

1

Fingerprint Image Analysis: Role of Orientation Patch and Ridge Structure Dictionaries

Anil K. Jain and Kai Cao

Michigan State University

1.1 Introduction

Biometric traits, such as palmprint (Duta et al. 2002) and fingerprint (Jain et al. 1997), refer to distinctive anatomical and behavioral characteristics for automatic human identification. Fingerprints, which are ridge and valley patterns on the tip of a human finger, are one of the most important biometric traits due to their known uniqueness and persistence properties (Maltoni et al. 2009). Since the advent of fingerprints for identifying and tracing criminals in Argentina in 1893 (Hawthorne 2008), fingerprints have been primarily used as evidence in law enforcement and forensics. After the first paper on automated fingerprint matching was published by Mitchell Trauring (Trauring 1963) in *Nature* in 1963, the Federal Bureau of Investigation (FBI) installed the first Automated Fingerprint Identification System (AFIS) in the 1980. Now large scale fingerprint recognition systems are not only used worldwide by law enforcement and forensic agencies, they are also beginning to be deployed in civilian applications, such as (i) the OBIM (formerly the US-VISIT) program by the Department of Homeland Security (Department of Homeland Security 2014), and (ii) India's Aadhar project (Planning Commission, Government of India 2014). In 2013, the TouchID system (Apple, Inc. 2014) in the Apple iPhone 5s for authenticating mobile phone users launched the application of fingerprint in mobile devices. Some major milestones in the history of fingerprint recognition are illustrated in Fig.1.1.

The purported uniqueness of fingerprints is characterized in terms of three levels of features (Maltoni et al. 2009) (see Fig. 1.2). Level 1 features include the general ridge flow and pattern configurations such as pattern type, ridge orientation and frequency fields, and singular points (core and delta points). While level 1 features are not sufficient for individualization, they can be used for exclusion (the outcomes of comparing a fingerprint pair are one of three possibilities: match, inconclusive and exclusion). Level 2 features mainly

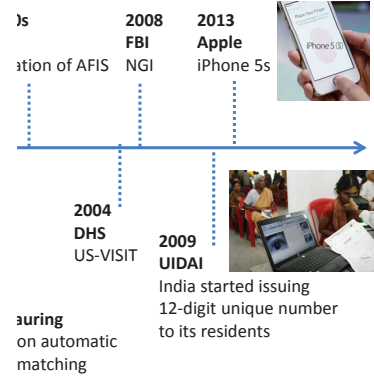


Figure 1.1: Some major milestones in fingerprint recognition.

refer to local ridge discontinuities, called minutia points; ridge endings and ridge bifurcations are the two most prominent types of minutia points. Level 3 features cover all other attributes at a fine level, such as width, shape, curvature and edge contours of ridges, pores, and incipient ridges. Level 3 feature extraction requires that the fingerprint images be acquired at a 1000 ppi resolution. Among all these fingerprint features, the set of minutia points (called minutiae) is regarded as the most distinctive and hence, is the most commonly used feature in fingerprint identification, both by human experts and AFIS.

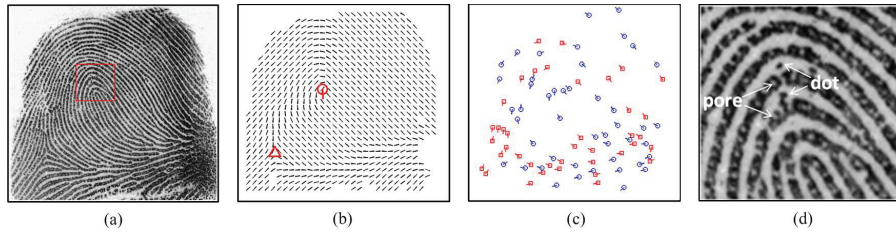


Figure 1.2: Illustration of fingerprint features at three different levels. (a) A gray-scale fingerprint image (NIST SD30, A002.01), (b) level 1 features: orientation field and singular points (core point shown as a circle and delta point shown as a triangle), (c) level 2 features: ridge ending minutiae (red squares) and ridge bifurcation minutiae (blue circles) and (d) level 3 features: pores and dots.

Based on the fingerprint image acquisition method and their source, fingerprints can be classified into three types, namely rolled, plain and latent (see Fig. 1.3). Rolled fingerprints are obtained by taking the impression from “nail to nail” in order to capture the complete ridge details of a finger. Plain fingerprints are acquired by pressing a fingertip onto a flat surface of either a paper for the inking method or a flatbed of a live-scan device (Maltoni et al. 2009). Latent fingerprints (or simply latents) refer to fingerprints lifted from the surfaces of objects, which are inadvertently touched or handled by a person. Compared

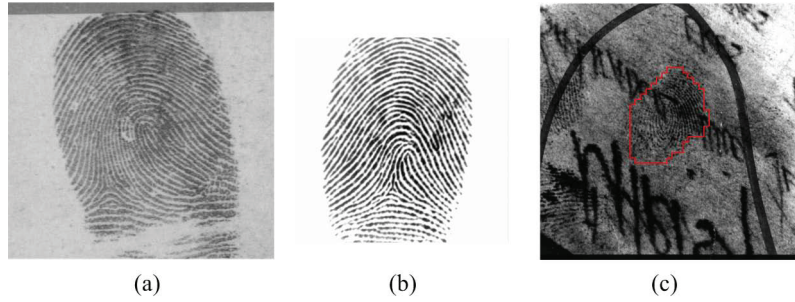


Figure 1.3: Three types of fingerprint images. (a) Rolled fingerprint (from NIST Special Database 4 (2014)), (b) plain fingerprint from (FVC2002 (2002)), and (c) latent fingerprint (from NIST Special Database 27 (2014)) with foreground (friction ridge pattern) highlighted by red curve.

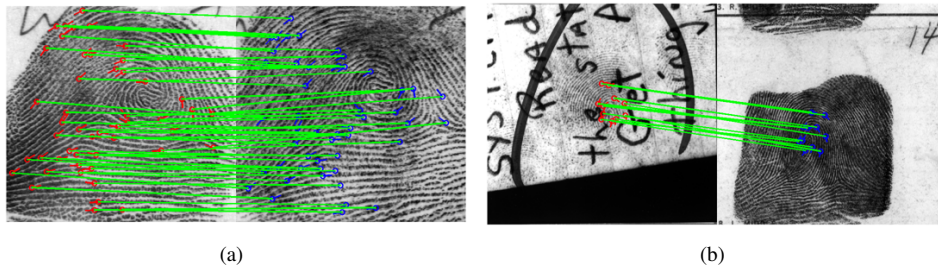


Figure 1.4: Examples of (a) rolled to rolled fingerprint matching and (b) latent to rolled fingerprint matching. Features in the rolled fingerprints shown here are extracted automatically by an AFIS, but features (minutiae, region of interest and singular points) in the latent were manually marked.

to rolled and plain fingerprints (or collectively called exemplar fingerprints) which are typically acquired in the presence of an officer or trained personnel, latents are generally of poor quality with incomplete ridge structure, background noise and non-linear distortion. Consequently, the accuracy of latent-to-exemplar matching is significantly lower than that of exemplar-to-exemplar matching. In NIST evaluations, the best performing AFIS achieved a true acceptance rate of 99.4% at a false acceptance rate of 0.01% for exemplar-to-exemplar fingerprint matching (Wilson et al. 2004). The best performing commercial latent matcher could only achieve a rank-1 identification rate of 63.4% in searching 1,114 latents against a background database containing 100,000 exemplar prints (Indovina et al. 2012). The search for the source of a latent is a challenging problem in terms of both the algorithmic efficiency and identification accuracy, especially when the reference or exemplar database (rolled or plain fingerprints) is extremely large. Fig. 1.4 shows examples of rolled-to-rolled fingerprint matching and latent-to-rolled fingerprint matching.

Given the difficulty of automatic latent matching, human intervention is unavoidable in

order to assess the value of latents as forensic evidence, mark features such as region of interest (ROI) and minutiae, and make a decision whether the latent has a match in the reference database given the candidate list (typically top 50 matches) generated by AFIS. Hence, latent examiners and AFIS work collaboratively in a framework called Analysis, Comparison, Evaluation and Verification (ACE-V) (Ashbaugh 1999). However, human involvement in latent examination has raised some concerns related to repeatability and reliability (Ulery et al. 2011, 2012). Furthermore, when the comparison time (between latent and exemplar prints) is limited, latent examiners are more likely to make an inconclusive matching decision (Dror et al. 2011). One of the priorities in the FBI's Next Generation Identification (NGI) program is to support the development of a lights-out¹ capability for latent identification (FBI- NGI 2014). An essential component necessary for achieving the lights-out capability is automatic feature extraction from latent fingerprints, this is necessary to (i) increase the throughput of latent matching systems, (ii) achieve repeatability of latent feature extraction and, (iii) improve the compatibility between features extracted in the latents and in the exemplar prints (Feng et al. 2013).

In order to achieve reliable feature extraction from latents, the latent images need to go through two main preprocessing steps: (i) segmentation to separate friction ridges from noisy background and (ii) fingerprint enhancement to enhance ridge and valley structures. Directional filtering, such as Gabor filtering (Hong et al. 1998), can adaptively improve the clarity of ridge and valley structures; the filters are tuned based on the local ridge orientation and frequency. Therefore, for latent enhancement it is essential to first obtain a good estimate of ridge orientation and frequency fields.

There is a rich body of literature on fingerprint segmentation (Chikkerur et al. 2007; Hong et al. 1998), orientation field estimation (Chikkerur et al. 2007; Hong et al. 1998; Mardia et al. 1997; Wang et al. 2007) and frequency field estimation (Chikkerur et al. 2007; Jiang 2000) for exemplar fingerprints. But these approaches do not work well on latent fingerprints since they did not consider (i) presence of structured noise, such as lines, markings, characters and speckles (see Fig. 1.3 (c)), which break the friction ridge pattern and hinder reliable feature extraction; and (ii) unclear fingerprint ridges in the foreground area. Some approaches have been proposed to specifically address the problem of latent fingerprint segmentation (Karimi-Ashtiani and Kuo 2008; Short et al. 2011; Zhang et al. 2013) and enhancement (Yoon et al. 2011). However, none of these approaches use the prior knowledge of ridge structure in fingerprints, resulting in only a marginal improvement in latent matching.

A dictionary, which is a set of words (or vectors) used to sparsely and linearly represent signals of the same dimension (called sparse coding), has been successfully applied to a number of signal processing problems, such as image denoising (Elad and Aharon 2006; Mairal et al. 2008b), classification (Lian et al. 2010; Mairal et al. 2008a), and face recognition (Liao et al. 2013; Wright et al. 2009). The dictionary learned from a set of training data is a collection of representative vectors of the training data. In this chapter, we investigate the use of dictionaries for the challenging problems in latent fingerprint image analysis, namely latent fingerprint segmentation and enhancement. Given that fingerprint patterns can be represented at two different levels (i.e., coarse representation for fingerprint ridge flow or orientation field, and fine representation for ridges and valleys), two dictionaries are developed: an

¹Lights-out identification refers to an AFIS requiring minimal or no human assistance in which a query fingerprint image is presented as input, and the output consists of a short candidate list (Indovina et al. 2009).

orientation patch² dictionary (Feng et al. 2013) and a ridge structure dictionary (Cao et al. 2014). An orientation patch dictionary, which contains only the orientation information in patches, is proposed to estimate the orientation field for latent fingerprint enhancement. A ridge structure dictionary, which contains ridge and valley patterns, is proposed for latent fingerprint segmentation (locating friction ridge pattern) and enhancement by estimating orientation and frequency fields. Experimental results on public domain latent fingerprint databases show that the dictionaries, learned from a large number of fingerprints, capture a domain-specific knowledge which is effective in improving the accuracy of latent fingerprint matching.

The rest of the chapter is organized as follows. In section 1.2, the method of dictionary construction for fingerprint patterns at orientation level and ridge level is described. The orientation patch dictionary for orientation field estimation in latent fingerprints is presented in section 1.3. The ridge structure dictionary for latent segmentation and enhancement is introduced in section 1.4. Conclusions and future research directions are described in section 1.5.

1.2 Dictionary Construction

Orientation patch and ridge structure dictionaries are both constructed offline from high-quality fingerprints to capture prior knowledge about fingerprint patterns.

1.2.1 Orientation Patch Dictionary Construction

To construct a dictionary of reference orientation patches, we used a set of 50 high quality fingerprints (referred to as reference fingerprints) in the NIST SD4 database (NIST Special Database 4 2014). All five major pattern types (plain arch, tented arch, left loop, right loop, and whorl) are covered by the reference fingerprints. The high-quality fingerprints are manually selected and their orientation fields (with block size 16×16 pixels) are estimated using VeriFinger 6.2 SDK (Neurotechnology Inc. 2012). A number of training orientation patches are obtained by sliding a window (of size $b \times b$ blocks) across the orientation field and its mirrored version for each reference fingerprint, where an orientation patch consists of $b \times b$ orientation elements and an orientation element refers to the dominant orientation in a block of size 16×16 pixels. Each orientation patch is rotated by 21 different angles $\{i \cdot 5^\circ, -10 \leq i \leq 10\}$ to generate additional training orientation patches to cover all possible directions in the latent fingerprints.

Given the training orientation patches, the orientation patch dictionary (shown in Fig. 1.5) is constructed by a *greedy* algorithm which is described below:

1. The first orientation patch in the training set is added to the dictionary, which is initially empty.
2. The next orientation patch in the training set, which is sufficiently different from all orientation patches in the dictionary, is added to the dictionary. The similarity measure between two orientation patches of size $b \times b$ is computed as n_s/b^2 , where n_s denotes the number of orientation elements whose difference is less than 10 degrees.

²An orientation patch refers to the block-wise orientation field of a fingerprint patch.

3. Repeat step 2 until all orientation patches have been considered.

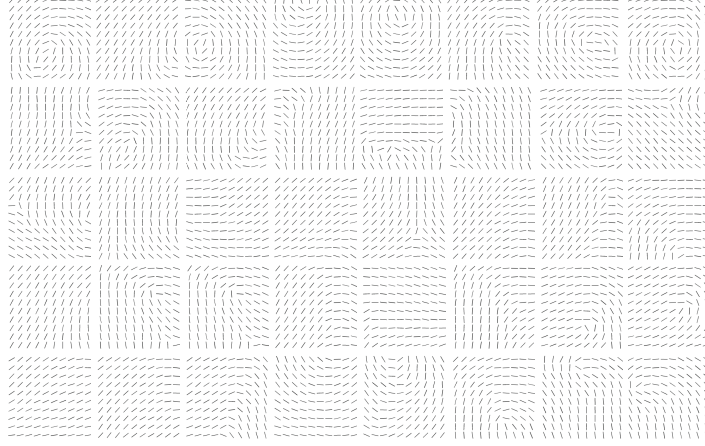


Figure 1.5: Examples of orientation patches in the dictionary; an orientation patch contains 10×10 orientation elements, and each orientation element corresponds to a block of 16×16 pixels.

The number of reference orientation patches in the dictionary depends on the number of reference fingerprints and the patch size. When the patch size is 10×10 blocks and 50 reference fingerprints are used, the number of reference orientation patches in the dictionary is around 23,000. While a larger size of the orientation patch is better for correcting errors in the initial orientation field, it would require a larger dictionary, which takes more time to search. To further demonstrate the impact of patch size, orientation fields corrected using different patch sizes are compared in Fig. 1.6, where an initial orientation patch is directly replaced by its closest orientation patch in the dictionary without considering compatibility between neighboring patches. The performance of the dictionary based approach improves with an increase in patch size. The estimation errors close to the fingerprint boundary are due to border effect (these patches contain very few foreground blocks with a friction ridge pattern).

1.2.2 Ridge Structure Dictionary Construction

The orientation patch dictionary characterizes ridge flow patterns of fingerprints which can be used for correcting an initial orientation field. However, the structure of ridges and valleys is ignored in the orientation patch dictionary. To remedy this, a ridge structure dictionary, which is learned directly from the fingerprint image patches, is introduced in this section. A large size of image patches will result in high dimensionality (a 4,096-dimensional vector for a 64×64 image patch, compared to 100-dimensional vector for a 10×10 orientation patch which covers a 160×160 image patch) and hence, a large dictionary. On the other hand, image patches of small size are not robust to structured background noise in the latent

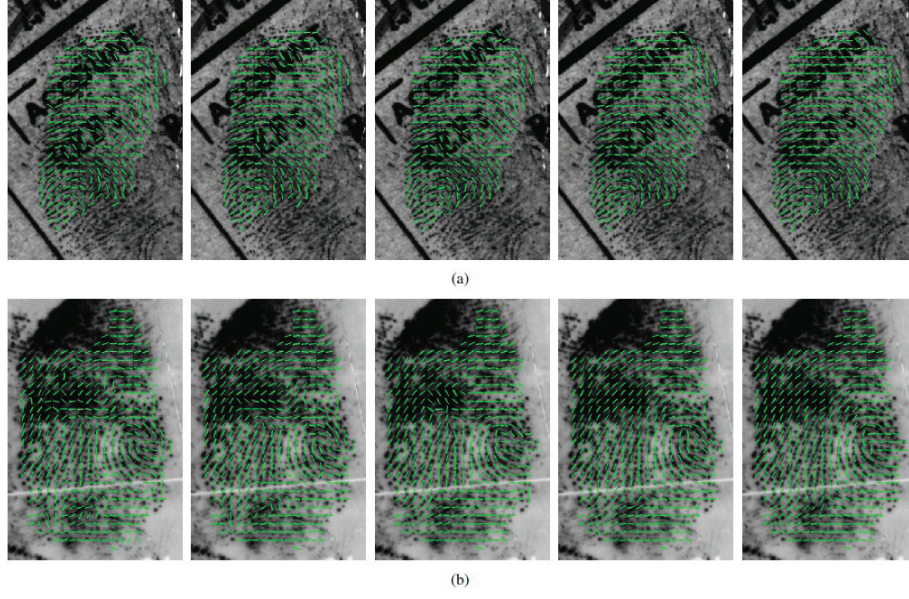


Figure 1.6: Orientation fields extracted from two latent fingerprint impressions ((a) and (b)) estimated using different patch sizes (increasing from left to right: 3×3 , 5×5 , 7×7 , 9×9 and 11×11). Only the nearest dictionary element of each initial orientation patch is considered here. Image courtesy of Feng et al. (2013).

images. Hence, two levels of ridge structure dictionaries are constructed: (i) a coarse-level dictionary with large patch size of 64×64 pixels, and (ii) 16 fine-level (orientation specific) dictionaries with small patch size of 32×32 pixels. The patch size for fine-level dictionaries is 32×32 pixels which covers about two ridges and valleys for 500ppi fingerprints and is robust to structured noise. The patch size for the coarse-level dictionary is twice the size of the fine-level dictionary to cover additional ridge structures.

Training Set Selection

A large number of high quality fingerprint patches from rolled fingerprints in NIST SD4 (NIST Special Database 4 2014) are selected for the dictionary construction as follows:

1. High quality fingerprint selection: NIST Fingerprint Image Quality (NFIQ) (Tabassi et al. 2004) is used to select 500 fingerprints of high quality³ (i.e., $\text{NFIQ} < 3$) in NIST SD4.
2. High quality patch selection: The block-wise orientation field and ridge quality map of the selected fingerprints are computed by MINDTCT (Garris et al. 2004). For the coarse-level dictionary, an image patch of size 64×64 pixels is included into the training set P^c if the average quality value of the image patch is larger than a predefined

³NFIQ ranges from 1 to 5, with 1 indicating the highest quality and 5 indicating the lowest quality fingerprint.

threshold T (T is set to 3.75, where the block ridge quality in MINDTCT ranges from 0 (the lowest quality) to 4 (the highest quality)). For the fine-level dictionary with the i -th orientation ($i = 1, \dots, 16$) orientation specific dictionary, an image patch is included in the training set P_i^f if it satisfies the following two conditions: (i) average quality value of the patch is larger than T , and (ii) average ridge orientation of the patch is within the range $[(i-1) \times \frac{\pi}{16}, i \times \frac{\pi}{16})$.

3. Vector normalization: Each patch p in the training sets is converted to a vector by concatenating the rows and normalized with mean of zero and standard deviation of one.

Let $P^c = \{p_j^c\}_{j=1}^{N^c}$ be the training set with N^c training patches for the coarse-level dictionary, and $P_i^f = \{p_{i,j}^f\}_{j=1}^{N_i^f}$, $i = 1, \dots, 16$, be the training set for the i -th fine-level dictionary, where N_i^f denotes the number of training patches for the i -th fine-level dictionary specified by ridge orientation. We then randomly select 80,000 image patches from P^c and 10,000 image patches from each P_i^f for dictionary learning.

Dictionary Learning

The goal of dictionary learning is to construct a dictionary D of size $N_P \times N_D$ that provides the best sparse representation for each patch in $P = \{p_j\}_{j=1}^N$, where N_P is the dimensionality of the patches in P , and N_D is the number of elements in the dictionary D . After the ridge dictionaries are constructed by K-SVD (Aharon et al. 2006), each dictionary element is normalized with mean of zero and standard deviation of one.

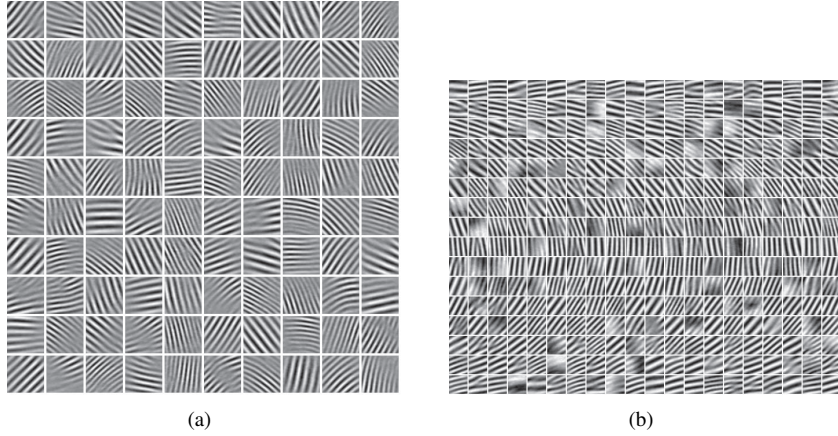


Figure 1.7: Examples of coarse and fine level dictionaries. (a) A subset of elements in the coarse-level dictionary, and (b) a subset of elements in the 16 orientation-specific dictionaries. The i -th row in (b) corresponds to the orientation range $[(i-1) \times \frac{\pi}{16}, i \times \frac{\pi}{16})$, $i=1, \dots, 16$.

A total of 17 different dictionaries are constructed by taking subsets selected from P^c and P_i^f , $i = 1, \dots, 16$ as the training sets. The number of elements N_D^c in the coarse-level

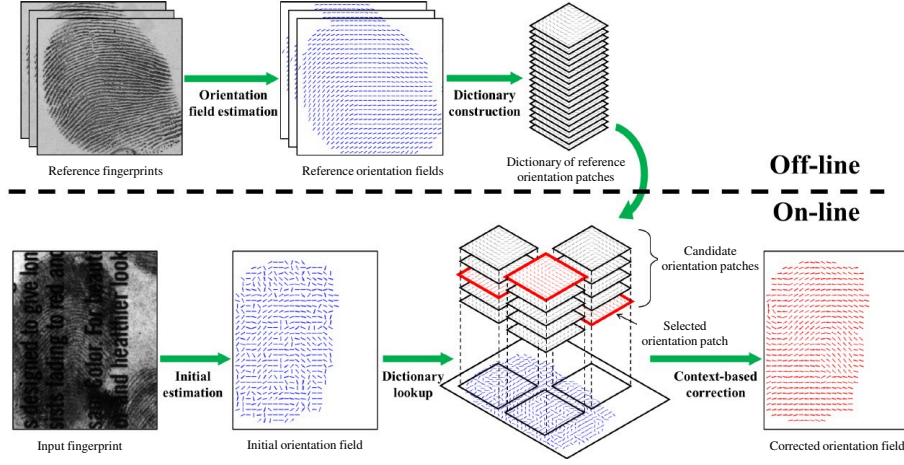


Figure 1.8: Flowchart of the orientation field estimation algorithm which consists of an off-line dictionary construction stage and an on-line orientation field estimation stage. Image courtesy of Feng et al. (2013).

dictionary is set to 1,024, and the total number of elements N_D^f in each fine-level dictionary is set to 64. Fig. 1.7 (a) shows a subset of dictionary elements in the coarse-level dictionary D^c and Fig. 1.7 (b) shows a subset of dictionary elements in the 16 fine-level dictionaries D_i^f .

1.3 Orientation Field Estimation Using Orientation Patch Dictionary

Given the orientation patch dictionary learned in section 1.2.1, the orientation field in the latent foreground is estimated using the following three steps (see Fig. 1.8):

1. *Initial estimation*: The initial orientation field is obtained using a local orientation estimation method, such as local Fourier analysis (Jain and Feng 2009).
2. *Dictionary lookup*: The initial orientation field is divided into overlapping patches. For each initial orientation patch, its six nearest orientation patches in the dictionary are selected as candidates for replacing the noisy initial orientation patch.
3. *Context-based correction*: The optimal combination of candidate orientation patches is found by minimizing the energy function which includes the similarity between an initial orientation patch and its selected reference orientation patch in the dictionary and the compatibility between neighboring reference orientation patches.

Details of the orientation field estimation algorithm are presented in the following sections.

1.3.1 Initial Orientation Field Estimation

The initial orientation field (16×16 pixel block size) is obtained by detecting a peak in the magnitude spectrum of the local image (Jain and Feng 2009). Other local estimation

approaches should also suffice for this initial estimation. Although the initial orientation field is typically very noisy due to the poor quality of latents, it should not be smoothed at this stage since the correct orientation elements may be degraded by noise in the neighboring regions. The initial orientation field is updated in the later stages by utilizing prior knowledge of fingerprints contained in the orientation patch dictionary.

1.3.2 Dictionary Lookup

Given an initial orientation patch that contains at least one foreground block, a number of candidate reference orientation patches from the dictionary are retrieved based on their similarity with the initial orientation patch. The similarity $S(\Theta, \Phi)$ between an initial orientation patch Θ and a reference orientation patch Φ is defined as

$$S(\Theta, \Phi) = n_s / n_f, \quad (1.1)$$

where n_f is the number of orientation elements in the initial orientation patch and n_s is the number of orientation elements whose differences are less than a predefined threshold (empirically set as $\pi/12$). However, for many initial orientation patches, the top candidate orientation patches of an initial orientation patch are quite similar to each other. In order to increase the probability of including the correct reference orientation patches in a short list, it is better to select a set of diverse candidates. A diverse set of n_c (empirically set as 6) candidates is selected from the top $10n_c$ initial candidates using the following greedy strategy:

1. Choose the first initial candidate orientation patch.
2. The next candidate patch is compared to each of the chosen candidates. If its similarity to all the chosen candidates is less than a predefined threshold (empirically set as 0.8), it is included in the list. Note that similarity is computed using only the foreground blocks in the initial orientation patch.
3. Repeat step 2 for all the initial candidates until n_c candidates have been chosen or all initial candidates have been checked.

As a result of the diversity heuristic, the correct orientation patch is more likely to appear in the candidate list, even if the initial orientation patch is very noisy or incomplete.

1.3.3 Context-based Orientation Field Correction

After dictionary lookup, we obtain a list of c_i ($1 \leq c_i \leq n_c$) candidate orientation patches, $\Phi_i = \{\Phi_{i,1}, \Phi_{i,2}, \dots, \Phi_{i,c_i}\}$, for an initial orientation patch Θ_i . Let r_i denote the index of the selected candidate for the patch i , and $\mathbf{r} = \{r_1, r_2, \dots, r_{n_p}\}$ be the vector of the indices of the selected candidates for all n_p foreground patches. Any combination of candidate indices could be a solution for the orientation field estimation. But, this leads to a large solution space, so we utilize contextual information to reduce the search space.

We address this problem by searching for optimal indices vector, \mathbf{r}^* , which minimizes an energy function $E(\mathbf{r})$. Choice of a proper energy function is crucial for the success of this method. We consider two factors in designing the energy function: (i) the similarity between the reference orientation patches and the corresponding initial orientation patches, and (ii)

the compatibility between neighboring reference orientation patches. The proposed energy function $E(\mathbf{r})$ is defined as

$$E(\mathbf{r}) = E_s(\mathbf{r}) + w_c E_c(\mathbf{r}) \quad (1.2)$$

where $E_s(\mathbf{r})$ and $E_c(\mathbf{r})$ denote the similarity term and compatibility term, respectively, and w_c (empirically set to 1 by the authors in (Feng et al. 2013)) is the weight of compatibility term. The similarity term is defined as

$$E_s(\mathbf{r}) = \sum_{i \in \mathcal{V}} (1 - S(\Theta_i, \Phi_{i,r_i})), \quad (1.3)$$

where \mathcal{V} denotes the set of foreground patches and $S(\cdot)$ is defined in Eq. (1.1). The compatibility term is defined as

$$E_c(\mathbf{r}) = \sum_{(i,j) \in \mathcal{N}} (1 - C(\Phi_{i,r_i}, \Phi_{j,r_j})), \quad (1.4)$$

where \mathcal{N} denotes the set of adjacent foreground patches which are four-connected neighbors.

The compatibility between two neighboring orientation patches Φ_{i,r_i} and Φ_{j,r_j} is measured by the similarity of orientations in the overlapping blocks. Let $\{\alpha_n\}_{n=1}^{N_o}$ and $\{\beta_n\}_{n=1}^{N_o}$ be the set of orientations in the N_o overlapping blocks of two orientation patches. The compatibility is computed as

$$C(\Phi_{i,r_i}, \Phi_{j,r_j}) = \frac{1}{N_o} \sum_{n=1}^{N_o} |\cos(\alpha_n - \beta_n)|. \quad (1.5)$$

To minimize the energy function in Eq. (1.2), the well-known loopy belief propagation algorithm (Blake et al. 2011) is used for optimization.

1.3.4 Experiments

The goal of an orientation field estimation algorithm is to obtain an accurate estimation of fingerprint orientation field for fingerprint enhancement and feature extraction and then to improve the fingerprint matching accuracy. The dictionary based algorithm is, therefore, evaluated in terms of the accuracy of orientation field estimation and the accuracy of fingerprint matching, respectively. The latent orientation field estimation and subsequent matching experiments are conducted on NIST SD27 (NIST Special Database 27 2014), which contains 258 latent fingerprint images (500 ppi). These latents have been classified into three different qualities, namely “Good”, “Bad” and “Ugly” (very bad), and their corresponding mated rolled fingerprints. The numbers of “Good”, “Bad” and “Ugly” latents are 88, 85 and 85, respectively. Fig. 1.9 displays examples of latent with these three qualities. To make the latent matching problem more realistic and challenging, 27,000 rolled fingerprints (file fingerprints) in the NIST SD14 database were also included in the background database.

In addition to the proposed orientation field estimation algorithm, two other approaches were included for comparison:

1. FOMFE: Combination of gradient-based local estimation and FOMFE-based global model (Wang et al. 2007).

2. STFT: Combination of STFT-based local estimation and low-pass filtering (Chikkerur et al. 2007).

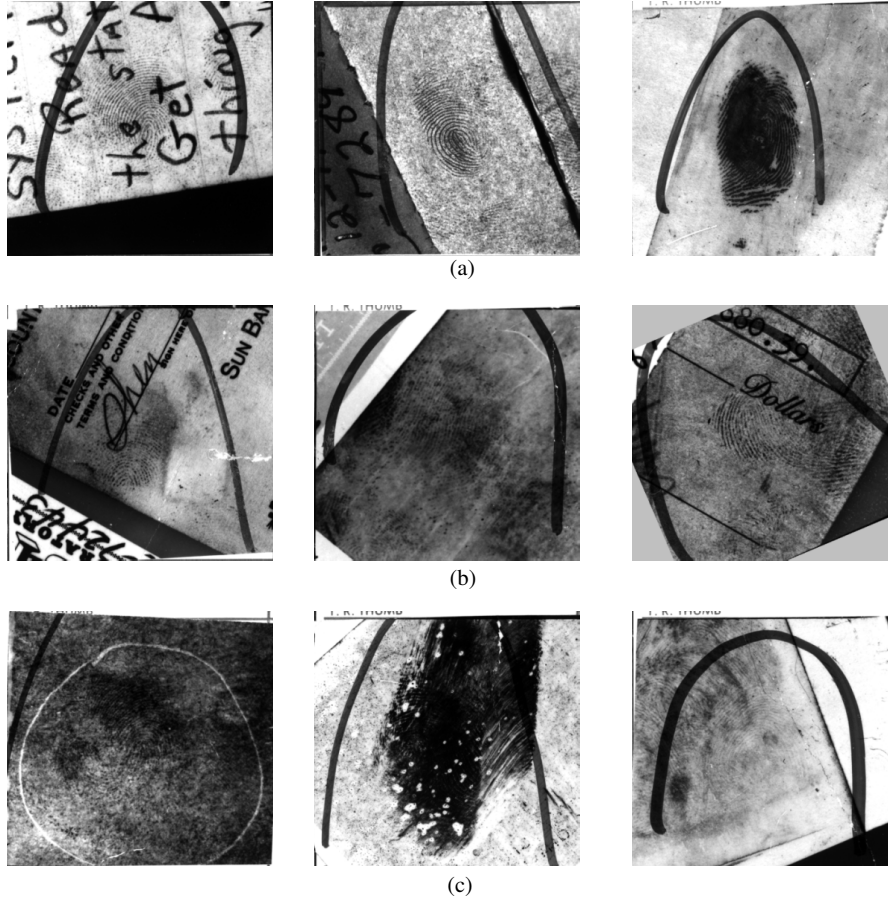


Figure 1.9: Examples of latents of different qualities. (a) Good, (b) bad and (c) ugly.

The accuracy of orientation field estimation algorithms is measured in terms of the average Root Mean Square Deviation (RMSD) (Turrone et al. 2011). The ground truth orientation fields were manually marked by one of the authors in Feng et al. (2013). Average RMSD of the dictionary based algorithm, FOMFE and STFT are computed on all the 258 latents in the NIST SD27 database and also on the three quality level subsets (Good, Bad and Ugly). Table 1.1 shows that the dictionary based algorithm outperforms FOMFE and STFT for all three subsets of latent fingerprints in NIST SD27.

In order to evaluate the matching performance, latent fingerprints are enhanced using a Gabor filter (Hong et al. 1998) whose frequency parameter is fixed at $1/9$ cycles

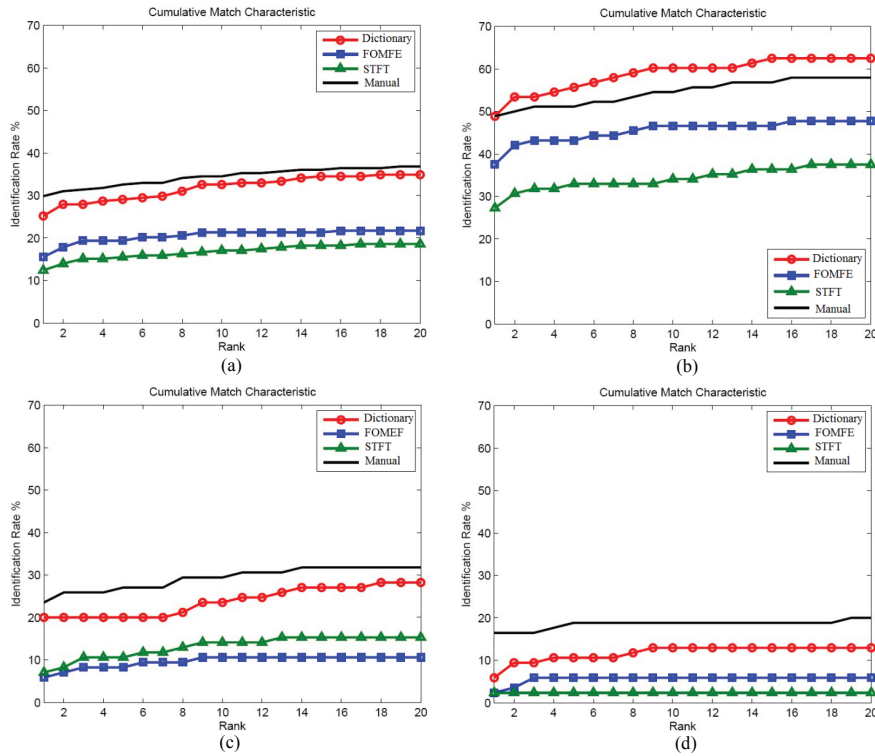


Figure 1.10: CMC curves of three orientation field estimation algorithms and the manual markup of orientation field on the NIST SD27 latent database: (a) all (258 latents), (b) good quality (88 latents), (c) bad quality (85 latents), and (d) ugly quality (85 latents). Images courtesy of Feng et al. (2013).

Table 1.1 Average estimation error (in degrees) of the orientation field estimation algorithm based on orientation patch dictionary and two competing algorithms on the latent fingerprints in the NIST SD27 Database

Algorithm	All	Good	Bad	Ugly
Orientation Patch Dictionary (Feng et al. 2013)	18.44	14.40	19.18	21.88
FOMFE (Wang et al. 2007)	28.12	22.83	29.09	32.63
STFT (Chikkerur et al. 2007)	32.51	27.27	34.10	36.36

per pixel, orientation parameter is tuned to the estimated orientation field, and standard deviations of the Gaussian envelope in x and y directions are fixed at 4. VeriFinger SDK 6.2 (Neurotechnology Inc. 2012) is used for feature extraction and matching.

Fig. 1.10 shows the Cumulative Match Characteristic (CMC) curves obtained from the three orientation field estimation algorithms and the manual markup (ground truth). The dictionary based algorithm consistently outperforms the other two algorithms (FOMFE and

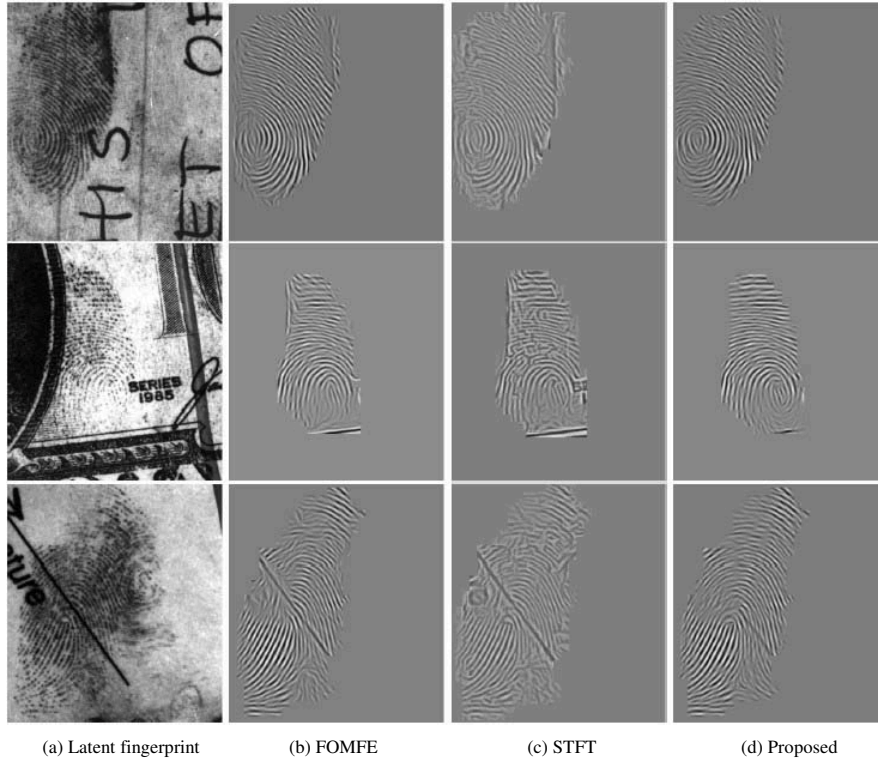


Figure 1.11: Enhanced images of three latent fingerprints in (a) using orientation fields estimated by (a) FOMFE, (b) STFT, and (c) the orientation patch dictionary based algorithm. Images courtesy of Feng et al. (2013).

STFT) on latents of all three quality levels. Three examples given in Fig. 1.11 compare the enhanced latent fingerprints using the orientation fields obtained by the dictionary based algorithm, FOMFE and STFT.

For many latents of good quality, the dictionary based algorithm outperforms the manual markup (see Fig. 1.10(b)). Our analysis of these examples demonstrates that the dictionary based algorithm has smaller deviation from true ridge orientation for good quality latents because it is difficult and time consuming for a fingerprint expert to accurately mark the complete orientation field in a latent. Manual markup still performs better on bad and ugly quality latents.

1.4 Latent Segmentation and Enhancement Using Ridge Structure Dictionary

Given the learned ridge structure dictionaries (section 1.2.2), latent segmentation and enhancement consists of the following steps (see Fig. 1.12):

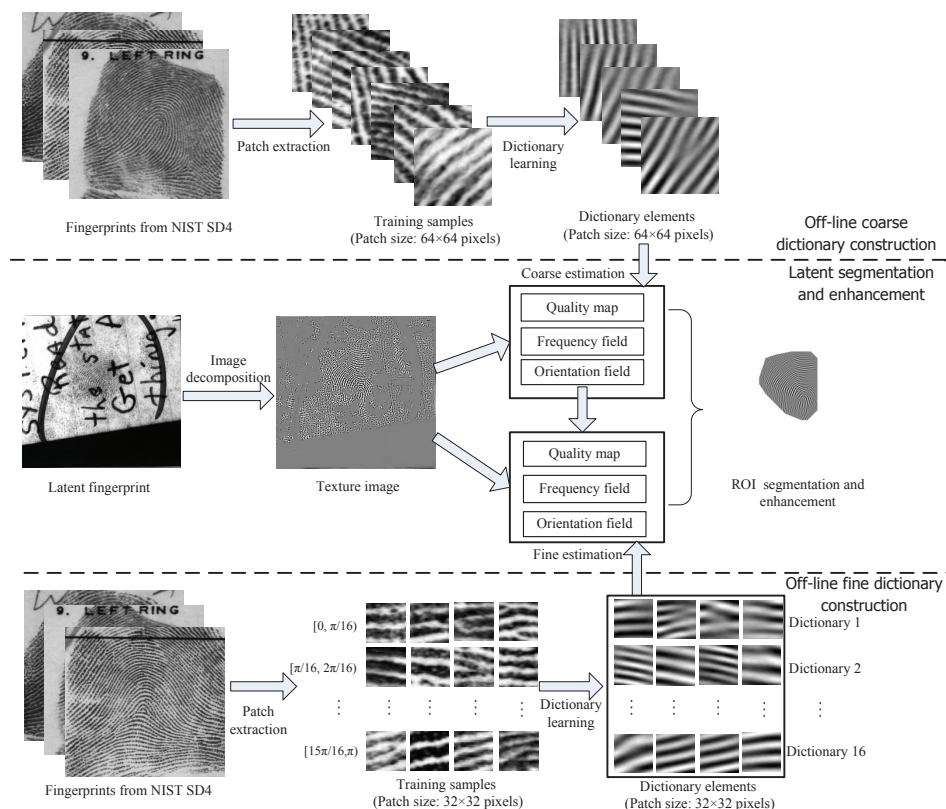


Figure 1.12: Overview of latent segmentation and enhancement algorithm based on ridge structure dictionary. The offline dictionary learning (top and bottom) and online latent segmentation and enhancement stage (middle) are shown.

1. *Decomposition*: Input latent is decomposed into cartoon and texture images using local total variations (Buades et al. 2010); the cartoon image which primarily consists of structured noise is discarded.
2. *Coarse-level estimation*: The coarse-level dictionary is used to estimate orientation and frequency fields on the texture image and assess coarse-level quality of the latent.
3. *Fine-level estimation*: Using coarse-level orientation field, select one fine-level dictionary out of the 16 fine-level dictionaries for each image patch in the texture image; this gives fine-level orientation and frequency fields and ridge quality map.
4. *Segmentation and enhancement*: The coarse-level quality map and fine-level quality map are fused for latent segmentation. In the foreground of texture image, a Gabor filter tuned to the orientation and frequency fields obtained in steps 2 and 3 is applied for latent enhancement.

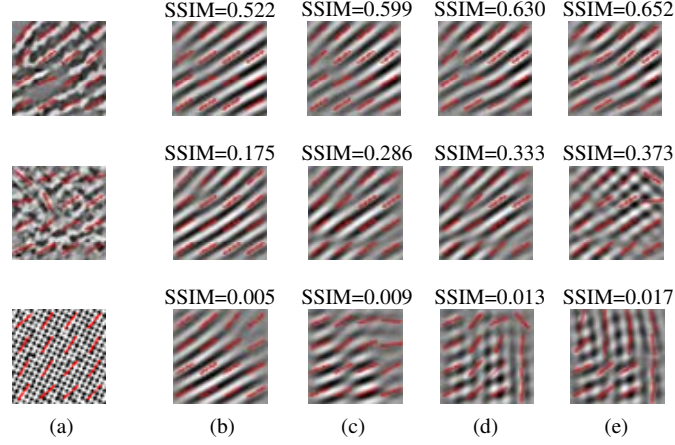


Figure 1.13: Patch reconstruction results (overlaid on orientation field) with different number of dictionary entries, T_1 . (a) Texture component of a high quality fingerprint patch (top), low quality fingerprint patch (middle), and non-fingerprint patch (bottom), (b), (c), (d) and (e) are the reconstruction results when $T_1 = 1, 2, 3, 4$, respectively. The SSIM indices between the given patch (column (a)) and the reconstructed patch with different value of T_1 are shown above the reconstructed patches.

1.4.1 Latent Image Decomposition

A latent fingerprint image, f , is decomposed into a cartoon (piece-wise smooth) image and a texture (oscillatory) image. The texture image primarily includes the ridge structure patterns, so it is kept for further latent segmentation and enhancement, while the cartoon image, viewed as structured noise, is discarded. We adopt the nonlinear decomposition method based on local total variation (LTV) proposed by Buades et al. (2010). Fig. 1.14(b) shows the texture component of three different latent images shown in Fig. 1.14(a); most of the structured noise in latents has been successfully removed by excluding the cartoon image and only the friction ridge pattern is retained in the texture image.

1.4.2 Coarse Estimates of Ridge Quality, Orientation and Frequency

Sparse Coding and Patch Quality

The texture image is divided into overlapping patches of size 64×64 pixels (P_L^c). Each patch has 64×48 or 48×64 overlapping pixels, with each of its four-connected neighboring blocks. Each patch $p \in P_L^c$ is converted to a vector by row concatenation and normalized with mean of zero and standard deviation of one. The sparse code α of p with respect to coarse-level dictionary D^c is obtained using orthogonal matching pursuit (Mallat and Zhang 1993). In general, the reconstructed patch \hat{p} is close to p if p is a fingerprint patch. In order to measure the similarity between \hat{p} and p , we have used the structural similarity index $\text{SSIM}(p, \hat{p})$ (Wang et al. 2004).

Fig. 1.13 compares the reconstructed patches using different values of dictionary elements and SSIM indices for two fingerprint patches (top and middle rows) and one non-fingerprint patch (bottom row). We observe that the value of SSIM indicates the quality of a patch in terms of fingerprint ridges. The quality of patch p , Q_p , is therefore defined by

$$Q_p = \text{SSIM}(p, \hat{p}). \quad (1.6)$$

A single dictionary element is selected for each image patch for the reconstruction. This is because (i) with just one element, the sparse code and SSIM index are easy to compute, and (ii) the orientation and frequency fields of \hat{p} can be computed off-line since \hat{p} is simply one of dictionary elements. Fig. 1.14(c) shows some examples of coarse-level quality maps when a single dictionary element is retrieved for reconstruction.

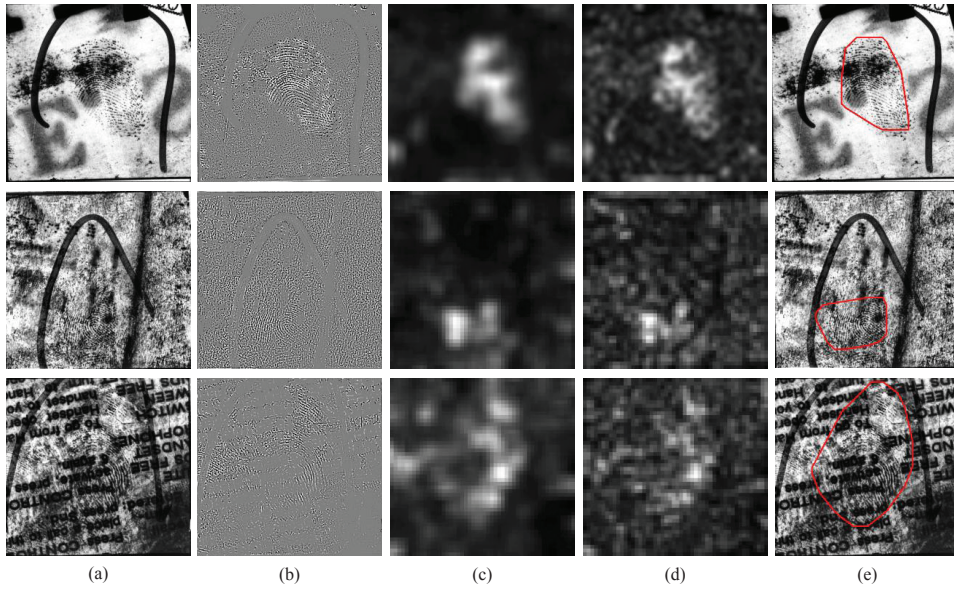


Figure 1.14: Illustration of latent fingerprint segmentation. (a) Gray-scale latent images, (b) texture component images, (c) coarse-level quality maps, (d) fine-level quality maps, and (e) segmentation results shown overlaid on the gray-scale latent images. The top, middle and bottom latent fingerprints in column (a) are of good, bad and ugly quality in NIST SD27. The contrast of the middle and bottom latent fingerprints has been adjusted for better visual quality.

Ridge Quality Map, and Orientation and Frequency Fields Estimation

As shown in Fig. 1.13, the reconstructed patches have better ridge quality. The orientation field and frequency field of patch p in the latent image can be obtained from the reconstructed patch \hat{p} . For a block b (16×16 pixels) in the latent covered by multiple patches, let

$\{q_i, \theta_i, f_i\}$ be the ridge quality, orientation and frequency of the i -th patch covering the block b . The coarse estimates of ridge quality Q_b^c , orientation θ_b^c and frequency f_b^c for block b are computed as:

$$Q_b^c = \frac{1}{n_b} \sum_{i=1}^{n_b} q_i, \quad (1.7)$$

$$\theta_b^c = \frac{1}{2} \tan^{-1} \left(\frac{\sum_{i=1}^{n_b} q_i \sin 2\theta_i}{\sum_{i=1}^{n_b} q_i \cos 2\theta_i} \right), \quad (1.8)$$

$$f_b^c = \frac{1}{\sum_{i=1}^{n_b} q_i} \sum_{i=1}^{n_b} q_i f_i, \quad (1.9)$$

where n_b is the number of patches covering the block b . In this case, a higher weight (q_i) is assigned to the patch with better ridge and valley structures. Other elaborate weighting strategies may lead to better results.

1.4.3 Fine Estimates of Ridge Quality, Orientation and Frequency

While the coarse-level dictionary is robust to local noise, it cannot extract detailed ridge information. Instead, small patch size dictionaries can be used to compute the fine-level quality map and fine-level orientation and frequency fields. The texture image is divided into smaller overlapping patches of size 32×32 pixels (P_L^f). Each patch has 32×16 or 16×32 overlapping pixels with each of its four-connected neighboring blocks. All patches in P_L^f are normalized with mean of zero and standard deviation of one. For each patch $p \in P_L^f$, a dominant orientation θ is used to select the corresponding orientation specific dictionary D_k^f , where $k = \lceil \frac{16 \times \theta}{\pi} \rceil$ and $\lceil \cdot \rceil$ is the ceiling operator. The closest dictionary element \hat{p} to p in D_k^f is selected, and the quality of the patch p is determined by the SSIM index between p and \hat{p} . For each 16×16 block b in the latent, the fine-level quality Q_b^f , orientation θ_b^f and frequency f_b^f are obtained from the covering patches using Eqs. (1.7), (1.8) and (1.9).

1.4.4 Segmentation and Enhancement

The final quality map Q is computed as the average of the coarse-level quality map and fine-level quality map. Q is then normalized to the range $[0,1]$ and a global threshold T_Q , determined by Otsu's method (Otsu 1979), is used to binarize the normalized quality map. The blocks with normalized quality less than T_Q are regarded as background; otherwise foreground. Morphological operations (dilation and opening) are then applied to remove small foreground blocks and fill holes inside the foreground. Finally, the convex hull of the set of foreground blocks determines the final segmentation result. Fig. 1.14(e) shows the segmentation results for latent fingerprints in NIST SD27.

In the foreground region, the texture image of a latent obtained from the decomposition is enhanced by Gabor filtering (Hong et al. 1998), where the orientation and frequency parameters of the filter are tuned based on the fine-level orientation field (θ^f) and the average frequency of coarse-level frequency field and fine-level frequency field ($\frac{f^f + f^c}{2}$); the standard deviations of the Gaussian envelope in x and y directions are set to 4.

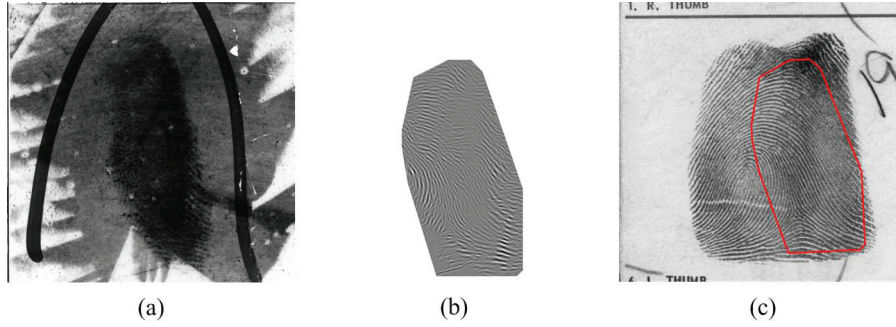


Figure 1.15: An example of latent segmentation and enhancement by the proposed algorithm. (a) A latent fingerprint (U286 from NIST SD27); (b) fully automatic segmentation of (a) by the proposed algorithm; (c) the true mate (rolled print) of (a) with the segmentation boundary in (c) outlined on the mate. By feeding the original latent in (a) and the segmented and enhanced latent in (b) into a commercial off-the-shelf (COTS) latent matcher (with a background database of 31,997 reference prints), the mated print is retrieved at ranks 4,152 and 2, respectively.

1.4.5 Experimental Results

Two latent databases are used for performance evaluation: NIST SD27 and the West Virginia University latent database (WVU DB) (iProBe 2014). The NIST SD27 contains 258 latent fingerprints with their mated rolled fingerprints. The WVU DB contains 449 latent fingerprints with their mated rolled fingerprints and an additional 4,290 rolled fingerprints. All these latent fingerprint images are 500 ppi images. The algorithm was implemented in MATLAB and C/C++ and run on a dual-core 2.66GHz, 4GB RAM machine running a Windows 7 operating system. The average computation time for segmentation and enhancement per latent is about 2.6 seconds for latents in NIST SD27 and 1.6 seconds for latents in WVU DB.

The ultimate goal of fingerprint segmentation and enhancement of latent images is to improve the latent matching performance. To make the latent matching problem more realistic and challenging, the background database size is expanded to 31,997 rolled fingerprints by including 27,000 rolled fingerprints in the NIST SD14, 258 rolled fingerprints in NIST SD27 and 4,739 rolled fingerprints in WVU DB.

The segmentation and enhancement algorithm based on ridge structure dictionary is evaluated by a state-of-the-art latent matcher (COTS) to determine whether the proposed algorithm is able to boost latent matching performance. The match scores from the two input images (original latent image in Fig. 1.15 (a) and segmented and enhanced image in Fig. 1.15 (b)) are fused by a weighted sum method; the weights for original latent image template and segmented and enhanced image templates are empirically set as 0.7 and 0.3, respectively. The resulting CMC curves for the COTS latent matcher on NIST SD27 and WVU DB are shown in Fig. 1.16. The fusion of match scores from these two inputs improves the rank-1 identification rate of COTS latent matcher on both NIST SD27 and WVU DB. Fusing outputs of diverse search templates extracted from different segmentation and enhancement

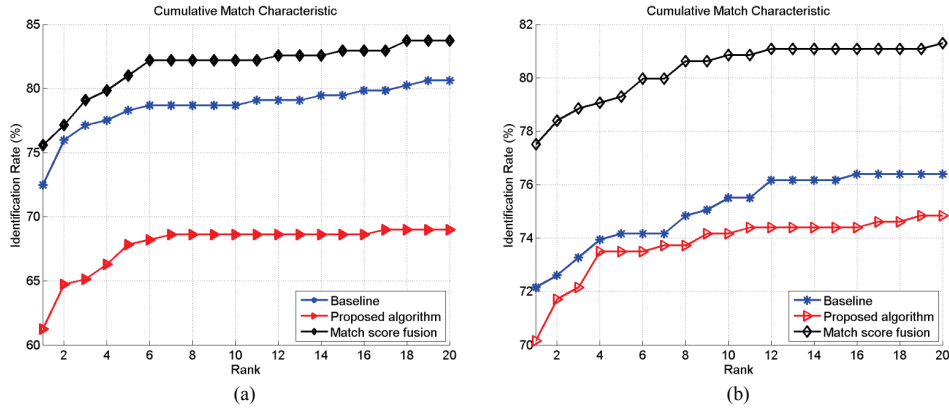


Figure 1.16: CMC curves of latent fingerprint identification with the COTS latent matcher on (a) NIST SD27 and (b) WVU DB. Images courtesy of Cao et al. (2014).

algorithms appears to be a good strategy to boost latent matching performance.

1.5 Conclusions and Future Work

Automatic Fingerprint Identification Systems (AFIS) have achieved extremely high matching accuracies in tenprint searches (rolled or plain fingerprints). For this reason, almost every law enforcement agency in the world relies on the use of AFIS to identify suspects and criminals. Further, there is a growing use of fingerprints to conduct background searches of applicants for visa and other government issued secure documents. However, latent fingerprint search is still a challenging problem due to the presence of complex background noise and poor quality of friction ridge structure that is typical for latent fingerprint images found at crime scene investigations. For this reason, human intervention, such as manual markup of minutiae and singular points, is a common practice for latent fingerprint identification.

A fully automatic latent identification (“lights-out” mode) is highly desired to alleviate the concerns about repeatability and reproducibility of latent examiners’ performance and increase the throughput of the latent matching process. Automatic feature extraction is one of the most crucial steps in “lights-out” latent identification. In this chapter, we have summarized the role of two types of dictionaries, orientation patch dictionary and ridge structure dictionary, as representations of prior knowledge about fingerprint patterns. We show how these dictionaries can be used in latent segmentation and enhancement. The orientation patch dictionary is used to update the initial orientation field in the input region of interest. The ridge structure dictionary is used for region of interest segmentation and enhancement. Experimental results on two different latent fingerprint databases demonstrate the advantages of our dictionary-based approach for fingerprint segmentation and enhancement.

In order to further improve the dictionary-based algorithms, we need to address the following issues:

1. Instead of simply using local orientation patch dictionary, global orientation field dictionary may further improve the accuracy of orientation field estimation.
2. A multiresolution approach should be considered to construct orientation patch dictionaries for both small and large friction ridge areas.
3. A robust ridge quality estimation for fingerprint images with low contrast (as in “dry” fingerprints) and background line structure noise is needed.

References

- Aharon M, Elad M and Bruckstein A 2006 K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing* **54**(11), 4311 – 4322.
- Ashbaugh D 1999 *Quantitative-Qualitative Friction Ridge Analysis: An Introduction to Basic and Advanced Ridgeology* CRC Press.
- Blake A, Kohli P and Rother C 2011 *Markov Random Fields for Vision and Image Processing* MIT Press.
- Buades A, Le T, Morel JM and Vese L 2010 Fast cartoon + texture image filters. *IEEE Transactions on Image Processing* **19**(8), 1978 – 1986.
- Cao K, Liu E and Jain A 2014 Segmentation and enhancement of latent fingerprints: A coarse to fine ridgestructure dictionary. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(9), 1847–1859.
- Chikkerur S, Cartwright AN and Govindaraju V 2007 Fingerprint enhancement using STFT analysis. *Pattern Recognition* **40**(1), 198–211.
- Dror IE, Wertheim K, Fraser-Mackenzie P and Walajts J 2011 The impact of human-technology cooperation and distributed cognition in forensic science: Biasing effects of AFIS contextual information on human experts. *Journal of Forensic Sciences* **57**(2), 343–352.
- Duta N, Jain AK and Mardia KV 2002 Matching of palmprints. *Pattern Recognition Letters* **23**(4), 477–485. In Memory of Professor E.S. Gelsema.
- Elad M and Aharon M 2006 Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing* **15**(12), 3736 – 3745.
- Feng J, Zhou J and Jain AK 2013 Orientation field estimation for latent fingerprint enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(4), 925–940.
- Garris MD, Tabassi E, Wilson CI, McCabe RM, Janet S and Watson CI 2004 NIST fingerprint image software 2.
- Hawthorne M 2008 *Fingerprints: Analysis and Understanding* CRC Press.
- Hong L, Wan Y and Jain A 1998 Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(8), 777–789.
- Indovina MD, Dvornychenko V, Hicklin RA and Kiebzinski GI 2012 Evaluation of latent fingerprint technologies: Extended feature sets (evaluation 2). *Technical Report NISTIR 7859, NIST*.
- Indovina MD, Dvornychenko VN, Tabassi E, Quinn GW, Grother PJ, Meagher S and Garris MD 2009 ELFT phase II - an evaluation of automated latent fingerprint identification technologies. *NISTIR 7577*.
- Jain A and Feng J 2009 Latent palmprint matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(6), 1032–1047.
- Jain A, Hong L and Bolle R 1997 On-line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**, 302–314.
- Jiang X 2000 Fingerprint image ridge frequency estimation by higher order spectrum *IEEE International Conference on Image Processing*, vol. 1, pp. 462–465.
- Karimi-Ashtiani S and Kuo CC 2008 A robust technique for latent fingerprint image segmentation and enhancement *IEEE International Conference on Image Processing*, pp. 1492 – 1495.
- Lian XC, Li Z, Wang C, Lu BL and Zhang L 2010 Probabilistic models for supervised dictionary learning *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2305 – 2312.
- Liao S, Jain AK and Li SZ 2013 Partial face recognition: Alignment-free approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(5), 1193–1205.
- Mairal J, Bach F, Ponce J, Sapiro G and Zisserman A 2008a Discriminative learned dictionaries for local image analysis *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8.
- Mairal J, Elad M and Sapiro G 2008b Sparse representation for color image restoration. *IEEE Transactions on Image Processing* **17**(1), 53 – 69.
- Mallat S and Zhang Z 1993 Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing* **41**(12), 3397 – 3415.
- Maltoni D, Maio D, Jain A and Prabhakar S 2009 *Handbook of Fingerprint Recognition* second edition edn Springer.
- Mardia K, Baczkowski A, Feng X and Hainsworth T 1997 Statistical methods for automatic interpretation of digitally scanned finger prints. *Pattern Recognition Letters* **18**(11-13), 1197 – 1203.

- Apple, Inc. 2014 iPhone 5s: About Touch ID security <https://support.apple.com/kb/HT5949>.
- Department of Homeland Security 2014 Office of biometric identify management <http://bias.dhs.gov/obim>.
- FBI- NGI 2014 http://www.fbi.gov/about-us/cjis/fingerprints_biometrics/ngi.
- FVC2002 2002 <http://bias.csr.unibo.it/fvc2002/>.
- iProBe 2014 <http://www.csee.wvu.edu/~ross/i-probe/>.
- Neurotechnology Inc. 2012 Verifinger <http://www.neurotechnology.com/verifinger.html>.
- NIST Special Database 27 2014 <http://www.nist.gov/srd/nisttd27.cfm>.
- NIST Special Database 4 2014 <http://www.nist.gov/srd/nisttd4.cfm>.
- Planning Commission, Government of India 2014 Unique Identification Authority of India <http://uidai.gov.in/>.
- Otsu N 1979 A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics* **9**(1), 62–66.
- Short NJ, Hsiao MS, Abbott AL and Fox EA 2011 Latent fingerprint segmentation using ridge template correlation *4th International Conference on Imaging for Crime Detection and Prevention*, pp. 1–6.
- Tabassi E, Wilson C and Watson C 2004 Fingerprint image quality. *NISTIR 7151*.
- Trauring M 1963 Automatic comparison of finger-ridge patterns. *Nature* **197**, 938–940.
- Turroni F, Maltoni D, Cappelli R and Maio D 2011 Improving fingerprint orientation extraction. *IEEE Transactions on Information Forensics and Security* **6**(3), 1002–1013.
- Ulery BT, Hicklin RA, Buscaglia J and Roberts MA 2011 Accuracy and reliability of forensic latent fingerprint decisions. *Proceedings of the National Academy of Sciences* **108**(19), 7733–7738.
- Ulery BT, Hicklin RA, Buscaglia J and Roberts MA 2012 Repeatability and reproducibility of decisions by latent fingerprint examiners. *PloS one* **7**(3), e32800.
- Wang Y, Hu J and Phillips D 2007 A fingerprint orientation model based on 2d fourier expansion (FOMFE) and its application to singular-point detection and fingerprint indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(4), 573–585.
- Wang Z, Bovik A, Sheikh H and Simoncelli E 2004 Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* **13**(4), 600–612.
- Wilson C, Hicklin RA, Korves H, Ulery B, Zoepfl M, Bone M, Grother P, Micheals R, Otto S and Watson C 2004 Fingerprint vendor technology evaluation 2003: Summary of results and analysis report. *NISTIR 7123*.
- Wright J, Yang A, Ganesh A, Sastry S and Ma Y 2009 Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(2), 210–227.
- Yoon S, Feng J and Jain A 2011 Latent fingerprint enhancement via robust orientation field estimation *2011 International Joint Conference on Biometrics (IJCB)*, pp. 1–8.
- Zhang J, Lai R and Kuo CC 2013 Adaptive directional total-variation model for latent fingerprint segmentation. *IEEE Transactions on Information Forensics and Security* **8**(8), 1261–1273.