Palmprint Recognition with Three Dimensional Features

QING Pei

Master of Science in Software Technology

The Hong Kong Polytechnic University

June 2012

Statement of Authorship

Except where reference is made in the text of this dissertation, this dissertation contains no material published elsewhere or extracted in whole or in part from a dissertation presented by me for another degree or diploma.

No other persons work has been used without due acknowledgement in the main text of the dissertation.

This dissertation has not been submitted for the award of any other degree or diploma in any other tertiary institution.

Name: QING Pei

Dated: 23/06/2012

Acknowledgements

Firstly, I would like to thank my advisor, David Zhang, for leading me into biometrics research, for insightful remarks and useful advice during my MSc. study, and for his advice on how to conduct scientific research. It has been an privilege to work under his supervision.

I would also like to thank Professor Lei Zhang for what I have learned in the Multimedia Computing course. The experience not only leads to a more systematic knowledge of the area, but also urges me to practice everything with real data.

During my time working on this dissertation, Dr. Wei Li helped me a lot in the data processing part. I want to thank him for saving me a huge amount of time dealing with the dataset.

Abstract

Three Dimensional (3D) palmprint has proved to be a significant biometrics for personal authentication. 3D palmprints are harder to counterfeit than 2D palmprints and more robust to variations in illumination and serious scrabbling on the palm surface. Previous work on 3D palmprint recognition has concentrated on local features such as texture and lines.

In this dissertation, the authentication process are improved using 3D features. The features are stable in samples from a single person over time and distinguishable among samples from different people. Three features adopted are Maximum Depth of palm center, Horizontal Cross-section Area and Radial Line Length from the centroid to the boundary of 3D palmprint horizontal cross-section of different levels. The feature set is combined as a column feature vector for matching. Support Vector Machine is introduced for higher efficiency.

Experiments are conducted on an existing 3D palmprint database of 8,000 samples. The results show that the proposed method is able to achieve an reasonable performance.

To my parents.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Palmprint has been increasingly recognized as unique and stable biometric characteristics for personal authentication. In the past decade, various methods based on two dimensional (2-D) palmprint have been studied in depth. The 2-D recognition techniques have proved to achieve high accuracy [1].

In recent years, three dimensional (3-D) palmprint recognition devices emerge and are quite promising because of the additional depth information gathered.

Although the devices has been out for more than two years, most previous matching algorithms treat 3-D information as a supplement to 2-D texture images and used joint matching techniques to increase accuracy [2, 3, 4, 5, 6]. Authentication with only the 3-D information has not been thoroughly studied. The amount of useful information carried by the 3-D data is still under investigation.

There are two major challenges:

1. 3-D devices, compared to 2-D ones, are lower in resolution.

2. The depth values are susceptible to movements of human hands and are therefore less stable than 2-D texture information of palmprints.

David et al. explore a 3-D palmprint recognition approach by exploiting the 3-D structural information of the palm surface [4, 5]. The structured light imaging is

used to acquire the 3-D palmprint data, from which several types of unique features, including mean curvature image, Gaussian curvature image, and surface type, are extracted. A fast feature matching and score-level fusion strategy is proposed for palmprint matching and classification. Wei et al. propose an efficient joint 2D and 3D palmprint matching scheme [3]. The principal line features and palm shape features are extracted and used to accurately align the palmprint, and a couple of matching rules are defined to efficiently use the 2D and 3D features for recognition. The experiments show that the proposed scheme can greatly improve the performance of palmprint verification. Wei et al. also present an efficient scheme for 3-D palmprint recognition [2]. They extract both line and orientation features after calculating and enhancing the mean-curvature image of the 3-D palmprint data. The two types of features are then fused at either score level or feature level for the final 3-D palmprint recognition.

Existing work has been done to utilize the 3-D information for palmprint classification and sorting. The global features proposed for that purpose are fast in matching speed but low in accuracy compared to 2-D techniques.

Three features are extracted from 3D palmprint depth data: Maximum Depth, Horizontal Cross-section Area and Radial Line Length. Theses features are combined together as a brief description of a 3D palmprint sample. Viewing the feature set as a column vector, various data processing techniques can be applied to improve the recognition efficiency such as dimension reduction and machine learning.

The rest of this work is organized as follows. Chapter 2 briefly introduces the existing research efforts related to this topic. Chapter 3 describes the entire data processing procedures to extract features and use them for 3D palmprint recognition. Chap-

ter 4 shows some experiment results. Chapter 5 concludes the work and points out potentials for some future work.

Chapter 2 Related Work

This work is based on a number of previous research efforts. The samples are collected with an existing hardware device. And there also exist some efforts on recognition based on local features.

2.1 Hardware

Different approaches are available for 3D imaging including laser scanning [7], viewpoint reconstruction [8] and structured light scanning [9]. Structured light scanning, compared to other approaches, is fast and accurate. For palmprint recognition, speed is an important factor. The verification or identification result must be given in a short time. Otherwise the system is barely suitable for real-world applications.

Projecting a narrow band of light onto a three-dimensionally shaped surface produces a line of illumination that appears distorted from other perspectives than that of the projector, and can be used for an exact geometric reconstruction of the surface shape (light section).

A faster and more versatile method is the projection of patterns consisting of many stripes at once, or of arbitrary fringes, as this allows for the acquisition of a multitude of samples simultaneously. Seen from different viewpoints, the pattern appears geometrically distorted due to the surface shape of the object.

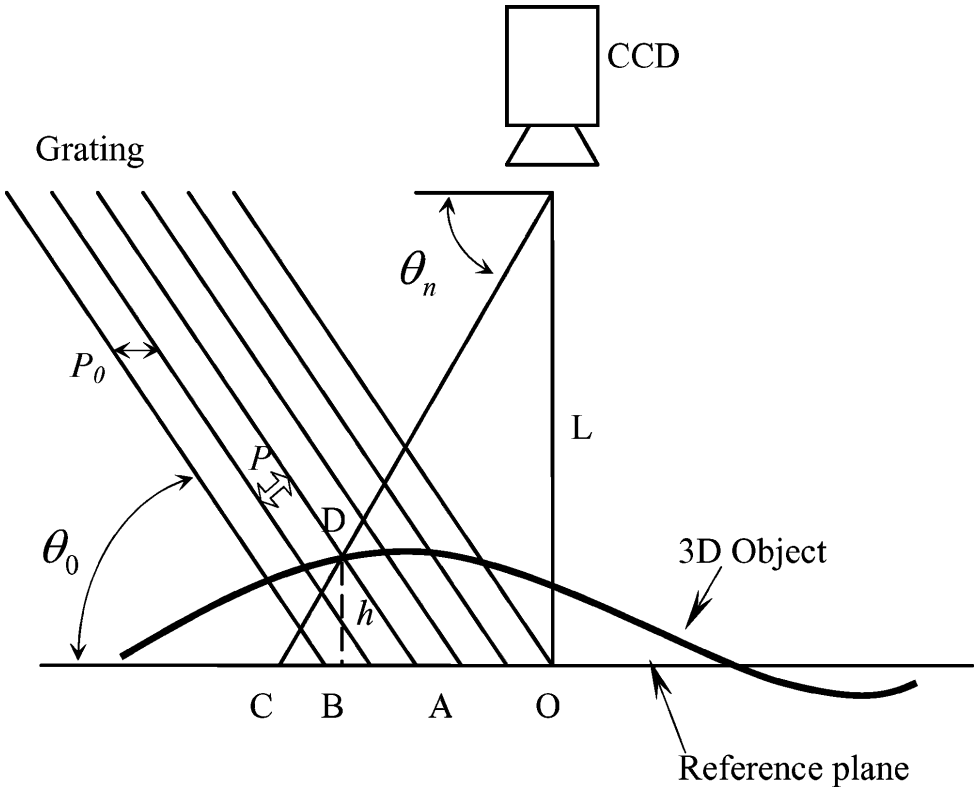


Figure 2.1: The principle of structured-light imaging[10]

Although many other variants of structured light projection are possible, patterns of parallel stripes are widely used. Figure 2.3 shows the geometrical deformation of a strip pattern projected onto a palm. The displacement of the stripes allows for an exact retrieval of the 3D coordinates of any details on the palm's surface.

Figure 2.2: 3D palmprint capturing device

Figure 2.2 shows the 3D palmprint capturing device designed by David et al. [5]. The system proposed has a resolution of 768x576.

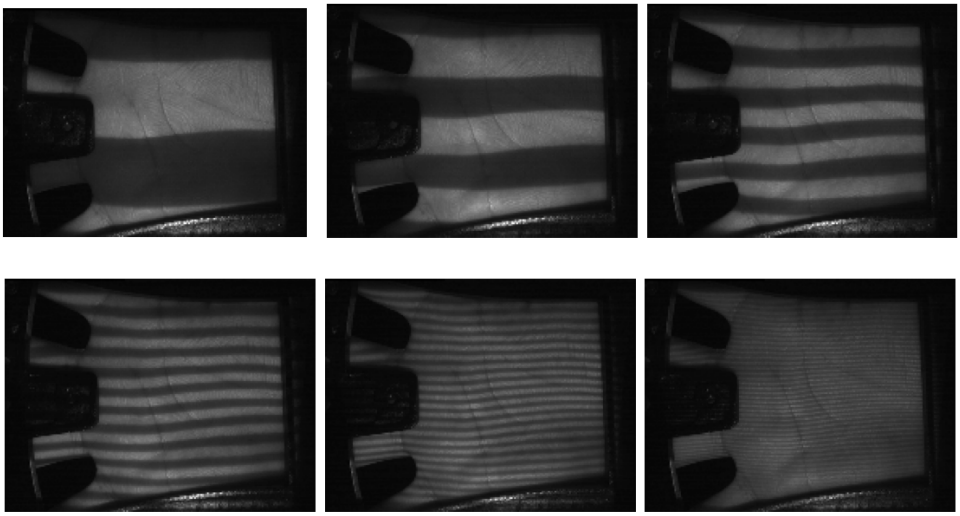


Figure 2.3: Sample patterns of the stripes on the palm [10]

2.2 Recognition Methods

Traditionally, palmprint recognition has made use of either high or low resolution 2D palmprint images. High resolution images are suitable for forensic applications [11] while low resolution images are suitable for civil and commercial applications [12]. Most current research use low resolution palmprint recognition and is either texture-based or line-based. The texture-based methods include PalmCode [12], Competitive Code [13] and Ordinal Code [14]. These methods use a group of filters to enhance and extract the phase or directional features which can represent the texture of the palmprint. Line-based methods use line or edge detectors to explicitly extract line information from the palmprint that is then used for matching. The representative methods include Derivative of Gaussian based line extraction [15] and Modified Finite Radon transform (MFRAT) based line extraction [16].

In recent years, 3D techniques have been applied to biometric authentication, such as 3D face [17] and 3D ear recognition [18]. Most recently, a structured-light imaging [9, 19] 3D palmprint system [4] was developed that captures the depth information of a palmprint. This information is then used to calculate the Mean and the Gaussian curvatures for use in 3D palmprint matching and recognition. To date, however, there has been no work with 3D palmprints that has extracted global shape features, which may be useful in classification and indexing. For fingerprint, according to the global ridge structure and singularities, it can be classified into five classes: arch, tented arch, left loop, right loop and whorl [20]. Wu et al. classified the palmprint into six classes according to the palmprint principal lines [21]. Besides the exclusive classification technique, the continuous classification technique is also widely used for indexing the database for personal identification [22].

Some general features were extracted for recognition including 3D Mean Curvature Image, 3D Gauss Curvature Image and 3D Surface Type [5, 10].

Chapter 3 Recognition Procedure

The process for feature extraction and matching are described in the following sections in the exact order they are applied to samples in the database.

3.1 Region of Interest Extraction

The raw 3D palmprint image consists of 768*576 pixels with a depth value at each point. The samples are captured by a 3D palmprint acquisition device based on structured light imaging [4]. Before any other processing, noisy pixels at the boundaries must be removed. The full resolution sample is cropped to a Region of Interest (ROI) with 400*400 resolution. A rectangle mask, from (234,68) to (634,468) is used to crop the ROI. Figure 3.1 shows a full resolution sample of a palmprint and Figure 3.2 shows the ROI cropped. To reduce the storage and computation requirements for further processing, the ROI is down-sampled to a resolution of 200*200.

The ROI data is stored in a 200 by 200 matrix,

$$D = d_{ij} | i = 1,2,\ldots,200; j = 1,2,\ldots,200 \tag{3.1}$$

where d_{ij} is the depth value of the i^{th} row and j^{th} column pixel of the ROI.

There exists more complicated ROI extraction methods taking advantage of the 2D approaches. 2D ROI is first obtained using the algorithm in [12]. As the 2D sample

Figure 3.1: Full resolution 3-D palmprint sample

captured is pixel-to-pixel matched to the 3D sample, using the exact 2D ROI for the 3D sample is trivial.

The cropping ROI extraction approach used here is much easier and faster than the one proposed in [4]. The reason to adopt this naive approach is that the features are invariant to rotation and translation. It is unnecessary to fully align every sample. Another consideration is that when the 3D information is the only data available, it is not feasible to apply the same 2D algorithm to the depth image.

Even if the samples are cropped to the center area, it is still possible that some noisy pixels are included. If the gradient

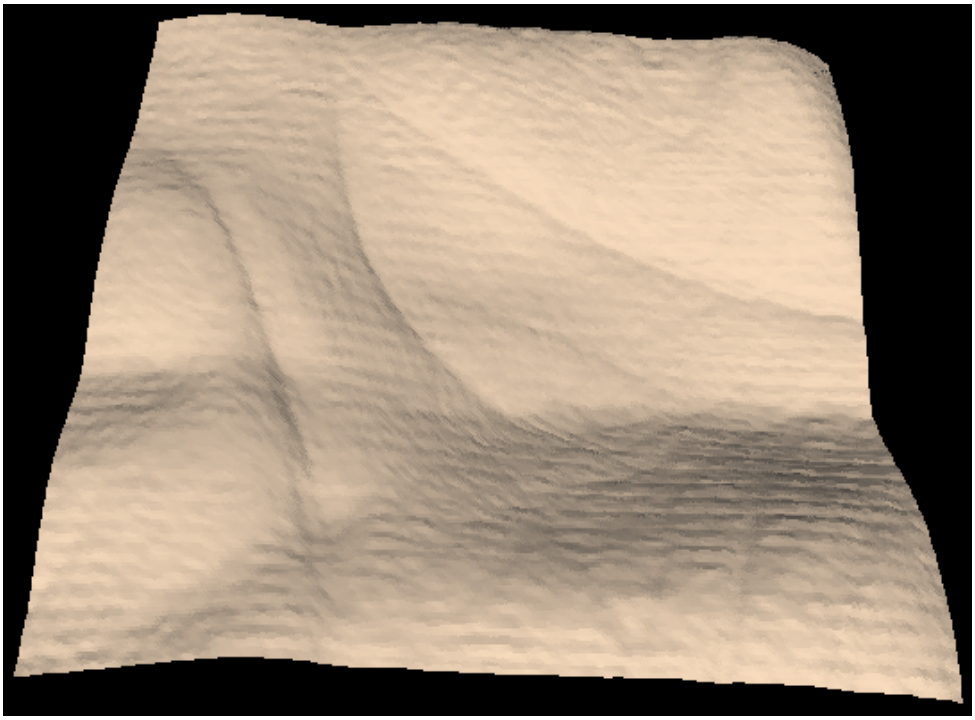


Figure 3.2: ROI from a 3-D palmprint sample

$$|\nabla D| = \sqrt{\left(\frac{\partial D}{\partial x}\right)^2 + \left(\frac{\partial D}{\partial y}\right)^2} \tag{3.2}$$

is larger than a given threshold, the pixel is regarded as noisy. Another 200 by 200 matrix,

$$M = m_{ij} | i = 1, 2, \ldots, 200; j = 1, 2, \ldots, 200 \tag{3.3}$$

is used to represent the mask, where $m_{ij} = 0$ marks the noisy pixels while $m_{ij} = 1$ marks the normal ones.

3.2 Feature Calculation

Using the ROI obtained from the original 3D palmprint data, three kinds of features to describe the shape of the 3D palmprint: Maximum Depth (MD) of palm center, the Horizontal Cross-section Area (HCA) of different levels and the Radial Line Length (RLL) from the centroid to the boundary of 3D palmprint horizontal cross-section of different levels.

3.2.1 Maximum Depth (MD)

MD means the maximum depth value of the 3D palm from a reference plane. The reference plane is decided using a rectangle obtained by experience. The top left and bottom right pixels of the rectangle are (65,6) and (136,35). These parameters are denoted as $R_s = 65, R_e = 136, C_s = 6$ and $C_e = 35$, i.e. the starting(ending) row(column). By examine a random 20 samples, gradient of this area is relatively small.

The depth of the reference plane is defined as the mean depth of the points contained by this rectangle

$$d_r = \frac{1}{\sum_{i=R_s}^{R_e} \sum_{j=C_s}^{C_e} m_{ij}} \sum_{i=R_s}^{R_e} \sum_{j=C_s}^{C_e} (d_{ij} \cdot d_{ij}) \tag{3.4}$$

where d_{ij} and m_{ij} are the elements defined in 3.1 and 3.3

After getting the depth of the reference plane, we find the maximum depth d_{max} in another region with $R_s = 41, R_e = 160, C_s = 65$ and $C_e = 190$.

$$d_{max} = \max_{i=R_s}^{R_e}(\max_{j=C_s}^{C_e}(d_{ij})) \tag{3.5}$$

The Maximum Depth (MD) is then defined as

$$MD = d_{max} - d_r \tag{3.6}$$

3.2.2 Horizontal Cross-section Area

When 3D ROI is examined in a contour view as shown in Figure 3.3, it is obvious that the regions of different given depth ranges can be used to describe a sample.

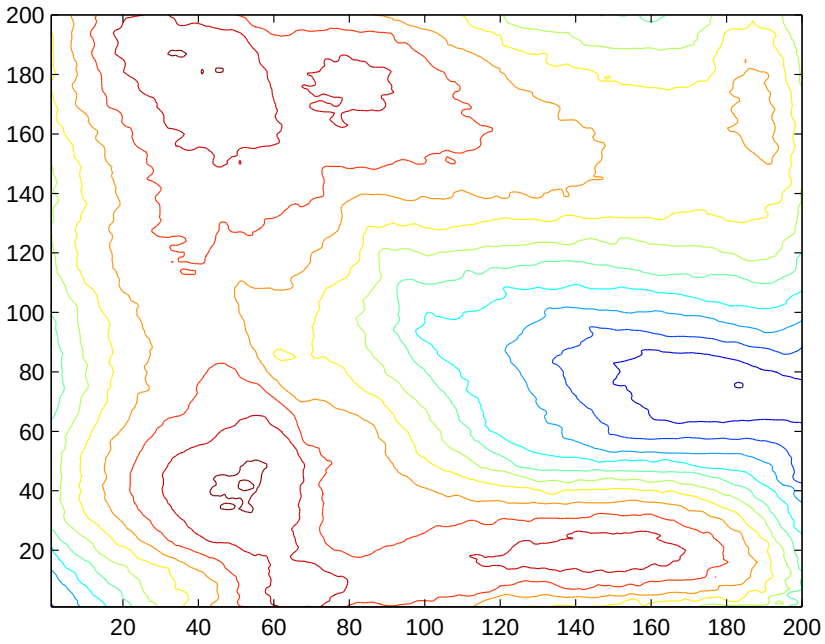


Figure 3.3: Contour view of a 3-D ROI

A group of equidistant horizontal planes cut the 3D ROI as shown in Figure 3.4. Figure 3.3 shows that most of the deeper level (enclosed with blue curves) are connected. These are more stable in response to noise or transformation.

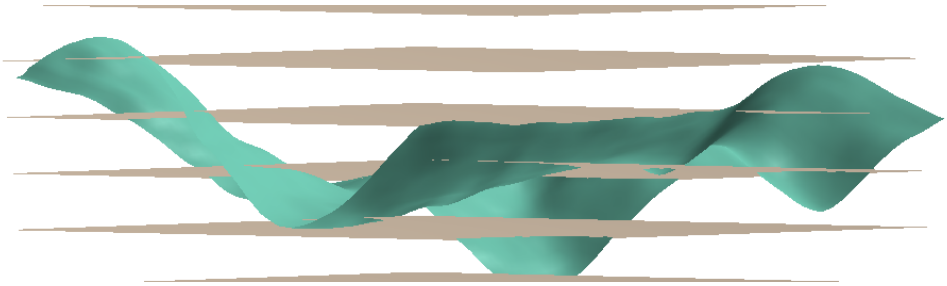


Figure 3.4: 3D ROI cut by parallel horizontal planes

To get a stable HCA, we take into consideration only the levels from the deepest point to the reference plane, defined in Section 3.2.1. Suppose we divide this region into N levels. Every level $G^k, k = 1,2,\ldots,N$ is described with a 200 by 200 matrix and calculated as

$$G_{ij}^k = \begin{cases} 1 & \text{if} d_{ij} > h \cdot (N-k+1)/N, \\ 0 & \text{otherwise} \end{cases} \quad k = 1,2,\ldots,N; i = 1,2,\ldots,200; j = 1,2,\ldots,200; \tag{3.7}$$

where d_{ij} is the depth value of the i^{th} row and j^{th} column pixel of the ROI defined in 3.1 and h is the palmprint depth defined by 3.6.

To stabilize the areas, the growth of higher level is constrained to its previous level except the first level. That is

$$L^k = \begin{cases} G^1 & k = 1 \\ G^k \cap (L^{k-1} \oplus \Theta^{k-1}) & k = 2,3,\ldots,N \end{cases} \tag{3.8}$$

where $\cap$ denotes logical AND, $\oplus$ denotes a morphological dilation operation and Θ^k is a disk morphological structuring element whose size can be calculated by $35 - 3 \times k$ (which is suitable for N = 8 by experience).

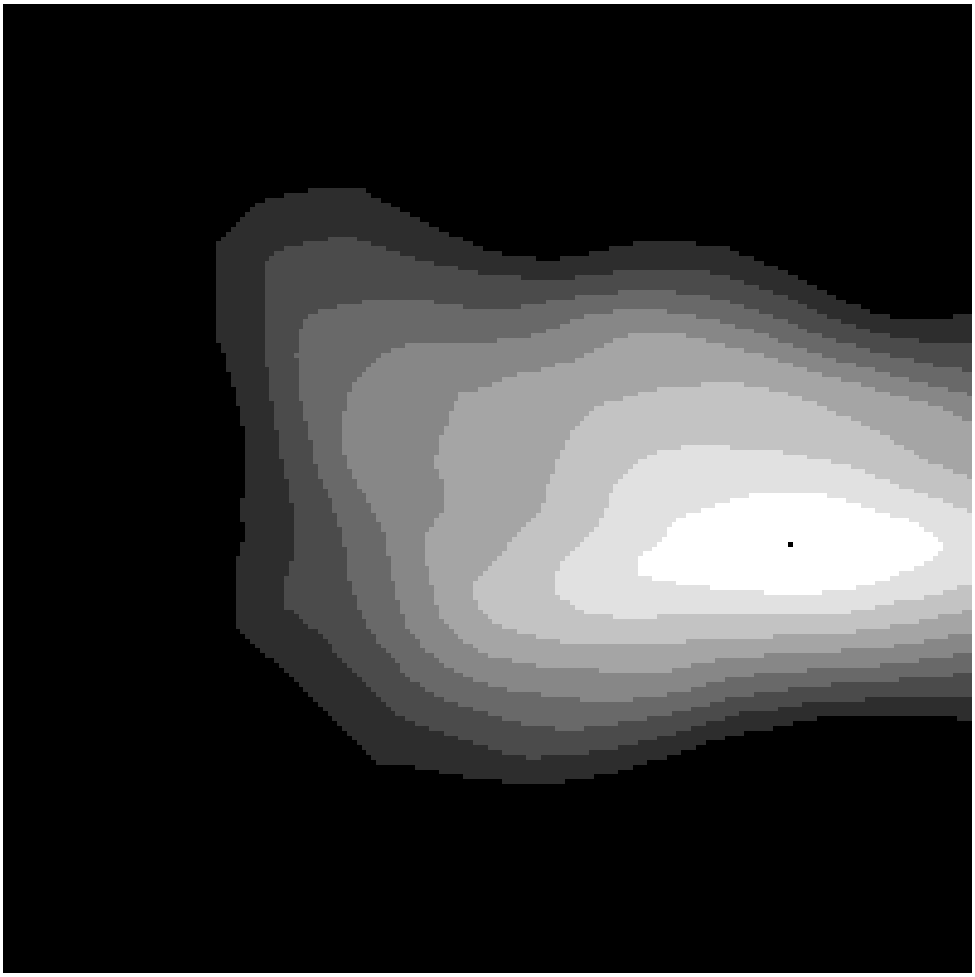


Figure 3.5: L^k shown stacked when $N = 8$

Figure 3.5 shows an example of all the levels stacked together.

Figure 3.6 and 3.7 shows the cross-sectional area feature from two samples collected from one palm. Figure 3.8 and 3.9 are extracted from two samples from another palm.

3.2.3 Radial Line Length (RLL)

HCA is a coarse summary of the ROI characteristics. Different sample may have a similar area but dramatically different shape or contour of that area. To describe this shape characteristic, we need more elements to represent the ROI. The Radial Line Length (RLL) is then introduced for this purpose.

First, we calculate the centroid of the first level L^1, thereafter we treat it as the reference point P_{ref} for all levels. Then, from P_{ref} we draw M radial lines which intersect with the contour of every level. RLL is defined as the distance from the intersection point to P_{ref}. The radial lines are distributed at equal angles. We record these radial lines from the inner layers to the outer layers starting with the horizontal direction by an M by N dimensional vector

$$R_i, i = 1, 2, \ldots, M \times N \tag{3.9}$$

where M is the number of radial lines and N is the number of cross-sections.

Figure 3.10(a) through 3.10(d) show some examples of RLL. This is a more detailed description than HCA as it takes the shape of each area into consideration.

The above three features are mainly determined by the central region of the palm. This region is certainly contained by the ROI described in Section 3.1 which makes

these features insensitive to translation and rotation. Although the RLL feature can be affected by rotation as the contours change smoothly, if the rotation is small then the variation of the RLL feature will also be small. Actually, there are some restricting pegs on the capture device which can guide the users to put their hands on the proper place as described in [4]. Furthermore, we assume the user is cooperative when collecting data as we aim at civil rather than law enforcement applications.

3.3 Feature Matching

The classification of biometrics speeds up the identification process by reducing the number of comparisons that must be made. There are two kinds of classification techniques: exclusive classification and continuous classification. Both fingerprint [20] and palmprint classifications [21] make use of exclusive classification. The main problem of this technique is that it uses only a small number of classes and the samples are unevenly distributed between them, with more than 90% of the samples being in just two or three classes. A further problem with exclusive classification is that when classification is performed automatically, it is necessary to handle errors and rejected samples gracefully, which is a hard problem in practice. In contrast, for continuous classification, samples are not partitioned into disjoint classes but rather associated with numerical vectors which represent features of the samples. These feature vectors are created through a similarity-preserving transformation so that similar samples are mapped into close points in the multi-dimensional space [23]. The continuous classification technique is adopted. As the features combining MD, HCA and RLL are high-dimensional, LDA is used for dimension reduction. Coarse-level matching and Ranking Support Vector Machine (RSVM) are applied to the low dimensional vectors for efficient palmprint recognition.

3.3.1 Dimension Reduction

LDA is a state-of-the-art dimensionality reduction technique widely used in classification problems. The objective is to find the optimal projection which simultaneously minimizes the within-class distance and maximizes the between-class distance, thus achieving maximum discrimination (Here, the class is used to denote the identity of the subjects, e.g. the samples collected from one palm are regarded as one class). However, the traditional LDA requires the within-class scatter matrix to be nonsingular, which means the sample size should be large enough compared with its dimension, but is not always possible. In this paper, we therefore adopt the orthogonal LDA (OLDA) proposed in [24], where the vectors of the optimal projection are calculated using the training database and the optimal projecting vectors are orthogonal to each other.

Suppose the 3D ROI has been divided to N levels and that M radial lines are used to represent the level contours. We can list the features as a column vector,

$$F = \{MD, A^1, A^2, \ldots, A^N, R^1, R^2, \ldots, R^{N\times M}\} \tag{3.10}$$

with $1 + N + N \times M$ rows. Given a training database which has n samples and k classes as $X = [X_1, X_2, \ldots, X_k]$, where $X_i \in \mathbb{R}^{(1+N+N\times M)\times n}, i = 1, 2, \ldots, k$ and $n = \sum_{i=1}^{k} n_i$, adopting OLDA [24] the optimal projection W can be calculated as follows.

First, the within-class scatter matrix S_w, the between-class scatter matrix S_b and total scatter matrix S_t can be expressed as

$$S_w = H_w H_w^T, S_b = H_b H_b^T, S_t = H_t H_t^T \tag{3.11}$$

where

$$H_w = \frac{1}{\sqrt{n}} \left[X_1 - m_1 \cdot e_1^T, \ldots, X_k - m_k \cdot e_k^T \right] \tag{3.12}$$

$$H_b = \frac{1}{\sqrt{n}} \left[\sqrt{n_1}(m_1 - M), \ldots, \sqrt{n_k}(m_k - m) \right] \tag{3.13}$$

$$H_t = \frac{1}{\sqrt{n}} (X - m \cdot e^T) \tag{3.14}$$

(9) where m_i is the centroid of the ith class X_i, m is the centroid of all the training samples X, $e_i = [1, 1, \ldots, 1]^T \in \mathbb{R}^n, i = 1, 2, \ldots, k$ and $e = [1, 1, \ldots, 1]^T \in \mathbb{R}^n$.

After calculating H_w, H_b and H_t, the reduced Singular Value Decomposition (SVD) is applied to H_t.

$$H_t \xrightarrow{\text{Reduced SVD}} U_r \Sigma_r V_r^T \tag{3.15}$$

Denote $B = \Sigma_r^{-1} U_r^T H_b$ and compute the SVD of B.

$$B \xrightarrow{\text{SVD}} U_B \Sigma_B V_B^T \tag{3.16}$$

Let

$$D = U_r \Sigma_r^{-1} U_B \tag{3.17}$$

$$q = rank(B) \tag{3.18}$$

and denote D_q the first q columns of the matrix D. Then, compute the QR decomposition of D_q.

$$D_q \xrightarrow{\text{QR decomposition}} QR \tag{3.19}$$

where Q is the desired orthogonal matrix and optimal projection, i.e. $W = Q$.

After getting the optimal projection W, we can map the $1 + N + N \times M$ dimensional vector F to a lower dimensional space

$$\tilde{F} = W^T F \tag{3.20}$$

(15) where $\tilde{F} = f_1, f_2, \ldots, f_\Gamma$ is a Γ dimensional vector with $\Gamma < 1 + N + N \times M$.

3.3.2 Coarse-level Matching

For palmprint recognition, we need a measurement of similarity between two samples. If the similarity is high, we have a high confidence to believe that the two samples are from the same person. Otherwise we reject to claim that.

After mapping the high-dimensional feature vector to Γ-dimensional vector $\tilde{F}$, the similarity between two samples can be calculated as

$$Similarity = \|\tilde{F}_1 - \tilde{F}_2\| = \sum_{i=1}^{\Gamma} (f_i^1 - f_i^2)^2 \tag{3.21}$$

There are other features extracted from the same dataset such as Mean Curvature Image in [4]. Figure 3.11 shows the ROI and MCI extracted from three different palms while Figure 3.12 show samples from the same palm. The corresponding matching score is defined as

$$Y = \frac{2 \sum_{i=1}^{n} \sum_{j=1}^{m} Z_d(i,j) \cap Z_t(i,j)}{\sum_{i=1}^{n} \sum_{j=1}^{m} Z_d(i,j) + \sum_{i=1}^{n} \sum_{j=1}^{m} Z_t(i,j)} \tag{3.22}$$

where symbol $\cap$ represents the logical AND operation, Z_d and Z_t are two binarized MCIs. Due to the fact that MCI is covariant to shifting, the actual matching process for MCI is repeated by shifting the image with 4 different displacement in 8 directions. A total of 33 matching scores are calculated and the maximum is adopted as the overall matching score.

Although MCI feature is more descriptive, it takes up more storage (200×200 floats to store the feature image). Compared to the computation in 3.21, the MCI matching process requires far more computation and is therefore significantly slower.

3.3.3 Support Vector Machine

Coarse-level matching scheme is a simple and easy way to reduce retrieval times. Its more useful for palmprint recognition if we can rank the candidate samples in the database in descending order according to the above features. Searching for the closest matches to a given query vector in a large database is time consuming if the vector is even moderately high-dimensional. Various methods have been proposed to speed up the nearest neighbor retrieval, including hashing and tree structures [25]. However, the complexity of these methods grows exponentially with increasing dimensionality [26]. Therefore, we have adopted the Ranking Support Vector Machine (RSVM) method [27], inspired by the approaches of Internet search engines, to rank the candidate samples in the database.

Given a query $q_k, k = 1,2,\ldots,n$ and a sample collection $D = \{d_1, d_2, \ldots, d_m$, the optimal retrieval system should return a ranking $r_k^*, k = 1,2,\ldots,n$ that orders the samples in D according to their relevance to the query.

In the palmprint recognition context, the query q and the sample d are the Γ-dimensional features as described above. If a sample d_i is ranked higher than $d_j$in some ordering r, i.e. $r(d_i) > r(d_j)$, then $(d_i, d_j) \in r$, otherwise $(d_i, d_j) \ni r$. Consider the class of linear ranking functions

$$(d_i, d_j) \in f_{\overrightarrow{w}}(q) \Leftrightarrow \overrightarrow{w}\phi(q, d_i) > \overrightarrow{w}\phi(q, d_j) \tag{3.23}$$

where $\overrightarrow{w}$ is a weighted vector that is adjusted by learning and $\phi(q,d)$ is a pairwise distance function describing the match between q and d can be defined as

$$\phi(q,d) = |q_i - d_i|, i = 1, 2, \ldots, \Gamma \tag{3.24}$$

Our goal is to find the optimal ranking function that will satisfy the maximum number of the following inequalities.

$$\forall (d_i, d_j) \in r_1^* \quad : \quad \overrightarrow{w}\phi(q_1, d_i) > \overrightarrow{w}\phi(q_1, d_j) \tag{3.25}$$

$$\ldots \tag{3.26}$$

$$\forall (d_i, d_j) \in r_n^* \quad : \quad \overrightarrow{w}\phi(q_n, d_i) > \overrightarrow{w}\phi(q_n, d_j) \tag{3.27}$$

It is easier to solve this problem if it is converted into to the following SVM classification problem by introducing non-negative slack variable $\xi_{i,j,k}$.

Hence, the problem is to minimize:

$$V(\overrightarrow{w}, \overrightarrow{\xi}) = \frac{1}{2}\overrightarrow{w} \cdot \overrightarrow{w} + C\sum \xi_{i,j,k} \tag{3.28}$$

subject to:

$$\forall (d_i, d_j) \in r_1^* \quad : \quad \overrightarrow{w}\left(\phi(q_1, d_i) - \phi(q_1, d_j)\right) \geq 1 - \xi_{i,j,1} \tag{3.29}$$

$$\ldots \tag{3.30}$$

$$\forall (d_i, d_j) \in r_1^* \quad : \quad \overrightarrow{w}\left(\phi(q_n, d_i) - \phi(q_n, d_j)\right) \geq 1 - \xi_{i,j,n} \tag{3.31}$$

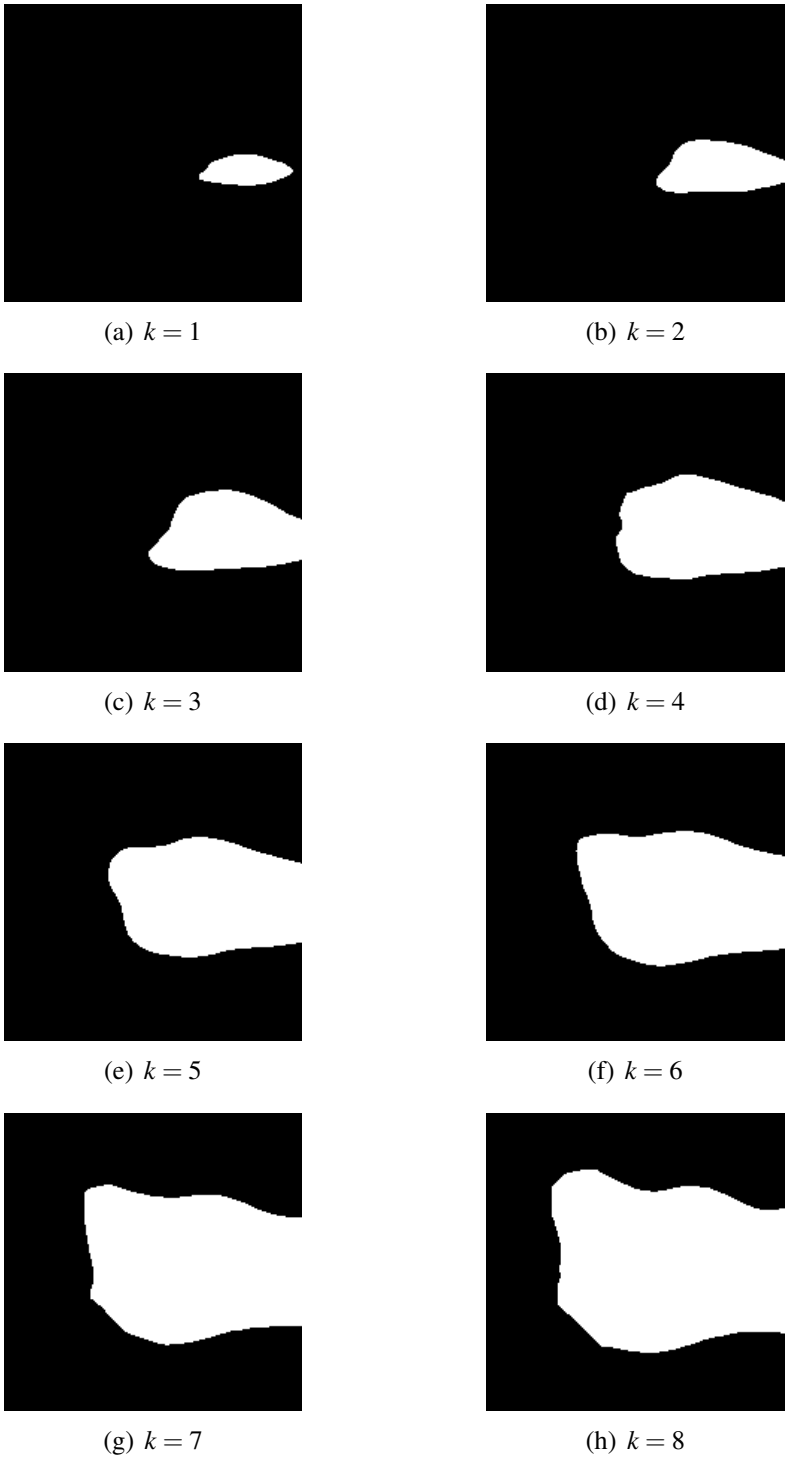
and

$$\forall i \forall j \forall k : \xi_{i,j,k} > 0 \tag{3.32}$$

where C is a parameter that allows trading-off margin size against training error and $C = 0.1$ set by experience.

In the training stage, it is the inner-class samples of a test sample that should be ranked higher than the inter-class samples, e.g. inner-class samples rank is 1 and inter-class samples rank is 0. We input the ranks together with the Γ-dimensional features into the RSVM algorithm to learn the optimal ranking function $f_{\rightarrow w^*}$. Given a new query q, the samples in the database can be sorted by their value of

$$rsv(q, d_i) =\rightarrow w^* \phi(q, d_i) \tag{3.33}$$



(a) $k = 1$ (b) $k = 2$

(c) $k = 3$ (d) $k = 4$

(e) $k = 5$ (f) $k = 6$

(g) $k = 7$ (h) $k = 8$

Figure 3.6: L^k for $k = 1, 2, \ldots, 8$ of a 3D ROI from a palmprint sample

(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

(d) $k = 4$

(e) $k = 5$

(f) $k = 6$

(g) $k = 7$

(h) $k = 8$

Figure 3.7: L^k for $k = 1, 2, \ldots, 8$ of a 3D ROI from another palmprint sample from the same person as in Figure 3.6

(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

(d) $k = 4$

(e) $k = 5$

(f) $k = 6$

(g) $k = 7$

(h) $k = 8$

Figure 3.8: L^k for $k = 1,2,\ldots,8$ of a 3D ROI from a palmprint sample from the a different person from that of Figure 3.6

(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

(d) $k = 4$

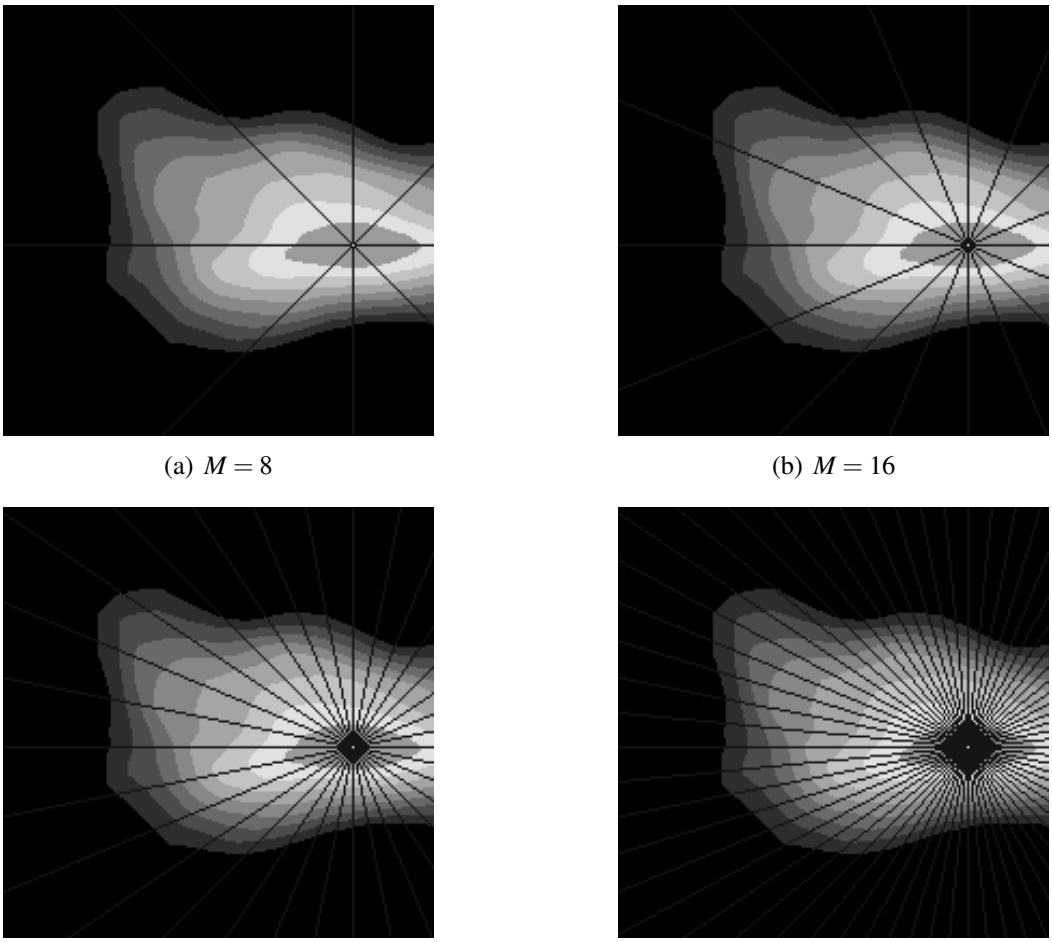
(e) $k = 5$

(f) $k = 6$

(g) $k = 7$

(h) $k = 8$

Figure 3.9: L^k for $k = 1, 2, \ldots, 8$ of a 3D ROI from another palmprint sample from the same person as in Figure 3.8



(a) $M = 8$ (b) $M = 16$

(c) $M = 32$ (d) $M = 64$

Figure 3.10: $M = 8, 16, 32, 64$ radial lines starting from the reference point

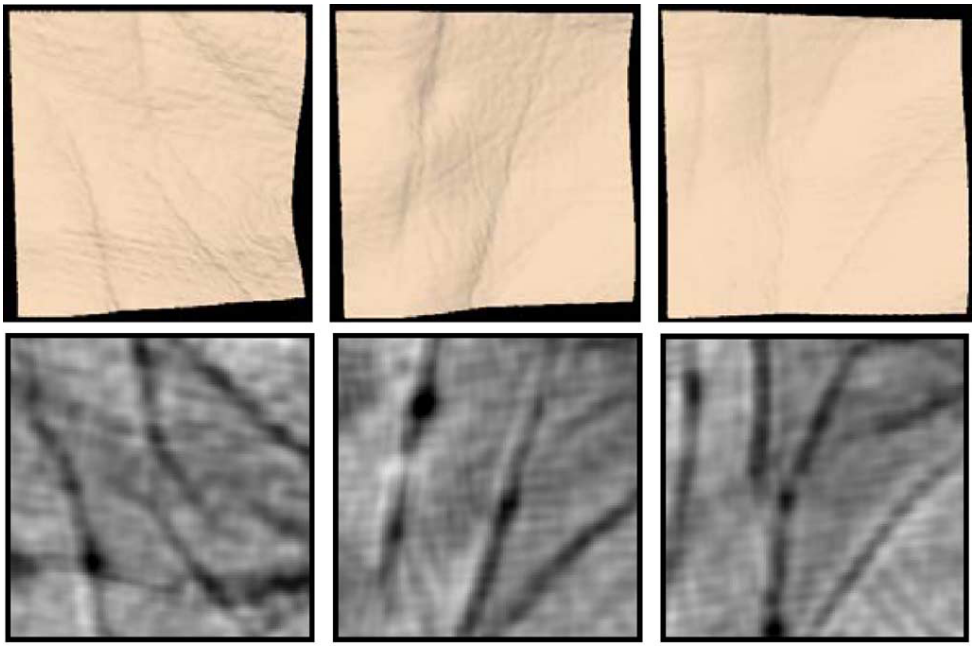


Figure 3.11: Mean Curvature Image for ROIs from three different palms

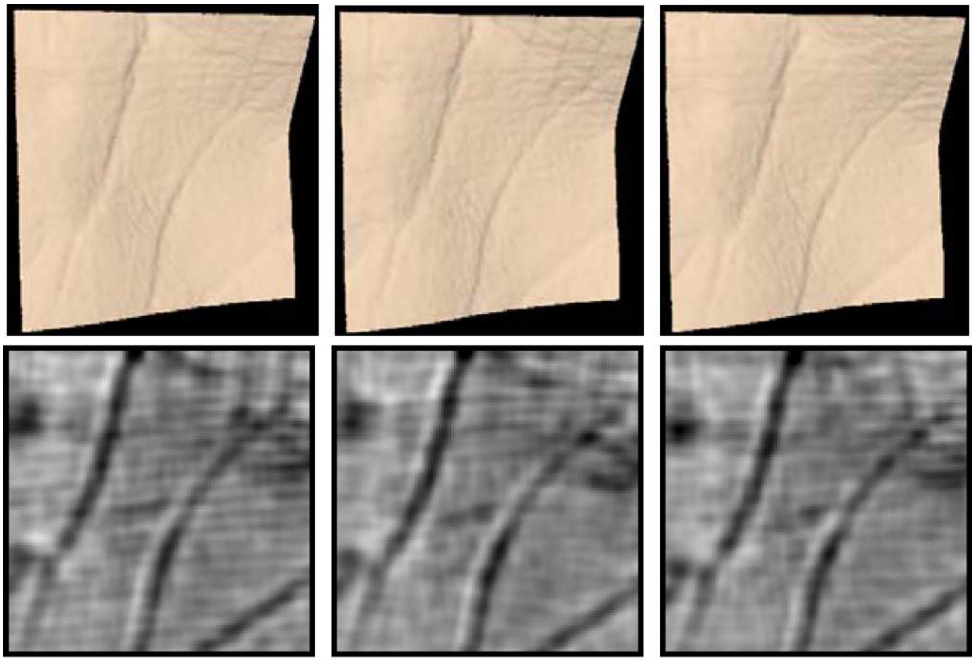


Figure 3.12: Mean Curvature Image for ROIs from three samples of the same palm

Chapter 4 Experimental Results

We used the 3D palmprint acquisition device developed in [4] to establish a 3D palmprint database containing 8000 samples collected from 400 palms. The 3D palmprint samples were collected in two separated sessions, 10 samples in each session. The average time interval between the two sessions is one month. The collection procedure required volunteers to put their palms naturally and without force on the device. As mentioned in Section 3.1, the original spatial resolution of the data was 768576. After ROI extraction, the central part (400400) was extracted and down-sampled to (200200) for feature extraction and recognition.

4.1 Optimizing Parameters

The database was divided into a training part (the first session of 4000 samples) and a testing part (the second session of 4000 samples). As described in Section 3.3.1, the dimension of the proposed features is $1+N+N\times M$. To select the value of M and N, a series of verifications on the training database were carried out where the class of the input palmprint was known. Each of the 3D samples was matched with the remaining samples in the training database. A successful match is where the two samples are from the same class. This is referred to as intra-class matching and the candidate image is said to be genuine. An unsuccessful match is referred to as inter-class matching and the candidate image is said to be an impostor. Treating the features as a point in the $1+N+N\times M$ dimension space, Euclidian distance is used as the matching score.

Table 4.1. The EER of verification for $N = 4, 8, 16$ and $M = 8, 16, 32, 64$

	M = 8 EER (%)	M = 16 EER (%)	M = 32 EER (%)	M = 64 EER (%)
N = 4	14.3	19.15	14.35	14.07
N = 8	14.2	16.3	12.32	12.54
N = 16	18.11	18.35	15.21	14.11

Table 4.1 shows the Equal Error Rate (EER) for $N = 4, 8, 16$ and $M = 8, 16, 32, 64$. The best result, as adopted in Chapter 3, is $N = 8$ and $M = 32$.

In order to balance accuracy and efficiency, we chose $N = 8$ and $M = 32$ in the following experiments. This means the features have $1 + N + N \times M = 265$ dimensions.

Table 4.2. EER of 3D palmprint verification with different feature sets

Global features	MD	HCA	RLL	MD+HCA+RLL
EER(%)	25.8	20.4	18.6	12.32

Table 4.2 shows the verification results by MD, HCA, RLL and their combined results. From the last column of Table 4.2 we can see using the combined three features will achieve a lower EER than each of the individual features.

As described in Section 3.3.1, we use the OLDA method to reduce features to a lower dimension Γ. To decide the optimal value of Γ, we carried out a series of recognition experiments on the 4000 sample training database. We divided this database into two equal parts and then chose the first five samples of every palm for training and set aside the rest for testing. As shown in Section 3.3.1, Γ is equal to q in (13). Instead of setting $q = rank(B)$, we set $q = 1, 2, \ldots, 10, 12, 15, 20, 30$.

Table 4.3 shows the recognition results.

Table 4.3. Recognition rate by OLDA for different dimensions

Γ	1	2	3	4	5	6	7	8	9	10	12	15	20	30
Recognition rate (%)	2	9.6	26	42.6	56.6	66.2	74.6	78.4	82.4	83.8	86	89.6	90.4	93.6

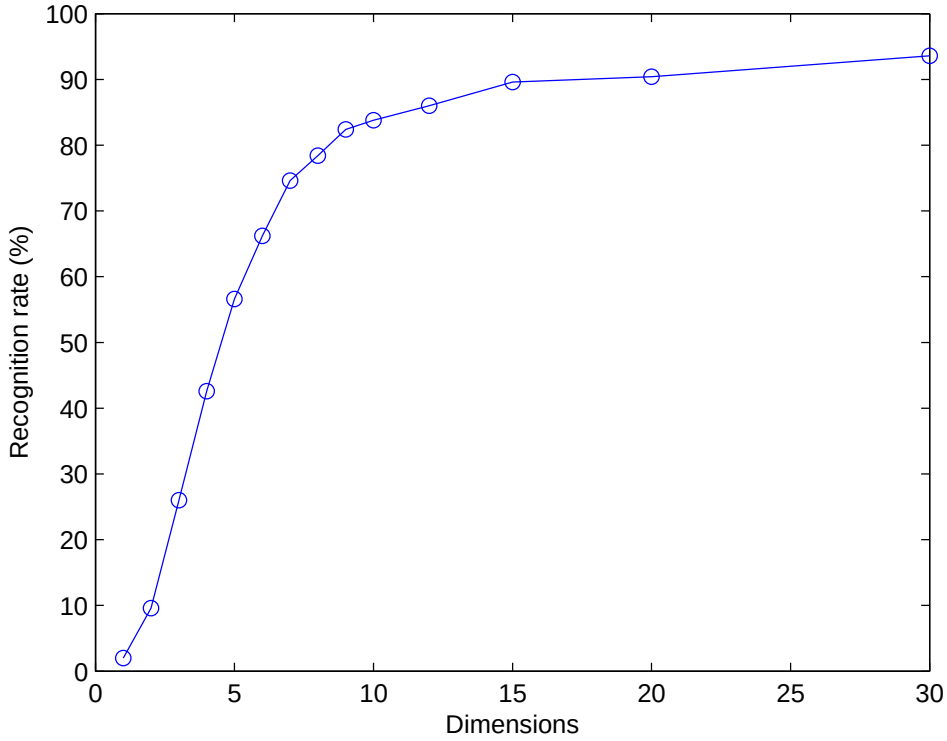


Figure 4.1: Recognition rate by OLDA for different dimensions

According to the curve shown in Figure 4.1, $\Gamma = 15$ is a good choice for the following experiments with balanced recognition rate and computation.

4.2 Genuine and Impostor Distributions

Figure 4.2 show the genuine and impostor distributions when the 15-dimensional features are applied to the 4000 training database when calculate the Euclidian distance.

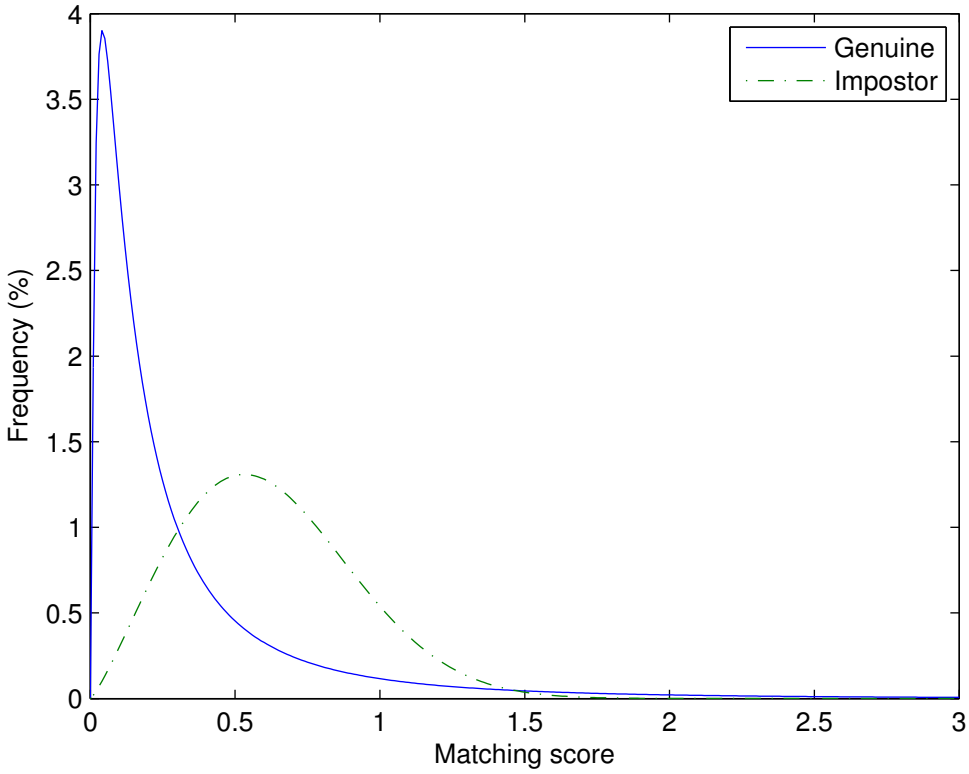


Figure 4.2: Sample distributions by matching the 15-dimensional feature vector

4.3 Recognition Performance

We next carried out the 3D palmprint classification and recognition experiments using the first sample of each class in the training database as a template and the 4000 samples in the testing database as probes, making a total of 400 templates and 4000 probes. The performance of classification and recognition is usually measured by error rate and penetration rate calculated in [23] as follows

$$\text{error rate} = \frac{\text{number of false match}}{\text{total number of probe}} \times 100\% \tag{4.1}$$

$$\text{penetration rate} = \frac{\text{number of accessed template}}{\text{total number of template in the database}} \times 100\% \qquad (4.2)$$

Obviously there is a trade-off between error rates and penetration rates. Generally speaking, if there is no classification, there are two retrieval strategies:

1. all of the templates in the database are visited and the template that gives the best matching score is regarded as the matched template, if the matching score is less than a given threshold Ψ_T

2. given a threshold Ψ_T, the search continues until a match is found that is below that threshold

Three 3D palmprint recognition matching approaches are used

1. no classification

2. coarse-level matching

3. RSVM

For no classification, we matched using the local feature MCI as described in [4]. The process we used for coarse-level matching is illustrated in Section 3.3 and involves fine-level matching using the local feature MCI. A single instance of coarse-level matching requires only $1/36000$ of the time it takes to do fine-level matching (coarse-level matching only needs 15 operations while fine-level matching must do $128 \times 128 \times (8 \times 4 + 1)$ operations, where $128 \times 128$is the size of ROI and $8 \times 4 + 1$

is the number of times the template is shifted plus the original unshifted case). For the above two approaches, the penetration rate and the error rate will vary with different thresholds Ψ_t. As for RSVM, we use the RSVM algorithm described in Section 3.3.3 to rank the templates in the database, and then match the top ρ percent by local feature MCI with the best matching score regarded as the matched template if this score is less than a given constant threshold Ψ_T. We can see from 4.2 that ρ is equal to the penetration rate. Given different thresholds Ψ_t and ρ, we carried out a series of 3D palmprint recognition experiments.

Table 4.4. Penetration rate and error rate with no classification

Penetration rate (%)	Error rate (%)
100	1.29
51.1	1.68
49.3	1.86
47.2	2.1
45.9	2.33
43.7	2.6
42.6	2.95
41.5	3.3
40.3	3.75
38.1	4.8
35.9	5.86

Table 4.5. Penetration rate and error rate using coarse-level matching

Penetration rate (%)	Error rate (%)
45.2	1.31
41.6	1.32
38.8	1.36
34.7	1.42
29.1	1.56
23.5	1.78
20.5	2.12
18.4	2.39
13.3	3.16
10.1	4.3
8.6	5.79

Table 4.6. Penetration rate and error rate using RSVM

Penetration rate (%)	Error rate (%)
30	1.3
27.5	1.33
25	1.37
22.5	1.42
20	1.49
17.5	1.63
15	1.87
12.5	2.48
10	3.35
7.5	4.41
5	5.88

Table 4.4, 4.5 and 4.6 difference in penetration rate and error rate using different recognition strategies. Even at an approximately equal error rate, the proposed coarse-level matching and RSVM approaches get a much lower penetration rate than the no classification approach. Obviously RSVM has the best performance but requires an additional off-line training process compared to coarse-level matching.

Figure 4.3 shows the results in a single plot.

Table 4.7. Running time of different recognition approaches

	No classification	Coarse-level matching	RSVM
Once feature extraction time	112ms	136ms	136ms
Once dimensionality reduction time	0ms	0.1ms	0.1ms
Ranking or coarse matching time for all templates in database	0ms	0.5ms	1.56ms
Once matching time by MCI	0.86ms	0.86ms	0.86ms
Total running time for one probe testing	456ms	292.09ms	240.86ms

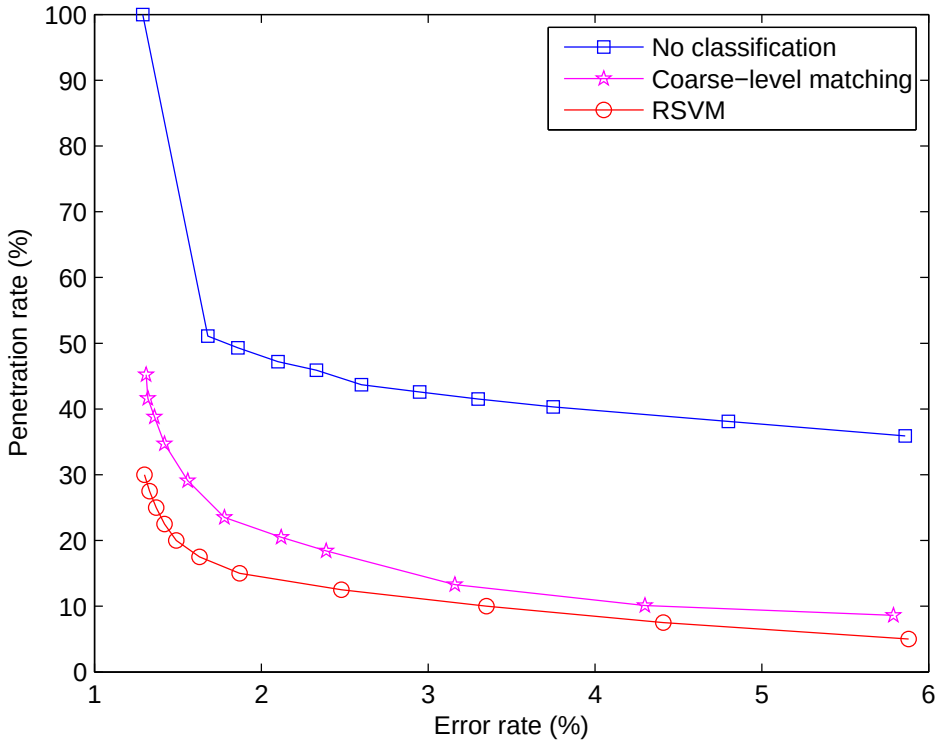


Figure 4.3: Penetration rate and error rate with different matching strategies

Table 4.7 shows the difference in time consumption for one probe. Due to the lower penetration rate, the running times are greatly reduced.

Chapter 5 Conclusions

In this chapter we conclude this dissertation by summarizing our contributions and discussing directions for future work.

5.1 Summary

In this dissertation, the authentication process are improved using 3D features. The features are stable in samples from a single person over time and distinguishable among samples from different people.

Three features adopted are Maximum Depth of palm center, Horizontal Cross-section Area and Radial Line Length from the centroid to the boundary of 3D palmprint horizontal cross-section of different levels. These cannot be extracted from 2D palmprints and are not correlated with local features, such as line and texture features. To make these features efficient for use in coarse classification, we treat them as a multi-dimensional vector and use OLDA to map it to a lower dimensional space.

We then improve the efficiency of 3D palmprint recognition using two proposed approaches, coarse-level matching and RSVM, both of which significantly reduce the penetration rate during retrieval.

Experiments are conducted on an existing 3D palmprint database of 8,000 samples. The results show that the proposed method is able to achieve an reasonable performance.

5.2 Future Work

As shown in Chapter 4, the error rate in recognition using only the (Maximum Depth, Horizontal Cross-section Area, Radial Line Length are much higher than the cases when Mean Curvature Image is used. The problem with the latter one is its computation time.

The most interesting direction for future work is to find more descriptive and efficient features for the 3D ROI.

References

[1] Adams Kong, David Zhang, and Mohamed Kamel. A survey of palmprint recognition. *Pattern Recognition*, 42(7):1408–1418, July 2009.

[2] W. Li, D Zhang, L. Zhang, G. Lu, and J. Yan. 3-D palmprint recognition with joint line and orientation features. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, (99):1–6, 2011.

[3] Wei Li, Lei Zhang, D Zhang, Guangming Lu, and Jingqi Yan. Efficient joint 2D and 3D palmprint matching with alignment refinement. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 795–801, 2010.

[4] D Zhang, Guangming Lu, Wei Li, Lei Zhang, and Nan Luo. Palmprint Recognition Using 3-D Information. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 39(5):505–519, 2009.

[5] D Zhang, Guangming Lu, Wei Li, Lei Zhang, and Nan Luo. Three Dimensional Palmprint Recognition using Structured Light Imaging. In *Biometrics: Theory, Applications and Systems, 2008. BTAS 2008. 2nd IEEE International Conference on*, pages 1–6, 2008.

[6] D Zhang, V. Kanhangad, N. Luo, and A. Kumar. Robust palmprint verification using 2D and 3D features. *Pattern Recognition*, 43(1):358–368, 2010.

[7] F Blais, M Rioux, and JA Beraldin. *Practical considerations for a design of a high precision 3-D laser scanner system*. SPIE Proceedings, 1988.

[8] Richard Hartley. Multiple view geometry in computer vision, 2000.

[9] M Halioua and H Liu. *Automated phase-measuring profilometry of 3D diffuse*

objects. Appl. Opt, 1984.

[10] Wei Li, Lei Zhang, and D Zhang. Three dimensional palmprint recognition. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, pages 4847–4852, 2009.

[11] A K Jain and Jianjiang Feng. Latent Palmprint Matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(6):1032–1047, 2009.

[12] D Zhang, WK Kong, and J You. Online palmprint identification. . . . , 2003.

[13] A.W.K. Kong and D Zhang. Competitive coding scheme for palmprint verification. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 1:520–523 Vol. 1, 2004.

[14] S. Zhenan, T. Tieniu, W. Yunhong, and Z. Li Stan. Ordinal palmprint representation for personal identification. *proceeding of IEEE*, 1:279–284, 2005.

[15] X. Wu, D Zhang, and K. Wang. Palm line extraction and matching for personal authentication. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 36(5):978–987, 2006.

[16] De-Shuang Huang, Wei Jia, and David Zhang. Palmprint verification based on principal lines. *Pattern Recognition*, 41(4):1316–1328, April 2008.

[17] Chafik Samir, Anuj Srivastava, and Mohamed Daoudi. Three-dimensional face recognition using shapes of facial curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1858–U1, 2006.

[18] Ping Yan and K.W Bowyer. Biometric Recognition Using 3D Ear Shape. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(8):1297–1308, 2007.

[19] HO Saldner. Temporal phase unwrapping: application to surface profiling of

discontinuous objects. *Applied optics*, 1997.

[20] SER Henry. Classification and uses of finger prints, 1900.

[21] Xiangqian Wu, David Zhang, Kuanquan Wang, and Bo Huang. Palmprint classification using principal lines. *Pattern Recognition*, 37(10):1987–1998, October 2004.

[22] A. Lumini, D. Maio, and D. Maltoni. Continuous versus exclusive classification for fingerprint retrieval. *Pattern Recognition Letters*, 18(10):1027–1034, 1997.

[23] D. Maltoni, D. Maio, and AK Jain. Handbook of Fingerprint Recognition. 2003.

[24] JP Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, 6:483–502, 2005.

[25] B. Matei, Y. Shan, H.S. Sawhney, Y. Tan, R. Kumar, D. Huber, and M. Hebert. Rapid object indexing using locality sensitive hashing and joint 3D-signature space estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1111–1126, 2006.

[26] H. Chen and B Bhanu. Efficient recognition of highly similar 3D objects in range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):172–179, 2009.

[27] Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM Request Permissions, July 2002.

Appendix A: Core Matlab Code

Create large arrays without asking for contiguous memory.

```
function result = createArrays(nArrays, arraySize, datatype)
%CREATEARRAYS
% This creates an cell array which does NOT require
% contiguous memory.
%
% To use it:
%   myArray = createArrays(numberOfArrays, [x y], 'single');
%
% To access the elements:
%   myArray{1}{2,3} = 10;
%   myArray{1} = zeros(500, 800, 'single');
    result = cell(1, nArrays);
    for i = 1 : nArrays
        result{i} = zeros(arraySize, datatype);
    end
end
```

Code to find the reference plane.

```
function [ d_ref ] = find_ref_plane( roi )
%FIND_REF_PLANE Find the depth of reference plane of an ROI
%   Detailed explanation goes here

        [~,roi_size] = size(roi);

        refplane = roi(1:(roi_size*0.25),
         (roi_size*0.25):(roi_size*0.9));

        d_ref = max(max(refplane));

end
```

Code to extract the X coordinates from original data.

```
clear;

prefix = '..\..\3D_palm';

%% File to process
load_dir = [prefix filesep '3DPalm_xyz'];
file_mask = [load_dir filesep '*.dat'];
file_list = dir(file_mask);
file_list = char({file_list.name});

%% Get X coordinates from file
tic;
dat_file = reshape(dlmread([load_dir filesep file_list(1,:)]),
                           768,576,3);
palmX=dat_file(:,1,1);

%% Save result to file
save_dir = ['output'];
save([save_dir filesep '3Dpalm_x.mat'],'palmX');
toc;
```

Code to remove outliers from a vector.

```
function [b,idx,outliers] = deleteoutliers(a,alpha,rep);
% [B, IDX, OUTLIERS] = DELETEOUTLIERS(A, ALPHA, REP)
%
% For input vector A, returns a vector B with outliers (at the
% significance level alpha) removed. Also, optional output
% argument idx returns the indices in A of outlier values. Optional
% output argument outliers returns the outlying values in A.
%
% ALPHA is the significance level for determination of outliers.
% If not provided, alpha defaults to 0.05.
%
% REP is an optional argument that forces the replacement of
% removed elements with NaNs to presereve the length of a.
%
% This is an iterative implementation of the Grubbs Test that
% tests one value at a time. In any given iteration, the tested
% value is either thehighest value, or the lowest, and is the
% value that is furthest from the sample mean. Infinite elements
% are discarded if rep is 0, or replaced with NaNs if rep is 1.
%
if nargin == 1
        alpha = 0.05;
        rep = 0;
elseif nargin == 2
        rep = 0;
elseif nargin == 3
        if ~ismember(rep,[0 1])
                error('Please enter a 1 or a 0 for optional argument rep.'
        end
elseif nargin > 3
        error('Requires 1,2, or 3 input arguments.');
end

if isempty(alpha)
        alpha = 0.05;
end

b = a;
b(isinf(a)) = NaN;

%Delete outliers:
outlier = 1;
```

```
43 while outlier
44         tmp = b(~isnan(b));
45         meanval = mean(tmp);
46         maxval = tmp(find(abs(tmp-mean(tmp))==
47                                     max(abs(tmp-mean(tmp)))
48                                    ));
49         maxval = maxval(1);
50         sdval = std(tmp);
51         tn = abs((maxval-meanval)/sdval);
52         critval = zcritical(alpha,length(tmp));
53         outlier = tn > critval;
54         if outlier
55                 tmp = find(a == maxval);
56                 b(tmp) = NaN;
57         end
58 end
59
60 if nargout >= 2
61         idx = find(isnan(b));
62 end
63 if nargout > 2
64         outliers = a(idx);
65 end
66 if ~rep
67         b=b(~isnan(b));
68 end
69 return
70
71 function zcrit = zcritical(alpha,n)
72 %ZCRIT = ZCRITICAL(ALPHA,N)
73 % Computes the critical z value for rejecting outliers
74 % (GRUBBS TEST)
75
76 tcrit = tinv(alpha/(2*n),n-2);
77 zcrit = (n-1)/sqrt(n)*(sqrt(tcrit^2/(n-2+tcrit^2)));
```

Code for batch feature extraction.

```
clear;

%% Files to process
save_dir = ['output'];

use_original_data = true;
% Toggle this for data source selection
% use_original_data = false;

if (use_original_data)
    data_dir = ['..' filesep 'rawdata'];
    file_mask = [data_dir filesep '*.zonly'];
    sample_width = 768;
    sample_height = 576;
    roi_size = 400;
else
    data_dir = ['..' filesep 'rawdata' filesep 'roi'];
    file_mask = [data_dir filesep '*.dat'];
    sample_width = 128;
    sample_height = 128;
    roi_size = 128;
end

file_list = dir(file_mask);
file_list = char({file_list.name});

num_of_samples = length(file_list);
sample_per_person = 10;
num_of_people = num_of_samples/sample_per_person;
% Toggle this for limited number of files
// num_of_people = 100;

feature_dimension = roi_size;
% feature_dimension = roi_size*roi_size;

features = zeros(num_of_people*sample_per_person,
                 feature_dimension);

%% Parallel read files
tic;
for current_person = 1:num_of_people
    disp(['Loading data from ' num2str(current_person) ' of \
```

```
43                  ' num2str(num_of_people) ' people.']);
44      for current_sample = 1:sample_per_person
45          file_id = (current_person - 1) * 10 + current_sample;
46          sample_filename = strtrim([data_dir filesep
47                                                      file_list(file_id,
48
49          mat = file2matrix(sample_filename,
50                                                  sample_width,
51                                                  sample_height);
52
53          if (use_original_data)
54              % Crop ROI from original data
55              mat = mat(235:(234+roi_size),69:(68+roi_size));
56          end;
57
58          % Extract feature for this sample
59          features(file_id,:) = calc_feature(mat);
60      end
61  end
62
63  save(['output' filesep 'features.mat'], 'features');
64  toc;
```

Code for curvature extraction.

```
function [K,H,Pmax,Pmin] = surfature(X,Y,Z),
% SURFATURE -  COMPUTE GAUSSIAN AND MEAN CURVATURES OF A SURFACE
%   [K,H] = SURFATURE(X,Y,Z), WHERE X,Y,Z ARE 2D ARRAYS OF POINTS
%   ON THE SURFACE.  K AND H ARE THE GAUSSIAN AND MEAN CURVATURES,
%   RESPECTIVELY.
%
%   SURFATURE RETURNS 2 ADDITIONAL ARGUEMENTS,
%
%   [K,H,Pmax,Pmin] = SURFATURE(...), WHERE Pmax AND Pmin ARE THE
%   MINIMUM AND MAXIMUM CURVATURES AT EACH POINT, RESPECTIVELY.


% First Derivatives
[Xu,Xv] = gradient(X);
[Yu,Yv] = gradient(Y);
[Zu,Zv] = gradient(Z);

% Second Derivatives
[Xuu,Xuv] = gradient(Xu);
[Yuu,Yuv] = gradient(Yu);
[Zuu,Zuv] = gradient(Zu);

[Xuv,Xvv] = gradient(Xv);
[Yuv,Yvv] = gradient(Yv);
[Zuv,Zvv] = gradient(Zv);

% Reshape 2D Arrays into Vectors
Xu = Xu(:);   Yu = Yu(:);   Zu = Zu(:);
Xv = Xv(:);   Yv = Yv(:);   Zv = Zv(:);
Xuu = Xuu(:); Yuu = Yuu(:); Zuu = Zuu(:);
Xuv = Xuv(:); Yuv = Yuv(:); Zuv = Zuv(:);
Xvv = Xvv(:); Yvv = Yvv(:); Zvv = Zvv(:);

Xu          =   [Xu Yu Zu];
Xv          =   [Xv Yv Zv];
Xuu         =   [Xuu Yuu Zuu];
Xuv         =   [Xuv Yuv Zuv];
Xvv         =   [Xvv Yvv Zvv];

% First fundamental Coeffecients of the surface (E,F,G)
E           =   dot(Xu,Xu,2);
F           =   dot(Xu,Xv,2);
```

```
G            =    dot(Xv,Xv,2);

m            =    cross(Xu,Xv,2);
p            =    sqrt(dot(m,m,2));
n            =    m./[p p p];

% Second fundamental Coeffecients of the surface (L,M,N)
L            =    dot(Xuu,n,2);
M            =    dot(Xuv,n,2);
N            =    dot(Xvv,n,2);

[s,t] = size(Z);

% Gaussian Curvature
K = (L.*N - M.^2)./(E.*G - F.^2);
K = reshape(K,s,t);

% Mean Curvature
H = (E.*N + G.*L - 2.*F.*M)./(2*(E.*G - F.^2));
H = reshape(H,s,t);

% Principal Curvatures
Pmax = H + sqrt(H.^2 - K);
Pmin = H - sqrt(H.^2 - K);
```

Code to test feature performance.

```
%% Load features
load(['output' filesep 'features.mat']);

[total,~] = size(features);

%% Split into training set and testing set
training_id=unique([
    1:10:total
    2:10:total
    3:10:total
    4:10:total
    5:10:total
    6:10:total
    ]);

trainingset = features(training_id,:);

test_id=unique([
    7:10:total
    8:10:total
    9:10:total
    10:10:total
    ]);

testingset = features(test_id,:);

%% Find match for each test input
nearest_id = knnsearch(trainingset, testingset, 'k', 10);

%% Interpret the match to person
[total_training,~] = size(trainingset);
training_sample_per_person = total_training / total * 10;
nearest_person = ceil(nearest_id/training_sample_per_person);

%% Check performance
[total_testing,~] = size(testingset);
testing_sample_per_person = total_testing / total * 10;

for i = 1:total_testing
    correct(i) = eq(nearest_person(i),
                    ceil(i/testing_sample_per_person));
end
```

```
43
44 correct = correct';
45
46 accuracy = sum(correct)/total_testing
```

Code to create mask according to gradient.

```
sizeH = 200; %
sizeW = 200; %
lnum = 8;

pixDis = 0.28;
% load facesMatrix64.mat;
temp = pixDis * ((sizeH-2)/2+0.5);
[X,Y] = meshgrid(-temp : pixDis : temp);

fid = fopen('../rawdata/Sub3D_I_3_0.dat', 'r');
% fid = fopen('../rawdata/Sub3D_II_100_0.dat', 'r');

Z = fread(fid, [sizeH,sizeW], 'double');
fclose(fid);

[fx, fy] = gradient(Z);
fxy = fx.^2 + fy.^2;

noiseP = find(fxy>0.1);
% 0.1 used for corrected and smoothed Sub3D,
% 1 used for original Sub3D

flag = ones(sizeH, sizeW); %1 for valid point, 0 for invalid point
flag(noiseP) = 0;
flag = 1 - flag;
se = strel('disk',5);
flag = imdilate(flag, se);
flag = 1 - flag;
noiseP = find(flag==0);
meanZ = sum(sum(Z .* flag)) / (sizeH*sizeW - length(noiseP));
Z(noiseP) = 5;
%neend't smooth
% Z = smooth(Z, 7);

%%for show the mask which get rid of bad quality region
% flag(1,:) = 0; flag(end,:) = 0;
% flag(:,1) = 0; flag(:,end) = 0;
% imshow(flag');
```

Code to determine the reference plane and MD.

```
%calculate the reference 0 plane
% use Z(6:35, 65:136) for calculate the mean value as reference 0
refRect = Z(6:35, 65:136);
refFlag = flag(6:35, 65:136);
refVal = sum(sum(refRect .* refFlag)) / sum(sum(refFlag));
Z = Z - refVal;

%search the min point in Z(65:190, 41:160)
rectforMin = Z(65:190, 41:160);
rectforMin(1:35, 1:25) = 5;
rectforMin(1:35, end-25:end) = 5;
% flagforMin = Z(65:190, 41:160);

palmH = min(min(rectforMin));
ind = find(Z == palmH);
% Z(ind) = 5;
```

Code to grow each level from the reference point.

```
%find the level regions from 0 to deepest point
%find the region > 0
palmH = -palmH;
Z = -Z;
Ln = 8;
step = palmH / Ln;
levelH = [0:step:palmH];

for i = 1:Ln
    L0 = zeros(sizeH, sizeW);
    L0( find( Z>=levelH(Ln-i+1) ) ) = 1;
    %the 1st level
    if i==1
        [L,num] = bwlabel(L0);

        if num > 1
            for j = 1:num
                indL = find(L==j);
                if length(find(indL==ind)) > 0
                    L0(:,:) = 0;
                    L0(indL) = 1;
                end
            end
        end
        Lp = L0;
        L0 = logical(L0');
        [x, y] = find(L0);
        temp = find(L0);
        mx = mean(x);
        my = mean(y);

%         saveIm(L0, i);
        figure;
        imshow(L0);
    else
        %dilate the Lp (previous level) and than & with L0
        se = strel('disk',35 - 3*i);
        L1 = imdilate(Lp, se);
        L0 = L0 & L1;
        Lp = L0;
        L0 = L0';
%         saveIm(L0, i);
```

```
43          figure;
44          imshow(L0);
45      end
46  end
```

Code to show the growed levels.

```
%for show
im_level = zeros(sizeH, sizeW); %for show the levels
%find the level regions from 0 to deepest point
%find the region > 0
palmH = -palmH;
Z = -Z';
Ln = 8;
step = palmH / Ln;
levelH = [0:step:palmH];
for i = 1:Ln
    L0 = zeros(sizeH, sizeW);
    L0( find( Z>=levelH(Ln-i+1) ) ) = 1;

    %the 1st level
    if i==1
        [L,num] = bwlabel(L0);
        if num > 1
            for j = 1:num
                indL = find(L==j);
                if length(find(indL==ind)) > 0
                    L0(:,:) = 0;
                    L0(indL) = 1;
                end
            end
        end
        Lp = L0;
        [x, y] = find(L0);
        temp = find(L0);
        mx = round(mean(x));
        my = round(mean(y));
        im_level(temp) = 155;

%         saveIm(L0, i);
%         figure;
%         imshow(L0);
    else
        %dilate the Lp (previous level) and than & with L0
        se = strel('disk',35 - 3*i);
        L1 = imdilate(Lp, se);
%         figure; imshow(Lp);
%         figure; imshow(L1);
        L0 = L0 & L1;
```

```
43  %           figure; imshow(L0);
44          Ltemp = L0 - Lp;
45  %           figure; imshow(Ltemp);
46          Lp = L0;
47          temp = find(Ltemp);
48          im_level(temp) = 255-(i-1)*30;
49  %           saveIm(L0, i);
50  %           figure;
51  %           imshow(L0);
52      end
53  end
54
55  im_level = uint8(im_level);
56  % imshow(im_level);
57  % imwrite(im_level, 'figures/im_levels.bmp');
```

Other functions.

```
function saveIm(data, nlevel)
filename = ['../rawdata/bmp/Sub3D_I_4_9',
                          '_' num2str(nlevel), '.bmp'];
imwrite(data, filename);


function [z] = smooth(z, fs)
% z -- 128*128
% n -- filter size
t = floor(fs/2)-1;
for i = 0:t;
    [m,n] = size(z);
    z = [z(:, 2*i+1) z(:,:) z(:, n-2*i)];
end

for i = 0:t;
    [m,n] = size(z);
    z = [z(2*i+1, :); z(:,:); z(m-2*i, :)];
end

h = ones(fs,fs)/(fs*fs);
z = filter2(h, z, 'valid');
```