

FingerCode: A Filterbank for Fingerprint Representation and Matching*

Anil K. Jain, Salil Prabhakar, Lin Hong,
Department of Computer Science and Engineering
Michigan State University, East Lansing, MI 48824
{jain,prabhaka,honglin}@cse.msu.edu

and Sharath Pankanti
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
sharat@watson.ibm.com

Abstract

With the identity fraud in our society reaching unprecedented proportions and with an increasing emphasis on the emerging automatic positive personal identification applications, biometrics-based identification, especially fingerprint-based identification, is receiving a lot of attention. There are two major shortcomings of the traditional approaches to fingerprint representation. For a significant fraction of population, the representations based on explicit detection of complete ridge structures in the fingerprint are difficult to extract automatically. The widely used minutiae-based representation does not utilize a significant component of the rich discriminatory information available in the fingerprints. The proposed filter-based algorithm uses a bank of Gabor filters to capture both the local and the global details in a fingerprint as a compact 640-byte fixed length FingerCode. The fingerprint matching is based on the Euclidean distance between the two corresponding FingerCodes and hence is extremely fast. Our initial results show identification accuracies comparable to the best results of minutiae-based algorithms published in the open literature [2]. Finally, we show that the matching performance can be improved by combining the decisions of the matchers based on complementary fingerprint information.

1 Introduction

With the advent of electronic banking, e-commerce, and smartcards and an increased emphasis on the privacy and security of information stored in various databases, automatic personal identification has become a very important topic. Accurate automatic personal identification is now needed in a wide range of civilian applications such as passport control, cellular telephones, automatic teller machines, and driver licenses. Traditional knowledge-based (password or Personal Identification Number (PIN)) and token-

*This work is partially supported by IBM Contract No. 111001069.



Figure 1: Minutiae features: a ridge ending (\square) and a bifurcation (\circ).

based (passport, drivers license, and ID card) identifications are prone to fraud because PINs may be forgotten or guessed by an imposter and the tokens may be lost or stolen. As an example, credit card fraud alone now costs more than 6 billion dollars annually. *Biometrics*, which refers to identifying an individual based on his or her physiological or behavioral characteristics is more reliable in differentiating between an authorized person and an imposter.

Among all the various biometrics (e.g., face, fingerprints, iris, etc.), fingerprint-based identification is the most mature and proven technique. A fingerprint is the pattern of ridges and furrows on the surface of the finger. The uniqueness of a fingerprint can be determined by the overall pattern of ridges and furrows as well as the local ridge anomalies (a ridge bifurcation or a ridge ending, called minutiae points (see Figure 1)). As fingerprint sensors are becoming smaller and cheaper [1], automatic identification based on fingerprint is becoming an attractive alternative/complement to the traditional methods of identification. The critical factor for the widespread use of fingerprints is in meeting the performance (e.g., matching speed and accuracy) standards demanded by

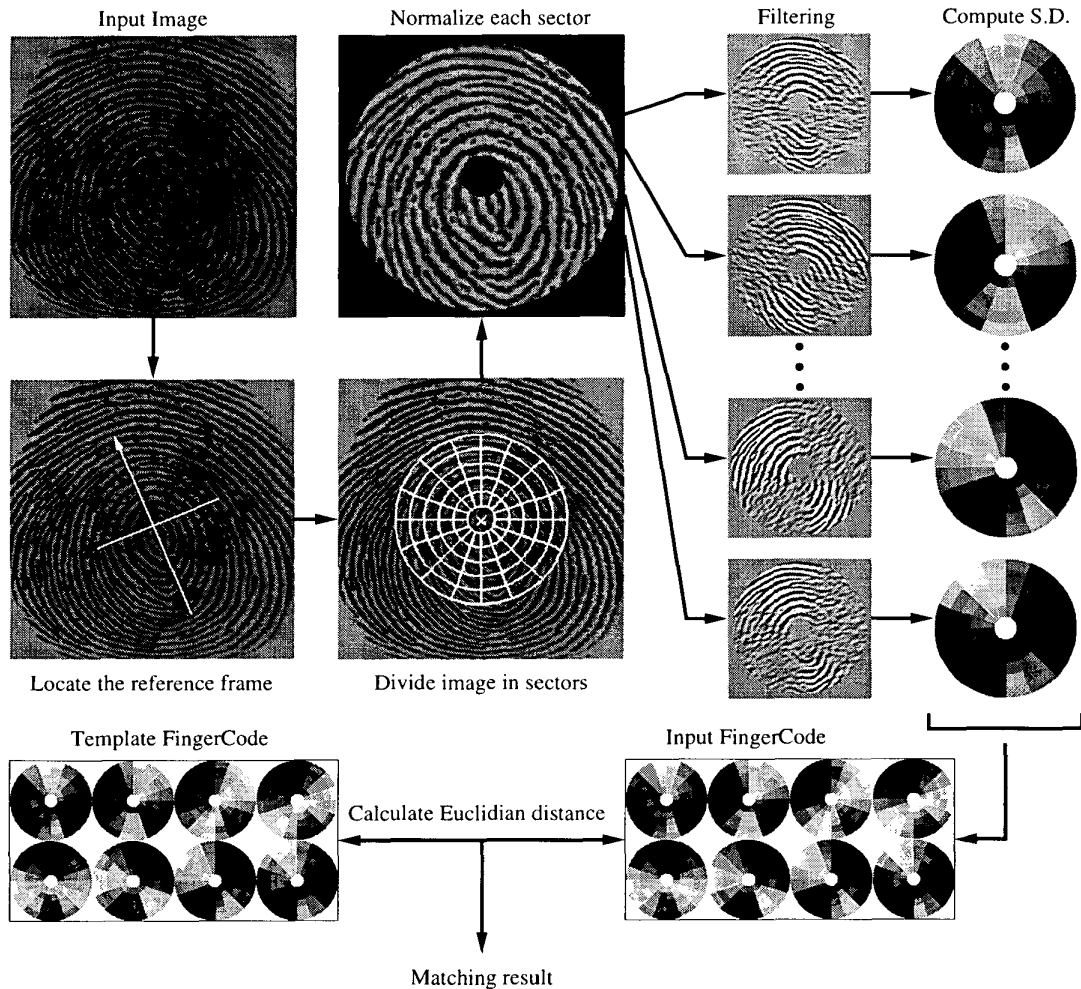


Figure 2: Flow diagram of our fingerprint identification system.

emerging civilian identification applications. Some of these applications (e.g., fingerprint-based smartcards) will also benefit from a compact representation of a fingerprint.

The popular fingerprint representation schemes have evolved from intuitive system design tailored for fingerprint experts who visually match fingerprints. These schemes are either predominantly local (e.g., minutiae-based fingerprint matching systems [2, 3]) or exclusively global (fingerprint classification based on Henry system [4, 5]). The minutiae-based automatic identification techniques first locate the minutiae points and then match their relative placements in a given finger and the stored template [2]. A good quality fingerprint contains between 60 to 80 minutiae, but different fingerprints have different number

of minutiae. The variable sized minutiae-based representation does not easily lend itself to indexing mechanisms; typical approaches [2] to match the minutiae from two fingerprints need to align the unregistered minutiae patterns of different sizes which makes them computationally expensive. The variable length of fingerprint representation (in terms of position and orientation of minutiae) makes it unsuitable to store the fingerprint on a smartcard. The global approach to fingerprint representation is typically used for indexing, but does not offer good individual discrimination. Further, the indexing efficacy of existing global representations is poor due to a small number of categories that can be effectively identified and a highly skewed distribution of the population in each category. Both these approaches utilize representations which cannot

be easily extracted from poor quality fingerprints.

It is desirable to explore representation schemes which combine global and local information in a fingerprint. We present a new representation for the fingerprints which yields a relatively short, fixed length code, called *FingerCode* suitable for matching as well as storage on a smartcard. The matching reduces to finding the Euclidean distance between these FingerCodes and hence the matching is very fast and the representation is amenable to indexing. We make use of both the global flow of ridges and furrows structure and the local ridge characteristics to generate a short fixed length code for the fingerprints while maintaining a high recognition accuracy.

The proposed scheme of feature extraction tessellates the region of interest of the given fingerprint image with respect to a *frame of reference* (Figure 2). A feature vector is composed of an ordered enumeration of the features extracted from the (local) information from each subimage (sector) specified by the tessellation. Thus, the feature elements capture the local information and the ordered enumeration of the tessellation captures the invariant global relationships among the local patterns. The local discriminatory information in the sector needs to be decomposed into separate components. Gabor filterbank is a well known technique to capture useful information in specific bandpass channels as well as decompose this information into orthonormal components in terms of spatial frequencies.

2 Filter-based Feature Extraction

The three main steps in our feature extraction algorithm are: (i) determine a reference frame for the fingerprint image, (ii) filter the image in eight different directions using a bank of Gabor filters, and (iii) compute the standard deviation of gray values in sectors around the reference point in filtered images to define the feature vector or the FingerCode.

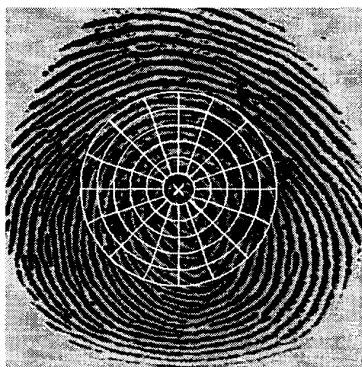


Figure 3: Reference point (\times), the region of interest and 80 sectors superimposed on a fingerprint.

Let $I(x, y)$ denote the gray level at pixel (x, y) in an $M \times N$ fingerprint image and let (x_c, y_c) denote the reference point. The region of interest is defined as the collection of all the sectors S_i , where the i^{th} sector S_i is computed in terms of parameters (r, θ) as follows:

$$S_i = \{(x, y) | b(T_i + 1) \leq r < b(T_i + 2), \theta_i \leq \theta < \theta_{i+1}, 1 \leq x \leq N, 1 \leq y \leq M\}, (1)$$

where $T_i = i \text{ div } k$, $\theta_i = (i \text{ mod } k)(2\pi/k)$, $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$, $\theta = \tan^{-1}((y - y_c)/(x - x_c))$, b is the width of each band and k is the number of sectors considered in each band. We consider five concentric bands around the detected reference point for feature extraction. Each band is 20-pixels wide ($b = 20$), and segmented into sixteen sectors ($k = 16$) (Figure 3). A 20-pixel wide band captures about 2 ridges and furrows on an average, in a 500 dpi fingerprint image. The innermost band is not used for feature extraction because the sectors in the region near the reference point contain very few pixels and, therefore, the standard deviation estimates in this region are not very reliable. Thus, we have a total of $16 \times 5 = 80$ sectors (S_0 through S_{79}). Eighty features for each of the eight filtered images give us a total of 640 (80×8) features per fingerprint image. Each feature can be quantized into 256 values and requires 1 byte of storage, so the entire feature vector requires only 640 bytes of storage.

It is difficult to rely on feature extraction based on explicit detection of structural features in (e.g., poor quality) fingerprints; features based on statistical properties of images are likely to degrade gracefully with the image quality deterioration. For this study, we rely on grayscale variance-based (e.g., standard deviation) features. The grayscale standard deviation in an image sector is indicative of the overall ridge activity. As noted in the Section 3, our matcher based on this simple statistical feature performs extremely well and we expect to achieve significantly better accuracies with more discriminative attributes.

It is desirable to obtain representations for fingerprints which are scale, translation, and rotation invariant. Scale invariance is not a significant problem since most fingerprint images could be scaled as per the spatial resolution (*dpi*) of the sensor. The rotation and translation invariance could be accomplished by establishing a reference frame based on the intrinsic fingerprint characteristics which are rotation and translation invariant.

2.1 Reference Frame Determination

Fingerprints have many conspicuous landmark structures and any combination of them could be used

for establishing a reference frame determination. We define the reference point of a fingerprint as the point of maximum curvature of ridges in the fingerprint image and the reference axis is defined to be the axis of local symmetry at the reference point. Reference point and reference axis together establish an invariant frame of reference for a given fingerprint. Reference frame detection relies on Poincaré index analysis and local symmetry detection, similar to the method used in [5].

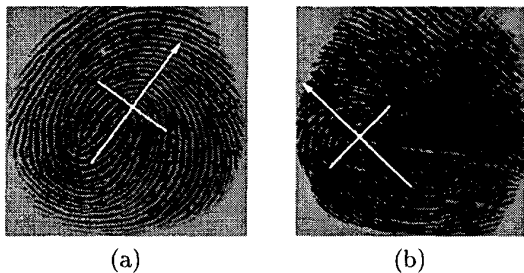


Figure 4: Determining the reference frame.

Our representation scheme tolerates imprecision in the estimates of reference frame. Since the fingerprints are *smoothly* flowing ridge patterns, the characteristics of local neighborhoods change gradually. A small perturbation (within one inter-ridge distance unit) is likely to change the representation only slightly, so the overall variations in the representation of a finger due to inaccurate localization is expected to remain small. The detected reference point could be as much as 12 pixels (approximately 1 inter-ridge distance units) away from its “true” location. Symmetry axis detection is very difficult in fingerprint images because the upper portion of the fingerprint has circular ridges. We do not rely on the symmetry axis to align the fingerprints. We achieve rotation invariance by rotating the FingerCode itself during the matching stage. Typical outputs of the reference frame determination algorithm are shown in Figure 4.

2.2 Filtering

To remove noise and enhance the ridge and furrow structures, we filter the fingerprint image in different directions using a bank of Gabor filters [6]. Fingerprints have local parallel ridges and furrows, and well-defined local frequency and orientation. Properly tuned Gabor filters can remove noise, preserve the true ridge and furrow structures, and provide information contained in a particular direction in the image. A minutia point is an anomaly in locally parallel ridges and this information is captured by the Gabor filters. An even symmetric Gabor filter has the following gen-

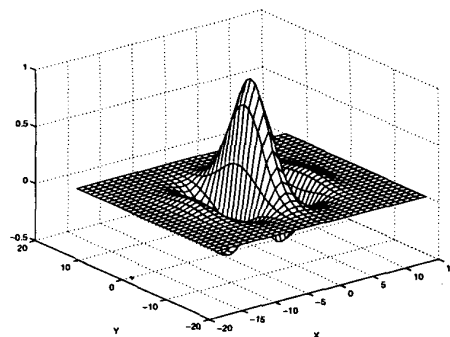


Figure 5: One of the eight Gabor filters (size = 33×33 , $f = 0.1$, $\delta_x = 4.0$, $\delta_y = 4.0$, $\theta = 0^\circ$) in the spatial domain.

eral form in the spatial domain:

$$G(x, y; f, \theta) = \mathcal{G}(x', y', \delta_x, \delta_y) \cos(2\pi f x'), \quad (2)$$

where $\mathcal{G}(x', y', \delta_x, \delta_y) = e^{-0.5((x'/\delta_x)^2 + (y'/\delta_y)^2)}$, $x' = x \sin \theta + y \cos \theta$, $y' = x \cos \theta - y \sin \theta$, f is the frequency of the sinusoidal plane wave along the direction θ from the x -axis, and δ_x and δ_y are the space constants of the Gaussian envelope along x and y axes, respectively.

In our experiments, we set the filter frequency f to the average ridge frequency ($1/K$), where K is the inter-ridge distance. The average inter-ridge distance is approximately 10 pixels in a 500 *dpi* fingerprint image. If f is too large, spurious ridges are created in the filtered image whereas if f is too small, nearby ridges are merged into one. We used eight different values for θ (0° , 22.5° , 45° , 67.5° , 90° , 112.5° , 135° , and 157.5°) with respect to the x -axis. One of these eight filters is shown in Figure 5. The region of interest in a fingerprint image is convolved with each of these eight filters to produce a set of eight filtered images. A fingerprint convolved with a 0° -oriented filter accentuates those ridges which are parallel to the x -axis and smoothes the ridges in the other directions. Filters tuned to other directions work in a similar way. These eight direction-sensitive filters capture most of the global ridge directionality information as well as the local ridge characteristics present in a fingerprint. We illustrate this by reconstructing a fingerprint image by adding together all the eight filtered images. The reconstructed image is similar to the original image but has been enhanced (Figure 6(h)). At least four directions are required to capture the entire global ridge information in a fingerprint (Figure 6(g)), but eight directions are required to capture the local characteristics. By capturing both the global and local information, the verification accuracy is improved although there is some redundancy among the eight fil-

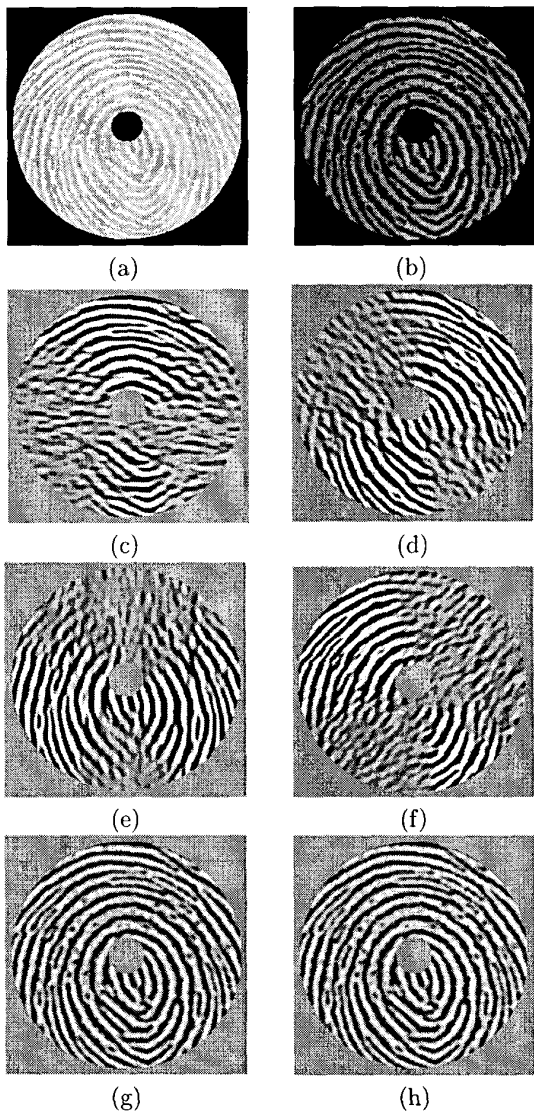


Figure 6: Normalized, filtered (only four orientations shown), and reconstructed fingerprint images. Only the central part of the fingerprint is shown. (a) area of interest (b) normalized image (c)-(f) 0° , 45° , 90° , and 135° filtered images; (g), (h) reconstructed images using four and eight orientation filters, respectively.

tered images. If the δ_x and δ_y (standard deviations of the Gaussian envelope) values are too large, the filter is more robust to noise, but is more likely to smooth the image to the extent that the ridge and furrow details in the fingerprint are lost. If the δ_x and δ_y values are too small, the filter is not effective in removing noise. The values for δ_x and δ_y were empirically determined and each is set to 4.0 (about half the average inter-ridge distance).

Before filtering the fingerprint image, we normalize the region of interest in each sector separately to a constant mean and variance. Normalization is performed to remove the effects of sensor noise and finger pressure differences. Let $I(x, y)$ denote the gray value at pixel (x, y) , M_i and V_i , the estimated mean and variance of sector S_i , respectively, and $N_i(x, y)$, the normalized gray-level value at pixel (x, y) . For all the pixels in sector S_i , the normalized image is defined as:

$$N_i(x, y) = \begin{cases} M_0 + \sqrt{\frac{(V_0) \times (I(x, y) - M_i)^2}{V_i}}, & \text{if } I(x, y) > M_i \\ M_0 - \sqrt{\frac{(V_0) \times (I(x, y) - M_i)^2}{V_i}}, & \text{otherwise,} \end{cases} \quad (3)$$

where M_0 and V_0 are the desired mean and variance values, respectively. Normalization is a pixel-wise operation which does not change the clarity of the ridge and furrow structures. If normalization is performed on the entire image, then it can not compensate for the intensity variations in different parts of the finger due to finger pressure differences. Normalization of each sector separately alleviates this problem. For our experiments, we set both M_0 and V_0 to a value of 100.

2.3 Feature Vector

Let $F_{i\theta}(x, y)$ be the θ -direction filtered image for sector S_i . For $\forall i = 0, 1, \dots, 79$ and $\theta \in [0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ, 112.5^\circ, 135^\circ, 157.5^\circ]$, the feature value is the standard deviation $V_{i\theta}$, defined as $V_{i\theta} = \sqrt{\frac{1}{K_i} \sum_{K_i} (F_{i\theta}(x, y) - P_{i\theta})^2}$, where K_i is the number of pixels in S_i and $P_{i\theta}$ is the mean of pixel values of $F_{i\theta}(x, y)$ in S_i . The standard deviation of each sector in each of the eight filtered images define the components of our feature vector. The 640-dimensional feature vectors (FingerCodes) for fingerprint images of two different fingers are shown as gray level images with eight disks, each disk corresponding to one filtered image (Figure 7). The gray level in a sector in a disk represents the feature value for that sector in the corresponding filtered image. Note that Figures 7(a) and (b) appear to be visually similar as

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.