction and matching strategy for 3D finger vein verification. Finally, we further improve the system's performance by fusing the texture features and the 3D geometric features of the finger. The results of a series of experiments strongly demonstrate that the proposed 3D finger vein verification system not only achieves better verification performance than traditional 2D finger vein verification systems but also has greater potential, especially regarding its ability to address the multipose problem.

This paper describes a preliminary exploration of a 3D finger vein verification system. Although the proposed approach achieves a promising performance for finger vein verification, a substantial amount of work remains to improve the verification accuracy and reduce time consumption. First, the image quality is largely related to the hardware condition, including both the camera performance and the light path design and control. An improvement in hardware design would result in improved image quality which, in turn, would improve the verification accuracy. Second, the 3D reconstruction and texture mapping procedures still introduce some errors that require more elaborate work. Third, the proposed feature extraction method is time consuming, and there is much room for further optimization. Finally, the size of our finger pose database is currently small compared to other databases; thus, we will continue to collect more full-view finger images in the future.

REFERENCES

[1] L. Yang, G. Yang, Y. Yin, and L. Zhou, "A survey of finger vein recognition," in *Proc. Chin. Conf. Biometric Recognit.*, 2014, pp. 234–243.

[2] J. Defranza and D. Gagliardi, *Introduction to Linear Algebra With Applications*. Long Grove, IL, USA: Waveland Press, 2015, pp. 199–200.

[3] S. Qiu *et al.*, "Finger-vein recognition based on dual-sliding window localization and pseudo-elliptical transformer," *Expert Syst. Appl.*, vol. 64, pp. 618–632, Dec. 2016.

[4] M. Vlachos and E. Dermatas, "Finger vein segmentation from infrared images based on a modified separable mumford shah model and local entropy thresholding," *Comput. Math. Methods Med.*, vol. 2015, no. 1, 2015, Art. no. 868493.

[5] E. C. Lee, H. Jung, and D. Kim, "New finger biometric method using near infrared imaging," *Sensors*, vol. 11, no. 3, pp. 2319–2333, 2011.

[6] T. D. Pham, Y. H. Park, D. T. Nguyen, S. Y. Kwon, and K. R. Park, "Nonintrusive finger-vein recognition system using NIR image sensor and accuracy analyses according to various factors," *Sensor*, vol. 15, no. 7, pp. 16866–16894, Jul. 2015.

[7] L. Yang, G. Yang, L. Zhou, and Y. Yin, "Superpixel based finger vein roi extraction with sensor interoperability," in *Proc. Int. Conf. Biometrics (ICB)*, May 2015, pp. 444–451.

[8] B. A. Rosdi, W. S. Chai, and S. A. Suandi, "Finger vein recognition using local line binary pattern," *Sensors*, vol. 11, no. 12, pp. 11357–11371, 2011.

[9] X. Meng, G. Yang, Y. Yin, and R. Xiao, "Finger vein recognition based on local directional code," *Sensors*, vol. 12, no. 11, pp. 14937–14952, 2012.

[10] W. Song, T. Kim, H. Kim, J. Choi, H. Kong, and S. Lee, "A finger-vein verification system using mean curvature," *Pattern Recognit. Lett.*, vol. 32, no. 11, pp. 1541–1547, 2011.

[11] J. Yang and Y. Shi, "Finger–vein ROI localization and vein ridge enhancement," *Pattern Recognit. Lett.*, vol. 33, no. 12, pp. 1569–1579, 2012.

[12] Y. Lu, S. J. Xie, S. Yoon, J. Yang, and D. S. Park, "Robust finger vein ROI localization based on flexible segmentation," *Sensors*, vol. 13, no. 11, pp. 14339–14366, 2013.

[13] M. S. M. Asaari, S. A. Suandi, and B. A. Rosdi, "Fusion of band limited phase only correlation and width centroid contour distance for finger based biometrics," *Expert Syst. Appl.*, vol. 41, no. 7, pp. 3367–3382, Jun. 2014.

[14] C. Zuo, K. Wang, and X. Song, "A new ROI extraction method of non-contact finger vein images," in *Biometric Recognition*. New York, NY, USA: Springer, 2013, pp. 288–297.

[15] B. Huang, Y. Dai, R. Li, D. Tang, and W. Li, "Finger-vein authentication based on wide line detector and pattern normalization," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 1269–1272.

[16] Y. Matsuda, N. Miura, A. Nagasaka, H. Kiyomizu, and T. Miyatake, "Finger-vein authentication based on deformation-tolerant feature-point matching," *Mach. Vis. Appl.*, vol. 27, no. 2, pp. 237–250, 2016.

[17] B. Huang, S. Liu, and W. Li, "A finger posture change correction method for finger-vein recognition," in *Proc. IEEE Symp. Comput. Intell. Security Defence Appl. (CISDA)*, Jul. 2012, pp. 1–7.

[18] C. B. Yu, H. F. Qin, and Y. Z. Cui, "Finger-vein image recognition combining modified Hausdorff distance with minutiae feature matching," *Interdiscipl. Sci. Comput. Life Sci.*, vol. 1, no. 4, pp. 280–289, 2009.

[19] F. Liu, G. Yang, Y. Yin, and S. Wang, "Singular value decomposition based minutiae matching method for finger vein recognition," *Neurocomputing*, vol. 145, pp. 75–89, Dec. 2014.

[20] W. Kang, Y. Lu, D. Li, and W. Jia, "From noise to feature: Exploiting intensity distribution as a novel soft biometric trait for finger vein recognition," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 4, pp. 858–869, Apr. 2018.

[21] Z. Ma, L. Fang, J. Duan, S. Xie, and Z. Wang, "Personal identification based on finger vein and contour point clouds matching," in *Proc. IEEE Int. Conf. Mechatron. Automat. (ICMA)*, Aug. 2016, pp. 1983–1988.

[22] Q. Zhang, B. Wang, and L. Wang, "Finger vein three-dimensional point cloud obtaining method and device and terminal," Google Patents 10 372 962 2B, Sep. 22, 2017.

[23] H. Qin and M. A. El-Yacoubi, "Deep representation for finger-vein image-quality assessment," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1677–1693, Aug. 2018.

[24] R. Das, E. Piciucco, E. Maiorana, and P. Campisi, "Convolutional neural network for finger-vein-based biometric identification," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 360–373, Feb. 2019.

[25] H. G. Hong, M. B. Lee, and K. R. Park, "Convolutional neural network-based finger-vein recognition using NIR image sensors," *Sensors*, vol. 17, no. 6, p. 1297, 2017.

[26] H. Qin and M. A. El-Yacoubi, "Deep representation-based feature extraction and recovering for finger-vein verification," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 8, pp. 1816–1829, Aug. 2017.

[27] H. Huang, S. Liu, H. Zheng, L. Ni, Y. Zhang, and W. Li, "DeepVein: Novel finger vein verification methods based on deep convolutional neural networks," in *Proc. IEEE Int. Conf. Identity Security Behav. Anal. (ISBA)*, Feb. 2017, pp. 1–8.

[28] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[29] M. Agarwal and N. Natarajan, "Inheritance relations of hexagons and ellipses," *College Math. J.*, vol. 47, no. 3, pp. 208–214, 2016.

[30] J.-Y. Bouguet, "MATLAB camera calibration toolbox," Caltech, Pasadena, CA, USA, Tech. Rep., 2000.

[31] J. Downs, *Practical Conic Sections: The Geometric Properties of Ellipses, Parabolas and Hyperbolas: Courier Corporation*. New York, NY, USA: IEEE, 1993.

[32] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1063–1074, Sep. 2003.

[33] G. Parziale, E. Diaz-Santana, and R. Hauke, "The surround imager: A multi-camera touchless device to acquire 3D rolled-equivalent fingerprints," in *Proc. Int. Conf. Biometrics*, 2006, pp. 244–250.

[34] B. D. Bradley, A. D. Chan, and M. J. D. Hayes, "A simple, low cost, 3D scanning system using the laser light-sectioning method," in *Proc. IEEE Instrum. Meas. Technol. Conf. (IMTC)*, May 2008, pp. 299–304.

[35] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Proc. IEEE 6th Int. Conf. Adv. Video Signal Based Surveill.*, Sep. 2009, pp. 296–301.

[36] Y. Wang, L. G. Hassebrook, and D. L. Lau, "Data acquisition and processing of 3-D fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 4, pp. 750–760, Dec. 2010.

[37] A. Kumar and C. Kwong, "Towards contactless, low-cost and accurate 3D fingerprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 681–696, Mar. 2015.

[38] D. York, "Least-squares fitting of a straight line," *Can. J. Phys.*, vol. 44, no. 5, pp. 1079–1086, May 1966.

[39] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861

[40] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 499–515.

[41] R. Wang, G. Wang, Z. Chen, Z. Zeng, and Y. Wang, "A palm vein identification system based on Gabor wavelet features," *Neural Comput. Appl.*, vol. 24, no. 1, pp. 161–168, Jan. 2014.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.

[45] Y. Yin, L. Liu, and X. Sun, "SDUMLA-HMT: A multimodal biometric database," in *Proc. Chin. Conf. Biometric Recognit.*, 2011, pp. 260–268.

[46] A. Kumar and Y. Zhou, "Human identification using finger images," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2228–2244, Apr. 2012.

Wenxiong Kang (M'10) received the M.S. degree from Northwestern Polytechnical University, Xi'an, China, in 2003, and the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2009. He is currently a Professor with the School of Automation Science and Engineering, South China University of Technology. His research interests include biometrics identification, image processing, pattern recognition, and computer vision.

Hongda Liu received the M.S. degree from the South China University of Technology, Guangzhou, China, in 2019. His research interests include biometrics identification, 3D reconstruction, and deep learning.

Wei Luo received the B.E. degree in automation from the South China University of Technology, Guangzhou, China, in 2018, where he is currently pursuing the M.S degree. His research interests include biometrics identification, computer vision, and deep learning.

Feiqi Deng was born in 1962. He received the Ph.D. degree in control theory and control engineering from the South China University of Technology, Guangzhou, in 1997. Since 1999, he has been a Professor with the South China University of Technology, where he is also the Director of the Systems Engineering Institute. He has published over 300 journal papers on the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, *Automatica*, the *SIAM Journal of Control and Optimization*, the *International Journal of Systems Science and Systems*, and *Control Letters*. His main research interests include stability, stabilization, and robust control theory of complex systems, including time-delay systems, nonlinear systems and stochastic systems. He is currently a member of Technical Committee on Control Theory (TCCT) and the Chinese Association of Automation. He is serving as the Chair for the IEEE CSS Guangzhou Chapter and IEEE SMC Guangzhou Chapter, a Vice Editor-in-Chief of the *Journal of South China University of Technology*, and a member of the Editorial Boards of the following journals: IEEE ACCESS, *Control Theory and Applications*, the *Journal of Systems Engineering and Electronics*, and the *Journal of Systems Engineering,* all about systems and control.