

FINGER PRINT REGONIZATION USING RIDGE FEATURES

Suresh.A,¹ Assistant Professor(Sr.Gr)/IT. Department of Information Technology,
Sri Ramakrishna Institute of Technology, Coimbatore-641 010.
Dr.P.Malathi²,Principal, Bharathiyar Institute of Engineering for Women, Salem

Abstract— this paper introduces a novel fingerprint matching algorithm using both ridge features and the conventional minutiae feature to increase the recognition performance against nonlinear deformation in fingerprints. The proposed ridge features are composed of four elements: ridge count, ridge length, ridge curvature direction, and ridge type. These ridge features have some advantages in that they can represent the topology information in entire ridge patterns existing between two minutiae and are not changed by nonlinear deformation of the finger. For extracting ridge features, it also defines the ridge-based coordinate system in a skeletonized image. With the proposed ridge features and conventional minutiae features (minutiae type, orientation, and position), this system propose a novel matching scheme using a breadth-first search to detect the matched minutiae pairs incrementally. Following that, the maximum score is computed and used as the final matching score of two fingerprints. Thus, it conclude that the proposed ridge feature gives additional information for fingerprint matching with little increment in template size and can be used in conjunction with existing minutiae features to increase the accuracy and robustness of fingerprint recognition systems.

Keywords— Bioinformatics, Ridge, Curvature, Authentication

I. INTRODUCTION

A fingerprint is the impression made by the papillary ridges on the ends of the fingers and thumbs. Fingerprints afford an infallible means of personal identification, because the ridge arrangement on every finger of every human being is unique and does not alter with growth or age. Fingerprints serve to reveal an individual's true identity despite personal denial, assumed names, or changes in personal appearance

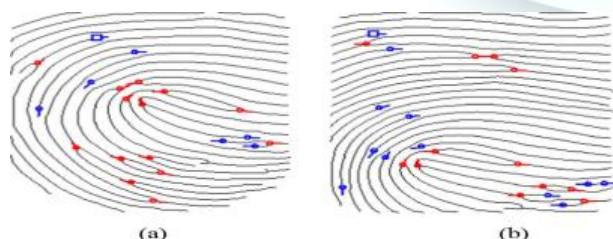


Fig 1.1 Example of skin distortions.

Resulting from age, disease, plastic surgery, or accident. The practice of utilizing fingerprints as a means of identification is an indispensable aid to modern law enforcement. Each ridge of the epidermis (outer skin) is dotted

with sweat pores for its entire length and is anchored to the dermis (inner skin) by a double row of peg like protuberances, or papillae. Injuries such as superficial burns, abrasions, or cuts do not affect the ridge

Structure or alter the dermal papillae, and the original pattern is duplicated in any new skin that grows. An injury that destroys the dermal papillae, however, will permanently obliterate the ridges. Fingerprint recognition has been widely adopted for user identification due to its reliable performance, usability, and low cost compared with other biometrics such as signature, iris, face, and gait recognition. It is used in a wide range of forensic and commercial applications, e.g., criminal investigation, e-commerce, and electronic personal ID cards. Although significant improvement in fingerprint recognition has been achieved, many challenging tasks still remain. Among them, nonlinear distortions, presented in touch-based fingerprint sensing, make fingerprint matching more difficult.

As shown in Fig1.1, even though these two fingerprint images are from the same individual, the relative positions of the minutiae are very different due to skin distortions. This distortion is an inevitable problem since it is usually associated with several parameters including skin elasticity, nonuniform pressure applied by the subject, different finger placement with the sensor, etc.

II.EXISTING SYSTEM

2.1.1 FINGERPRINT PREPROCESSING

Before extracting the proposed ridge features, need to perform some preprocessing steps (see Fig.2.1). These steps include typical feature extraction procedures as well as additional procedures for quality estimation and circular variance estimation. first divide the image into 8×8 pixel blocks. Then, the mean and variance values of each block are

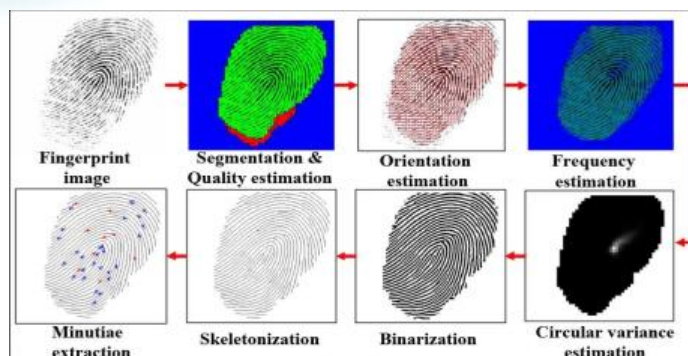


Fig.2.1 Overall Preprocessing steps

calculated to segment the fingerprint regions in the image. The Gabor filter is applied to enhance the image and obtain a skeletonized ridge image. Then, the minutiae (end points and bifurcations) are detected in the skeletonized image.

The quality estimation procedure is performed in order to avoid extracting false minutiae from poor quality regions and to enhance the confidence level of the extracted minutiae set. Furthermore, in regions where ridge flows change rapidly, such as the area around a singular point, it is hard to estimate the ridge orientations accurately or to extract the thinned ridge patterns consistently. Therefore, to detect regions which have large curvature, then apply circular variance estimation. The circular variance of the ridge flows in a given block is Calculated as follows

$$:Var(\theta) = 1 - 1/n [(\sum_{i=1}^n \cos \theta_i)^2 + (\sum_{i=1}^n \sin \theta_i)^2]$$

Where θ_i and n represent the estimated orientation of the i^{th} block and the number of neighboring blocks around the i^{th} block, respectively. In this project, it uses eight neighboring blocks. Quality estimation and circular variance values are used to avoid generating feature vectors in poor quality regions or in regions around singular points. Moreover, some post processing steps to remove falsely extracted ridges, such as short ridges and bridges. So the system can then extract the ridge structures consistently against various noise sources

pixel value will not be affected much. The affected value are very less compare than conventional watermarking.

2.2 PROPOSED SYSTEM

PROPOSED RIDGE-BASED COORDINATE SYSTEM

2.2.1 RIDGE FEATURE EXTRACTION

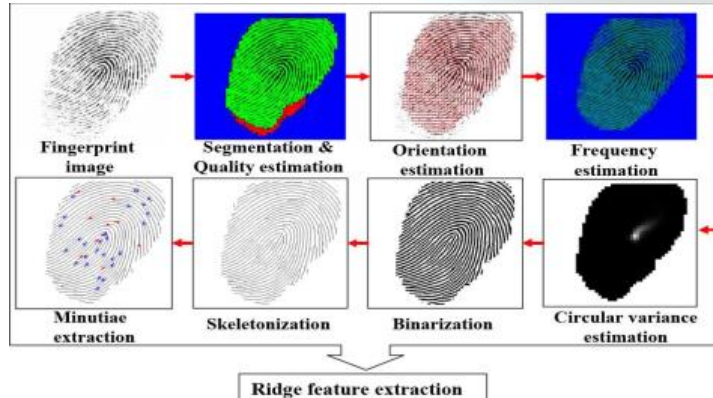
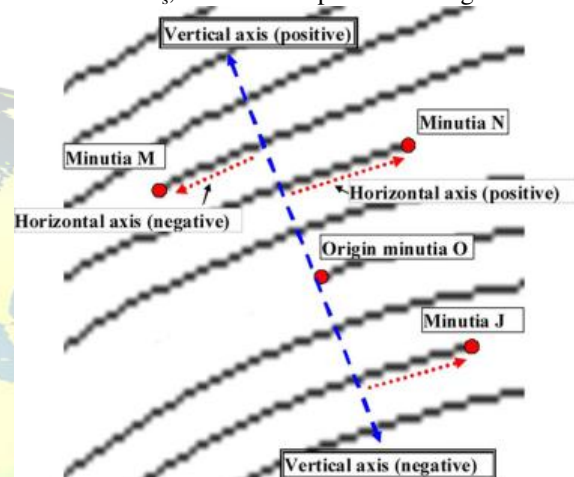


Fig.2.2 Preprocessing steps and Ridge Feature Extraction

After performing the preprocessing steps, obtain the skeletonized ridges and minutiae information from the fingerprint image. Then define ridge coordinates and extract ridge features between two minutiae. As shown in Fig.2.3, each ridge-based coordinate system is defined by a minutia (called origin) and vertical and horizontal axes starting from the origin minutia. First, the vertical axis is defined by drawing a line passing through the origin and orthogonal to the orientation of the origin. The axis also traverses the ridge flows orthogonally.

$$V_s = \text{sign}(\vec{O} \times \vec{V}_n)$$

where \vec{V}_s , \vec{O} and \vec{V}_n represent the sign of the vertical



axis, the minutia orientation vector, and the unit vector of the vertical axis, respectively. Thus determine the positive and the negative side of the vertical axis by checking the sign value of V_s .

To represent the relative position of the minutiae according to the origin, horizontal axes should be defined. The horizontal axes are defined as ridges intersecting the vertical axis. To define the sign of each horizontal axis, the cross product between the vectors pointing from the intersection to the vertical and horizontal axes is calculated as follows:

$$H_s = \text{sign}(\vec{H}_n \times \vec{V}_n)$$

where H_s , \vec{H}_n and \vec{V}_n represent the sign of the horizontal axis, the vector pointing from the intersection to the horizontal and the vertical axis, respectively. In the ridge-based coordinate system, the ridge features that describe the relationship between the origin (minutia in Fig.2.3) and an arbitrary minutia are described as follows:

$$\vec{V} = (rc, rl, rcd, rt)$$

Where rc , rl , rcd , and rt represent the ridge count, ridge length, ridge curvature direction, and ridge type, respectively. These four components form a ridge-based feature vector between two minutiae and this feature vector is used in the



matching process. In the following sections, it will explain in detail these ridge features were selected and the methods for extracting these features.

RIDGE FEATURE EXTRACTION

In the general ridge count methods the number of ridges that intersect the straight line between two minutiae in the

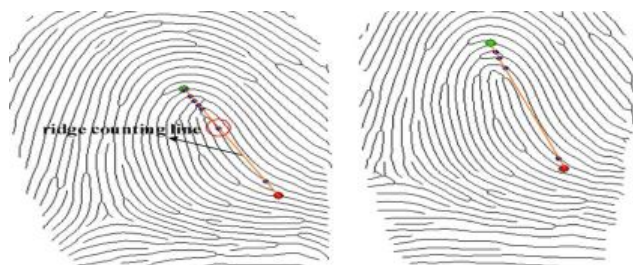


Fig.2.4 Examples of ridge-counting errors using the general ridge counting methods

spatial domain is counted. However, when the ridge-counting line is parallel to the ridge structures, the line may meet the same ridge at one point, at more than two points, or at no point, due to skin deformation (see Fig.2.4).

Therefore, unlike existing ridge-counting methods, here, the ridge count (rc) is calculated by counting the number of ridges along the vertical axis until the axis meets the ridge attached to the neighboring minutia. The vertical axis is perpendicular to the ridge structures. Thus, the counted numbers are less affected by skin deformation than in the results of the general ridge counting methods. Furthermore, to increase the discriminating power of the ridge count (rc) feature, it also consider the direction of the ridge count line. The ridge count (rc) is not always a positive number and the sign of the ridge count follows the sign of the vertical axis. If two minutiae are directly connected by the same ridge, the ridge count would be zero. The ridge length (rl) is the distance on the horizontal axis from the intersection of the vertical and

than 16 pixels. Therefore, it can set the threshold of the ridge length feature to determine the same fingerprint as 16 pixels.

The ridge length value also has a sign and follows the sign of the related horizontal axis to improve the discriminating power.

To use more topology information in ridge patterns for matching, the ridge curvature direction is also considered. As shown in Fig.2.5, even though the ridge count and ridge length values are very similar, the shapes of the ridge patterns may be different [Fig.2.5 (a), Fig.2.5 (b)]. The ridge curvature direction is defined as follows:

$$rcd = \text{sign}(\sum_{i=1}^N \vec{v}_i \times \vec{v}_{i-1})$$

Where \vec{v}_i represents the i^{th} vector between the sampling points along the horizontal axis from the intersection of the vertical and horizontal axes to the minutia (see Fig.2.5) and represents the number of sampling points. In this project, set the sampling point every 8 pixels on the ridges. Then, by checking the sign of this value, we can determine the ridge curvature direction.

The ridge curvature direction feature is robust to skin deformation but some errors may still occur. First, ridges may have more than two inflection points, which makes it hard to define This feature. Second, some ridges are too straight to define a curved direction. Therefore, to avoid the error caused by more than two inflection points, to empirically limit the maximum length of ridges to 80 pixels Fig.2.6 Examples of ridge types. Additionally, to avoid the error caused by a straight ridge, defined the ridge curvature direction as 0. Due to the feature extraction error, skin condition changes, and different finger pressures, end points may appear as bifurcations and vice versa. Therefore, considering these facts and to further improve the discriminating power of ridge features, the ridge type (rt) is used as one of the ridge features instead of a minutia type. To determine the ridge type (rt), each minutia is first classified as an end point or a bifurcation. If a minutia is an end point, there is only one ridge belonging to the minutia. If a minutia is a bifurcation, there are three ridges connected to the minutiae. Next, the type of ridge associated with the minutia is determined as one of four types according to the type of the minutia and the relative position of the ridges. As shown in Fig.2.6, if a minutia is an end point, the ridge type is defined as E.

Fig 2.5 Ridge curvature direction (a) concave shape (b) convex shape

horizontal axis to a minutia. As shown in the figure, the absolute differences of ridge length elements are mostly less

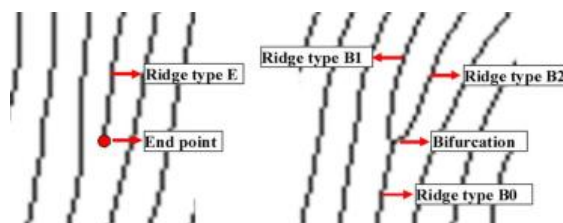
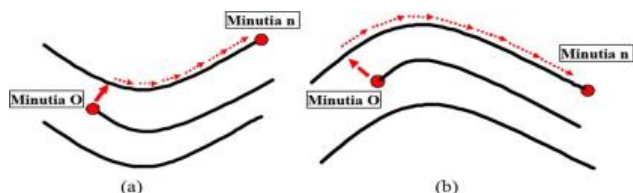


Fig.2.6 Examples of ridge types

In a bifurcation case, the three ridges are labeled by checking the angle between each ridge and the minutia orientation. A triangle is created by three points on the ridges (equidistant from the bifurcation). If the vertex of the triangle is not on the shortest side of the triangle, then the ridge belongs to the vertex and is defined as type B_0 . The other two ridges are classified as type B_1 and B_2 , moving in a clockwise direction from B_0 . Generally speaking, ridge type can change only into ridge type B_1 or B_2 . However, type E cannot be converted into type B_0 . Therefore, this information used in the fingerprint matching.

FINGERPRINT MATCHING

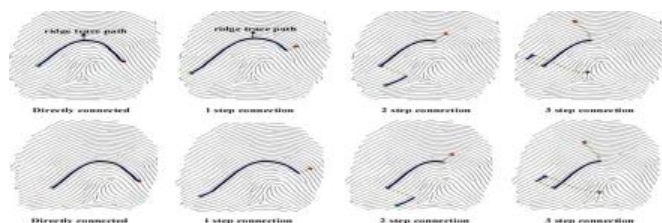
The ridge feature vectors between the minutiae in the ridge coordinate system can be expressed as a directional graph whose nodes are minutiae and whose edges are ridge feature vectors.

Thus, the graph matching methods to utilize the ridge feature vectors in fingerprint matching. They first defined the local neighborhood of each minutia, called K-plet, which consists of the K-nearest minutiae from a center minutia. The comparison of two K-plets is performed by computing the distance between the two strings obtained by concatenating the neighboring minutiae, sorted by their radial distance with respect to the center minutia. Neighborhoods are matched by dynamic programming and a match of local neighborhoods is propagated with a breadth-first fashion. Thus, this matching scheme to the ridge-based coordinate system, since the ridge-based coordinate system can be represented as a graph and each coordinate system makes a local neighborhood.

Dynamic programming is applied to find the optimal solution in matching two string sequences in the enrolled and input ridge-based coordinates. The ridge feature vectors in a ridge-based coordinate system are arranged in the order of their ridge count feature component (rc), then the order is invariant intrinsically. Therefore, the feature vectors in a ridge-based coordinate system can be stored as the elements of an ordered sequence. Thus, all the enrolled and input ridge-based coordinates are compared one by one and a similarity score is computed for the dynamic programming. The similarity score is based on the Bayesian decision rule and is calculated as follows:

$$\begin{cases} \text{score} = P(w_1 / X), & \text{when } P(w_1 / X) > P(w_2 / X) \\ \text{score} = 0 & \text{Otherwise} \end{cases}$$

Where X is the absolute difference between two feature vectors, w_1 is the correctly matched class, and w_2 is the incorrectly matched class. In order to calculate the posterior probability, it assumed that the prior probabilities of w_1 and w_2 are equal. For the ridge feature vector, the three feature elements (ridge count, ridge length, and ridge curvature direction) are used to calculate the scores and the ridge type feature is used to check the validity of the candidate pairs. After that, select the top degree of matched ridge-based coordinate pairs. In this project, set the value N as 10. For every initially matched pair, then perform a



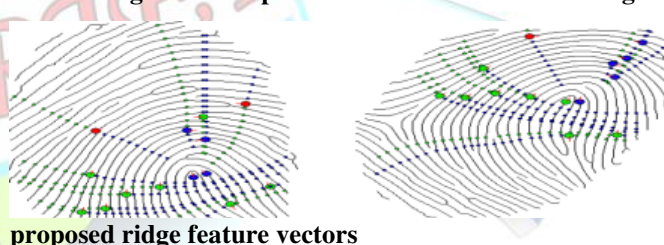
BFS to increment the match for other neighboring ridge-coordinate systems. However, there is not always a path for every minutiae pair because it do not extract ridge features in the fingerprint regions which have low quality or a high curvature.

Fig.2.7 Examples of corresponding ridge feature vectors according to number of connection steps (upper and lower row images are from the same finger).

Therefore, find a detour path to perform the BFS. For example, even if it is not possible to directly extract the ridge feature vector between minutia and due to the absence of a path, it is still possible to obtain the ridge feature vector by including minutia (as). Fig.2.7 shows some examples of the corresponding ridge feature vectors using the detour, as the number of connection steps increases. Then check the validity of the matched coordinate pairs using the relative position and orientation of the minutiae used in conventional Minutiae-based matching. If the relative position and orientation of the minutiae in the coordinate pair are also matched, it can be sure that these minutiae are correctly matched.

Then count the number of matched minutiae and store them. Finally, after the execution of the BFS procedure for every initial matched pair, to find the maximum number of matched minutiae between two fingerprints. Fig.2.8 shows an example of matched minutiae using the proposed method. As shown in the figure, even if two impressions of the same finger

Fig.2.8 Examples of Matched minutiae using the



are different due to skin distortion, many minutiae are matched correctly.

To compute the matching score, it must consider both the degree of overlap between two impressions and the degree of similarity of the overlapped region. Thus, the matching score can be computed as follows:

$$S_m = \frac{L \times L}{m_0 \times n_0} \times \frac{L \times L}{N_1 \times N_1}$$

Where L, N_1 and N_2 are the number of matched minutiae, the number of minutiae in an input image, and the number of minutiae in a template image, respectively. And are the numbers of minutiae in the overlapping regions of the query and template images, respectively. The overlapped regions are where two



fingerprints intersect after the linear transformation (translation and rotation) using the matched minutiae.

additional procedures for quality estimation and circular variance estimation.

3.1 PROBLEM DEFINITION

Fingerprint recognition has been widely adopted for user identification due to its reliable performance, usability, and low cost compared with other biometrics such as signature, iris, face, and gait recognition. It is used in a wide range of forensic and commercial applications, e.g., criminal investigation, e-commerce, and electronic personal ID cards.

Although significant improvement in fingerprint recognition has been achieved, many challenging tasks still remain. Among them, nonlinear distortions, presented in touch-based fingerprint sensing, make fingerprint matching more difficult.

This project introduces a novel fingerprint matching algorithm using both ridge features and the conventional minutiae feature to increase the recognition performance against nonlinear deformation in fingerprints.

3.2 OVERVIEW

Fingerprints are classified in a three-way process: by the shapes and contours of individual patterns, by noting the finger positions of the pattern types, and by relative size, determined by counting the ridges in loops and by tracing the ridges in whorls. The information obtained in this way is incorporated in a concise formula, which is known as the individual's fingerprint classification.

Minutia, the technique of fingerprinting, involves cleaning the fingers in benzene or ether, drying them, and then rolling the balls of each over a glass surface coated with printer's ink. Each finger is then carefully rolled on prepared cards according to an exact technique designed to obtain a light gray impression with clear spaces showing between each ridge so that the ridges may be counted and traced. Simultaneous impressions are also taken of all fingers and thumbs.

3.3. SYSTEM IMPLEMENTATION

3.3.1 MODULES

3.3.2 FINGERPRINT PREPROCESSING

Before extracting the proposed ridge features, the system has to perform some preprocessing steps. These steps include typical feature extraction procedures as well as

3.3.3 SEGMENTATION AND QUALITY ESTIMATION

Segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known as superpixels). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics.

The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s). When applied to a stack of images, typical in Medical imaging, the resulting contours after image segmentation can be used to create 3D reconstructions with the help of interpolation algorithms like marching cubes. Image quality is a characteristic of an image that measures the perceived image degradation (typically, compared to an ideal or perfect image). Imaging systems may introduce some amounts of distortion or artifacts in the signal, so the quality assessment is an important problem.

3.3.4 BINARIZATION

A binary image is a digital image that has only two possible values for each pixel. Typically the two colors used for a binary image are black and white though any two colors can be used. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color. In the document scanning industry this is often referred to as bi-tonal.

Binary images are also called bi-level or two-level. This means that each pixel is stored as a single bit (0 or 1). The names black-and-white, B&W, monochrome or monochromatic are often used for this concept, but may also designate any images that have only one sample per pixel, such as grayscale images. In Photoshop parlance, a binary image is the same as an image in "Bitmap" mode.

A binary image is usually stored in memory as a bitmap, a packed array of bits. A 640×480 image requires 37.5 KB of storage. Because of the small size of the image files, fax machines and document management solutions usually use this format.



Binary images can be interpreted as subsets of the two-dimensional integer lattice Z^2 ; the field of morphological image processing was largely inspired by this view.

3.4 RIDGE FEATURE EXTRACTION

After performing the preprocessing steps, obtain the skeletonized ridges and minutiae information from the fingerprint image.

3.4.1 ORIENTATION

Orientation fields can be used to describe interleaved ridge and valley patterns of fingerprint image, providing features useful for fingerprint recognition. In computer vision and image processing a common assumption is that sufficiently small image regions can be characterized as locally one-dimensional, e.g., in terms of lines or edges.

For natural images this assumption is usually correct except at specific points, e.g., corners or line junctions or crossings, or in regions of high frequency textures. However, what size the regions have to be in order to appear as one-dimensional varies both between images and within an image. Also, in practice a local region is never exactly one-dimensional but can be so to a sufficient degree of approximation.

Image regions which are one-dimensional are also referred to as simple or intrinsic one-dimensional (i1D).

Given an image of dimension d ($d = 2$ for ordinary images), a mathematical representation of a local i1D image region is

$$f(x) = g(x \cdot \bar{n})$$

where f is the image intensity function which varies over a local image coordinate \mathbf{x} (a d -dimensional vector), g is a one-variable function, and \bar{n} is a unit vector.

The intensity function f is constant in all directions which are perpendicular to \bar{n} . Intuitively, the orientation of an i1D-region is therefore represented by the vector \bar{n} . However, for a given f , \bar{n} is not uniquely determined. If

$$\bar{n} = -\bar{n}$$

$$g(x) = g(-x)$$

then f can be written as

$$f(x) = g(x \cdot \bar{n})$$

which implies that $\bar{n} = -\bar{n}$ also is a valid representation of the local orientation.

In order to avoid this ambiguity in the representation of local orientation two representations have been proposed

- The double angle representation
- The tensor representation

The double angle representation is only valid for 2D images ($d=2$), but the tensor representation can be defined for arbitrary dimensions d of the image data.

3.4.2 GABOR FILTER

In image processing, a Gabor filter, named after Dennis Gabor, is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave. The Gabor filters are self-similar: all filters can be generated from one mother wavelet by dilation and rotation.

Its impulse response is defined by a harmonic function multiplied by a Gaussian function. Because of the multiplication-convolution property (Convolution theorem), the Fourier transform of a Gabor filter's impulse response is the convolution of the Fourier transform of the harmonic function and the Fourier transform of the Gaussian function. The filter has a real and an imaginary component representing orthogonal directions. The two components may be formed into a complex number or used individually.

Complex

$$g(x,y;\lambda,\theta,\sigma,\gamma) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \exp\left(i\left(2\pi\frac{x}{\lambda} + \psi\right)\right)$$

Real

$$g(x,y;\lambda,\theta,\sigma,\gamma) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \cos\left(2\pi\frac{x}{\lambda} + \psi\right)$$

Imaginary

$$g(x,y;\lambda,\theta,\sigma,\gamma) = \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \sin\left(2\pi\frac{x}{\lambda} + \psi\right)$$

where

$$x = x \cos \theta + y \sin \theta$$

and

$$y = -x \sin \theta + y \cos \theta$$

In this equation, λ represents the wavelength of the sinusoidal factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, x is the phase offset, σ is the sigma of the Gaussian envelope and γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function



4.0 FINGERPRINT MATCHING

The ridge feature vectors between the minutiae in the ridge coordinate system can be expressed as a directional graph whose nodes are minutiae and whose edges are ridge feature vectors.

4.4.1 THRESHOLDING

Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images

A. METHOD

During the thresholding process, individual pixels in an image are marked as "object" pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as "background" pixels otherwise. This convention is known as threshold above. Variants include threshold below, which is opposite of threshold above; threshold inside, where a pixel is labeled "object" if its value is between two thresholds; and threshold outside, which is the opposite of threshold inside. Typically, an object pixel is given a value of "1" while a background pixel is given a value of "0." Finally, a binary image is created by coloring each pixel white or black, depending on a pixel's labels.

1) THRESHOLD SELECTION

The key parameter in the thresholding process is the choice of the threshold value (or values, as mentioned earlier). Several different methods for choosing a threshold exist; users can manually choose a threshold value, or a thresholding algorithm can compute a value automatically, which is known as automatic thresholding. A simple method would be to choose the mean or median value, the rationale being that if the object pixels are brighter than the background, they should also be brighter than the average. In a noiseless image with uniform background and object values, the mean or median will work well as the threshold, however, this will generally not be the case. A more sophisticated approach might be to create a histogram of the image pixel intensities and use the valley point as the threshold.

The histogram approach assumes that there is some average values for both the background and object pixels, but that the actual pixel values have some variation around these average values. However, this may be computationally expensive, and image histograms may not have clearly defined valley points, often making the selection of an accurate threshold difficult

In such cases a unimodal threshold selection algorithm may be more appropriate. One method that is relatively simple, does not require much specific knowledge of the image, and is robust against image noise, is the following iterative method

1. An initial threshold (T) is chosen, this can be done randomly or according to any other method desired.

2. The image is segmented into object and background pixels as described above, creating two sets:

1. $G_1 = \{f(m,n): f(m,n) > T\}$ (object pixels)
2. $G_2 = \{f(m,n): f(m,n) < T\}$ (background pixels) (note, $f(m,n)$ is the value of the pixel located in the m^{th} column, n^{th} row)

3. The average of each set is computed.

1. $m_1 =$ average value of G_1
2. $m_2 =$ average value of G_2

4. A new threshold is created that is the average of m_1 and m_2

1. $T' = (m_1 + m_2)/2$

5. Go back to step two, now using the new threshold computed in step four, keep repeating until the new threshold matches the one before it (i.e. until convergence has been reached).

This iterative algorithm is a special one-dimensional case of the k-means clustering algorithm, which has been proven to converge at a local minimum meaning that a different initial threshold may give a different final result.

2) ADAPTIVE THRESHOLDING

Thresholding is called adaptive thresholding when a different threshold is used for different regions in the image. This may also be known as local or dynamic thresholding.

4.5 DYNAMIC PROGRAMMING

It is applied to find the optimal solution in matching two string sequences in the enrolled and input ridge-based coordinates.

4.5.1 SKELETONIZATION

Skeletonization is a process for reducing foreground regions in a binary image to a skeletal remnant that largely preserves the extent and connectivity of the original region while throwing away most of the original foreground pixels.

In shape analysis, skeleton (or topological skeleton) of a shape is a thin version of that shape that is equidistant to its boundaries. The skeleton usually emphasizes geometrical and topological properties of the shape, such as its connectivity, topology, length, direction, and width. Together with the distance of its points to the shape boundary, the skeleton can also serve as a representation of the shape (they contain all the information necessary to reconstruct the shape).



4.5.2 BREADTH FIRST SEARCH

BFS is an uninformed search method that aims to expand and examine all nodes of a graph or combination of sequences by systematically searching through every solution. In other words, it exhaustively searches the entire graph or sequence without considering the goal until it finds it. It does not use a heuristic algorithm

From the standpoint of the algorithm, all child nodes obtained by expanding a node are added to a FIFO (i.e., First In, First Out) queue. In typical implementations, nodes that have not yet been examined for their neighbors are placed in some container (such as a queue or linked list) called "open" and then once examined are placed in the container "closed".

5.0 ALGORITHMS USED

ALGORITHM 1: RIDGE FEATURE EXTRACTION

Here, Ridge features are extracted for the fingerprints.

Steps:

- 1) Perform preprocessing steps and extract a ridge image from a fingerprint.
- 2) Traverse the ridge-valley structures along the vertical axis from each minutia origin.
 - a) If the vertical axis intersects with the ridges attached to a minutia, extract ridge features (ridge count, ridge length, ridge curvature direction, and ridge type) from the origin to the minutia and form a ridge feature vector between the origin and the minutiae.
 - b) Keep traversing all the ridges until one of three terminating conditions is satisfied (see below).
- 3) If all minutiae are used as the origin minutiae, terminate the procedure. Otherwise, return to step 2).

The termination conditions include the following three cases:

- 1) The vertical axis reaches a background region in the fingerprint image.
- 2) The vertical axis reaches a poor quality region in the fingerprint image.
- 3) The vertical axis reaches a high circular variance region in the fingerprint image.

ALGORITHM 2: FINGERPRINT MATCHING ALGORITHM

Here, the stored fingerprints are matched with the enrolled fingerprint.

Steps:

- 1) Initially match any pair of ridge-based coordinate systems extracted from the enrolled fingerprint image and the input fingerprint image using dynamic programming.
- 2) Select the top N degree of matched ridge-based coordinate pairs.

3) For every initially matched pair, a breadth-first search (BFS) is performed to detect the matched ridge-based coordinate pairs incrementally.

4) Check the validity of the matched coordinate pairs using the relative position and orientation of the minutiae and count the number of matched minutiae.

5) Iterate steps 3) and 4) N times and then return the maximum number of matched minutiae.

6) Compute the matching score.

CONCLUSION & FUTURE ENHANCEMENT

6.0 CONCLUSION

This project implements a novel fingerprint matching algorithm using both ridge features and the minutiae. The ridge features consist of four elements (ridge count, ridge length, ridge curvature direction, and ridge type) that describe the relationship between the minutiae. With the ridge features and conventional minutiae features (minutiae type, orientation, and position), a novel matching scheme is implemented which uses BFS to detect the matched minutiae pairs. Hence that the ridge features give additional information for fingerprint matching against non-linear deformations in fingerprints with little increment of template size.

7.0.2 FUTURE ENHANCEMENT

Future research will try to incorporate ridge features into the state-of-the-art minutiae-based matchers for further improvement of the matching performance. Also, this matching method needs to be improved for images with a small foreground area and those of low quality. Therefore, in future work will develop the use of global knowledge of fingerprints, such as singular point position, to enhance the matching accuracy and also develop a robust preprocessing method to reduce enhancement errors. Moreover, these ridge features can be used in other applications. In the area of fingerprint identification, it is important to be able to extract alignment-free features since it needs no time to align a query feature set with the N enrolled feature sets one by one.

The ridge features are invariant to any transform, thus they can be used in addition to conventional alignment-free features in the fingerprint identification or cancellable fingerprint area where cancellable fingerprints are without a fiducial corresponding pair such as a core point, it is difficult to align a transformed feature set with an enrolled one.

REFERENCES

- [1] Choi.H, Choi.K and Kim.J (JUN 2011) 'Fingerprint Matching Incorporating Ridge Features with inutiae,' IEEE Trans. Inf. Forensics Security, vol.6, no.2
- [2] Bazen.A.M and Gerez.S.H,(Aug. 2003) 'Fingerprint matching by thin-plate spline modelling of elastic



- deformations,' Pattern Recognit., vol. 36, no. 8, pp. 1859–1867.
- [3] Cappelli.R, Maio.D, and Maltoni.D, "Modeling plastic distortion in fingerprint images," in Proc. ICAPR, 2001, pp. 369–376.
- [4] Chen.X, Tian.J, Yang.X, and Zhang.Y,(Jun. 2006) 'An algorithm for distorted fingerprint matching based on local triangle feature set,' IEEE Trans. Inf. Forensics Security, vol. 1, no. 2, pp. 169–177.
- [5] Dorai.C, Ratha.N, and Bolle.R,(Jun. 2000) 'Detecting dynamic behavior in compressed fingerprint videos: Distortion,' in Proc. CVPR, Hilton Head, SC, pp. 2320–2326.
- [6] Jiang.X and Yau.W.Y,(Sep. 2000) 'Fingerprint minutiae matching based on the local and global structures,' in Proc. 15th Int. Conf. Pattern Recognition, Barcelona, Spain, vol. 2, pp. 1038–1041.
- [7] Lee.D, Choi.K, and Kim.J,(Aug. 2002) 'Arobust fingerprint matching algorithm using local alignment,' in Proc. 16th Int. Conf. Pattern Recognition, Quebec City, Que., Canada, vol. 3, pp. 803–806.
- [8] Luo.X.P, Tian.J, and Wu.Y(Sep. 2000), 'Aminutia matching algorithm in *fingerprint verification*,' in Proc. 15th ICPR, vol. 4, pp. 833–836.
- [9] Maltoni.D, Maio.D, Jain.A.K, and Prabhakar.S,(2003). 'Handbook of Fingerprint Recognition.' New York: Springer-Verlag.
- [10] Ratha.N.K and R. M. Bolle.R.M, "Effect of controlled acquisition on fingerprint matching," in Proc. 14th ICPR, 1998, vol. 2, pp. 1659–1661.



A.SURESH is the author of this paper he was born in the year of 17, July 1980 in Coimbatore. The author completed B.E electrical and electronics engineering degree programme at Government College of Technology, Coimbatore, after that he joined M.E computer science and engineering at Sri

Krishna college of engineering and technology. The author have good experience in trouble shooting computer, he has worked as system administrator at GRG School of applied technology Coimbatore. After that he joined as Assistant Professor/IT at Sri Ramakrishna institute of technology, Coimbatore.