Pattern Anal Applic
DOI 10.1007/s10044-009-0160-3

THEORETICAL ADVANCES

A comparative study on feature extraction for fingerprint classification and performance improvements using rank-level fusion

Uday Rajanna · Ali Erol · George Bebis

Received: 13 May 2008 / Accepted: 11 September 2008

Abstract Fingerprint classification represents an important preprocessing step in fingerprint identification, which can be very helpful in reducing the cost of searching large fingerprint databases. Over the past years, several different approaches have been proposed for extracting distinguishable features and improving classification performance. In this paper, we present a comparative study involving four different feature extraction methods for fingerprint classification and propose a rank-based fusion scheme for improving classification performance. Specifically, we have compared two well-known feature extraction methods based on orientation maps (OMs) and Gabor filters with two new methods based on "minutiae maps" and "orientation collinearity". Each feature extraction method was compared with each other using the NIST-4 database in terms of accuracy and time. Moreover, we have investigated the issue of improving classification performance using rank-level fusion. When evaluating each feature extraction method individually, OMs performed the best. Gabor features fell behind OMs mainly because their computation is sensitive to errors in localizing the registration point. When fusing the rankings of different classifiers, we found that combinations involving OMs improve performance, demonstrating the importance of orientation information for classification purposes. Overall, the best classification results were obtained by fusing orientation map with orientation collinearity classifiers.

Keywords Fingerprint classification · Orientation field · Minutiae map · Orientation collinearity · Gabor features · Rank-level fusion

U. Rajanna (✉) · A. Erol · G. Bebis
Computer Vision Laboratory, University of Nevada, Reno, NV, USA
e-mail: rajanna@cse.unr.edu

A. Erol
e-mail: aerol@cse.unr.edu

G. Bebis
e-mail: bebis@cse.unr.edu

1 Originality and contribution

This paper presents a comparative study involving four different feature extraction methods for fingerprint classification. In addition, it presents a rank-based fusion scheme for improving classification performance. Our work is original and contributes to improving research on fingerprint classification.

2 Introduction

Fingerprint matching is among the most important and reliable methods for the identification of a person. There are two main applications involving fingerprint matching: fingerprint verification and fingerprint identification. While the goal of fingerprint verification is to confirm the identity of a person, the goal of fingerprint identification is to establish the identity of a person. In general, fingerprint identification involves comparing a query fingerprint with a large number of fingerprints stored in a database, which is time consuming. To reduce search time and lower computational complexity, fingerprint classification is often employed to partition the database into smaller subsets [1]. The key idea is assigning a given fingerprint to a broad category using high-level features such as ridge density and ridge direction. During identification, a query fingerprint

Published online: 28 April 2009

Fig. 1 Major Henry classes: **a** Whorl (W), **b** Left loop (L), **c** Right loop (R), **d** Arch (A), **e** Tented Arch (T)

needs to be matched only against fingerprints belonging to the same category with the query.

A recent review on fingerprint classification methods can be found in [2]. Commonly, fingerprints are classified into five major classes, known as Henry classes, namely *Whorl*, *Left Loop*, *Right Loop*, *Arch* and *Tented Arch* (see Fig. 1). Non-linear distortions resulting from skin elasticity, sensor noise and the presence of intrinsically low-quality fingerprint images make fingerprint classification a challenging problem. These distortions result in small inter-class and large intra-class variability among the five different Henry classes. Therefore, several systems have been proposed to deal with this challenging problem [3].

Extracting a set of distinguishable features is critical for fingerprint classification. The main feature that defines the Henry classes is the ridge flow pattern, which in principle can be characterized by the number and types of singularities in the direction field (i.e., ridge flow field). Core and delta points are the main features used in rule-based approaches such as the one proposed by Kawagoe and Tojo [4]. However, these systems suffer from failures when the singularities are missing or cannot be extracted. For example, core and delta points may be missing due to incorrect placement of the finger or might not be extracted reliably due to low image quality.

Many studies rely on more robust features representing the global or local ridge patterns in fingerprints. Wang [5] has introduced one such approach making use of orientation field information. In [6], ridges represented by B-spline curves were employed for the same purpose. A structural approach using partitioning of the orientation field into homogeneous regions has been proposed in [7, 8]. Prabhakar et al. [9] proposed a set of Gabor features showing promising results. In general, ridge flow or structure-based features have proven to be the most reliable and accurate means for automated fingerprint classification, because they are inherently more tolerant to noise.

Nevertheless, there have also been efforts to employ other features. In [10], Fitz et al. introduced frequency-based features to perform classification. However, their method was tested on a very small data set of 40 fingerprints, which does not give a good indication of the generalization ability and robustness of the algorithm. Several systems making use of minutiae information in fingerprints have also been proposed. One such study presented a feature extraction method based on the position, location and orientation associated with minutiae points [11]. In a different study [12], genetic programming was used to learn a set of features for classification.

This study presents a comparative analysis of several different feature extraction methods for fingerprint classification. Specifically, we compared two well-known methods, based on orientation maps (OMs) and Gabor filters, with two other methods, introduced in this study, based on minutiae maps (MMs) and orientation collinearity. To compare each approach, we used a k-nearest neighbor (k-NN) classifier as in [9]. Besides evaluating each feature extraction method individually, we also investigated the issue of improving the accuracy of fingerprint classification using rank-level fusion. We report improved classification results by fusing OMs with orientation collinearity. Our experiments were conducted using the NIST-4 database, which has now become a benchmark database in literature for testing fingerprint classification techniques.

The rest of the paper is organized as follows: Sect. 2 reviews the feature extraction methods used in our

comparison study. Section 3 presents our classification approach using *k*-NN classifiers. The data set used in our experiments is described in Sect. 4. Our experimental results using each feature extraction method individually as well as using rank-level fusion are presented in Sect. 5. Finally, our conclusions and plans for future research are presented in Sect. 6.

3 Feature extraction methods

Based on previous work on fingerprint classification, we chose to compare two well-known feature extraction methods based on OMs [13] and Gabor filters [9]. In addition, we considered two more methods, introduced in this study, based on MMs and orientation collinearity. MMs have been motivated by the ideas presented in [2], while the main idea behind orientation collinearity is a coarse representation of OMs. Each of the feature extraction methods considered here requires a registration step to provide translation invariance. In this study, translation invariance was achieved using core detection. Ideally, one should also account for rotation changes; however, all fingerprints in the NIST-4 database have already been normalized with respect to rotation. Next, we describe the core detection and feature extraction algorithms used in this study.

3.1 Core point detection

Accurate and reliable core point detection in fingerprint images is a critical issue that affects the performance of many fingerprint classification and recognition systems. Several different algorithms have been proposed in literature for detecting and extracting the core points reliably. In this study, we experimented with two known core point extraction algorithms and a hybrid approach.

The first algorithm extracts the core points using a method similar to the one reported by Novikon and Kot [14]. According to this method, a core point is defined as the crossing point of lines normal to the ridges as shown in Fig. 2. Detecting the crossing point is very robust; however, it does not always lie close to the true core point. The second algorithm extracts the core points using the Poincaré index [5]. The Poincaré index is a tool for detecting and classifying singularities in vector fields. It can be applied on fingerprint OMs with minor modifications and has shown to have high accuracy, but low robustness.

To improve core point extraction, we also experimented with a hybrid method, which reduces false positives by combining the outputs of the above two algorithms. In particular, although the Poincaré index method is very accurate, it can produce many false core and delta points when the orientation map is noisy. We used several

Fig. 2 Core point by intersection of ridge normals

heuristics to filter out false core points based on their location relative to the crossing point and the boundary of the image. In particular, we discarded the core points that had the distance from the image boundary or the crossing point less than a fixed number of pixels (e.g., 20 pixels). Among the remaining Poincaré index-based singularities, we took the one closest to the crossing point as the registration point.

3.2 Minutiae maps

Minutiae refer to the bifurcation or termination points of ridges on the finger surface. They are mainly utilized in fingerprint matching since their distribution on the fingerprint provides a unique signature for an individual [15]. Ross et al. [11] have investigated the problem of reconstructing fingerprint images from the minutiae locations and directions. Our motivation to use MMs in this study was to examine whether minutiae contain enough information for fingerprint classification. Our experimental results indicate that there is some correlation between the distribution of minutiae and corresponding fingerprint classes; however, this information alone is not sufficient for highly accurate fingerprint classification.

In our experiments, we extract the minutiae using the Verifinger library tool kit [16]. Then, we represent them using their X, Y image coordinates as well as their orientation (i.e., average direction of surrounding ridges). To represent minutiae distribution information, we detect the core point and define a region of interest around it. In this study, we assume a circular region centered at the core point. Then, we tessellate the circular region using a methodology similar to [9]. To determine the radius of the circular region, we do not consider the whole image, since there is high noise around the image boundary. Moreover, most useful information is contained around the core point.

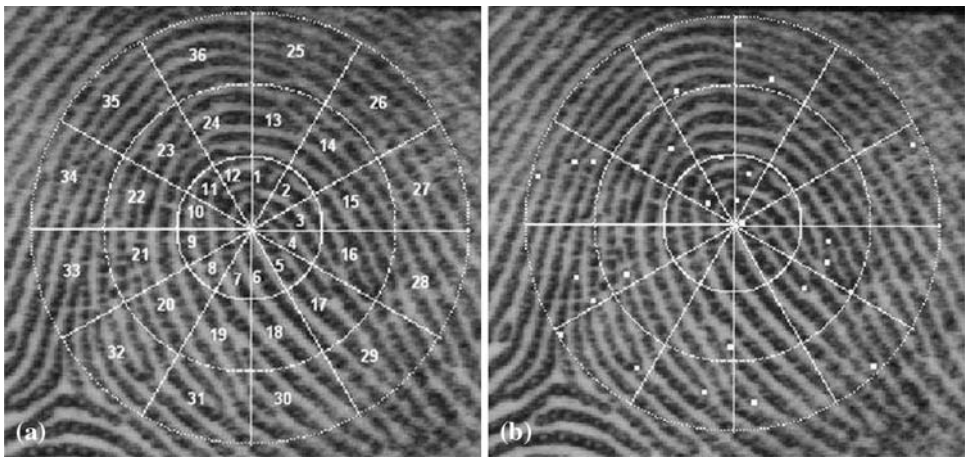

Fig. 3 Spatial tessellation and detected minutiae overlaid on a fingerprint image. In polar coordinates, the angle and radius axes are quantized in steps of 30 and 40 pixels, respectively

Figure 3 shows a tessellation example along with the minutiae overlaid on the fingerprint image. Two types of features are extracted from each sector: (1) the number of minutiae inside the sector, normalized by the total number of minutiae inside the largest circle, and (2) the average minutiae orientation within each sector.

3.3 Orientation maps

Orientation maps describe the ridge flow in a fingerprint image. Usually, the image is partitioned into non-overlapping square blocks and each block is processed to determine the dominant orientation inside it as shown in Fig. 4. This forms an important representation that serves many purposes. In the classification domain, the singularities of the field are helpful in registering translation-variant features. After registration, the map itself contains important information for determining the class of a fingerprint. Here, a square-shaped sub-region, centered at the core point, is used to form the feature vector.

Fig. 4 Orientation map

In our experimentation and analysis, we employed two different orientation estimation methods. The first one is based on the PCASYS algorithm [17], which operates in the frequency domain. In this case, the frequency spectrum of each block is analyzed to determine the strongest orientation inside the block. The PCASYS algorithm uses 16×16 blocks, therefore, it computes low-resolution OMs (i.e., 32×32 in our case). The second method estimates the orientation map using a gradient-based approach, which is the most common and well-known methodology [13]. In this case, the gradient vectors inside each block are analyzed to determine the dominant orientation in each block. The unit vector that is most orthogonal to the gradient vectors inside each block gives the dominant orientation. The gradient-based method can produce higher resolution maps (i.e., 64×64 in our case) since it uses 8×8 blocks.

It should be mentioned that orientation information cannot be estimated in background regions of the fingerprint images. These regions can be detected using the variance of the intensity in each block; when the variance falls below a threshold, the block is flagged as background. We utilize this information in our distance calculations. Specifically, we have adopted two different schemes for computing the distance between two OMs. In the first scheme, we consider all available information, including that of the background blocks. In this case, the distance is calculated as follows:

$$D(x,y) = \sum_{i=1}^{N} 1 - |\cos(x[i] - y[i])| \quad (1)$$

where x and y denote the feature vectors.

In the second scheme, the background blocks were regarded as "don't care" components and were not considered in our distance calculations. That is, only the average distance between corresponding non-background components was taken into consideration as follows:

$$D(x,y) = \left(\sum_{i \in N} 1 - |\cos(x[i] - y[i])| \right) / |N| \quad (2)$$

where N denotes the set of feature vector components without a "don't care" value in x and y.

3.4 Orientation collinearity maps (OCMs)

Several studies, including [9], have reported that OMs are quite sensitive to noise. The main motivation for introducing OCMs was to obtain a coarser representation of orientation information, which would be less sensitive to noise. In this context, OCMs take into consideration the inherent continuity that exists between adjacent cells in OMs. Specifically, by examining the average orientation information of adjacent cells, we assign a label to each cell based on the degree of collinearity of the corresponding orientation directions. This process allows us to build a set of templates for each fingerprint class, which are then used for classification purposes. This is illustrated in Fig. 5 where each cell corresponds to a block used in orientation map estimation.

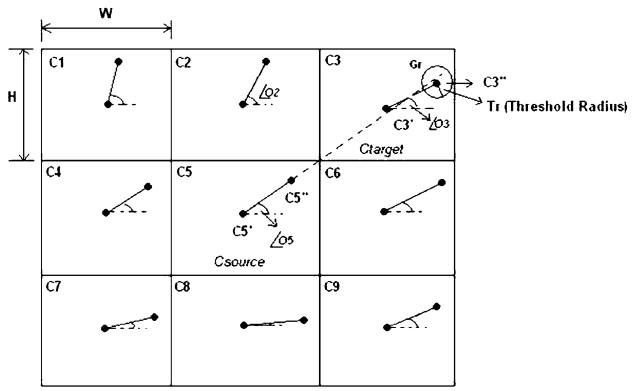

Fig. 5 Illustration of orientation collinearity

Specifically, let us consider a 3×3 neighborhood of blocks $\{C_1, \ldots, C_9\}$ and denote the unit orientation vectors at the center of each block C_i as $C_i^{'}, i = 1, \ldots, 9$. The main idea is examining whether the orientation vector corresponding to cell C_5 is collinear with the orientation vectors corresponding to the cells C_i in the neighborhood of C_5 as illustrated in Fig. 5. Once we have determined that the orientation of the center block is consistent with that of a surrounding block, we label both of them using one of four labels corresponding to the direction of the line connecting them (i.e., 0, 45, 90, 135).

The resulting labels can be visualized for different Henry classes in Fig. 6. As it can be observed, orientation collinearity creates easy-to-distinguish patterns for different classes. We construct four different feature vectors from the collinearity labels using a circular tessellation around the core point similar to the case of MMs (see Fig. 3). The first feature vector corresponds to all the labels inside the largest circle. The other feature vectors are constructed using local features of sectors in the tessellation (i.e., the mean, the median, and the mode of the labels inside each sector).

3.5 Gabor feature maps (GFMs)

In [9], Prabhakar et al. proposed using a Gabor filter bank for feature extraction. Their system uses a circular tessellation (see Fig. 3) centered at a point 40 pixels below the core point. The amount of shift from the true core point was determined experimentally and the resulting tessellation was argued to contain more class information. In each sector, the features extracted were the outputs of the Gabor filters. It should be mentioned that extracting Gabor

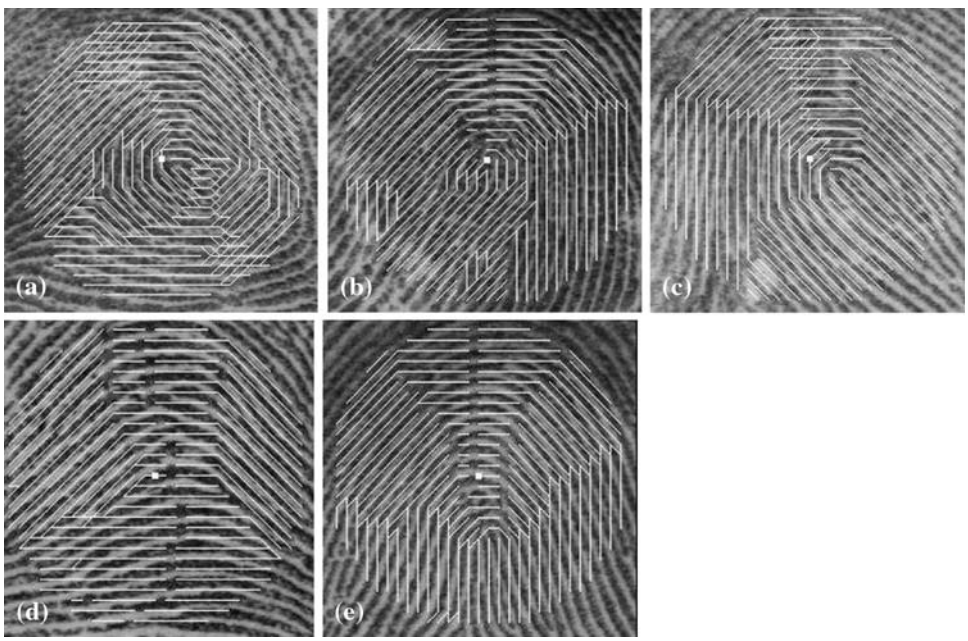

Fig. 6 Visualization of orientation collinearity labels for different Herny classes: **a** Whorl (W), **b** Left loop (L), **c** Right loop (R), **d** Arch (A), **e** Tented Arch (T)

features is more time consuming than any of the other three methods as discussed in Sect. 5.

During preprocessing, the fingerprint image was first normalized by normalizing the contrast in each sector of the tessellation. Then, a bank of Gabor filters tuned to equally spaced orientations were applied. The variance of the Gabor filter outputs in each sector was used to construct a feature vector. More details can be found in [9]. In our implementation, we used the same parameters (e.g., Gabor filter variance, angular and radial quantization steps, etc.) provided in [9] to extract the Gabor features. However, the core point detection algorithm used was different from the one used in [9].

4 Classification approach

We employed a k-NN classifier as a common platform not only to compare the different feature extraction methods, but also to produce comparable results with those reported in [9]. This is a simple classifier, which has its roots in non-parametric estimation [18]. The k-NN rule first finds the k nearest neighbors to an input pattern in the feature space. Then, it assigns the input pattern to the class which is more frequently represented among the k nearest neighbors. Alternatively, the top two classes can be retrieved by finding the classes that have the highest and second highest counts among the nearest neighbors.

It should be mentioned that the methodology presented in [9] employs a two-stage classification scheme. The idea is to decompose the five-class problem into a set of 10 two-class problems. According to this scheme, the first stage employs a k-NN classifier, while the second stage employs 10 neural network (NN) classifiers. Given an input, the purpose of the first stage is to choose the two, most likely, classes using a k-NN classifier. In the second stage, the appropriate NN is chosen to distinguish between the two most likely classes.

In this study, all feature extraction methods have been compared using a one stage k-NN classifier. In each experiment, we evaluated the ability of the k-NN classifier to predict the correct class to a given input by considering both the top and top two classes. Although we did not experiment with a two-stage classification scheme, it would be reasonable to assume that adding a second stage would improve accordingly the classification performance of all the methods compared in this study.

5 Data set

In our experiments, we used the NIST-4 database, which consists of 4,000 512 × 512 images of rolled fingerprint impressions scanned at 500 dpi. Each finger in the database has two impressions. The images in the NIST-4 database are numbered $f0001$ through $f2000$ and $s0001$ through $s2000$. Each number represents a different finger and the prefixes f and s denotes the first and second impressions of the same finger. Each image is manually labeled with one or more of the five classes shown in Fig. 1. Almost 17% of the images in the database have more than one class labels. We form our training set using the first impressions and the test set consists of the second impressions In the training set, we make use of only the first label. During testing, however, we use all the labels and consider the output of the classifier to be correct if it matches with any one of the labels of the test fingerprint. This testing scheme is consistent with common practices followed by other researchers, including [9], in comparing classification results on the NIST-4 database.

It is worth noting that the NIST-4 database does not have a natural class distribution. Since the frequency of the hardest to distinguish classes (i.e., arch and tented-arch) are much lower than the others, the accuracy figures reported here should be expected to be higher on more realistic data sets.

6 Experimental results

Next, we present our experimental results and comparisons by considering each feature extraction method individually as well as by fusing them using rank-level fusion.

6.1 Results using MMs

Figure 7a, b illustrates classification accuracy using MMs. The vertical axis in each graph corresponds to classification accuracy, while the horizontal axis represents the number of nearest neighbors. In our experiments, some of the samples were rejected due to invalid spatial tessellations. An invalid tessellation occurs when it falls outside the image; that is, the detected core point is close to the boundary. In MMs, we used a maximum radius of tessellation equal to 120 pixels. In this case, the total rejection rate was 1.77% with 36 training samples and 35 testing samples rejected.

The curves shown in Fig. 7a, b are for different tessellation parameters dR and dA, which denote the angular and radial quantization steps, respectively. The parameters that gave the best results were $dR = 40$ pixels and $dA = 30$. When $k = 10$, the top-two classes of accuracy is close to 86%. Although these results are not satisfactory, they do indicate that there is a correlation between minutiae distribution and fingerprint classes.

6.2 Results using OMs

Using OMs, we tested the two different distance measures discussed in Sect. 2.3. Our results indicated that discarding the background blocks improves accuracy as shown in

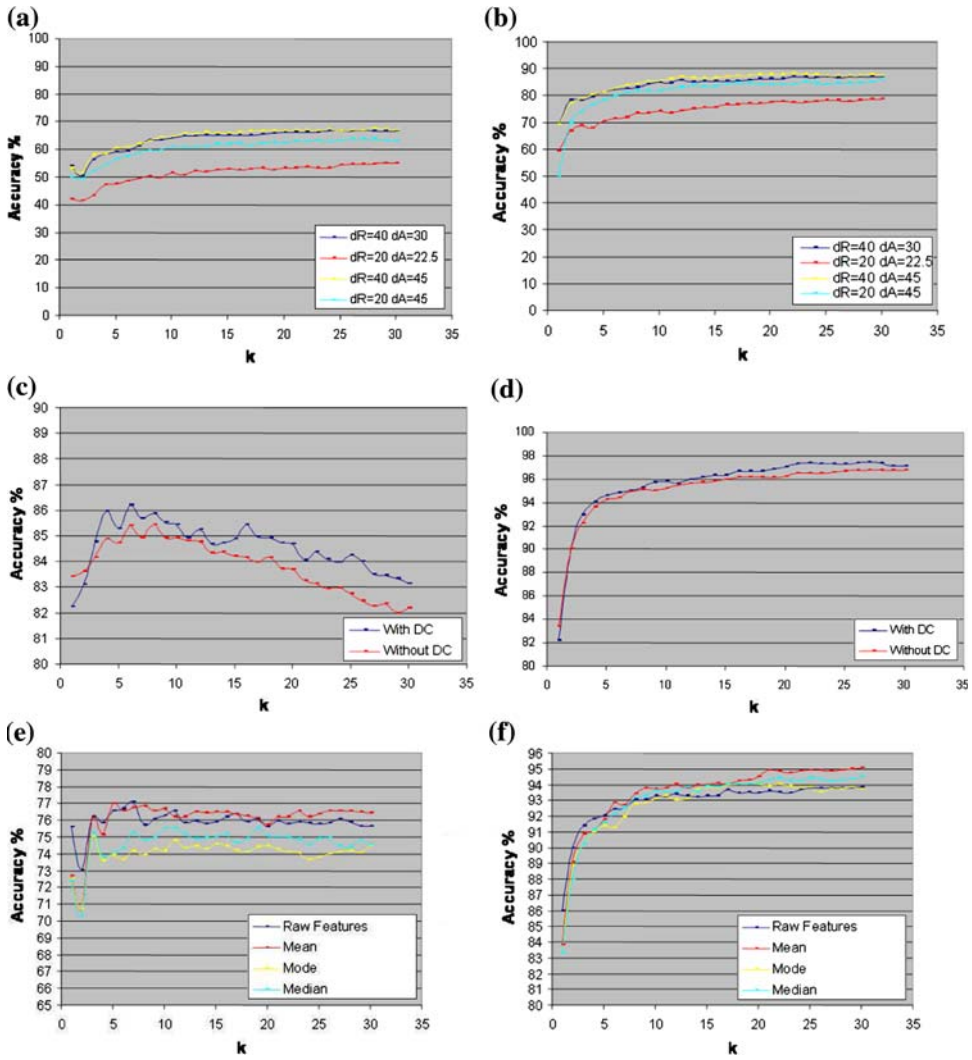

Fig. 7 Results using **a** minutiae maps: top-class accuracy, **b** minutiae maps: top-two-classes accuracy, **c** orientation maps: top-class accuracy, **d** orientation maps: top-two-classes accuracy, **e** orientation collinearity: top-class accuracy, and **f** orientation collinearity: top-two-class accuracy. *DC* Don't care

Fig. 7c, d. The total rejection rate was 1.75% with 37 training images and 33 testing images rejected. The maximum radius of spatial tessellation was set to 120 pixels. The feature vectors consisted of 14×14 sub-regions centered at the core point. The results obtained illustrate the significance of orientation information for fingerprint classification. It is worth mentioning that OMs have been found to be sensitive to noisy images in the NIST-4 database [9]. However, our results indicate that OMs perform quite well using rejection rates similar to those in [9].

6.3 Results using OC

Figure 7e, f reports classification accuracy using orientation collinearity with different attributes of spatial tessellation (i.e., mean, mode, median and raw features). The parameters used for the spatial tessellation were $dA = 18$ and $dR = 40$ pixels. These results indicate that the mean orientation in each sector gives the best performance. Using the above quantization parameters, the size of the resulting feature vectors was 60, which is much smaller than the 192-dimensional feature vectors used in [9]. When $k = 10$, the top-class accuracy was close to 77%, while the top-two classes accuracy was 93.7%. These results alone are not as good as the ones reported in literature (i.e., [6, 9, 19]); however, one has to keep in mind the lower dimensionality of the feature vectors.

6.4 Results using Gabor features

In these experiments, we used the tessellation parameters reported in [9]. Specifically, the maximum radius of tessellation was 140 pixels, $dA = 45$ and $dR = 20$ pixels. The innermost sectors in the tessellation were ignored as in [9]. Using these parameters, we ended up with 48 features for each Gabor filter orientation. Given four different filter orientations, we had a total of 192 features.

Results for the top class and top two classes are provided in Fig. 8. The overall rejection rate was 3.62% (i.e., 72 training images and 73 testing images were rejected due to

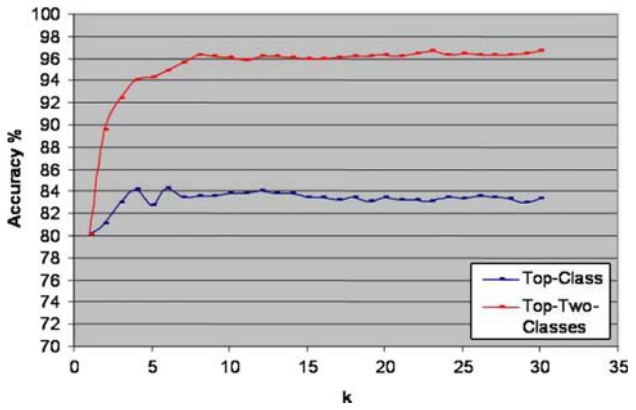

Fig. 8 Top-class and top-two-classes accuracy using Gabor features

invalid tessellations). The reported rejection rate in [9] was 1.8% which indicates that their core point extraction algorithm was probably more robust than ours. As we can see from Fig. 8, when $k = 10$, the top-class accuracy was 83.86% while the top-two-classes accuracy was 96.1%. When comparing these results with those based on OMs, the top-class accuracy was 85.43% and the top-two-classes accuracy was 95.77%. However, the best performance in the case of OMs was obtained when $k = 6$; in this case, the top-class accuracy was 86.2%, while the top-two-classes accuracy was similar for both methods.

6.5 Processing time

Besides considering classification accuracy, it is also important to take into account the time complexity of each feature extraction method. Feature extraction using Gabor filters and orientation collinearity was carried out in Matlab v6.5.0, while minutiae extraction and orientation map extraction were implemented in C. Table 1 provides a comparison among the different methods. As it can be observed, the average processing time taken for Gabor filters is much higher than any other feature extraction method.

6.6 Rank-level fusion

To improve classification accuracy, we investigated a rank-level fusion approach, which can be implemented efficiently when fusing the outputs of several k-NN classifiers. The key idea is fusing the nearest neighbors of different k-NN classifiers, each employing a different type of features. Specifically, given an input image, a k-NN classifier outputs its k nearest neighbors. To fuse the results of two of more k-NN classifiers, we combine the nearest neighbors of each classifier into a single vector. Then, we assign the input to the class, which is most frequently represented among the combined nearest neighbors. The top two classes can be also retrieved by finding the classes that have the highest and second highest counts among the combined nearest neighbors. Figure 9 depicts this idea in the case of two k-NN classifiers.

Although we experimented with fusing together the rankings of different classifiers, the best results were always obtained when including in the fusion the rankings of the OM or OC classifier. When fusing the rankings of MM and OM classifiers in the case of top-class accuracy (see Fig. 10a), fusion was slightly worse than using the OM classifier alone. In the case of top-two-classes accuracy (see Fig. 10b), however, fusion outperformed the OM classifier from $k = 1$ to $k = 10$. Similar observations were made when fusing MM with OC classifiers. When fusing the rankings of OM and OC classifiers, we obtained significant improvements in the case of top-two-classes accuracy. In the case of top-class accuracy, fusion slightly outperformed the OM classifier from $k = 1$ to $k = 10$. To keep computational requirements low, we did not consider Gabor features for fusion purposes. Also, fusing together more than two classifiers did not yield significant improvements to justify the higher computational requirements.

Table 2 summarizes the results of the four different feature extraction methods compared here for $k = 9$. Both top-class and top-two-classes accuracies are reported. As it can be observed, both top-class and top-two-classes accuracy of the MM and OC classifiers do not compare well with the accuracy of the OM and Gabor features classifiers. Fusing MM with OM rankings yields an accuracy, which approaches that of OM but not exceeding it, both for top-class and top-two-classes accuracy. Fusing OM and OC rankings improves accuracy by 2% in the case of top-two-classes but no significant improvements were observed in the case of top-class accuracy. It should be mentioned that

Table 1 Time processing comparisons

Method	Average processing time (s)
Gabor features	5.6
OM	0.03
MM	0.30
OC	2.29

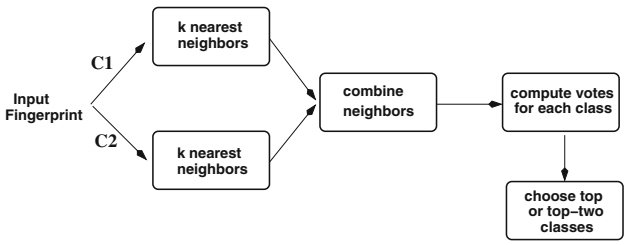

Fig. 9 Rank-level fusion scheme

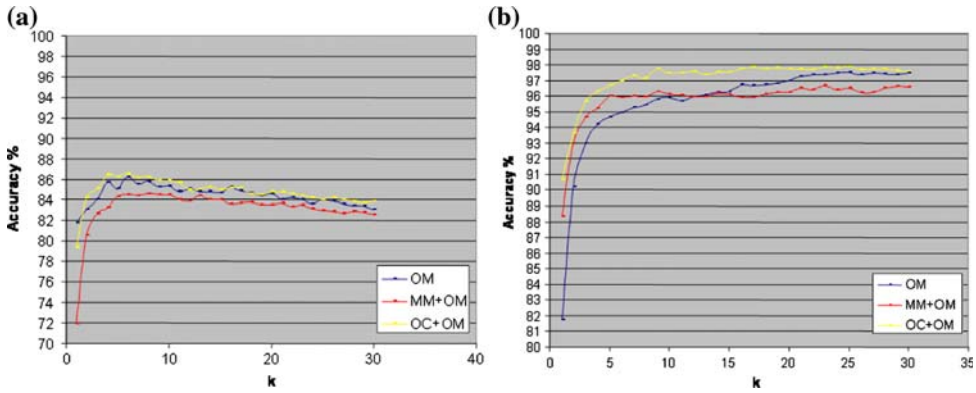

Fig. 10 Results using rank-level fusion: **a** top-class accuracy, **b** top-two-classes accuracy

Table 2 Summary of Results

Method	Top-class (%)	Top-two-classes (%)
OC + OM	85.852	97.761
OM	85.532	95.721
MM + OM	84.478	96.285
Gabor	83.601	96.211
OC	76.524	93.75
MM	64.533	85.163

the combination of OM with OC outperforms the classifier using Gabor features while having much lower computational requirements too.

7 Conclusions

We performed a comparative study of four different feature extraction methods for fingerprint classification and reported the results on their accuracy and time requirements. Our results indicate that OMs have the best performance, both in terms of accuracy and time. Gabor features fell behind OMs in terms of classification accuracy due to their sensitivity to localization errors of the core point. We also experimented with a simple rank-level fusion scheme to improve the classification accuracy. Our experimental results indicate that fusing the rankings of OM and OC classifiers improves accuracy in the case of top-two-classes classification. Further improvements in classification accuracy will be the focus of our future work. In this context, we plan to investigate a multi-stage approach with the *k*-NN classifier in the first stage and a support vector machine (SVM) in the second stage as in [3]. Moreover, we believe that improving the robustness of core point extraction would lead to higher classification rates. Finally, there are dependencies among the features used in our experiments; for example, they contain orientation information. We plan to investigate these dependencies using feature selection techniques [20] to identify which features are most important for classification.

References

1. Ratha N, Karu K, Chen S, Jain A (1996) A real-time system for large fingerprint databases. IEEE Trans Pattern Anal Mach Intell 18(8):799–813
2. Yager N, Amin A (2004) Fingerprint classification: a review. Pattern Anal Appl 7:77–93
3. Maltoni D, Maio D, Jain AK, Prabhakar S (2003) Handbook of fingerprint recognition. Springer, Berlin
4. Kawagoe M, Tojo A (1984) Fingerprint pattern classification. Pattern Recognit 17(3):295–303
5. Candela G (1995) Pcasys-a pattern-level classification automation system for fingerprints. NIST technical report NISTIR 5647
6. Chong M, Ngee T, Jun L, Gay K (1997) Geometric framework for fingerprint image classification. Pattern Recognit 30(7–9):1475–1488
7. Cappelli R, Lumini A, Maio D, Maltoni D (1999) Fingerprint classification by directional image partitioning. IEEE Trans Pattern Anal Mach Intell 21(5):402–421
8. Cappelli R, Maio D, Maltoni D (2002) A multi-classifier approach to fingerprint classification. Pattern Anal Appl 5:136–144
9. Jain A, Prabhakar S, Hong L (1999) A multichannel approach to fingerprint classification. IEEE Trans Pattern Anal Mach Intell 21(4):348–359
10. Ruta D, Gabrys B (2000) An overview of classifier fusion methods. Comput Inform Syst 7(1):1–10
11. Ross A, Shah J, Jain A (2005) Towards reconstructing fingerprints from minutiae points. In: Proceedings of SPIE conference on biometric technology for human identification II
12. Tan X, Bhanu B, Lin Y (2005) Fingerprint classification based on learned features. IEEE Trans Syst Man Cybern Part C: Appl Rev 35(3):287–300
13. Wang S (2002) Fingerprint classification by directional fields. In: IEEE international conference on multimodal interfaces
14. Noviko S, Kot V (1998) Singular feature detection and classification of fingerprints using hough transform. In:Proceedings of SPIE 3346, sixth international workshop on digital image processing and computer graphics: applications in humanities and natural sciences, pp 259–269
15. Yager N, Amin A (2004) Fingerprint verification based on minutiae features: a review. Pattern Anal Appl 7:94–113

16. NL (Lithuania), Verifinger fingerprint identifications system. http://www.neurotechnologija.com/verifinger.htm
17. Candela G et al (1995) Pcasys-a pattern-level classification automation system for fingerprints. NIST technical report NISTIR 5647
18. Duda R, Hart P, Stork D (2000) Pattern classification, 2nd edn. Wiley, New York
19. Senior A (2001) A combination fingerprint classifier. IEEE Trans Pattern Anal Mach Intell 23(10):1165–1174
20. Jain A, Duin R, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37

Author Biographies

Uday Rajanna received the Bachelors in Engineering degree in Computer Science and Engineering from Bangalore University, India in 2001 and the Master of Science degree from the University of Nevada, Reno in 2006, where he worked at the Computer Vision Laboratory (CVL) of the Department of Computer Science and Engineering. Currently he is working as a software engineer at Maxim Healthcare Services in Baltimore, Maryland with a background in designing applications for Data Mining, Statistical Machine Learning and Information Retrieval systems. His research interests include Computer Vision, Artificial Intelligence, Robotics and Pattern Recognition

.

Ali Erol received the B.S. degree in electrical and electronics Engineering from Bilkent University, Turkey in 1991 and the M.Sc. and Ph.D. degrees in electrical and electronics engineering from Middle East Technical University, Turkey in 1995 and 2001, respectively. He worked as a post-doctoral fellow in the Computer Vision Laboratory (CVL) of the Department of Computer Science and Engineering at the University of Nevada, Reno (UNR). Currently, he is working as a research scientist in Ocali Software, Turkey for the development of an image-based 3D reconstruction software. His research interests include computer vision, image processing and pattern recognition.

George Bebis received the B.S. degree in mathematics and M.S. degree in computer science from the University of Crete, Greece in 1987 and 1991, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Central Florida, Orlando, in 1996. Currently, he is an Associate Professor with the Department of Computer Science and Engineering at the University of Nevada, Reno (UNR) and Director of the UNR Computer Vision Laboratory (CVL). His research interests include computer vision, image processing, pattern recognition, machine learning, and evolutionary computing. His research is currently funded by NSF, NASA, ONR, and Ford Motor Company. Dr. Bebis is an associate editor of the *Machine Vision and Applications Journal*, and serves on the Editorial Board of the *Pattern Recognition Journal* and the *International Journal on Artificial Intelligence Tools*. He has served on the program committees of various national and international conferences, and has organized and chaired several conference sessions. In 2002, he received the Lemelson Award for Innovation and Entrepreneurship. He is a member of the IEEE and the IAPR Educational Committee.