A COMPARATIVE STUDY OF FINGERPRINT MATCHING TECHNIQUES

Chapter 2 - Literature Review

ID	FULL NAME
10736694	Terence Ugo Nacciarone Quashie
10729461	Abdul-Aziz Abubakar Saddick

Contents

Table of Figures

Fingerprint Matching Algorithms

A fingerprint matching algorithm is one that revolves around comparing previously stored templates of fingerprints against candidate fingerprints usually for the purpose of authentication (Wang, Gavrilova, Luo, & Rokne, 2006). The algorithm allows for a person to be identified or verified

Two types of algorithms are prevalent; these are:

1. Scale-Invariant Feature Transformation, and
2. Minutiae-based Fingerprint Recognition and Matching

Scale-Invariant Feature Transformation

Scale-Invariant Feature Transformation **(SIFT)** is a computer vision algorithm to describe, detect and match local features in digital images. It locates certain keypoints and then furnishes them with **quantitative information** or **descriptors** that can be used for object recognition. (Wechsler & Li, 2014) The Scale-Invariant Feature Transform (SIFT) algorithm consists of two successive and independent operations. These are

i. The detection of interesting points (keypoints)
ii. The extraction of a descriptor associated to each of the keypoints.

SIFT is useful for fingerprint matching because it produces unique descriptors on each fingerprint making it a convenient technique to match two fingerprint images. Due to the nature of these descriptors, comparison between samples can be undertaken (regardless of how distorted the image being used for comparison is) because a match is still generated.

Descriptors

Descriptors are quantitative information about a keypoint. These descriptors are invariant against various transformations such as image translation, rotation and scaling which might make images look different although they represent the same object. (Edmund, 2016)

SIFT descriptors have also proved to be robust to a wide family of image transformations including viewport changes, noise, blur, scene deformation and contrast changes while remaining discriminative enough for matching purposes. Since these descriptors are robust, they are usually used for matching pairs of images. (Otero, Delbracio, & Mauricio, 2014).

Keypoints

Keypoints are interesting points whose center position and characteristic scale are accurately located. For each keypoint, the size, center and orientation are normalized. Due to this normalization the keypoints remain invariant to any translation, scale change or rotation.

SIFT detects a series of keypoints from multiscale image representation. This multiscale representation consists of a set of increasingly blurred images. From these keypoints, descriptors are generated.

Video stabilization is another popular application of the SIFT method, however, the scope of this research will be limited to image recognition, **more specifically finger print image recognition**.

Procedures involved in obtaining keypoints and descriptors

To obtain these keypoints and descriptors, various steps and frameworks are constructed and used.

The Gaussian Scale-Space Construction

Keypoints in SIFT are invariant to scale-changes. In order to attain this scale invariance, SIFT is built on a Gaussian Scale-Space.

A Gaussian Scale-Space is a multiscale image representation simulating the family of all possible zoom-outs through increasingly blurred versions of the input image (Otero, Delbracio, & Mauricio, 2014)This blurring process simulates the loss of detail produced when a scene is photographed from farther and farther. The scale-space, therefore provides SIFT with scale invariance as it can be interpreted as the simulation of a set of snapshots of a given scene taken at different distances. This scale space is constructed by applying a variable gaussian operator on an input image. This is also known as **Gaussian Blurring**.

Gaussian blurring is a technique used by the SIFT algorithm to properly prepare images for scale space construction. This technique is a widely used effectively in graphics software, typically to reduce image noise or image graininess. However, in computer-vision-based algorithms, it is used as a preprocessing technique in order to enhance images at different scales (Haddad & Akansu, 1991)

In other areas of computer vision, the Gaussian blur is also used as a way to detect edges. Since most edge detection algorithms are sensitive to noise, the gaussian blur serves as a way to reduce the noise in order to make edge detection more accurate.

The Gaussian blur technique will be used to prepare fingerprint images for the detection of where the fingerprint arc, whorl or loop begins.

Difference of Gaussians

Another refinement technique used is the Difference of Gaussians **(DoG)**. This refers to a feature enhancement technique that involves the subtraction of one Gaussian Blurred version of an original image from another less blurred version of the original (Davidson, 2016).This results in a set of Gaussian-Smoothed images known as **octaves**

The Difference of Gaussians technique ensures that the spatial information that lie between the range of frequencies are preserved between the two blurred images, these include visibility of edges and any other keypoints present in the digital image.

3D extrema

In SIFT, ***candidate keypoints*** are defined as the **3D extrema** of the normalized scale-space. The extrema are detected by observing each image point in the **Difference of Gaussian (DoG)**. A point is decided as a local minimum or maximum when its value is smaller or larger than all its surrounding neighbor points by a certain amount. If an extremum is decided as unstable or is placed on an edge. It is removed because it can not be reliably detected again with small variation of the viewport or lighting changes.

Calculation of 3D extrema

Continuous 3D extrema of the digital DoG are calculated in two steps. Firstly, the 3D discrete extrema are extracted from each octave with pixel precision. Next, their location is refined through interpolation of the digital DoG by using a quadratic model such as the **Taylor expansion**.

The resulting image is then compared to its neighbors to detect the 3D discrete maxima and minima. These comparisons are possible due to the auxiliary images in the DoG.

Although this process produces **candidate keypoints** that can be used, it is prone to noise and as such produces unstable detections. The Candidate keypoints chosen may therefore be flawed since they are constrained to the sampling grid.

Filtering Unstable Candidate Keypoints

Noisy images produce erroneous candidate keypoints thereby making them unstable and unlinked to any particular structure in the image.

SIFT attempts to eliminate these false detections by discarding those candidate keypoints found outside the DoG threshold on distance ratio defined as *(distance to nearest candidate keypoint neighbor / distance to second nearest candidate keypoint neighbor)***.**

Unstable key-points are those on the edges of the image. These candidate keypoints are difficult to precisely locate due to the fact that an edge is invariant to translations along its principal axis. Such detections do not help define covariant keypoints and are also discarded.

Orientation Assignment

After a set of stable candidate keypoints have been generated, an orientation is assigned to each of these keypoints to make them invariant to rotation (Singh, 2019). This is done by computing the magnitude and orientation for each keypoint and then constructing a histogram to determine the peak orientation for a particular keypoint.

An example is shown below in ***Figure 1***

35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

Figure 1 depicting pixel matrix

Consider the matrix above which is a matrix of pixels. To compute the orientation and magnitude for the pixel in red, the gradients in both the x and y directions are calculated as follows

$$Gx\ (gradient\ of\ x) is\ obtained\ by\ subtracting\ 46\ from\ 55\ giving\ us\ Gx = 9$$
$$Gy\ (gradient\ of\ y) is\ obtained\ by\ subtracting\ 42\ from\ 56\ giving\ us\ Gy = 14$$

$$Magnitude = \sqrt{(G_x)^2 + \left(G_y\right)^2} = 16.64$$
$$\Phi = atan(Gy\ /\ Gx) = atan(1.55) = 57.17$$

The magnitude represents the intensity of the pixel while the orientation represents the direction of the pixel. From this a histogram is created by plotting the magnitude and orientation value for all the pixels to obtain the peak orientation. An example is shown below in ***Figure 2****;*

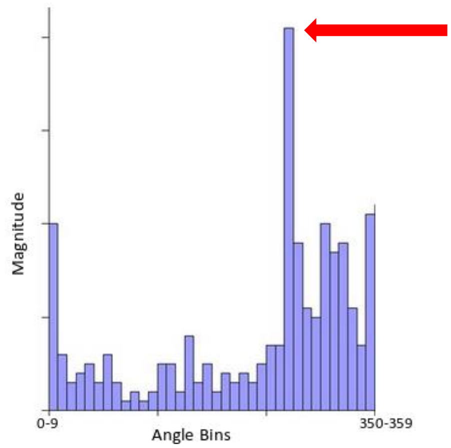

Figure 2 depicting histogram for orientation assignment

At some point the histogram peaks (arrowed in figure 2) and from this the orientation of the keypoint is determined making it invariant to rotations.

Keypoint Descriptor

From the unique orientation-invariant keypoints, ***Descriptors*** are obtained. Descriptors fall into two categories:

i. Descriptors based on properties of the image that are already rotation-invariant
ii. Descriptors based on a normalization with respect to the reference orientation

For the scope of this project, we will use descriptors based on ***category (i)*** (Singh, 2019).

Matching of Descriptors

Descriptors between two images are matched by identifying their nearest neighbors. In the event that the descriptors are too close to each other due to image noise, the ratio of the closest distance to second closest distance is taken.

The standard ratio for this distance is 0.8 and if they are greater than this, the points are rejected. This ensures that 90% of false matches are eliminated while only discarding 5% of correct matches. (Tyagi, 2019)

Using descriptors for matching purposes work due to the following reasons:

- Keypoints are extracted at different scales and blur levels and all subsequent computations are performed within the scale space framework. This makes the descriptors invariant to image scaling and small changes in perspective.
- Computation relative to a reference orientation makes the descriptors robust against rotation.
- The descriptor information is stored relative to the keypoint position and thus invariant against translations.
- Many potential keypoints are discarded if they are deemed unstable or hard to locate precisely. The remaining keypoints are thus relatively immune to image noise.
- The histograms are normalized meaning the descriptors do not store the magnitudes of the gradients, only their relationships to each other. This makes the descriptors invariant against global, uniform illumination changes.

- The histogram values are 'thresholded' to reduce the influence of large gradients. This makes the information partly immune to local, non-uniform changes in illumination. (Edmund, 2016)

Use of SIFT on Fingerprint Images

Since SIFT keypoints are limited only by the condition of the local minima or maxima in a given scale space, a large number of keypoints on a fingerprint image are detected. These keypoints are determined by a set of parameters including the number of octaves. Typical fingerprints contain up to a thousand keypoints (Park, Pankanti, & Jain, 2008).

Use of SIFT on fingerprint images can be summarized into three (3) steps summarized in the flow chart below in ***Figure 3***

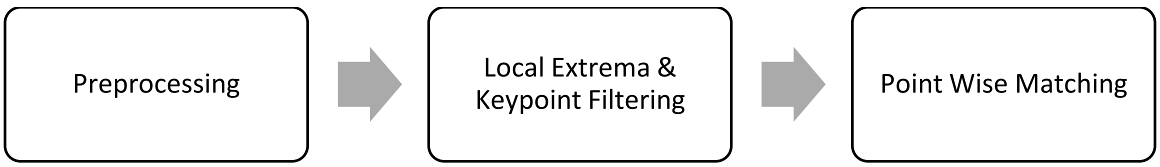

Figure 3

Preprocessing (Obtaining scale space)

The first phase, which involves obtaining the scale space of the fingerprint image, is done by blurring the images using a **Gaussian blurring** method to simulate the different zoom levels of the image indicated by the orange arrow ( ) in figure 4. Further blurring is done as indicated on its left neighbor or by the green arrow ( ) as shown in ***Figure 4.***

To further make the scale space robust, the images can be scaled-down to better simulate different zoom levels as indicated by the blue arrow ( )***. Each*** row is known as an **octave**. Image representation of generating scale space shown in ***Figure 4*** below.

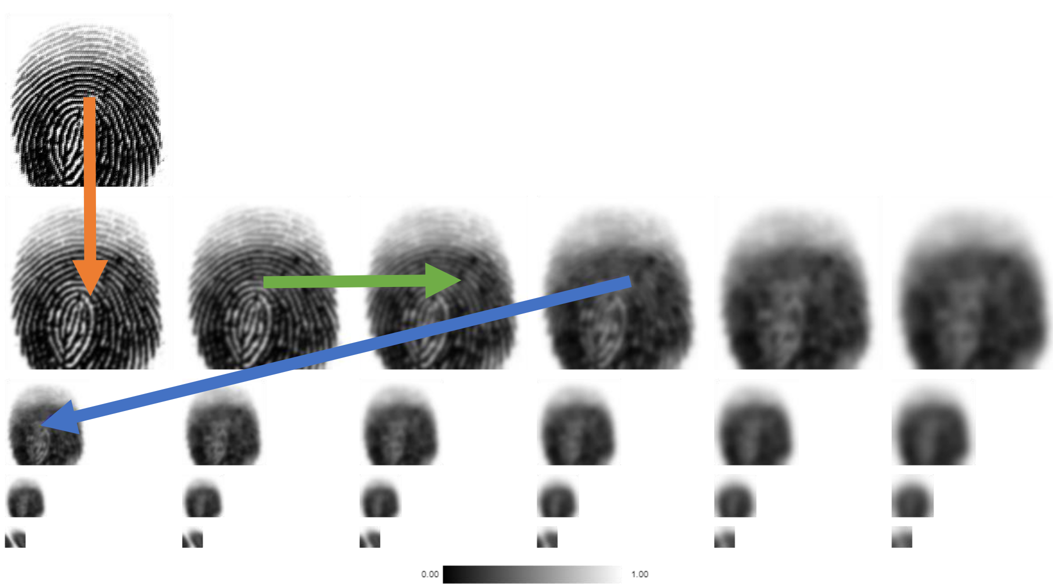

Figure 4 Scale space of fingerprints

Local Extrema & Descriptor Extraction

To find the local Extrema in this phase, we first compute the **Difference of Gaussians** (DoG). This can be done by gray-scaling each image in the octave as shown in Figure 5 and calculating the difference between each pixel in the adjacent image as shown in ***Figure 6***Figure 5

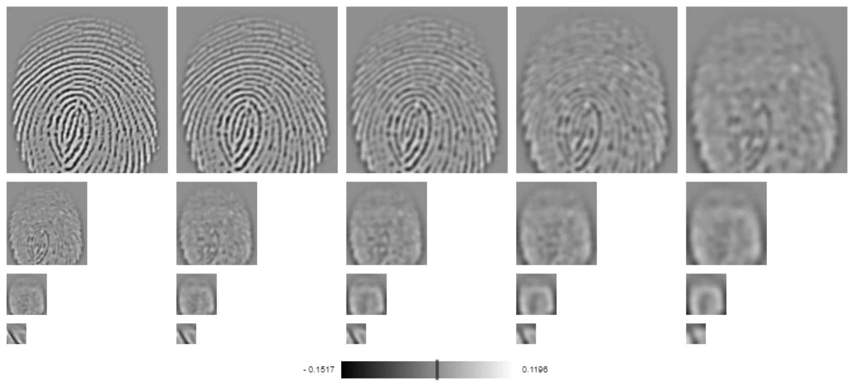

Figure 5 Grayscaling images

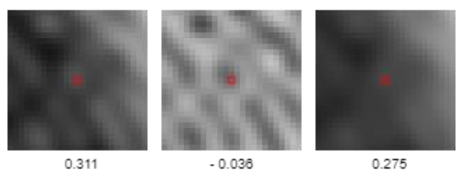

Figure 6 Calculated differences in pixel

The extremum/candidate keypoint then becomes a pixel whose gray value is larger than all of its neighboring pixels as shown below in ***Figure 7***.

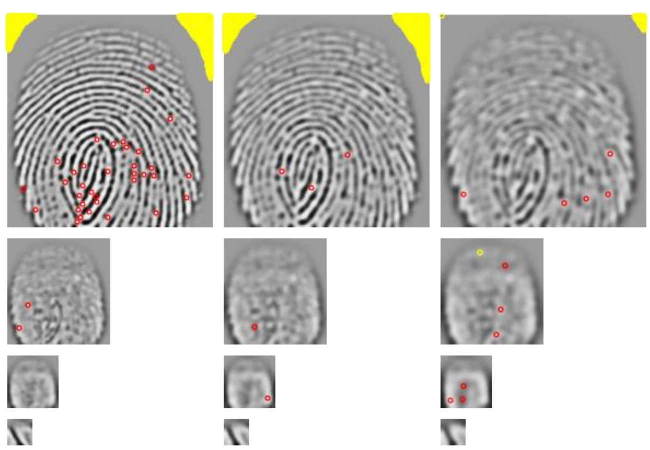

Figure 7 Candidate keypoints/extrema

Area marked in yellow ( ) are indeed extrema/candidate keypoints, but their absolute values are so low that they are discarded

These extrema usually exist as a result of image noise. The extrema are then filtered using the quadratic **Taylor expansion** of scale space-function. This is an iterative process that refines the location of a keypoint.

Following the first filtering, keypoints that lie on the edges are then identified and discarded. These points are not invariant to translations parallel to edge direction hence they are discarded.

In the example being used here, the remaining keypoints are shown in ***Figure 8*** below

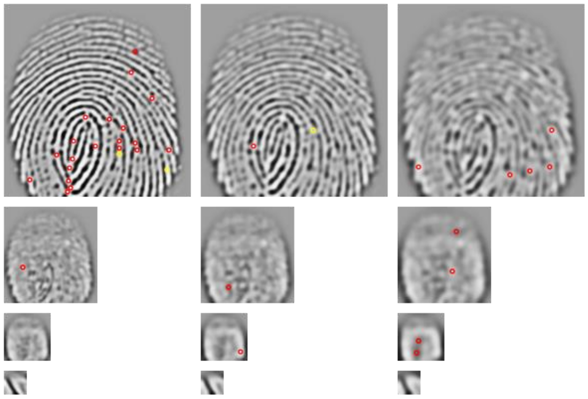

Figure 8 rRefined keypoints

A reference orientation is now calculated making the keypoints invariant to rotation as described earlier in ***Orientation Assignment*** resulting in descriptors.

Descriptors that do not have enough pixels to compute a reference orientation are discarded. Descriptors without a dominating orientation are also discarded resulting in the image as shown in ***Figure 9***

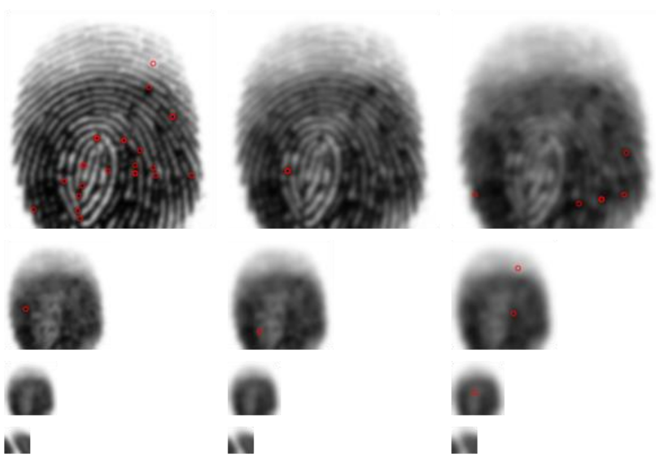

Figure 9 New keypoint after orientation assignment

Point Wise Matching

This final set of descriptors are then compared to the descriptors of another image even if they are depicted with different illumination, slightly distorted or with a different perspective. This comparison is effective/valid as discussed in the section under ***Matching of Descriptors*** above.

Figure 10 illustrates this comparison between a fingerprint and its distorted version.

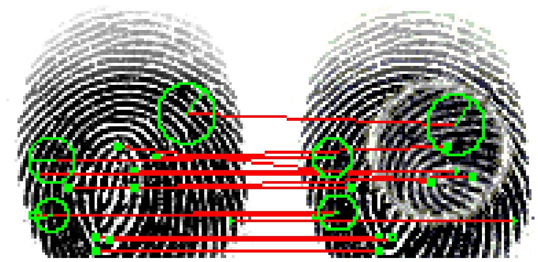

Figure 10 Matching of two descriptors

Minutiae-based Fingerprint Extraction and Recognition

The minutiae-based fingerprint extraction and recognition algorithm is a **set of techniques** used to extract feature points or descriptors known as **"minutiae"**. Minutiae are small plot points on a fingerprint. They include characteristics such as **ridge bifurcation, the ridge ending** and **the orientation**. A ridge bifurcation is defined as the point where a ridge diverges into branch ridges, whereas a ridge ending is the point where a ridge ends abruptly. The orientation on the other hand is defined as the local ridge orientation of the associated ridge (Zaeri, 2011)

The number of minutiae points on each fingerprint sample differ from one fingerprint image to another. Depending on the resolution of the fingerprint image, one fingerprint can generate up to 100 minutiae features (Zaeri, 2011)

These extracted minutiae features are then compared to the feature points of other fingerprint template images for fingerprint recognition and matching. The minimum number of matching pairs of minutiae required to declare that two template fingerprints match is at least 12 minutiae features (Amina, Dominique, & Youcef, 2022). Procedures involved in extracting minutiae features

To obtain these minutiae points, a series of frameworks are constructed and used. These are explained below.

Normalization

In an ideal fingerprint image, lines of ridges flow in a constant direction. However, due to certain conditions such as, cuts and wetness of the skin, incorrect finger pressure when taking a reading or image noise generated as a result of the sensor, poor quality images are generated which may lead to the generation of multiple false minutiae hence the need for normalization. (Cao & Wang, 2016)

Normalization, also referred to as histogram stretching or contrast stretching, is a process which involves changing the range of pixel intensity values (Fisher, Perkins, Walker, & Wolfart, 2003). The end result is a high contrast image with much clearer details**.** In the case of fingerprint images, it makes finer details such as ridges much more "refined".

In minutiae extraction, one of the most popular choices of normalization is the Gabor Filter. **The Gabor filter** is a linear filter used as a normalization technique. It also functions as a tool for feature extraction, edge detection and texture analysis. The Gabor filter amplifies a band of frequencies and rejects the others (Shah, 2018). By rejecting some of these color frequencies, the variation between gray values along the ridges are also reduced thereby making it easier to have more contrast in the image.

Binarization/Segmentation

Image binarization is the process of taking a normalized grayscale image and converting all its pixels it black and white, essentially reducing the color information from 255 shades of gray to two colors, black and white with values of 0 and 255 respectively.

Binarization is done to preserve the characteristics of the ridge structure while removing some of the cohesion between patterns. (Erwin, Karo, Sari, Aziza, & Putra, 2019). The next step after normalization is usually to binarize that image.

The process of binarization works by finding a threshold value in the color histogram. The threshold value is a value that divides the histogram into two parts. Each representing one of two layers; the background layer and the object itself, and case, the pattern contained in the fingerprint.

The threshold value cannot be accurately calculated but a good prediction can be made by dividing the image into a $w \times w$ block and its mean and variance, calculated for each block.

Next, the average the mean and variance of all the blocks are calculated. From this average, the relative mean and variance are also calculated. The threshold value for foreground, in this case the ridges of the fingerprint, will be represented as the average of the variance and the background, the average mean. (Al-Najjar & Sheta, 2008)

Fast Fourier Transform

The Fast Fourier Transform is an image enhancement technique used to transform an image between the spatial and frequency domain**.** Following the binarization process, some of the ridges may be broken as a result of having their pixel values greater than the threshold value and as such assigned values of 255 or white. The Fast Fourier Transform (FFT) is then used to reconnect these broken ridges.

Any image represented in a frequency domain has two major components.

- High Frequency components which correspond to the edges in the image
- Low Frequency components which correspond to the smooth regions of the image

Fast Fourier transformation decomposes the source image into its spectral information. For instance, its **directional information** which includes respective sines and cosines which reveal a repeating pattern within the image. This orientation data allows the reconnection of broken ridges.

By transforming the source image into the frequency domain, the FFT preserves all original data. (Amina, Dominique, & Youcef, 2022) in addition to further removing noise that may exist in the image

Dominant frequencies refer to a sinusoid which consists of the following:

- Spatial Frequency – Deals with brightness of the image
- Magnitude – Deals with contrast
- Phase – Refers to color information

Thinning

Thinning is an image processing technique that involves reducing the thickness of each line of ridge pattern till the width is shrunk enough to become a single or one (1) pixel. This process is done to identify the exact pattern of the fingerprint which in turn makes feature/minutiae extraction possible. (Al-Ani, 2013)

For an image to be considered properly thinned, each ridge should be thinned to its central pixel. After an image is thinned, further removal of pixels is not possible.

Minutiae Extraction

In this stage, all minutiae feature points, that is, ridge ending and bifurcations have been refined and are ready for extraction. Appropriate minutiae feature points are extracted by applying a filter of a 3×3 matrices over the region of interest (ROI) in the thinned image by following some rules:

i. If the central pixel is '1' and the sum of pixels inside the block is '2', then that central pixel is a ridge termination

ii. If the central pixel is ‘1’ and the sum is ‘4’ then the central pixel is a bifurcation

iii. If the central pixel is some other value than ‘1’, then the central pixel is a regular pixel with no unique point (Al-Najjar & Sheta, 2008)

Region of interest (ROI)
Region of interest (ROI) is the area of the thinned image of interest. The region of interest is found by applying some morphological operations such as opening area, filling, erosion or closing. Specifying the ROI allows for minutiae suppression outside the region. (Al-Najjar & Sheta, 2008)

Minutiae Orientation
Once the minutiae points have been determined, the orientation for both the ridge termination and bifurcation are calculated.

Termination Orientation: This is found by using a table of 3×3 dimensions for different angles of theta. The positions of pixels connected to the center in a 3×3 ROI block. By keeping the edge pixels, the first non-zero pixels are taken and its location compared to the table to find the corresponding angle for that termination.

Bifurcation Orientation: For each bifurcation, there are three bounding non-zero pixels hence the procedure for termination orientation calculation is applied three times instead of once. (Al-Najjar & Sheta, 2008)

Terminating False Minutiae
False minutiae consist of broken ridges, minutiae adjacent to each other and minutiae near the borders. Algorithms such as the Fast Fourier Transform aim to reduce the generation of these points, however, as they are not perfect, some false minutiae will still be generated. (Maddala & Tangellapally, 2010). False minutiae adversely affect matching and as such they need to be trimmed down or removed.

Minutiae Matching
Once the false minutiae have been trimmed. A set of the pixel coordinates and orientation angles are generated from the source image. This data is then compared to the data set of the target fingerprint to determine if the source and the target fingerprint images match. (Al-Najjar & Sheta, 2008)

Using Minutiae Extraction Method on Fingerprint Images
The following set of images depict the various phases of the minutiae extraction method

Figure 111
fingerprint sample

Figure 2
Normalized Image

Figure 3
Binarized Image

Figure 4
thinned Image

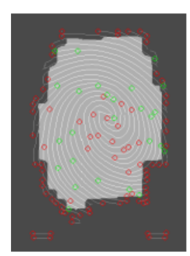

Figure 5
Region of Interest

Figure 6
Extracted Minutiae

References

Al-Ani, M. S. (2013, February 2). A Novel Thinning Algorithm for Fingerprint Recognition. *International Journal of Engineering Sciences*, 43-48.

Al-Najjar, Y., & Sheta, A. (2008). MINUTIAE EXTRACTION FOR FINGERPRINT RECOGNITION. *2008 5th International Multi-Conference on Systems, Signals and Devices* (pp. 1-5). As-Salt: INPROCEEDINGS.

Amina, Dominique, M., & Youcef, Z. (2022). Fast Fourier transform for fingerprint enhancement and features extraction with adaptive block size using orientation fields. *ITM Web of Conferences*, 10.

Cao, L., & Wang, Y. (2016, May 31). Fingerprint Image Enhancement and Minutiae Extraction Algorithm. Linnaeus, Li, Sweden.

Davidson, M. W. (2016, May 17). *Difference of Gaussians Edge Enhancement.* Retrieved from Micro.Magnet: https://micro.magnet.fsu.edu/primer/java/digitalimaging/processing/diffgaussians/index.html

Edmund. (2016). *SIFT - Scale-Invariant Feature Transform.* Retrieved from Weitz: http://weitz.de/sift/

Erwin, Karo, N. N., Sari, A. Y., Aziza, N., & Putra, H. K. (2019). The Enhancement of Fingerprint Images using. *Journal of Physics: Conference Series*.

Fisher, R., Perkins, S., Walker, A., & Wolfart, E. (2003). *Contrast Stretching.* Retrieved from HIPR2: https://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm

Haddad, R. A., & Akansu, A. N. (1991). A Class of Fast Gaussian Binomial Filters for Speech and Image Processing. *Transactions on Signal Processing*, 723-727.

Maddala, S., & Tangellapally, S. R. (2010, September). Implementation and Evaluation of NIST Biometric Image Software For Fingerprint Recognition. Blekinge, Sweden.

Otero, R., Delbracio, I., & Mauricio. (2014). Anatomy of the SIFT Method. *Image Processing On Line*, 370--396.

Park, U., Pankanti, S., & Jain, A. (2008). Fingerprint Verification Using SIFT Features . *SPIE Defense and Security Symposium*.

Shah, A. (2018, June 17). *Through The Eyes of Gabor Filter.* Retrieved from Medium: https://medium.com/@anuj_shah/through-the-eyes-of-gabor-filter-17d1fdb3ac97

Singh, A. (2019, October 9). *A Detailed Guide to the Powerful SIFT Technique for Image Matching.* Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/#:~:text=SIFT%20helps%20locate%20the%20local,detection%2C%20scene%20detection%2C%20etc.

Tyagi, D. (2019, March 16). *Introduction to SIFT(Scale Invariant Feature Transform).* Retrieved from Medium: https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40

Wang, C., Gavrilova, M., Luo, Y., & Rokne, J. (2006, January). *An efficient algorithm for fingerprint matching.* Retrieved from ResearchGate: https://www.researchgate.net/publication/220927699_An_efficient_algorithm_for_fingerprint_matching/citations

Wechsler, H., & Li, F. (2014). Chapter 10 - Biometrics and Robust Face Recognition. In M. Kaufmann, *Conformal Prediction for Reliable Machine Learning* (pp. 189-215). 189-215: Kaufmann, Morgan.

Zaeri, N. (2011). *Minutiae-based Fingerprint Extraction and Recognition.* Rijeka: IntechOpen. doi:10.5772/17527