EURASIP Journal on Applied Signal Processing 2005:13, 2043–2053

Coarse Fingerprint Registration Using Orientation Fields

Neil Yager

School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia
Email: nyager@cse.unsw.edu.au

Adnan Amin

School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia
Email: amin@cse.unsw.edu.au

Received 9 December 2003; Revised 2 August 2004

The majority of traditional research into automated fingerprint identification has focused on algorithms using minutiae-based features. However, shortcomings of this approach are becoming apparent due to the difficulty of extracting minutiae points from noisy or low-quality images. Therefore, there has been increasing interest in algorithms based on nonminutiae features in recent years. One vital stage in most fingerprint verification systems is registration, which involves recovering the transformation parameters that align features from each fingerprint. This paper investigates the use of orientation fields for registration; an approach that has the potential to perform robustly for poor-quality images. Three diverse algorithms have been implemented for the task. The first algorithm is based on the generalized Hough transform, and it works by accumulating evidence for transformations in a discretized parameter space. The second algorithm is based on identifying distinctive local orientations, and using these as landmarks for alignment. The final algorithm follows the path of steepest descent in the parameter space to quickly find solutions that are locally optimal. The performance of these three algorithms is evaluated using an FVC2002 dataset.

Keywords and phrases: fingerprint registration, fingerprint verification, orientation fields, biometrics, FVC2002.

1. INTRODUCTION

Fingerprints have been used as a means of personal identification for over a century. Traditionally, the driving force behind advancements in fingerprint technology has been law enforcement agencies and forensic scientists. Using fingerprints lifted at a crime scene to identify suspects can be a crucial step during a criminal investigation. Consequently, massive fingerprint databases have been collected by law enforcement agencies around the world. For example, the FBI maintains the world's largest fingerprint database, containing more than 200 million prints. The administration and querying of such large databases relies heavily on automated systems, thereby motivating the early research efforts in the field.

Another application of fingerprint-based identification that has emerged more recently is biometric systems. Biometrics is the automatic identification of an individual based on his or her physiological or behavioral characteristics. The ability to accurately identify or authenticate an individual based on these characteristics has several advantages over traditional means of authentication such as knowledge-based (e.g., password) or token-based (e.g., key) authentication [1]. Example applications of biometric systems include building access systems, ATM authentication, and welfare disbursements. Due to its security-related applications and the current world political climate, biometrics has become the subject of intense research by both private and academic institutions.

Due to the continuing needs of law enforcement and interest from the developers of biometric systems, efficient automated fingerprint identification systems (AFISs) are becoming increasingly widespread and are being extensively researched by the pattern recognition community. Most fingerprint matching algorithms are based to some degree on matching small fingerprint details known as minutiae (see Section 2). Minutiae are notoriously difficult-extract from noisy or low-resolution images, causing many difficulties for minutiae-based algorithms [2]. Therefore, in recent years there has been increasing interest in algorithms based on nonminutiae features.

Fingerprint registration is a vital stage in most fingerprint matching algorithms. In order to determine the degree of similarity between two fingerprints, it is first necessary to align the prints so that corresponding features may be matched. This paper introduces the novel idea of using

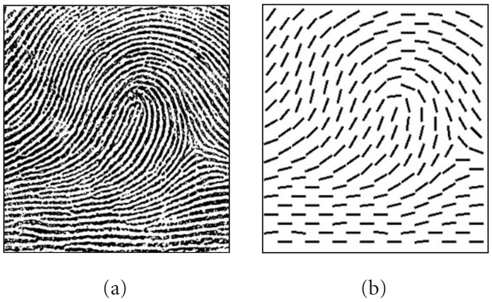

Figure 1: (a) A fingerprint and (b) its orientation field.

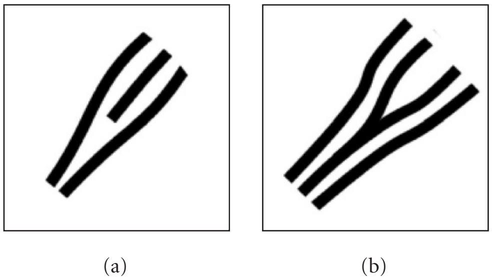

Figure 2: Fingerprint minutiae. (a) A ridge ending. (b)A bifurcation.

orientation fields for fingerprint registration, which has the potential to perform robustly for poor-quality fingerprint images. Three orientation field registration algorithms are presented. The first algorithm is based on the generalized Hough transform, and it works by accumulating evidence for transformations in a discretized parameter space. The second algorithm is based on identifying distinctive local orientations, and using these as landmarks for alignment. The final algorithm follows the path of steepest descent in the parameter space to quickly find a solution that is locally optimal. A framework for the extensive evaluation of these algorithms is developed, and experiments are conducted using a publicly available fingerprint database.

The paper begins with background on fingerprint structure in Section 2, and a literature review of existing fingerprint registration techniques in Section 3. Registration using orientation fields is introduced in Section 4, along with three algorithms implementing this concept. Section 5 describes the experiments that were conducted to evaluate the registration algorithms, and is followed by the results and an analysis in Section 6. Finally, Section 7 closes with some concluding remarks and comments regarding future directions for this research.

2. FINGERPRINT STRUCTURE

The skin on the palm and fingers of the human hand has the unique property of being corrugated by a pattern of narrow ridges and valleys. Of particular interest are the ridges and valleys at the tips of fingers, known as fingerprints. Figure 1a shows a sample finger print.

There are two main levels of structure in a fingerprint. The first level of structure is the global pattern of ridges and valleys, and this pattern is often summarized using orientation fields. Orientation fields contain information about the local average directions of fingerprint ridges (see Figure 1b).

The second level of structure occurs in local fingerprint regions. Local ridge discontinuities, known as a *minutiae*, have little effect on the global ridge-valley pattern. However, it is the existence and locations of these minutiae that embody much of a fingerprint's individuality. For this reason, they are the most important and common discriminating feature used by human experts. There are two basic types of minutiae: ridge endings and bifurcations. Ridge endings are places where ridges terminate and bifurcations are locations where a single ridge separates into two ridges (see Figure 2).

Singularities are another important fingerprint structure that have both global and local properties. Globally, a singularity is a region of a fingerprint where the ridge pattern makes it visually prominent. There are two types of fingerprint singularities: cores and deltas. Locally, a core is the turning point of an inner-most ridge and a delta is a place where two ridges running side-by-side diverge.

3. FINGERPRINT REGISTRATION

Fingerprint matching algorithms are based on comparing features from one print against those from another fingerprint, and this process is usually composed of several stages. One stage that is present in most matching algorithms is registration. In order to compare two fingerprints to determine their similarity, it is first necessary to find and remove the relative transformation between the prints. This process is known as registration. To be specific, the goal of fingerprint registration is to find the translation and rotation parameters that align two fingerprints and their corresponding features.[1]

3.1. Minutiae-based registration

The traditional approach to fingerprint registration is based on aligning minutiae features [2]. Given two fingerprint images, all of the minutiae are extracted from each print and their location, orientation, and type are recorded. Registration is based on aligning these two minutiae sets. This is essentially point pattern matching; a well-known problem that often arises in pattern recognition and computer vision. In the case of matching minutiae sets, there are a few complications [2]. First of all, every time a fingerprint is obtained, a different area of the fingerprint surface may be captured. Therefore, alignment should be based only on the overlap area of the prints and the corresponding minutiae subsets. Secondly, missing and spurious minutiae are common when

[1] In our case, we assume the images have the same resolution and do not consider scaling. Furthermore, we do not account for nonlinear deformations.

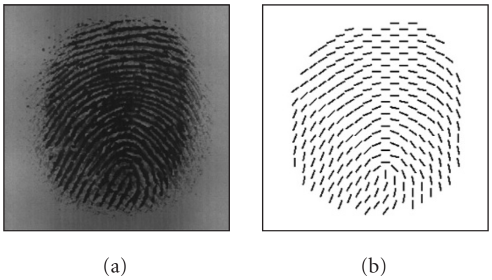

Figure 3: (a) A poor-quality fingerprint. Minutiae extraction from (a) would be difficult, however an accurate orientation field (b) has been computed.

the fingerprint image quality is low (see Figure 3a). Therefore, the alignment algorithm must allow some minutiae to be unmatched, even in the area of overlap. Finally, it is well known that fingerprints deform when pressed against a flat surface. This nonlinear distortion changes the locations and orientations of the minutiae, making it impossible to find a perfect alignment of the sets. Therefore, most registration algorithms attempt to find an alignment that minimizes these errors. After taking these issues into consideration, it is not surprising that finding the optimal alignments can be very difficult.

Ratha et al. [3] estimate the rotation and translation parameters using the generalized Hough transform. The Hough transform is a common tool in image analysis that is normally used for detecting lines in point sets, but it can be generalized for point pattern matching. The space of all possible transformations is discretized into a finite set of values, and this is known as the parameter space. For each pair of potentially corresponding minutiae (one from the test set, and one from the reference set) the translation and rotation necessary to align them is calculated. Evidence for this translation and rotation is accumulated. After testing all possible matching minutiae pairs, the translation and rotation parameters with the most accumulated evidence are selected. The main drawback of this approach is that it is very computationally expensive. As the size of the minutiae sets grow, the number of potentially matching minutiae pairs to be considered increases exponentially.

3.2. Incorporating supplementary features to aid minutiae registration

Due to the large number of possible translations, rotations, and distortions, aligning fingerprints has a high computational overhead. One common approach used to deal with this complexity is to incorporate supplementary information from other fingerprint features to aid the alignment. The following registration algorithms use various features to reduce the number of plausible parameters.

Local structural features

Hrechak and McHugh use a representation of fingerprints that is based on local structural relationships among minutiae [4]. For each minutia, the number and types of other minutiae within a given radius are recorded. This information can be used to find potential matches in another minutiae set when the local structure is distinctive, and these potential matches can be used to recover the registration parameters. Several other approaches based on local structural features can be found in the literature [5, 6, 7, 8, 9, 10].

Ridge shape

Jain et al. use ridge information as an aid for alignment [11, 12]. When extracting minutiae, the shape and location of its associated ridge is also stored. For each possible minutiae pair (one from each point set), if the associated ridges are similar, the minutiae sets are translated so that the candidate minutiae pair are at the same place, and then rotated so that the associated ridges are aligned. This provides a coarse alignment of the minutiae sets.

Transformation parameter clustering

Germain et al. have developed a matching algorithm that uses various properties of minutiae triplets [13]. Originally developed as a method to index large fingerprint databases, it employs a clever method to align minutiae sets. For each fingerprint in the database, keys are generated based on minutiae triplets. Given three minutiae, a rotation and translation invariant key is constructed using the distance between minutiae pairs, ridge-counts between minutiae pairs, and minutiae orientation angles. For a query fingerprint, the same keys are generated and used to find similar minutiae triplets in the fingerprints database. For each pair of similar keys, the rigid transformation parameters needed to align them are calculated, and evidence for these transformations is accumulated. This process is known as transformation parameter clustering.

Pixel intensities

A fingerprint verification system presented by Kovacs-Vajna uses the pixel intensities from the local area of a minutia to help find minutiae matches [14]. Minutiae regions from the test and reference images are compared to find potential correspondences. The main drawback of this approach is that the pixel values around a minutiae point can vary for several reasons, and dealing with rotation is difficult.

Singularities

All fingerprint classes have at least one core point except for the plain arch. Since core points are common, they can be used as aid for fingerprint alignment. Zhang and Wang have explored this possibility [15]. First the core points from the two fingerprint images are detected using a multiresolution algorithm, and are then used as landmarks for registration. Basically, if a corresponding core point from each image can be found, it determines the translation parameter. Structural features of minutiae close to the core point are used to calculate the rotation parameter. This is a powerful and fast approach to fingerprint alignment assuming that core points

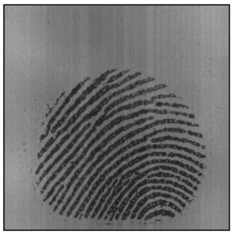

Figure 4: A fingerprint image that does not contain a core or delta point. Registration methods that require locating a singularity will fail for images such as this.

can be reliably extracted from fingerprint images. In some cases (such as plain arches), there will be no existing core point, so the definition of a core point must be broadened to include a well-defined position in all fingerprints. A more serious problem with this approach occurs when the capture surface is small and does not include the true core point at all (see Figure 4). Therefore, a robust fingerprint matching system should not rely solely on singularities for fingerprint registration.

Orientation fields

Tico and Kuosmanen have developed a registration algorithm that records orientation-field information from the neighborhood of all minutiae [16]. They define an orientation based descriptor that is used to find likely minutiae correspondences between the two minutiae sets. Registration is then based on the straightforward alignment of these probable correspondences.

Another system using orientation field information has been developed by Tico and Kuosmanen [17]. K points are selected from the test fingerprint and the orientation of their surrounding ridges is computed. The alignment parameters are estimated by searching for the best fit of these K orientations over the entire orientation field of the reference fingerprint. The location and orientation of minutiae from both sets are used to restrict the number of possible transformations that need to be considered.

3.3. Nonminutiae-based registration

All of the registration algorithms discussed in Sections 3.1 and 3.2 are based to some degree on aligning minutiae points. However, due to the problems with this approach, listed in Section 3.1, there is certainly value in algorithms based entirely on nonminutiae features. Surprisingly, there are very few registration algorithms that adhere to this concept.

Jain et al. have developed a fingerprint identification system that is based on nonminutiae features [18]. Their approach uses a bank of Gabor filters to capture information about the orientation and frequency of the fingerprint ridge patterns. Registration is implicitly based on locating a point of reference in fingerprints, defined to be the point of maximum curvature of the concave ridges. The authors admit that the reference point is difficult to locate for low-quality images. More seriously, if the capturing surface is small, this reference point may not even be captured at all. In this case, the system is unable to perform matches. Another drawback of the approach is that the feature vector is not rotationally invariant. This is addressed by storing multiple feature vectors for each fingerprint in the database, which has obvious storage and efficiency disadvantages. Park et al. have suggested methods to improve the feature extraction and alignment stages of this algorithm [19].

Another system based entirely on nonminutiae features has been proposed by Ross et al. [20]. The feature extraction for this system is also based on a bank of Gabor filters. Alignment is based on correlating feature maps, which has the advantage of not depending on the extraction of landmark reference points. However, the system does not calculate rotation parameters, which is a major weakness.

4. ORIENTATION FIELD REGISTRATION

4.1. Advantages

The approach explored in this paper is the use of orientation fields for alignment. To our knowledge, this is a novel approach and there have been no studies using this method published previously. There are several advantages to this technique over the traditional methods of fingerprint registration.

(i) There is no reliance on the existence of landmarks, such as singularities or minutiae. This is important as these features may not always be captured in the area of overlap. As long as there is some overlap between the two fields, the alignment may proceed independent of the fingerprint regions that have been captured. Furthermore, there are no a priori assumptions about the amount of overlap necessary.

(ii) Orientation fields have been widely researched [21, 22] and can be reliably calculated for low-quality images, even when other features can not be extracted, as illustrated in Figure 3.

(iii) Calculating orientation fields has a relatively low computational cost. Conversely, extracting other features (such as minutiae points) may require a lot of processing (and often extensive image enhancement).

(iv) Orientation fields are a high-level, coarse feature, and therefore, alignment will not be hindered by the presence of nonlinear distortions in the prints.

(v) Orientation fields are used extensively in fingerprint algorithms (e.g., for image enhancement, ridge map extraction, singularity extraction, etc.) and so they will likely be available without any additional computation.

The primary disadvantage of using orientation fields to guide registration is that they will only be able to provide a coarse alignment. Orientation field elements contain local averages, and this imposes a fundamental limitation on the accuracy of the alignment (depending on the size of the orientation field elements). Therefore, for applications requiring a high degree of accuracy, other methods would need to be employed to fine-tune the parameters generated by orientation field registration. However, even in these cases, having initial parameter estimates may be very beneficial.

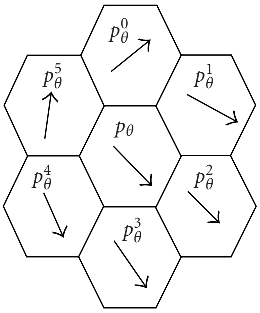

FIGURE 5: A hexagonal orientation field element and its neighbors.

4.2. Definitions

For this study, every pixel in the fingerprint image is assigned an orientation value. The orientations are restricted to the range $[-(1/2)\pi, (1/2)\pi)$. Therefore, the two directions α and $\alpha + \pi$ lie along the same orientation. The orientation field partitions the image into hexagonal elements (see Figure 5), which has the advantage that the distance to the center of the 6 neighboring elements is almost constant (as opposed to the 8 neighbors of the square grid case). For each orientation field element, the orientation is calculated as the average of the orientations of each pixel. After averaging, each pixel in the element is reassigned this value. The orientation field can be summarized by a set of vectors (one for each element) with a position and orientation:

$$\mathcal{P} = \left\{ (p_x^1, p_y^1, p_\theta^1), \ldots, (p_x^{|\mathcal{P}|}, p_y^{|\mathcal{P}|}, p_\theta^{|\mathcal{P}|}) \right\}. \tag{1}$$

The hexagonal elements used in this study have sides 12 pixels long. For an image of size 388 by 374 pixels, this partitions the image into around 700 elements.

Define the function λ of the orientations α and β to be the minimum counterclockwise rotation necessary to align one orientation with the other. If we assume $\alpha \geq \beta$, we calculate λ as follows:

$$\lambda(\alpha, \beta) = \min \{(\alpha - \beta), (\beta - \alpha + \pi)\}, \tag{2}$$

which takes a value in the range $[0, (\pi/2)]$.

The output of registration is the transformation parameters $(\theta, \Delta x, \Delta y)$ that are applied to one fingerprint to align it with another. Let $\mathcal{P}$ be a representation of the reference orientation field, and let $\mathcal{Q}$ be a representation of the query orientation field. We define the quality of an alignment to be the degree of similarity between the orientations of elements at the same locations in $\mathcal{P}$ and $\mathcal{Q}$ after the transformation has been applied to $\mathcal{Q}$. Assume $\mathcal{Q}'$ is the orientation field obtained by applying transformation parameters to the query fingerprint. The cost function C of $\mathcal{P}$ and $\mathcal{Q}'$ is

$$C(\mathcal{P}, \mathcal{Q}') = \left(\frac{1}{N}\right) \sum_{p,q'} \lambda(p_\theta, q'_\theta), \tag{3}$$

where $p \in \mathcal{P}$, $q' \in \mathcal{Q}'$, the location (q'_x, q'_y) is within element p in $\mathcal{P}$, and N is the number of overlapping element pairs. This function assigns a cost to each possible transformation $(\theta, \Delta x, \Delta y)$ in the parameter space (known as the cost surface). Low values indicate a high degree of consistency between the elements in each set, while high cost values are obtained when the respective orientations differ widely. For example, Figure 6e would receive a low cost, while Figure 7e would receive a high one. Although the cost of an alignment is defined in terms of the vector sets $\mathcal{P}$ and $\mathcal{Q}$, it should be kept in mind that the transformation is applied to every pixel of the query image (and its associated orientation).

4.3. Algorithms

Three diverse algorithms have been implemented for evaluation: the generalized Hough transform, alignment of distinctive local orientations, and steepest descent on the cost surface.

Generalized Hough transform

The Hough transform is a common tool in image analysis that is normally used for detecting lines in point sets. However, it can be generalized to solve other problems [23]. This algorithm is known as the generalized Hough transform, and is one approach that has been applied to minutiae-based alignment (see Section 3.1).

The basic idea behind the generalized Hough transform is that the space of all possible parameters is discretized into bins used to collect evidence for the respective transformations. Our GHT algorithm proceeds as follows: for each pair of potentially matching orientation field elements, the transformation parameters needed to align them are calculated, and evidence for these parameters is accumulated. It should be noted that orientation field elements are restricted to the range $[-(1/2)\pi, (1/2)\pi)$, so for a given pair of elements there are two possible rotations to align them (differing by π) and evidence for both must be accumulated. After this has been completed, the parameters with the most accumulated evidence are returned.

There are several implementation details to consider for the GHT algorithm. First of all, there is a tradeoff when determining the size of the accumulator bins in the discretized parameter space. If the bins are too large, the algorithm will be unable to find accurate parameters. On the other hand, if the bins are too small, the parameters may rarely coincide, leading to little evidence for any parameters. Another consideration is the accumulation of neighboring elements to account for small errors. Due to the discrete nature of the orientation field, there is a limited number of transformations that are considered. It is possible that the optimal parameters are not accumulated exactly by any of these transformations. However, these other transformations will tend to cluster around the optimal solution in the parameter space. Therefore, it is common to accumulate neighboring elements in the parameters space. This allows the full range of transformations to be considered, even though they may not have been accumulated directly.

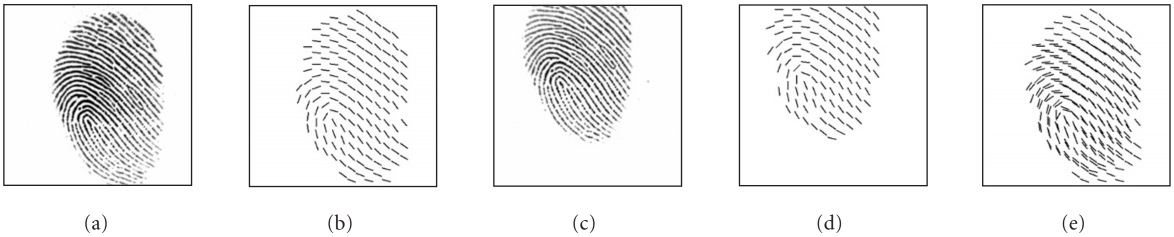

FIGURE 6: A genuine matche and the result of its registration. (a) Reference print. (b) Reference OF. (c) Query print. (d) Query OF. (e) Registration result.

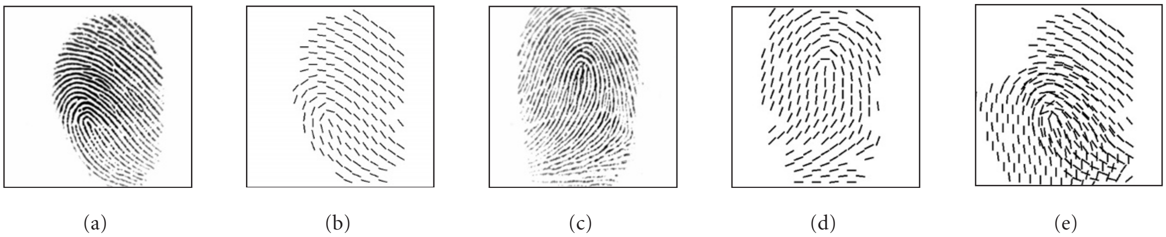

FIGURE 7: An imposter matche and the result of its registration. (a) Reference print. (b) Reference OF. (c) Query print. (d) Query OF. (e) Registration result.

Distinctive local orientations

One approach to fingerprint registration is based on using fingerprint singularities as landmarks [15]. Singularities are often located using orientation fields because the orientations surrounding core and delta points have distinctive patterns. However, there are two main drawbacks to this approach: the singularities may be difficult to extract for poor-quality images, and may not even be present if the capturing surface is small (as in Figure 4).

A more general algorithm looks for any distinctive patterns in the orientation field. This has the advantage that singularities do not need to be located. However, if they are present, it is most likely that they will be used to guide the registration.

Consider an orientation field element p with neighboring elements p^i where i is the neighbor's index for $i = 0$ to 5 (see Figure 5). A feature vector for p that characterizes the local orientations can be defined as

$$\mathbf{f}_p = \{\lambda(p_\theta, p_\theta^0), \lambda(p_\theta, p_\theta^1), \dots, \lambda(p_\theta, p_\theta^5)\}. \quad (4)$$

The feature vector (4) is translation invariant, but not rotation invariant. In order to achieve rotational invariance, we define *max* to be the index i of the $\lambda(p_\theta, p_\theta^i))$ with the greatest λ value for $i = 0$ to 5. A revised translation and rotation invariant feature vector is

$$\mathbf{f}_p = \left\{ \left\{ \left(\lambda(p_\theta, p_\theta^{(\max + i) \bmod 6} \right) \right\}_{i=0}^{5} \right\}. \quad (5)$$

For the orientation fields $\mathcal{P}$ and $\mathcal{Q}$, a feature vector is calculated for each element in the field. *Distinctive patterns* are those that have high values in the feature vector. For example, for most fingerprints large areas of the orientation field are fairly uniform, creating feature vectors with all six values close to 0. However, for nonuniform areas of the orientation field (such as those near singularities), the feature vectors will contain some high values making them distinctive. The alignment algorithm first uses a simple search to find similar distinctive patterns from $\mathcal{P}$ and $\mathcal{Q}$, and uses these pairs to calculate translation and rotation parameters. The parameters leading to the lowest cost (according to (3)) are selected.

Steepest descent

The steepest descent algorithm takes a very different approach from the previous two as it directly tries to minimize the cost function. The algorithm traces a path along the cost surface in the parameter space until a local minimum is found. The algorithm begins with an initial estimate for the parameters and evaluates its cost according to (3). Recall that (3) assigns a cost to every possible transformation $(\theta, \Delta x, \Delta y)$. The algorithm then evaluates a sample of points within close proximity in the parameter space and evaluates their costs. The parameters that give the greatest decrease in cost are selected, and this is repeated until no neighbors have a lower cost (i.e., a local optimum has been found).

The initial parameters estimate can be chosen in several ways. For this application, an approach that works well is to choose the translation parameters that align the centers of mass of the two orientation fields. This is a very fast calculation, and provides a good initial estimate. The initial rotation parameters are set to -15°, 0°, and 15°, and the best result is chosen. Once again, this is a reasonable choice if the rotation parameter is expected to be small.

5. EVALUATION AND EXPERIMENTS

The dataset used for the experiments is DB1 from the FVC2002 competition [24], which contains a wide variety of fingerprint image qualities. This database was captured using an optical sensor and contains 110 unique fingers, with 8 impressions of each finger. The competition organizers selected a set of 2800 genuinely matching pairs and 4950 nonmatching pairs for evaluation. A variety of performance measures are calculated, the details of which can be found in [24]. The algorithms have been implemented in Java and the experiments were conducted on a 2.5 GHz PC with 512 MB of RAM. The method used for calculating the orientation field is the gradient-based method presented by Bazen and Gerez [22].

The evaluation of alignment algorithms is difficult, so four different experimental methods have been devised and executed.

Visual

Visual comparisons give some indication performance (see Figures 6 and 7), however they are difficult to test using automated methods. The difficulty is largely due to the lack of ground truth registration parameters. Finding and labeling optimal alignments by hand is a very laborious process, and is somewhat subjective. This inhibits directly evaluating the results of the registration algorithms. Despite this, an extensive manual evaluation has been conducted. A random sample of 300 genuine matches were selected and registered using each algorithm. For each result, it was decided whether or not the results were a reasonably accurate registration of the two fingerprints. Due to the fact that only a portion of the database was sampled, a 95% confidence interval for the standard error is also reported.

Fingerprint verification system

A less subjective method to get a measure of performance is by constructing a fingerprint verification algorithm based on orientation field alignment. This will give some indication of the relative accuracy of the algorithms. Equation (3) is the cost function that measures the quality of an alignment. Specifically, it measures the average orientation difference between overlapping elements. A match score M between orientation fields $\mathcal{P}$ and $\mathcal{Q}'$ can be defined simply as

$$M(\mathcal{P},\mathcal{Q}') = 1 - \left(\frac{2}{\pi}\right)C(\mathcal{P},\mathcal{Q}'), \tag{6}$$

which produces a score in the range 0 to 1 (with 1 being a perfect match). We make the assumption that high scores represent correct alignments for genuine matches, and low scores are obtained when aligning imposters. If the match score is higher than a given threshold, the prints are labeled as a *match*, otherwise a *nonmatch*. A useful measure that is often used to summarize such system is the equal error rate (EER). The EER is the value that results when the false match rate is equal to the false nonmatch rate as the threshold is varied.

We can use the performance of this verification algorithm as an indirect measure of how well the prints are being aligned. However, it should be kept in mind that the following situations are possible and will affect this mode of evaluation.

(i) A correct alignment for genuine matches may be given a relatively low score for a variety of reasons (such as noise).

(ii) An incorrect alignment may be found for genuine matches and still be given a high score.

(iii) We assume that genuine matches should lead to high match scores, and similarly we assume that, in general, high match scores will not be obtained for imposter matches. However, this may not always be the case as it is possible for imposter matches to receive high scores if the overall ridge-valley patterns are similar (which is not uncommon [25]). This will affect the accuracy of the proposed verification system negatively, however it is not an indication that the registration algorithm is failing.

Due to these considerations, the EER should not be interpreted as the error rate of the registration algorithm.

This method of evaluation has the benefit of answering another interesting question; namely, how much discriminatory information is contained in a fingerprint's orientation field? Classification systems (such as Henry's classification) partition fingerprints into mutually exclusive categories based on the overall pattern in orientation fields. However, approximately 95% of prints fall into only three categories (the left loop, right loop, and whorl). Therefore, these schemes have little ability to distinguish individual prints from one another. A fingerprint identification system based solely on orientation field information may give some insight into the expected performance bounds of fingerprint classification systems.

Hybrid minutiae and orientation field verification

Another experiment has been designed to test the benefit of using orientation fields as a supplementary feature for fingerprint verification. The minutiae-based matching algorithm described in Section 3.1 based on the generalized Hough transform has been implemented. This algorithm is used for registration and a match score based on the number of corresponding minutiae is calculated [3]. Equation (6) is then used to calculate another match score based on the orientation field alignment using the registration parameters derived by the minutiae alignment algorithm. Instead of one match score, we now have two; one based on minutiae correspondences and one based on orientation field alignment. The match scores are combined by having two thresholds, and if either match score is above their respective threshold, the prints are labeled as a match. The goal of this experiment is to determine whether the EER of the system will drop with the introduction of this supplementary matching feature.

Table 1: Averages and standard deviations of the genuine and imposter match score distributions.

Algorithm	Genuine	Imposter
SD	83 ± 9	49 ± 18
GHT	80 ± 10	42 ± 20
DLO	76 ± 16	37 ± 22

Table 2: Algorithm accuracy, EER, and time.

Algorithm	Accuracy	EER	Average run time (ms)
SD	98.9% ± 1.1%	8.6%	215
GHT	95.2% ± 2.5%	9.6%	4510
DLO	93.8% ± 2.9%	12.4%	6

Visual comparison with minutiae registration results

As mentioned above, a minutiae-based registration algorithm has been implemented. A final experiment was conducted to compare the results of its registration results with the methods proposed in this paper. For each of the 2800 genuine matches, registration parameters were calculated using the minutiae-based algorithm and the SD algorithm proposed in this paper. The parameters generated by these algorithms were compared with each other, and when they were close it was assumed that they represented an accurate alignment of the prints. This is a valid assumption because the features used for registration are completely different, and so it is very unlikely that both methods would chose similar, but erroneous, parameters. For prints that generated significantly different registration parameters, a visual examination classified this result as one of the following: the SD algorithm is correct while the minutiae algorithm is incorrect, the minutiae algorithm is correct and the SD algorithm is incorrect, or both the SD and minutiae algorithms are incorrect. By comparing the size of these classes, this mode of evaluation allows a direct comparison between the methods proposed in this paper and the more traditional minutiae-based methods.

6. RESULTS AND DISCUSSION

Table 1 contains some statistics on the match scores obtained by the registration algorithms. As expected, the average match score for genuine matches are higher than for imposter matches. Furthermore, the standard deviations are smaller, showing that most genuine matches receive a high score, while imposter match scores are more variable. This is not surprising as it is certainly reasonable for imposter matches to receive a variety of high and low scores depending on arbitrary similarities between the orientation fields.

Table 2 contains the results of the accuracy experiments by manual evaluation. The EERs for the verification experiments and average running times for the algorithms can also be found in Table 2. The running times listed do not include the time necessary for preprocessing and feature extraction.

The best performance in terms of accuracy is attained by the SD algorithm. This is confirmed by both the manual evaluation and the lower EER. The choice of initial parameters seems to work very well, allowing the SD algorithm to quickly find a good solution without getting stuck in local minimums. The average running time for this algorithm is reasonable; not as quick as DLO, but much less that GHT.

It is somewhat surprising that the SD algorithm outperforms the GHT algorithm since the GHT algorithm conducts the most extensive search of the parameter space. One reason that the GHT sometimes fails is that it does not give any preference to alignments with low rotation parameters. Therefore, it sometimes finds alignments with a low cost by using a large rotation. However, in practice the rotation parameters are usually relatively small. Our experiments have shown that for the FVC2002 DB1 database, most valid registration rotations are within ±20°. Since SD's initial parameters are -15°, 0°, and 15°, it has a preference for alignments with small rotations (which are more often correct). Another factor is that the SD algorithm directly tries to minimize the cost function. Conversely, the GHT algorithm does not consider the cost; it attempts to maximize the number of consistent orientation field elements. The former seems to be a better approach.

An obvious disadvantage of the GHT algorithm is its slow running time of almost 5 seconds. This running time will increase quickly ($\mathcal{O}(n^2)$) as the number orientation field elements n increases (e.g., due to larger prints or smaller elements). Furthermore, the GHT algorithm has a large space complexity. A three-dimensional accumulator array must be set up, and this grows exponentially as the accumulator elements get smaller. For example, consider a parameter space that is discretized to 360 degrees for the rotation parameter, 250 pixels for each of the x and y translation parameters, and contains a two-byte counter. This would require over 40 MB of memory, making it infeasible for environments with limited resources (such as smart cards).

The accuracy and EER results of the DLO algorithm are poor in comparison with the other algorithms. The primary advantage of this method is that it is very fast and it requires little memory. The feature vectors for the orientation elements can be calculated quickly, and efficient search algorithms can be used to find potential matches. Recovering the transformation parameters is also a very fast computation. Another advantage is that it does not require too much overlap between the prints. As long as some mutual distinctive patterns exist, the algorithm will be successful. However, the algorithm seems to suffer from the same problems (although not to the same degree) as algorithms based on aligning singularities. Since singularities create the most distinctive orientation patterns, if they do not exist or are not captured, there is little information to guide the alignment. For example, consider the fingerprint and orientation field shown in Figure 8. In this case, a large area of the fingerprint has been captured, but the orientation field is relatively homogeneous. The DLO algorithm struggles in situations such as

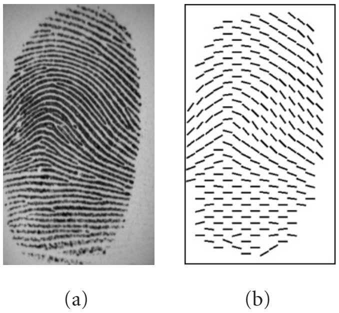

Figure 8: A fingerprint (a) whose orientation field (b) does not contain any distinctive local patterns.

these, while the GHT and SD algorithms perform well. The DLO approach is based entirely on local patterns, whereas the other two use information from the whole print to guide the alignment. The results suggest that the global approach is more effective.

The EER rate obtained by the SD algorithm is comparable with those from some other verification systems in the literature. For example, a recent paper published an EER of 8.36% for a system using a directional filter bank [26]. The authors also implemented a system using a Gabor filter bank [18], and obtained an almost identical EER of 8.37%. Both of these results are comparable to the results obtained by the SD algorithm, despite being much more complicated.

The results of the hybrid minutiae and orientation field verification systems are as follows. Using only minutiae correspondences, the EER of the system was 7.5%. However, after the introduction of the orientation field features, the EER dropped to 3.4%. This is a significant improvement, and demonstrates the profit that results from using orientation fields as a matching feature.

Finally, experiments were conducted to compare the registrations generated by the SD algorithm with those generated by a minutiae-based algorithm. Out of the 2800 genuine matches tested, the two methods agreed (within a given tolerance threshold) on 96% of the registrations. Of the remaining 4%, 45% of the orientation field registrations were correct, 38% of the minutiae registrations were correct, and for the remaining 17% both algorithms failed (this is equivalent to 0.7% all genuine matches). One point that should be mentioned is that in most cases the minutiae results are more accurate than the proposed algorithms (due to the averaging nature of an orientation field). The two methods have different strengths and weaknesses. For example, the orientation field algorithm struggles with prints that have circular patterns (i.e., whorl prints). In these cases it is difficult to determine the correct rotation parameter. These prints do not pose a problem for the minutiae-based algorithm. Conversely, the minutiae algorithm often fails for noisy prints (with unreliable minutiae information) or prints with a small number of minutiae in the area of overlap. A combination of these factors can cause both algorithms to fail (see Figure 9a), however the most common cause of failure for both algorithms is a small amount of overlap between the prints (see Figures 9b and 9c).

7. CONCLUSIONS AND FUTURE DIRECTIONS

A novel method of fingerprint registration has been presented based on the alignment of orientation fields. Three algorithms implementing this concept have been presented, and their performance has been evaluated using an FVC2002 database.

All three of the algorithms presented have been conceptually simple, yet effective. Overall, the SD algorithm has the best performance, with an accuracy of approximately 98.9%. The SD algorithm is also reasonably fast. Therefore, this algorithm is superior to the DLO and GHT algorithms, and would be the preferred method for most practical applications. The primary application of the algorithms presented in this paper is for any system requiring coarse alignment parameters. The algorithms will be particularly useful when the fingerprint image quality is unreliable.

This research also demonstrates an interesting application beyond fingerprint registration. Many identification systems consider only minutiae features when matching prints. Therefore, it is possible that imposter prints are labeled as matches even when the overall shape of the ridge patterns is very different. A simple but effective modification to these systems is to include orientation fields as an additional matching feature. Combining independent feature sets for identification improves the performance of such systems significantly. In our experiments, the EER of a strictly minutiae-based verification system was reduced from 7.5% to 3.4%. In most cases the orientation field will have already been calculated during preprocessing, so the additional computation is negligible.

Classification schemes based on categorizing the overall orientation field pattern into mutually exclusive classes (such as Henry's classification) are difficult to implement due to large intraclass and small interclass variation [25]. Even when their performance is high, these experiments show that there is more discriminatory information available in orientation fields than is currently being exploited. Some continuous classification schemes based on orientation fields have been suggested [27, 28]. The results from this paper support that this is a promising approach.

One idea to extend this research would be to develop registration methods that are guided by both orientation fields and minutiae. In the experiments, it was shown that 99.3% of genuinely matching pairs could be registered correctly by at least the minutiae or SD algorithm. However, neither algorithm in isolation could achieve this accuracy. Therefore, an algorithm that exploits both feature sets has the potential to be very powerful. Another future direction is to develop methods to fine-tune the coarse registration parameters generated by these algorithms. This can be done by using either minutiae or nonminutiae features. Finally, the ultimate extension of this research would be to develop nonrigid registration algorithms that model the nonlinear distortions present in fingerprints.

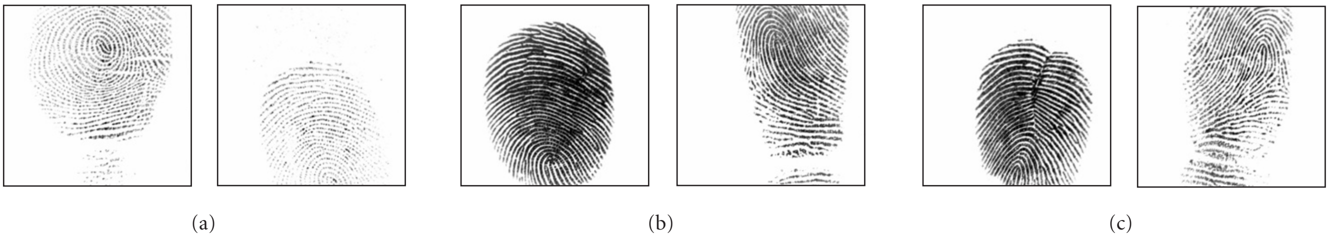

Figure 9: Genuine matches that could not be registered by neither the SD algorithm nor the minutiae-based algorithm.

REFERENCES

[1] A. K. Jain, L. Hong, and S. Pankanti, “Biometrics: promising frontiers for emerging identification market,” *Communications of the ACM*, vol. 43, no. 2, pp. 91–98, 2000.

[2] N. Yager and A. Amin, “Fingerprint verification based on minutiae features: a review,” *Pattern Analysis and Applications*, vol. 7, no. 1, pp. 94–113, 2004.

[3] N. K. Ratha, K. Karu, S. Chen, and A. K. Jain, “A real-time matching system for large fingerprint databases,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 8, pp. 799–813, 1996.

[4] A. K. Hrechak and J. A. McHugh, “Automated fingerprint recognition using structural matching,” *Pattern Recognition*, vol. 23, no. 8, pp. 893–904, 1990.

[5] Z. Chen and C. H. Kuo, “A topology-based matching algorithm for fingerprint authentication,” in *Proc. 25th Annual IEEE International Carnahan Conference on Security Technology*, pp. 84–87, Taipei, Taiwan, October 1991.

[6] A. Wahab, S. H. Chin, and E. C. Tan, “Novel approach to automated fingerprint recognition,” *IEE Proceedings on Vision, Image and Signal Processing*, vol. 145, no. 3, pp. 160–166, 1998.

[7] D. P. Mital and E. K. Teoh, “An automated matching technique for fingerprint identification,” in *Proc. IEEE Conference on Emerging Technologies and Factory Automation (EFTA ’96)*, vol. 1, pp. 87–92, Kauai, Hawaii, USA, November 1996.

[8] X. Jiang and W.-Y. Yau, “Fingerprint minutiae matching based on the local and global structures,” in *Proc. 15th IEEE International Conference on Pattern Recognition (ICPR ’00)*, vol. 2, pp. 1038–1041, Barcelona, Spain, September 2000.

[9] N. K. Ratha, R. Bolle, V. D. Pandit, and V. Vaish, “Robust fingerprint authentication using local structural similarity,” in *Proc. 5th IEEE Workshop on Applications of Computer Vision (WACV ’00)*, pp. 29–34, Palm Springs, Calif, USA, December 2000.

[10] R. Qun, T. Jie, H. Yuliang, and C. Jiangang, “Automatic fingerprint identification using cluster algorithm,” in *Proc. 16th IEEE International Conference on Pattern Recognition (ICPR ’02)*, vol. 2, pp. 398–401, Quebec City, Quebec, Canada, August 2002.

[11] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, “An identity-authentication system using fingerprints,” *Proc. IEEE*, vol. 85, no. 9, pp. 1365–1388, 1997.

[12] A. K. Jain, L. Hong, and R. Bolle, “On-line fingerprint verification,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 4, pp. 302–314, 1997.

[13] R. S. Germain, A. Califano, and S. Colville, “Fingerprint matching using transformation parameter clustering,” *IEEE Comput. Sci. Eng. Mag.*, vol. 4, no. 4, pp. 42–49, 1997.

[14] Z. M. Kovacs-Vajna, “A fingerprint verification system based on triangular matching and dynamic time warping,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 11, pp. 1266–1276, 2000.

[15] W. Zhang and Y. Wang, “Core-based structure matching algorithm of fingerprint verification,” in *Proc. 16th IEEE International Conference on Pattern Recognition (ICPR ’02)*, vol. 1, pp. 70–74, Quebec City, Quebec, Canada, August 2002.

[16] M. Tico and P. Kuosmanen, “Fingerprint matching using an orientation-based minutia descriptor,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 25, no. 8, pp. 1009–1014, 2003.

[17] M. Tico and P. Kuosmanen, “A remote authentication system using fingerprint,” *International Journal of Image and Graphics*, vol. 3, no. 3, pp. 425–446, 2003.

[18] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, “Filterbank-based fingerprint matching,” *IEEE Trans. Image Processing*, vol. 9, no. 5, pp. 846–859, 2000.

[19] C.-H. Park, J.-J. Lee, and K.-H. Park, “Fingerprint matching based on directional image feature in polar coordinate system,” in *Proc. 10th International Conference on Computer Analysis of Images and Patterns (CAIP ’03)*, vol. 2756 of *Lecture Notes in Computer Science*, pp. 293–300, Groningen, the Netherlands, August 2003.

[20] A. Ross, J. Reisman, and A. K. Jain, “Fingerprint matching using feature space correlation,” in *Proc. Post-European Conference on Computer Vision Workshop on Biometric Authentication*, vol. 2359 of *Lecture Notes in Computer Science*, pp. 48–57, Copenhagen, Denmark, June 2002.

[21] J. Gu and J. Zhou, “A novel model for orientation field of fingerprints,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR ’03)*, vol. 2, pp. 493–498, Madison, Wis, USA, June 2003.

[22] A. M. Bazen and S. H. Gerez, “Systematic methods for the computation of the direction fields and singular points of fingerprints,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 7, pp. 905–919, 2002.

[23] D. Ballard, “Generalized hough transform to detect arbitrary patterns,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 3, no. 2, pp. 111–122, 1981.

[24] FVC2002, November 2003, http://bias.csr.unibo.it/fvc2002/.

[25] N. Yager and A. Amin, “Fingerprint classification: a review,” *Pattern Analysis and Applications*, vol. 7, no. 1, pp. 77–93, 2004.

[26] C.-H. Park, J.-J. Lee, M. J. T. Smith, S.-I. Park, and K.-H. Park, “Directional filter bank-based fingerprint feature extraction and matching,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 74–85, 2004.

[27] A. Lumini, D. Maio, and D. Maltoni, “Continuous versus exclusive classification for fingerprint retrieval,” *Pattern Recognition Letters*, vol. 18, no. 10, pp. 1027–1034, 1997.

[28] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, “Fingerprint classification by directional image partitioning,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 5, pp. 402–421, 1999.

Neil Yager was born in Sydney, Australia, in 1978. He moved to Canada at a very early age, but has since returned to Australia. He is currently a Ph.D. student in the School of Computer Science and Engineering, University of New South Wales (UNSW), Sydney. His primary research interests are in the fields of image analysis and pattern recognition. In particular, his emphasis is on biometrics and automated fingerprint identification. He has obtained a B.S. degree in computer science from the UVic, Victoria, Canada, in 2000, and a M.Eng.Sc. degree from UNSW in 2002.

Adnan Amin received his Doctorate d'Etat (D.Sc.) in computer science from the University of Nancy I, France, in 1985. From 1981 to 1985, Dr. Amin was an Assistant Professor at the University of Nancy II. From 1985 to 1987, he worked in INTEGRO, Paris, as Head of Pattern Recognition Department. From 1987 to 1990, he was an Assistant Professor at Kuwait University and joined the School of Computer Science and Engineering, University of New South Wales (UNSW), Australia, in 1991 as a Senior Lecturer. Dr. Amin's research interests are in pattern recognition, document image analysis and recognition, neural networks, and machine learning. He has written more than 100 technical papers in these areas. He is an Associate Editor for the International Journal on Document Analysis and Recognition, Pattern Analysis and Applications, International Journal of Pattern Recognition and Artificial Intelligence, and International Journal of Image and Graphics. He was on the program committee of many conferences and served as a referee for numerous journals and scientific organizations. He is a Member of IEEE, ACM, and IAPR Societies.