applied sciences

MDPI

Article

Robust Fingerprint Minutiae Extraction and Matching Based on Improved SIFT Features

Samy Bakheet [1], Shtwai Alsubai [2,*], Abdullah Alqahtani [2] and Adel Binbusayyis [2]

[1] Faculty of Computers and Artificial Intelligence, Sohag University, Sohag 82524, Egypt; samy.bakheet@fci.sohag.edu.eg
[2] College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al Kharj 11942, Saudi Arabia; aq.alqahtani@psau.edu.sa (A.A.); a.binbusayyis@psau.edu.sa (A.B.)
* Correspondence: sa.alsubai@psau.edu.sa

Abstract: Minutiae feature extraction and matching are not only two crucial tasks for identifying fingerprints, but also play an eminent role as core components of automated fingerprint recognition (AFR) systems, which first focus primarily on the identification and description of the salient minutiae points that impart individuality to each fingerprint and differentiate one fingerprint from another, and then matching their relative placement in a candidate fingerprint and previously stored fingerprint templates. In this paper, an automated minutiae extraction and matching framework is presented for identification and verification purposes, in which an adaptive scale-invariant feature transform (SIFT) detector is applied to high-contrast fingerprints preprocessed by means of denoising, binarization, thinning, dilation and enhancement to improve the quality of latent fingerprints. As a result, an optimized set of highly-reliable salient points discriminating fingerprint minutiae is identified and described accurately and quickly. Then, the SIFT descriptors of the local key-points in a given fingerprint are matched with those of the stored templates using a brute force algorithm, by assigning a score for each match based on the Euclidean distance between the SIFT descriptors of the two matched keypoints. Finally, a postprocessing dual-threshold filter is adaptively applied, which can potentially eliminate almost all the false matches, while discarding very few correct matches (less than 4%). The experimental evaluations on publicly available low-quality FVC2004 fingerprint datasets demonstrate that the proposed framework delivers comparable or superior performance to several state-of-the-art methods, achieving an average equal error rate (EER) value of 2.01%.

Keywords: fingerprint minutiae; SIFT feature detection; feature matching; FVC2004 database; EER

check for updates

Citation: Bakheet, S.; Alsubai, S.; Alqahtani, A.; Binbusayyis, A. Robust Fingerprint Minutiae Extraction and Matching Based on Improved SIFT Features. *Appl. Sci.* **2022**, *12*, 6122. https://doi.org/10.3390/app12126122

Academic Editors: Phivos Mylonas, Katia Lida Kermanidis and Manolis Maragoudakis

Received: 27 April 2022
Accepted: 8 June 2022
Published: 16 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Biometrics is often identified as the science of recognizing an individual through his physical/behavioral traits in addition to physiological characteristics. The characteristics that can be used by biometric systems typically involve fingerprint recognition, facial identification, voice recognition and handwriting recognition systems. Among all biometric techniques, fingerprint recognition is the most widely used for personal identification systems, due to its relative permanence and uniqueness [1]. Due to the relatively high level of fingerprint accuracy among all the biometric traits, recent years have witnessed a fairly substantial upswing in the use of many digital fingerprint reading devices in our day-to-day lives. However, these modern devices are being used increasingly for a wide variety of purposes, e.g., for the attendance of the staff before and after their work as a login password in computers or the key to a locker, etc. Fingerprints are thought to be excellent individualizing evidence because they are permanent from birth to death and very unique for each individual (i.e., the probability of two fingerprints being the same is 64 billion to 1.2, according to mathematical assumptions). Furthermore, they are easy to verify and leave marks on every object a person touches. This makes fingerprint-based

biometric security devices highly popular in sensitive high-security areas such as banks, correctional facilities, jewellers, prisons and military establishments [2].

Feature extraction is basically a dimensionality reduction process, whose ultimate goal is to gain the most relevant information from the original data in a high-dimensional space and represent that information in a lower dimensional space. When the input data to an algorithm are too large to be processed, and it is suspected to be excrescent (many data, but not much information), then the input data should be converted into a reduced set of representative features (also called feature vector). In other words, through the process of extracting features, the input high-dimensional data are converted to a small set of low-dimensional features to identify useful information [3]. If the extracted features are carefully selected, it is highly expected that the pertinent information will be extracted from the input data by the reduced feature set, so as to effectively perform a given task by utilizing this reduced representation rather than the full-size input data. A fingerprint recognition system is an automatic pattern recognition system that typically consists of three fundamental stages: image pre-processing, feature extraction and fingerprint matching [4]. A good feature set contains rich information that can effectively distinguish an object from other objects (i.e., being able to identify an object). At this stage, it is necessary as much as possible to prevent the creation of different feature codes for the objects in the same class [5]. Latent fingerprints are merely partial impressions of the finger's ridge pattern inadvertently left after fingertips contact a surface, which involve ridges and valleys. In a fingerprint, ridges are presented as black lines, whereas the valleys are presented as a white area among the ridges [6]. Fingerprint features can be broadly divided into two categories:

1. *Global features:* These features form a special pattern of ridge and valleys, called singularities or Singular Point (SP), and they can further be divided into three types: loop, delta, and whorl. The significant points are the core and the delta. The core is defined as the most points on the innermost ridges, and the delta is defined as the central point where three different trend flows converge (see Figure 1a). It can be argued that these features provide the most useful and crucial information for fingerprint classification, fingerprint matching, and fingerprint alignment [7,8].
2. *Local features:* At the local level, ridge characteristics, collectively called minutia, represent the most widely used features to match fingerprints. There are several types of minutiae, but for practical purposes, just two types of minutiae [9,10] are considered as the two most prominent ridge characteristics: ridge ending (point where ridge ends abruptly) and ridge bifurcation (point where a ridge forks or diverges into branch ridges), as shown in Figure 1b,c.

The primary objective of this work is the development of a fully automated minutiae extraction and matching framework for fingerprint identification and verification purposes, in which an adaptive SIFT detector is applied to high-contrast fingerprints preprocessed by means of denoising, binarization, thinning, dilation and enhancement to improve the quality of latent fingerprints. As a result, an optimized set of highly-reliable salient points discriminating fingerprint minutiae is identified. The SIFT descriptors of the local keypoints in a given fingerprint are then matched with those of the stored templates using a brute force algorithm. Finally, a dual-threshold filter is adaptively applied, which can potentially eliminate almost all the false matches, while losing less than 4% of correct matches. The remainder of the paper is organized as follows: in Section 2, we briefly review closely related prior work on feature extraction and matching algorithms developed for fingerprint recognition. The details of the proposed methodology for fingerprint minutiae extraction and matching and its components and stages are elaborately described in Section 3. Section 4 provides the experimental results and discusses their implications. Finally, concluding remarks and avenues for future research are drawn in Section 5.

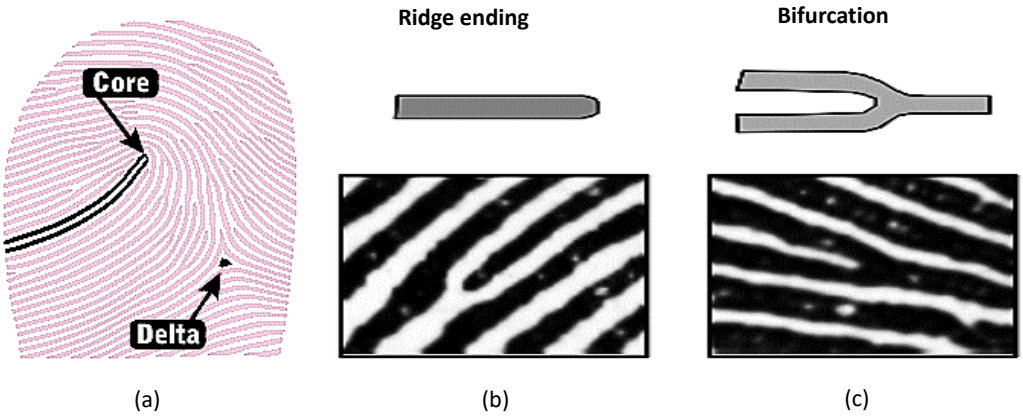


Figure 1. Typical Minutiae structures: (**a**) core and delta singularities; (**b**) ridge ending; and (**c**) ridge bifurcation.

2. Prior Work

In the fingerprint recognition literature, a variety of feature extraction algorithms have been proposed for identifying remarkable features. For example, in [11], Bader and Sagheer developed two computer vision algorithms (i.e., FAST and Harris) to extract features (i.e., corner points) from the finger vein image, where the fingerprint patterns are matched based on the differences between corners represented in the form of points using the Manhattan distance. The false match rate (FMR) and false non-match rate (FNMR) are then minimized to get the optimum threshold that triggers the final decision. The results of the study confirmed that the use of two adaptive algorithms concurrently reduces the error rate and helps to build a reliable system of finger vein identification. In [12], an improved Harris–SIFT image matching algorithm is proposed. First, the feature points of the image are extracted by the Harris corner detection operator. Then, the 28-dimensional increasing homocentric square window is applied to describe the neighborhood information of key feature points. Euclidean distance is typically employed as a similarity measure function in the matching process. Finally, simulation results demonstrated the validity of the improved algorithm, igniting a new thought for the research into the image matching.

The fingerprint feature extraction process aims at finding various minutiae points in a fingerprint to use them further for fingerprint matching. However, the minutiae extraction process can be difficult and not accurate because a fingerprint image may be contaminated due to noise present in the fingerprint image. This results in a lot of minutiae point candidates. Thus, an optimal preprocessing technique is needed to minimize the number of key minutiae points and to get only those key points that can be further used to match the fingerprint [13]. In this direction, recently a study conducted by Singh and Kaurr [14] proposed a methodology for fingerprint minutia extraction using morphological operations, where the number of minutiae points is calculated by using three different methods, and the result is then analyzed. The original image is compared against two preprocessed images; one is obtained by using a dilation operation and the other is obtained by first applying hole filling, followed by a dilation.

In addition, Singh et al. [15] proposed a new fingerprint feature detection algorithm, where it has been found that the presence of artifacts or noise in fingerprint images leads to a lot of spurious minutiae. To tackle this problem, a good strategy for feature extraction has been devised to extract the valid minutiae points in fingerprints and at the same time avoid extracting spurious minutiae points. The presented method could effectively perform the template matching [16] to find out bifurcation and termination of ridges. A smoothing algorithm is developed to find ridges in the fingerprint images with the help of eight different masks. It is a process of making a binary image of ridges from the grayscale fingerprint image. The experimental results verified the validity and accuracy of the algorithm in terms of genuine acceptance rate (GAR), false acceptance rate (FAR), and false rejection rate (FRR).

Furthermore, in [17], an algorithm based on Harris corner detector was proposed for extracting fingerprint minutiae. At the beginning, the Harris corner detector is applied to detect minutiae points and extract high curvature dots from the enhanced fingerprint image. In the later postprocessing stage, false or spurious minutiae are removed, based on the space distribution of minutiae. The type of minutiae is judged using a neighboring gray level information. The exact orientation of minutiae is decided by initial orientation and style. Compared to typical minutiae extraction algorithms, the presented approach needs to not convert the fingerprint image to a binary image and submit the resultant binary image to a thinning process. Instead, the minutiae are directly extracted from the gray-level fingerprint image. This can reduce the processing time and promote efficiency. Experimental results on the FVC2002 latent fingerprint database demonstrated that the presented method is not only fast and fairly reliable, but also fit to use in reality.

In [18], a fingerprint-based authentication approach introduced, by means of fingerprint enhancement, feature extraction and matching techniques. Firstly, the contrast of the small tiles existing in the fingerprint image is enhanced by using an adaptive variant of histogram equalization called Contrast Limited Adaptive Histogram Equalization (CLAHE) along with a combination of Gabor filters [19] and fast Fourier transform (FFT). Then, the improved fingerprint is authenticated by picking a small amount of information from some local interest points—so-called 'minutiae features'. In order to render significantly improved feature detection results, a hybrid combination of SURF and Harris corner detection algorithms is applied to the thinned binary fingerprint image. For fingerprint matching, the Euclidean Distance between the SURF-Harris descriptors of two feature points is used as the similarity criterion of the two matched fingerprints. To automatically remove false matches and incorrect match points, the authors applied an iterative algorithm called RANdom SAmple Consensus (RANSAC). The extensive experiments conducted on the two publicly accessible FVC2002 DB1 and FVC2000 DB1 fingerprint databases demonstrated the efficiency and effectiveness of the proposed method in achieving average recognition rates of 95% and 92.5% for FVC2002 DB1 and FVC2000 DB1 databases, respectively.

3. Proposed Methodology

In this section, the proposed minutiae-based feature extraction and matching framework for robust fingerprint recognition is presented, detailing its major modules, including image preprocessing, minutiae feature extraction, and template matching. A block diagram depicting various modules and workflow within the presented framework is shown in Figure 2. Minutiae points are local descriptive features of a fingerprint image, which are heavily relied upon to match fingerprints accurately. These minutiae points are necessary and sufficient to determine the uniqueness of a fingerprint image. A good-quality fingerprint image typically has 25 to 80 minutiae, depending on the resolution of the fingerprint scanners and the finger position on the sensor [20].

3.1. Image Preprocessing

In fingerprint recognition, the performance of fingerprint minutiae extraction heavily depends upon the quality of the input fingerprint image. Typical preprocessing steps prior to fingerprint minutiae extraction involve binarization, noise removal, and fingerprint segmentation [20,21]. From a fingerprint image, good minutiae points can precisely be located from the thinned ridges. However, in practice, it is not yet possible to extract good minutiae points accurately from a fingerprint image, since a significant percentage of an acquired fingerprint image is of poor-quality due to several factors, including acquisition device conditions (e.g., dirtiness, humidity, pattern location, and orientation), individual artifacts (e.g., skin environment, age, skin tensility, and pressure), etc.

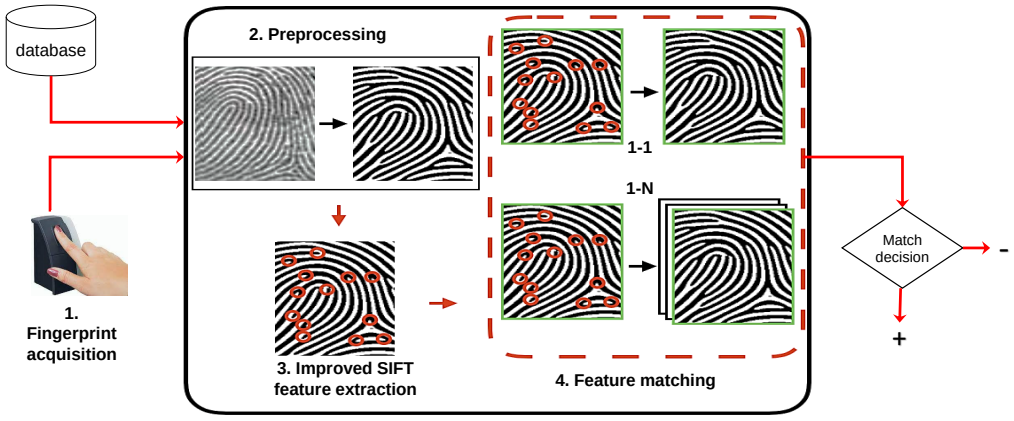


Figure 2. A functional block diagram of the automated fingerprint recognizer.

Moreover, the ridge structures in very low quality fingerprint image are completely corrupted or not always well-defined; therefore, they can not be correctly detected. This leads to a significant number of spurious minutiae and, at the same time, a large percentage of genuine minutiae may be ignored. Therefore, fingerprint image preprocessing for contrast enhancement and illumination correction is a crucial step in any scheme for automated minutiae extraction and matching [22]. In this work, the fingerprint image preprocessing stage that primarily aims at improving minutiae features in the fingerprint image through noise removal and contrast enhancement to reliably extract the minutiae features and potentially minimize the chances of false or spurious minutiae to be detected in the highly corrupted regions [1]. The procedure for image preprocessing is elucidated in some detail in the following subsections below.

3.1.1. Fingerprint Enhancement Using Contextual Filtering

In this work, an enhancement technique based on contextual iterative filters is designed for preprocessing the fingerprint images, which proceeds in four distinct steps: (i) ROI estimation (ii) ridge-visibility enhancement, (iii) ridge-pattern enhancement, and (iv) image binarization. In the first step, an algorithm based on the local standard deviation [23,24] is applied to the input fingerprint image for evaluating the local variance of the image intensity, followed by a thresholding operation for removing unwanted finger shadows. Finally, a mask binary image that reliably identifies the fingerprint ROI is obtained. The second step is performed based on a Homomorphic filtering technique, in which the background image B_I is first estimated by applying a morphological opening operation with a mask s to the input image I. Then, the image R_I representing the fingerprint ridges can readily be obtained as: $I_R = I - B_I$. In order to effectively suppress the noise present in the image, a nonlinear equalization is preformed by applying the logarithm to the ridge image I_R, yielding an improved image $I_L(x,y) = \log(I_R(x,y))$. In the third step, the enhancement of the ridge pattern is performed, similarly to that described in [23], by computing ridge frequency and orientation maps, followed by the application of a bank of Gabor filters optimally tuned to the computed local ridge maps that result in an image I_E containing a minimum amount of background texture details [25]. The fourth and last step in the preprocessing stage is an adaptive image binarization process, which permits effectively reducing the noise in the edges of the ridge pattern described in the image I_E, and reliably estimating the potential minutiae points. For obtaining a uniform contrast between ridges and valleys, the logarithm of I_E is taken as: $I(x,y) = \log(I_E(x,y))$. After establishing the histogram of the image I_I, a binary image of the ridge pattern I_B is obtained as follows:

$$I_B(x,y) = \begin{cases} 0 & \text{if } I_I(x,y) \leq \arg\max_i(H(i)) \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

where H is the histogram of the image I_I. A simplified outline of fingerprint image enhancement based on contextual filters is presented in Figure 3.

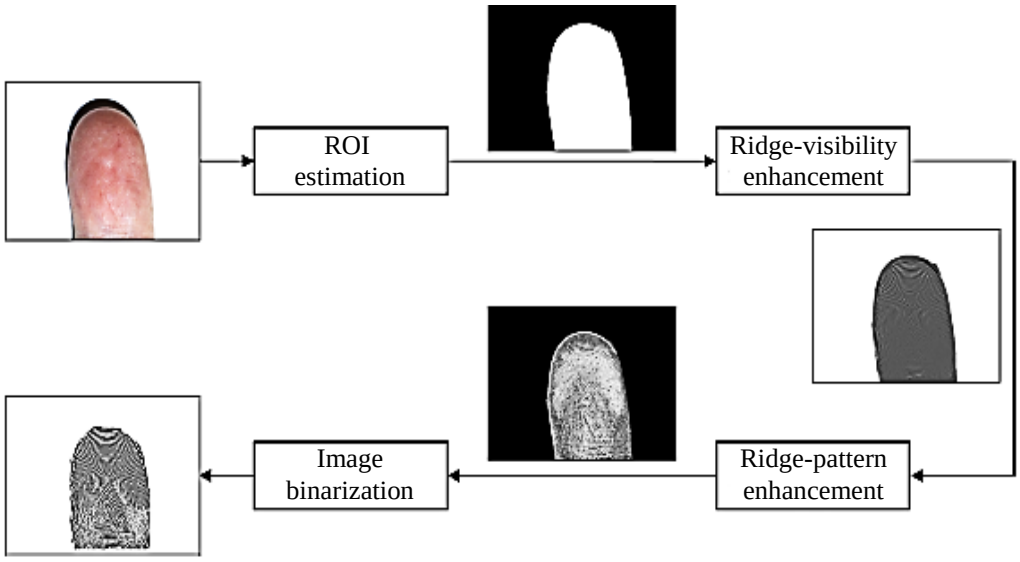


Figure 3. Simplified outline of fingerprint enhancement based on contextual filters.

Additionally, some morphological operations (e.g., erosion and closing) are performed to fill out the holes by smoothening the ridges' surface in the fingerprint image and merging the narrow gaps among the fingerprint ridges [1].

3.1.2. Thinning and Dilation

Erosion (or thinning) and dilation (or thickening) are two morphological operations applied to the fingerprint binary image. As the fingerprint ridges are relatively thick, it is desirable for subsequent minutiae feature extraction and analysis to thin the ridges, so that each is a single pixel thick [1]. The thinning algorithm essentially consists of removing contour points of connected components in a fingerprint image to produce their skeleton [26,27]. In order to extract a skeleton of the fingerprint, a skeletonization process is performed by applying the Zhang–Suen thinning algorithm developed by Zhang and Suen in [28] to the binary image. The Zhang–Suen thinning algorithm is a simple and efficient technique and is one of the most popular adaptive thinning algorithms in the literature, which maintains a 3×3 sized block and consists of two sub-iterations. It is an iterative algorithm that computes the skeleton of an image by removing all the contour points of the image except those belonging to the skeleton [29].

More formally, suppose black pixels are represented by 1's and white pixels are represented by 0s. The algorithm finds connected components in a binary image by working on every dark pixel (e.g., $p1$) that can have eight neighbors (see Figure 4). To preserve the connectivity of the skeleton, each iteration is further divided into two sub-iterations. In the first subiteration, the contour point $p1$ is removed from the digital pattern, if the following conditions are held:

$$2 \leq N(p1) \leq 6, \quad M(p1) = 1, \quad p2 \times p4 \times p6 = 0, \quad p4 \times p6 \times p8 = 0.$$

where $N(p1)$ and $M(p1)$ denote the number of 0 to 1 transitions from $p2$ to $p9$ in a clockwise direction and the number of non-zero neighbors of $p1$, respectively. That is, $N(p1) = p2 + p3 + \ldots + p8 + p9$.

P9	P2	P3
P8	P1	P4
P7	P6	P5

Figure 4. A 3 × 3 sized block.

In the second subiteration, the contour point p1 is removed from the digital pattern, if the following conditions are met:

$$2 \leq N(p1) \leq 6, \quad M(p1) = 1, \quad p2 \times p4 \times p8 = 0, \quad p2 \times p6 \times p8 = 0.$$

An example of fingerprint thinning is given in Figure 5a. Dilation is one of basic morphological operations, which is typically applied to binary images to enhance image features by enlarging the boundaries of the segmented image objects and fill in the holes within these objects. In this work, the dilation process is used to increase the width of the fingerprint ridges in a way such that, if there is a gap of one or two pixels between any two ridges, then they can be joined to form a single ridge. Such a gap of one or two pixels between fingerprint ridges can be due to any error of some sort [1]. For fingerprint dilation, we create a function that looks at every pixel of the binary image and, in the meantime, all the neighboring pixels of the pixel under consideration are checked as well. Consequently, when a pixel is selected, all of its neighbors are checked and, if any one of them is black, the value of that pixel is changed to black so that this results in removing small holes in the image and also joining ridges that have at most a 2-pixel gap [24]. A binary fingerprint image after the dilation operation is shown in Figure 5b. After performing thinning and dilation operations, the resultant image is almost well-suited for feature extraction by removing all noises present in the image. However, there still remains very small patterns in the fingerprint image that need to be eliminated in order to perform more robust minutiae feature extraction [1]. This can be performed simply by calculating the number of pixels in each ridge and any ridge having fewer than a certain number of pixels (e.g., $\leq$20 pixel) is removed from the fingerprint image (see Figure 5c).

(**a**) (**b**) (**c**)

Figure 5. Main steps of fingerprint preprocessing: (**a**) thinning, (**b**) dilation and (**c**) unwanted pattern removal.

3.2. Minutiae Feature Extraction

After finishing all the aforementioned preprocessing operations, the enhanced fingerprint image is obtained, in which the minutiae feature points can be easily located [1]. In this work, the robust minutiae feature points of the enhanced fingerprint image are determined by using an adaptive version of the SIFT detector [30] to obtain a sparse set of frames (or local keypoints) from the fingerprint image. The keypoints are viewed as oriented disks attached to blob-like structures of the fingerprint image under consideration [31]. As the image scales, rotates, and/or translates, these keypoints could help track image objects and thus the deformation. The effect of such deformation on the feature appearance is excluded by canonization, i.e., by mapping the keypoints to a canonical disk [32–34]. In order to search for fingerprint blobs on a multiple scale, the SIFT based detector constructs a scale space defined as a function $F(x, \sigma)$ of spatial and scale variables, where $x \in \mathbb{R}^2$ and $\sigma \in \mathbb{R}_+$ are the spatial and scale coordinates, respectively. The domain of the scale-space variable σ is sampled at discretized steps in logarithmic values:

$$\sigma(s, o) = \sigma_0 2^{o+s/S}, \; o \in \mathbb{Z}, \; s = 0, \ldots, S-1 \tag{2}$$

where o, s, $S \in \mathbb{N}$ and $\sigma_0 \in \mathbb{R}_+$ are the octave index, scale index, scale resolution and base scale offset, respectively. Notice that octaves of negative index are likely to be obtained. Then, a resolution defined as a function of the octave can be used for sampling the spatial coordinate x on a lattice, as follows:

$$x = 2^o x_o, \qquad o \in \mathbb{Z}, \quad x_o \in [0, \ldots, N_o - 1] \times [0, \ldots, M_o - 1] \tag{3}$$

where x_o and (N_o, M_o) denote the spatial index and spatial resolution of octave o, respectively. The resolution of the octaves can be obtained from the resolution (M_0, N_0) of the base octave $o = 0$, as follows:

$$N_o = \lfloor \frac{N_0}{2^o} \rfloor, \qquad M_o = \lfloor \frac{M_0}{2^o} \rfloor \tag{4}$$

However, some scale levels can be usefully stored twice across different octaves, by allowing the parameter s to be negative or larger than S. More formally, let $[s_{\min}, s_{\max}]$ and $[o_{\min}, o_{\min} + O - 1]$ be the ranges of s and o, respectively, where O denotes the number of octaves.

The SIFT detector and descriptor make use of two scale spaces, namely a Gaussian space and a Difference of Gaussian (DoG) space. The *Gaussian scale-space* of an image $I(x)$ can be obtained from the 'zero-scale' image by a Gaussian convolution:

$$G(x, \sigma) \triangleq (g_\sigma * I)(x) \tag{5}$$

where the scale σ is typically sampled a particular way to reduce redundancy. In practice, it is assumed that the fingerprint image is nominally pre-smoothed σ_n, accordingly $G(x, \sigma) = (g_{\sqrt{\sigma^2 - \sigma_n^2}} * I)(x)$. As pointed out in [30], successive convolutions need to be performed by small Gaussian kernels to allow the pyramid to be incrementally computed. Another scale space that the detection algorithm makes use of is the *difference of Gaussians* which is progressively calculated from the scale derivative of $G(x, \sigma)$ along the scale coordinate σ:

$$D(x, \sigma(s, o)) \triangleq G(x, \sigma(s+1, o)) - G(x, \sigma(s, o)) \tag{6}$$

For computing the octave $o = -1$, the fingerprint image is enlarged by a factor of, e.g., 2 through bilinear interpolation (applied to the enlarged image $\sigma_n = 1$). For scale space extrema detection at all scales, the DoG space possesses $s \in [-1, S]$. As the DoG space is derived from the differentiation of the Gaussian scale space, the latter has a scale index in the range of $s \in [-1, S+1]$. With regard to the parameter O, it should be set as high as possible to cover all octaves. Thus, keypoints are extracted by picking local-extremum points in

3D neighborhoods of $D(x,\sigma)$, where the extrema are efficiently selected by looking at $9 \times 9 \times 9$ neighborhoods of samples. Since the octave is represented by a 3D array, the index k is mapped to scale space indexes (x_1, x_2, s), as follows:

$$k - 1 = x_2 + x_1 M_o + (s - s_{\min}) M_o N_o \tag{7}$$

The index k can alternatively be mapped to a subscript (i, j, l) by using:

$$x_1 = j - 1, \qquad x_2 = i - 1, \qquad s = l - 1 + s_{\min} \tag{8}$$

The actual size of a spatial bin is $m\sigma$, where σ is the scale of the keypoint, and $m = 3.0$ is a nominal factor (see Figure 6).

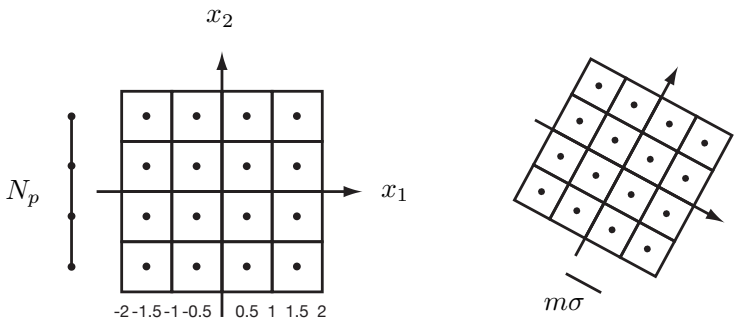


Figure 6. SIFT descriptor layout for 4×4 sub-regions.

Due to the way such extrema are detected, the following inequality constraints are strictly satisfied: $1 \leq x_2 \leq M_o - 2$, $1 \leq x_1 \leq N_o - 2$ and $s_{\min} + 1 \leq s \leq s_{\max} - 1$. As the interest is in both local maxima and minima, the process is then repeated for $-G(x,\sigma)$. For sub-pixel refinement, a test with a threshold on the intensity $D(x,\sigma)$ is simultaneously applied on the peakedness of the extremum to exclude weak points and/or edge points (see Figure 7).

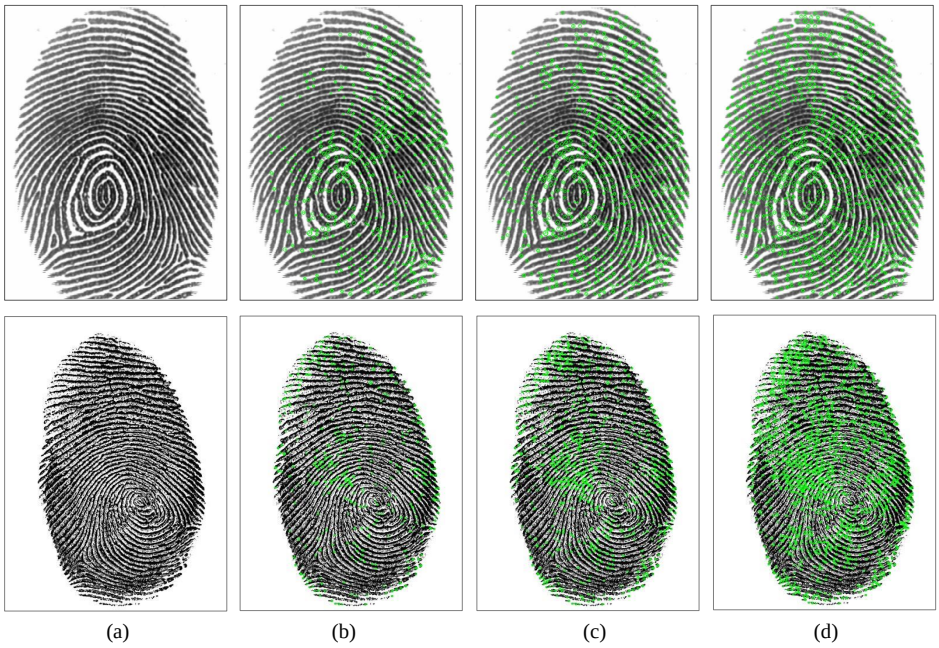


Figure 7. SIFT minutiae detection for various threshold values: (**a**) source image and detected frames at threshold values of (**b**) 5.0, (**c**) 7.5, and (**d**) 10.0.

The orientation θ of a detected keypoint (x, σ) is obtained as a predominant orientation of the gradient around the keypoint calculated as a quadratically interpolated maximum of the histogram of gradient orientations $\angle \nabla G(x_1, x_2, \sigma)$ around the keypoint. The histogram is then weighted by the magnitude of the gradient $|\nabla G(x_1, x_2, \sigma)|$, using a Gaussian window of standard deviation 1.5σ centered on the keypoint. After arranging data in bins, a moving average filter is applied to smooth the constructed histogram before computing the maximum. Besides the global maximum, there is also a need to retain all the local maximum of a value over 0.8% of the maximum. Hence, multiple SIFT frames would be obtained for each location and scale. Two examples of SIFT keypoint detection are shown in Figure 8 below.

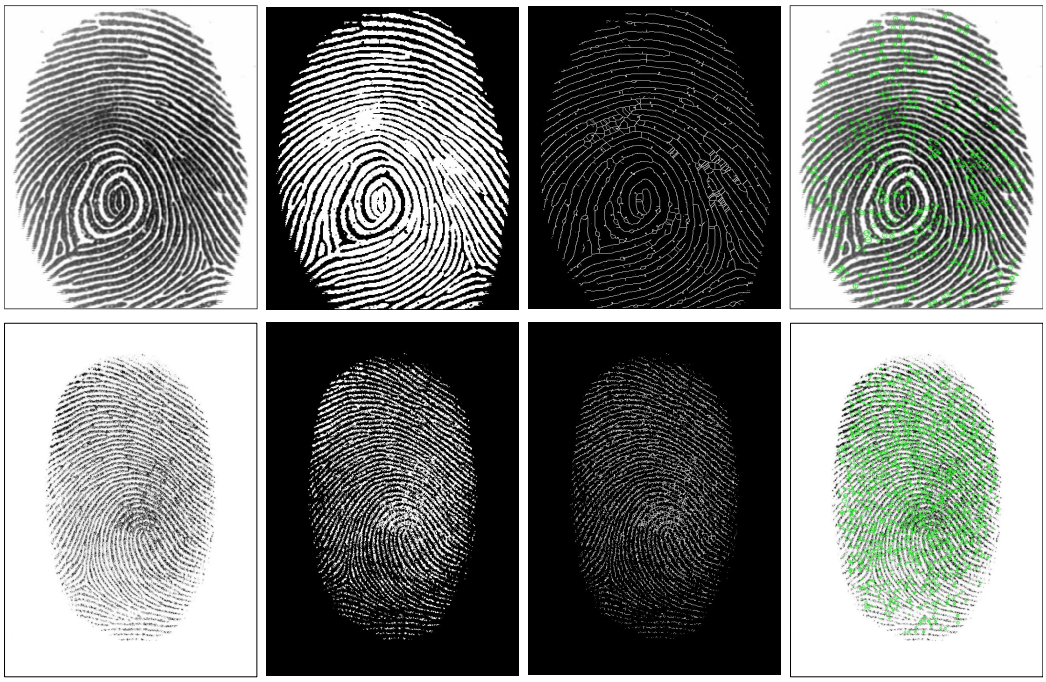


Figure 8. SIFT minutiae feature detection (**a**) original image, (**b**) binary image, (**c**) thinned image, and (**d**) detected SIFT minutiae keypoints (pixels shown in green color).

The SIFT descriptor of a detected keypoint (x, σ) is a local statistic of gradient orientations of $G(\cdot, \sigma)$, which is computed from a weighted 3D histogram of gradient orientations. More formally, the SIFT descriptor is expressed as a weighted and interpolated histogram of the gradient orientations and locations within a patch around the keypoint, where the histogram domain is represented in tuples: $(x, \theta) \in \mathbb{R}^2 \times \mathbb{R}/\mathbb{Z}$. Moreover, the bins constitute a 3D lattice with $N_p = 4$ spatial bins along each spatial direction, $N_o = 8$ bins for the gradient orientation and a total of $N_p^2 N_o = 128$ components. The window $H(x)$ is a Gaussian with a deviation of $N_p/2$, i.e., half of the spatial bin range.

For invariance purposes, the histogram is then projected on the image domain, based on the local reference frame around the keypoint. The spatial dimensions are then multiplied by a factor of $m\sigma$, where m and σ are a nominal factor (set to 3.0 by default) and the scale of the keypoint, respectively. The layout is rotated as well to ensure that x_1 is aligned to the keypoint orientation θ. The resultant histograms are further weighted by the gradient modulus, and the contributions of their gradient orientations are smoothly distributed using a trilinear interpolation into adjacent histogram bins to avoid boundary effects in which the descriptor changes abruptly as a sample orientation shifts smoothly from one bin to another.

3.3. Minutiae Feature Matching

From the above description of the feature extraction module, it is evident that the SIFT algorithm can generally be seen as a local image operator that takes a given image and transforms it into a large collection of local feature vectors. Hence, the feature matching procedure between feature descriptors of two fingerprint images essentially involves computing the Euclidean distance between each descriptor of the first image and each descriptor of the second image in Euclidean space [35]. To use this local operator for fingerprint recognition purposes, it is applied on two fingerprint images, i.e., a test and template image.

To find corresponding features between the two images, which could lead to fingerprint recognition, several feature matching approaches can be applied. Based on the Nearest Neighborhood (NN) procedure, for each feature a_i in the feature set of the query fingerprint image, the entire reference database is queried to find the most similar stored feature b_i with the smallest Euclidean distance to the feature a_i. A pair of corresponding features (a_i, b_i) is typically termed a match $M(a_i, b_i)$ [36]. To check if this match is positive or negative, a certain predefined threshold is considered. For matching, when the ratio of the Euclidean distance of the nearest-neighbor to the Euclidean distance of the next nearest-neighbor exceeds the predefined threshold, the matched feature is rejected. On the other hand, if the Euclidean distance between two feature vectors a_i and b_i does not exceed the value of the threshold, the match $M(a_i, b_i)$ is labelled as positive and stored as a valid match. As shown in [30], Euclidean distance refers to the distance of keypoints in the feature space.

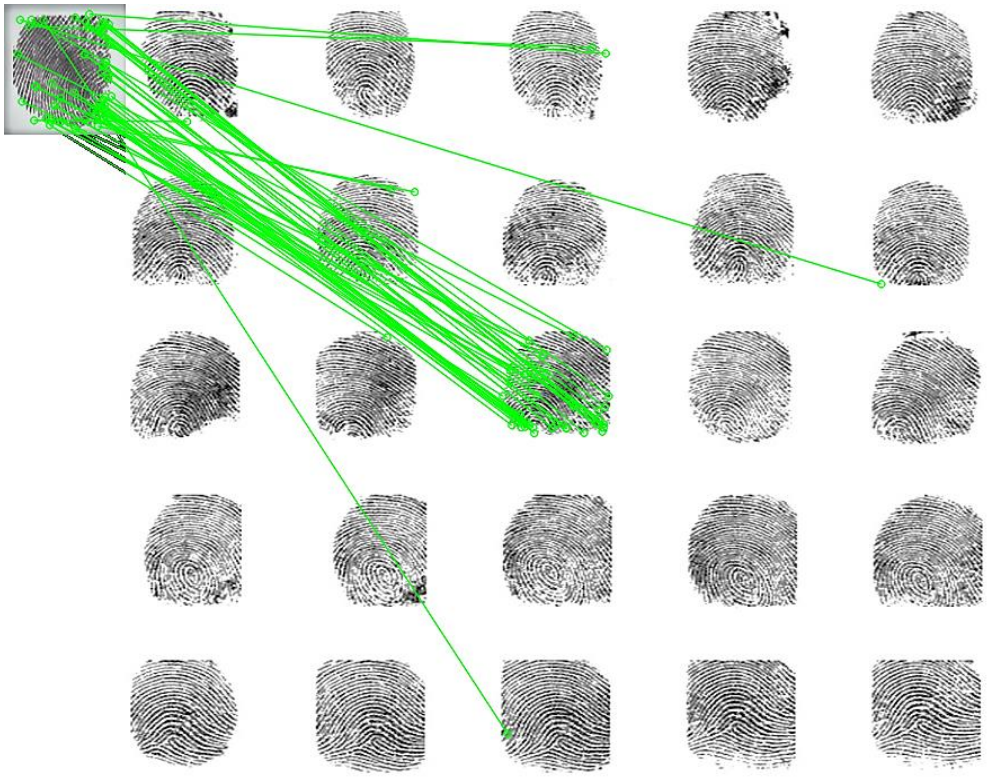
The keypoints (i.e., features) in image space are transformed into a multi-dimensional space based on their characteristics such as gradients, orientations, magnitude, locations and brightness, where each feature is represented by a feature vector. Then, the Euclidean distance between the two feature vectors **a** and **b** is defined as:

$$D(\mathbf{a}, \mathbf{b}) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \ldots + (a_n - b_n)^2} = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{9}$$

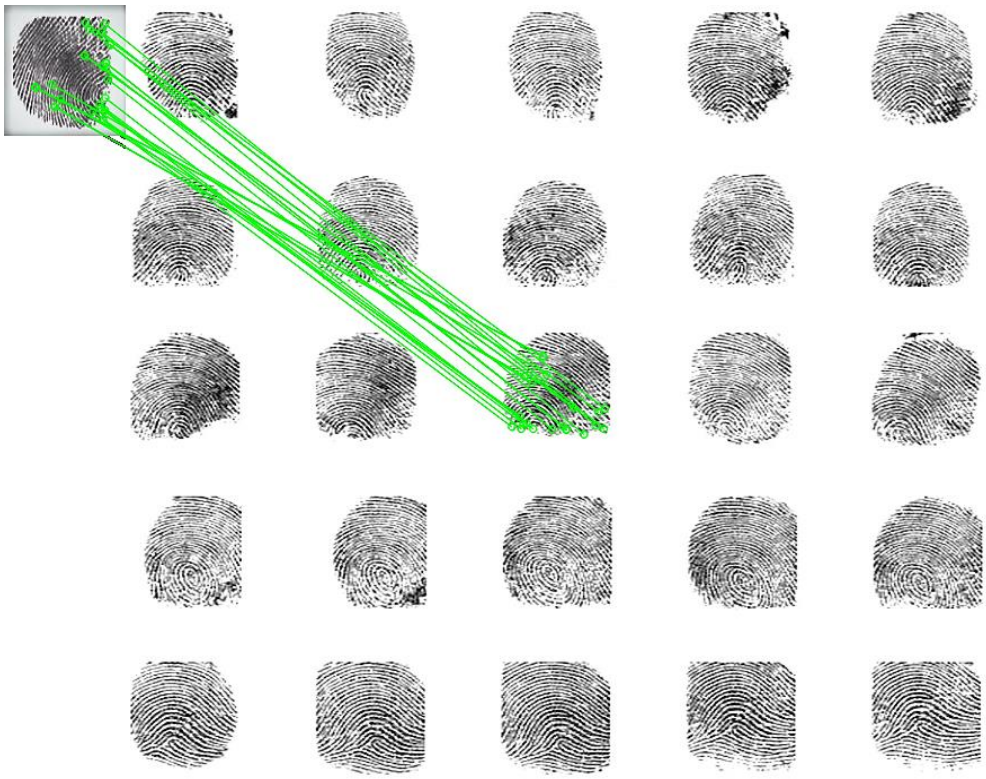
where n denotes the number of features in the feature set. The value of $D(\mathbf{a}, \mathbf{b})$ is then compared with a predefined threshold. If it is greater than the threshold, the matched keypoint will be discarded. However, the computation of the pairwise Euclidean distances between all feature points can be prohibitively expensive. As a result, it is very useful, especially for computational purposes, to find inner products of vectors in the feature spaces [37]. This will greatly reduce the computational burden and, in the meantime, retain the feature robustness. As the distance between feature vectors is probably going to be similar, mismatch might take place, but the angle is constantly different [38]. On the other hand, cosine similarity that measures the similarity between two vectors in the feature space can be obtained by the cosine of the angle between the two vectors. Formally, having computed the inner product of a pair of feature vectors, then it is straightforward to compute the inverse cosine between each pair of feature vectors as follows:

$$\begin{aligned} \mathbf{a} \cdot \mathbf{b} &= \sum_{i=1}^{n} a_i b_i = a_1 b_1 + a_2 b_2 + \ldots + a_n b_n \\ \theta &= \arccos\left(\frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} \right) \end{aligned} \tag{10}$$

Then, the nearest neighbor is checked to see if it has an angle below the predefined threshold ratio, i.e., $\theta < \tau_\theta$. It is worth mentioning that, in the original SIFT matching algorithm, only the nearest neighbor distance is compared against other distances and the smallest distance value is selected, whereas, in this improved SIFT matching algorithm, angles between feature vectors are compared. Additionally, the ratio of descriptor distances is employed for *outlier rejection* to reduce the number of false positive matches (see Figure 9 below).



(**a**)



(**b**)

Figure 9. Fingerprint minutiae matching results using (**a**) original SIFT algorithm and (**b**) improved SIFT algorithm.

As can be seen in Figure 9, the improved SIFT algorithm gives prominently better results in terms of matching quality and robustness compared to the original SIFT matching algorithm, particularly when the fingerprint image contains much local similar characteristics. However, despite discarding a few correct matches, nearly all mismatches are removed.

4. Experimental Results and Analysis

In this section, the results of the experiments and simulations conducted to verify the validity and performance of the proposed framework for fingerprint identification are presented. The experiments were carried out independently on a publicly available fingerprint database, namely FVC2004 [39] from the Fingerprint Verification Competition (FVC); a series of competition organised by the University of Bologna (Italy) which aims at establishing benchmarks for evaluating the performance of fingerprint recognition systems. FVC2004 is basically a multi-database, where each component database was created from fingerprints captured with a different sensor technology. In FVC2004 fingerprint database, there are four distinct datasets (i.e., DB1, DB2, DB3, and DB4), each containing 110 different fingerprint images with eight impressions per finger (resulting in a total of $110 \times 8 = 880$ fingerprints). Each component database is further divided into two disjoint subsets (A and B), so that the subset A contains the first 100 fingers and eight impressions per finger (i.e., 800 impressions in total), which is commonly used for the performance evaluation of fingerprint verification systems, whereas set B that is made available to allow parameter tuning before the submission of the algorithms only has the last 10 fingers (80 impressions).

The fingerprints in FVC2004 database mainly depend upon image quality, size and the source sensor type used for acquiring the fingerprints. Fingerprint images are available in a TIFF format with 8-bit gray-level depth, and a resolution of about 500 dpi. The image size varies depending on the database. During performance evaluation, only homogeneous fingerprint images that belong to the same component database are matched to each other. Fingerprint samples taken from FVC2004 fingerprint database are shown in Figure 10.

Figure 10. A sample of fingerprint images from the FVC2004 fingerprint database.

Taking into account that the fingerprint data included in FVC2004 database were collected by using different fingerprint sensing technologies, including optical, semiconductor, thermal, and tactile, and also the variations in image size and resolution are particularly apparent in this database. In addition, in this database, fingerprints were not acquired in realistic settings according to a formal protocol, but instead primarily characterized by the presence of different distortions (e.g., rotations, scalings, translations, and poor quality in resolution) within fingerprints of the same person's finger. Taking all these challenging aspects mentioned above into consideration, this dataset represents a great choice for the testing and validation of the proposed fingerprint matching framework in extreme conditions. Additionally, a publicly available fingerprint dataset presented in [40] was used for feature matching subsystem, where these fingerprint images acting as the fingerprint biometric data undergo minutiae points feature extraction.

The performance of the proposed framework is evaluated on the FVC2004 benchmark fingerprint database (described above) by the Equal Error Rate (EER) that can be simply attained from the ROC (Receiver Operating Characteristic) curve; a plot of FMR against FNMR. The EER refers to the point on the ROC curve where FMR and FNMR are equal; a lower EER value generally indicates a better performance of the fingerprint verification

system. A plot of the ROC curve that quantitatively shows the performance of the proposed framework is presented in Figure 11. In real-world applications, a fingerprint recognition system usually operates away from the EER point by reducing the FMR in order to ensure a high level of security. Note that the FMR and FNMR are inversely related and there is a strict trade-off between them; therefore, it is of particular interest to evaluate how the FNMR is affected. Moreover, it could also be argued here that the matching performance (in terms of EER) can be greatly affected by a large variation in true matches against the false matches.

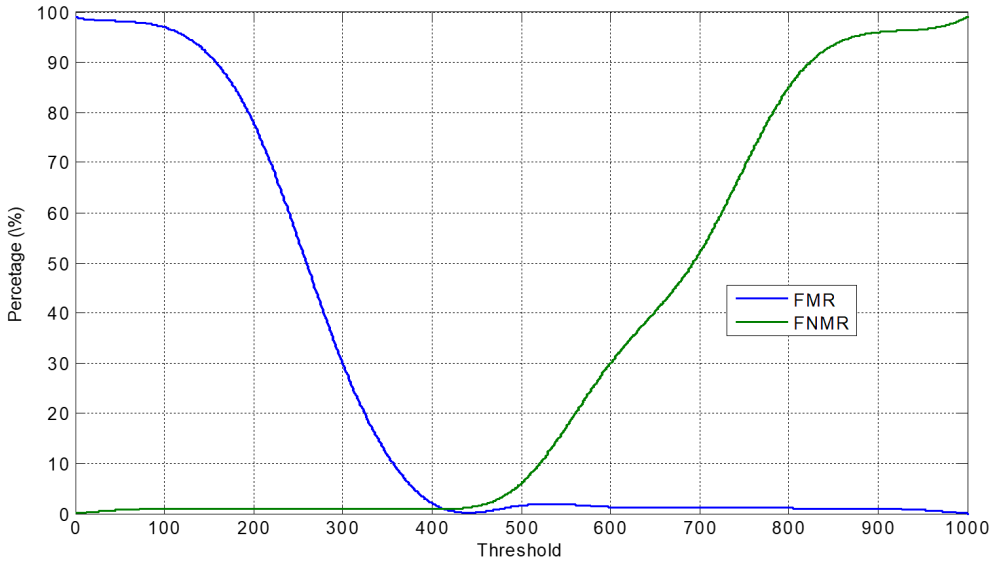


Figure 11. A plot of the ROC curve between FMR and FNMR for the FVC 2004 fingerprint database.

A comparison of the matching performance in terms of EER (%) between the proposed framework and some closely related works is presented in Table 1.

Table 1. A summary of the performance comparison between the proposed framework and most closely related techniques.

Work	Techniques	EER (%)
Proposed method	Improve SIFT Features	02.01
Ali and Prakash [41]	Fingerprint Shell	02.02
Arunalatha et al. [42]	Dictionary Learning	02.04
Francesco et al. [43]	Orientation Extraction	02.06
Alam et al. [44]	Fingerprint template	02.07
Jucheng et al. [45]	Two-Stage Enhancement Scheme	02.19
Bartunek et al. [46]	Pre-processing	02.40
Carsten [47]	Curved Gabor Filters	11.97

From the figures presented in Table 1, one can see that the proposed framework performs effectively in comparison with other techniques, achieving the best EER value of 2.01%. In addition, there is a close matching performance (in terms of EER) between our framework and those previously presented in [42,43,45,46]. The average execution time of the fingerprint recognition algorithm, including enhancement, feature extraction and matching, is about 674 ms, so that it can run sufficiently fast for real-time operation, since the additional computational costs for the fingerprint enrollment are negligible besides the realtime SIFT feature extraction and matching. The average recognition time taken by the proposed technique has been compared with that of several closely related techniques, as shown in Table 2.

Table 2. Average time taken comparison of various closely related techniques.

Techniques	Elapsed Time (ms)
Proposed Technique	674
Ali and Prakash [41]	632
Arunalatha et al. [42]	745
Francesco et al. [43]	957
Alam et al. [44]	1075
Jucheng et al. [45]	1119
Carsten [47]	1149

The presented fingerprint recognition system is implemented for much of its framework using Microsoft Visual Studio 2017 development tools and OpenCV Vision Library to realize real-time digital image processing and automatic object recognition. All tests and evaluations were preformed on a PC with an Intel® Core(TM) i7 CPU—2.60 GHz with Turbo Boost Technology, 8 GB RAM, and running Windows 10 Pro (64-bit) as the operational system.

5. Conclusions

This paper has introduced an automated minutiae extraction and matching framework for fingerprint identification and verification purposes, The proposed framework has followed a stepwise procedure as follows: first, multiple preprocessing operations including denoising, binarization, thinning, dilation, and enhancement are performed on the input fingerprint image to obtain high accuracy minutiae data. An improved SIFT detector is then applied to high-contrast fingerprints to detect an optimized set of highly-reliable salient points discriminating fingerprint minutiae and describe them accurately and quickly. Then, the SIFT descriptors of the local key-points in a given fingerprint are matched with those of the stored templates using a brute force approach. Finally, an adaptive dual-threshold filter is applied to remove false matches, while preserving the correct one. Experimental results on the public FVC2004 fingerprint datasets have demonstrated that the presented framework admits highly competitive or even much better performance than several state-of-the-art methods in terms of EER and robustness. Furthermore, the results of the performance evaluation indicated that the framework is easy to perform, inexpensive, relatively fast, and can offer enough finger ridge detail to allow for excellent fingerprint comparison for identification and verification purposes, which can be deemed as the most important managerial implication emerging from this study. Nevertheless, a possible limitation of this study is that the validation process has been performed using a single database of a relatively small number of latent fingerprint images. Future work will be along two main axes. The first will be further extension of this framework by incorporating advanced deep learning paradigm for better recognition performance. The second will concentrate on conducting more experiments for testing and evaluation of our approach on challenging latent fingerprints unintentionally left by subjects at crime scenes.

Author Contributions: Conceptualization, S.B.; methodology, S.B.; software, S.B.; validation, S.B. and S.A.; formal analysis, S.B. and S.A.; project administration, S.A., A.A. and A.B.; funding acquisition, S.A., A.B. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Singh, P.; Kaur, L. Fingerprint Feature Extraction using Ridges and Valleys. *Int. J. Eng. Res. Technol.* **2015**, *4*, 1330–1334.
2. Grosz, S.A.; Engelsma, J.J.; Liu, E.; Jain, A.K. C2CL: Contact to Contactless Fingerprint Matching. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 196–210. [CrossRef]
3. Bakheet, S.; Al-Hamadi, A. Chord-length shape features for license plate character recognition. *J. Russ. Laser Res.* **2020**, *41*, 156–170. [CrossRef]
4. Ali, S.F.; Khan, M.A.; Aslam, A.S. Fingerprint matching, spoof and liveness detection: Classification and literature review. *Front. Comput. Sci.* **2021**, *15*, 151310. [CrossRef]
5. Kumar, G.; Bhatia, P.K. A Detailed Review of Feature Extraction in Image Processing Systems. In Proceedings of the 2014 Fourth International Conference on Advanced Computing & Communication Technologies, Rohtak, India, 8–9 February 2014; pp. 5–12.
6. Duan, Y.; He, K.; Feng, J.; Lu, J.; Zhou, J. Estimating 3D Finger Pose via 2D-3D Fingerprint Matching. In Proceedings of the 27th International Conference on Intelligent User Interfaces, Helsinki, Finland, 22–25 March 2022.
7. Bakheet, S.; Al-Hamadi, A. Robust hand gesture recognition using multiple shape-oriented visual cues. *EURASIP J. Image Video Process* **2021**, *2021*, *26*. [CrossRef]
8. Mali, K.; Bhattacharya, S. Fingerprint recognition using global and local structures. *Int. J. Comput. Sci. Eng.* **2011**, *3*, 161–172.
9. Alonso-Fernandez, F.; Bigun, J.; Fierrez, J.; Fronthaler, H.; Kollreider, K.; Ortega-Garcia, J. Fingerprint recognition. In *Guide to Biometric Reference Systems and Performance Evaluation*; Springer: London, UK, 2009; pp. 51–88.
10. Bakheet, S.; Al-Hamadi, A. A framework for instantaneous driver drowsiness detection based on improved HOG features and naïve Bayesian classification. *Brain Sci.* **2021**, *11*, 240. [CrossRef]
11. Bader, A.S.; Sagheer, A.M. Finger Vein Identification Based On Corner Detection. *J. Theor. Appl. Inf. Technol.* **2018**, *96*, 2696–2705.
12. Cao, Y.; Pang, B.; Liu, X.; Shi, Y. An Improved Harris-SIFT Algorithm for Image Matching. In Proceedings of the International Conference on Advanced Hybrid Information Processing, Harbin, China, 17–18 July 2017; pp. 56–64.
13. Bakheet, S.; Al-Hamadi, A. A Discriminative Framework for Action Recognition Using f-HOL Features. *Information* **2016**, *7*, 68. [CrossRef]
14. Singh, P.; Kaur, L. Fingerprint feature extraction using morphological operations. In Proceedings of the International Conference on Advances in Computer Engineering and Applications, Cebu, Philippines, 15–17 December 2015; pp. 764–767.
15. Singh, K.; Kaur, K.; Sardana, A. Fingerprint feature extraction. *Int. J. Comput. Sci. Technol.* **2011**, *2*, 237–241.
16. Bakheet, S.; Al-Hamadi, A.; Mofaddel, M.A. Recognition of Human Actions Based on Temporal Motion Templates. *Br. J. Appl. Sci. Technol.* **2017**, *20*, 1–11. [CrossRef]
17. Lian, Q.; Zhang, J.; Chen, S. Extracting fingerprint minutiae based on Harris corner detector. *Opt. Tech.* **2008**, *34*, 383–387.
18. Bakheet, S.; Al-Hamadi, A.; Youssef, R. A Fingerprint-Based Verification Framework Using Harris and SURF Feature Detection Algorithms. *Appl. Sci.* **2022**, *12*, 2028. [CrossRef]
19. Bakheet, S.; Al-Hamadi, A. Hand gesture recognition using optimized local Gabor features. *J. Comput. Theor. Nanosci.* **2017**, *14*, 1380–1389. [CrossRef]
20. Thakkar, D. Minutiae Based Extraction in Fingerprint Recognition. Available online: https://www.bayometric.com/minutiae-based-extraction-fingerprint-recognition/ (accessed on 10 October 2017).
21. Sadek, S.; Abdel-Khalek, S. Generalized α-Entropy Based Medical Image Segmentation. *J. Softw. Eng. Appl.* **2014**, *7*, 62–67. [CrossRef]
22. Bakheet, S.; Al-Hamadi, A. A Hybrid Cascade Approach for Human Skin Segmentation. *Br. J. Math. Comput. Sci.* **2016**, *17*, 1–14. [CrossRef]
23. Hong, L.; Wan, Y.; Jain, A. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 777–789. [CrossRef]
24. Sadek, S.; Al-Hamadi, A.; Michaelis, B.; Sayed, U. A New Method for Image Classification Based on Multi-level Neural Networks. In Proceedings of the International Conference on Signal and Image Processing (ICSIP'09), Amsterdam, The Netherlands, 23–25 September 2009; pp. 197–200.
25. Bakheet, S.; Al-Hamadi, A. Computer-Aided Diagnosis of Malignant Melanoma Using Gabor-Based Entropic Features and Multilevel Neural Networks. *Diagnostics* **2020**, *10*, 822. [CrossRef]
26. Patel, M.B.; Parikh, S.M.; Patel, A.R. An Improved Thinning Algorithm For Fingerprint Recognition. *Int. J. Adv. Res. Comput. Sci.* **2017**, *8*, 1238–1244. [CrossRef]
27. Hall, R.W. Fast parallel thinning algorithms: Parallel speed and connectivity preservation. *Commun. ACM* **1989**, *32*, 124–131. [CrossRef]
28. Zhang, T.Y.; Suen, C.Y. A Fast Parallel Algorithms For Thinning Digital Patterns. *Commun. ACM* **1984**, *27*, 236–239. [CrossRef]
29. Kocharyan, D. A modified fingerprint image thinning algorithm. *Am. J. Softw. Eng. Appl.* **2013**, *2*, 1–6.
30. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *2*, 91–110. [CrossRef]
31. Vedaldi, A. An Implementation of SIFT Detector and Descriptor. 2008. Available online: http://cs.tau.ac.il/~turkel/imagepapers/ (accessed on 8 March 2022).
32. Lee, Y.; Lee, D.H.; Park, J.H. Revisiting NIZK-Based Technique for Chosen-Ciphertext Security: Security Analysis and Corrected Proofs. *Appl. Sci.* **2021**, *11*, 3367. [CrossRef]
33. Dospinescu, O.; Brodner, P. Integrated Applications with Laser Technology. *Inform. Econ.* **2013**, *17*, 53–61. [CrossRef]

34. Agarwal, D.; Garima; Bansal, A. A Utility of Ridge Contour Points in Minutiae-Based Fingerprint Matching. In Proceedings of the International Conference on Computational Intelligence and Data Engineering, Hyderabad, India, 8–9 August 2020.
35. Jiayuan, R.; Yigang, W.; Yun, D. Study on eliminating wrong match pairs of SIFT. In Proceedings of the IEEE 10th International Conference on Signal Processing Proceedings, Ljubljana, Slovenia, 18–20 September 2010; pp. 992–995. [CrossRef]
36. Omercevic, D.; Drbohlav, O.; Leonardis, A. High-Dimensional Feature Matching: Employing the Concept of Meaningful Nearest Neighbors. In Proceedings of the 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 14–20 October 2007; pp. 1–8. [CrossRef]
37. Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988; pp. 147–151.
38. Moravec, H. Towards Automatic Visual Obstacle Avoidance. In Proceedings of the 5th International Joint Conference on Artificial Intelligence (IJCAI'77), Cambridge, MA, USA, 22–25 August 1977; Volume 1, p. 584.
39. Maio, D.; Maltoni, D.; Cappelli, R.; Wayman, J.L.; Jain, A.K. FVC 2004: Third Fingerprint Verification Competition. In Proceedings of the International Conference on Biometric Authentication, Hong Kong, China, 15–17 July 2004; Volume 3072, pp. 1–7.
40. Lakshmanan, R.; Selvaperumal, S.; Chow, M. Integrated Finger Print Recognition Using Image Morphology and Neural Network. *Int. J. Adv. Stud. Comput. Sci. Eng.* **2014**, *3*, 40–48.
41. Ali, S.; Prakash, S. 3Dimensional Secured Fingerprint Shell. *Pattern Recognit. Lett.* **2019**, *126*, 68–77. [CrossRef]
42. Arunalatha, J.S.; Tejaswi, V.; Shaila, K.; Anvekar, D.; Venugopal, K.R.; Iyengar, S.S.; Patnaik, L.M. FIVDL: Fingerprint Image Verification using Dictionary Learning. *Procedia Comput. Sci.* **2015**, *54*, 482–490. [CrossRef]
43. Turroni, F.; Maltoni, D.; Cappelli, R.; Maio, D. Improving Fingerprint Orientation Extraction. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 1002–1013. [CrossRef]
44. Alam, B.; Jin, Z.; Yap, W.S.; Goi, B.M. An alignment-free cancelable fingerprint template for bio-cryptosystems. *J. Netw. Comput. Appl.* **2018**, *15*, 20–32. [CrossRef]
45. Yang, J.; Xiong, N.; Vasilakos, A.V. Two-Stage Enhancement Scheme for Low-Quality Fingerprint Images by Learning from the Images. *IEEE Trans. Hum.-Mach. Syst.* **2013**, *43*, 235–248. [CrossRef]
46. Bartunek, J.S.; Nilsson, M.; Sallberg, B.; Claesson, I. Adaptive Fingerprint Image Enhancement With Emphasis on Preprocessing of Data. *IEEE Trans. Image Process.* **2013**, *22*, 644–656. [CrossRef] [PubMed]
47. Gottschlich, C. Curved-Region-Based Ridge Frequency Estimation and Curved Gabor Filters for Fingerprint Image Enhancement. *IEEE Trans. Image Process.* **2012**, *21*, 2220–2227. [CrossRef] [PubMed]