Segmentation of Fingerprint Images

Article · December 2001

2 authors:

Asker M. Bazen

Sabih H. Gerez
University of Twente

Segmentation of Fingerprint Images

Asker M. Bazen and Sabih H. Gerez

University of Twente, Department of Electrical Engineering,
Laboratory of Signals and Systems,
P.O. box 217 - 7500 AE Enschede - The Netherlands
Phone: +31 53 489 3156 Fax: +31 53 489 1060
E-mail: a.m.bazen@el.utwente.nl

Abstract— **An important step in an automatic fingerprint recognition system is the segmentation of fingerprint images. The task of a fingerprint segmentation algorithm is to decide which part of the image belongs to the foreground, originating from the contact of a fingertip with the sensor, and which part to the background, which is the noisy area at the borders of the image.**

In this paper, an algorithm for the segmentation of fingerprints is presented. The method uses three pixel features, being the coherence, the mean and the variance. An optimal linear classifier is trained for the classification per pixel, while morphology is applied as postprocessing to obtain compact clusters and to reduce the number of classification errors.

Manual inspection shows that the proposed method provides accurate high-resolution segmentation results. Only 6.8% of the pixels is misclassified while the postprocessing further reduces this ratio. Experiments show that the proposed segmentation method and manual segmentation perform equally well in rejecting false fingerprint features from the noisy background.

Keywords— **Image processing, fingerprint image segmentation, pixel features, coherence, linear classification, neural network.**

I. Introduction

An important step in an automatic fingerprint recognition system is the *segmentation* of fingerprint images. Segmentation is the decomposition of an image into its components. A captured fingerprint image usually consists of two components, which are called the *foreground* and the *background*. The foreground is the component that originated from the contact of a fingertip with the sensor. The noisy area at the borders of the image is called the background. The task of the fingerprint segmentation algorithm is to decide which part of the image belongs to the foreground and which part to the background.

Accurate segmentation is especially important for the reliable extraction of features like *minutiae* and *singular points*. Most feature extraction algorithms extract a lot of false features when applied to the noisy background area. Therefore, the main goal of the segmentation algorithm is to discard the background, and thus reduce the number of false features.

Several approaches to fingerprint image segmentation are known from literature. In [1], the fingerprint is partitioned in blocks of 16×16 pixels. Then, each block is classified according to the distribution of the gradients in that block. In [2], this method is extended by excluding blocks with a gray-scale variance that is lower than some threshold. In [3] the gray-scale variance in the direction orthogonal to the orientation of the ridges is used to classify each 16×16 block. In [4], the output of a set of Gabor filters is used as input to a clustering algorithm that constructs spatially compact clusters. In [5], fingerprint images are segmented based on the coherence, while morphology is used to obtain smooth regions.

This paper is organized as follows. First, Section II discusses pixel feature extraction. Then, Section III presents a classifier and Section IV proposes postprocessing methods for the reduction of classification errors. Finally, Section V presents some experimental results.

II. Pixel Feature Extraction

The first step in the development of an algorithm for fingerprint image segmentation is the selection of *pixel features*[1] on which the segmentation will be based. We have selected three features that may contain useful information for segmentation. These features are the *coherence*, the *local mean* and the *local variance* of the fingerprint image. For noise reduction, the features are averaged by a Gaussian window W with $\sigma = 6$, providing a combination of both localized and aver-

[1]Note that the term 'feature' is used here to refer to properties of individual pixels whereas it was used earlier to refer to properties of the entire (foreground) image (such as the list of minutiae locations). In the rest of this paper, the correct meaning of 'feature' should become clear from the context.

aged features.

The coherence gives a measure how well the gradients are pointing in the same direction. Since a fingerprint mainly consists of parallel line structures, the coherence will be considerably higher in the foreground than in the background. In a window W around a pixel, the coherence is defined as:

$$Coh = \frac{|\sum_W (G_{s,x}, G_{s,y})|}{\sum_W |(G_{s,x}, G_{s,y})|} = \frac{\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}}{G_{xx} + G_{yy}} \tag{1}$$

where $(G_{s,x}, G_{s,y})$ is the squared gradient, $G_{xx} = \sum_W G_x^2$, $G_{yy} = \sum_W G_y^2$, $G_{xy} = \sum_W G_x G_y$ and (G_x, G_y) is the local gradient. More information on the coherence can be found in [5], where the coherence is used as the only pixel feature for segmentation of fingerprint images, and in [6].

The average gray value is the second pixel feature that might be useful for the segmentation of fingerprint images. For most fingerprint sensors, the ridge-valley structures can be approximated by black and white lines, while the background, where the finger does not touch the sensor, is rather white. This means that the mean gray value in the foreground is in general lower, i.e. darker gray, than it is in the background. Using I as the intensity of the image, the local mean for each pixel is given by:

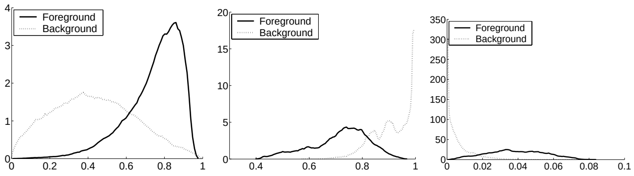
$$Mean = \sum_W I \tag{2}$$

The variance is the third pixel feature that can be used. In general, the variance of the ridge-valley structures in the foreground is higher than the variance of the noise in the background. The variance is for each pixel given by:

$$Var = \sum_W (I - Mean)^2 \tag{3}$$

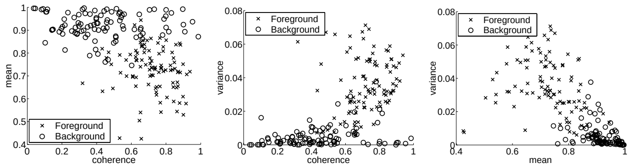
To evaluate the usefulness of these pixel features, their probability density functions have been determined for both the foreground and the background area. For this experiment, Database 2 of the Fingerprint Verification Competition (FVC2000) has been used [7]. This database has been acquired from untrained volunteers with a capacitive sensor. Examples of fingerprints from this database are shown in Figure 3.

In order to obtain the distributions of these three pixel features in both the foreground and the background area, 30 fingerprint images (`100_1.tif` - `29_1.tif`) have been segmented manually. The distributions of *Coh*, *Mean* and *Var* are shown in Figure 1, while the joint distributions of the combinations of two features are depicted in Figure 2.



(a) Coherence (b) Mean (c) Variance

Fig. 1. Distributions of the pixel features in the foreground and background areas of Database 2.



(a) Coherence - Mean (b) Coherence - Variance (c) Mean - Variance

Fig. 2. Joint distributions of the pixel features for foreground and background areas of Database 2.

III. Classification

Many segmentation algorithms use unsupervised clustering algorithms to assign feature vectors to either foreground or background [4], [8]. However, in this paper we will follow a supervised approach since examples of the pixel features in both areas are available. Using this method, a classification algorithm can be constructed that minimizes the probability of misclassifying feature vectors.

Many different classification algorithms exist that can be applied to this problem. One can for instance think of K-nearest neighbor, neural networks, etc. to find the optimal decision boundaries [9]. However, since we want to apply the classifier to all pixels in a fingerprint image, it is very important to use a classification algorithm that has the lowest computational complexity possible. We have therefore chosen to use a linear classifier, which tests a linear combination of the features, given by:

$$v = \mathbf{w}^T \mathbf{x} = w_0 \, Coh + w_1 Mean + w_2 \, Var + w_3 \tag{4}$$

where v is the value to be tested, $\mathbf{w} = [w_0 \; w_1 \; w_2 \; w_3]^T$

is the weight vector and $\mathbf{x} = [Coh\ Mean\ Var\ 1]^T$ is the feature vector. Then, using class ω_1 for the foreground, class ω_0 for the background and $\hat{\omega}$ for the assigned class, the following decision function is applied:

$$\hat{\omega} = \begin{cases} \hat{\omega}_1 & \text{if } \mathbf{w}^T\mathbf{x} > 0 \\ \hat{\omega}_0 & \text{if } \mathbf{w}^T\mathbf{x} \leq 0 \end{cases} \tag{5}$$

This classifier is essentially a linear neuron of which the output is given by $v = \mathbf{w}^T\mathbf{x}$ [10]. This classifier is trained by first setting the desired responses d. Here, we choose $d_{\omega_1} = 1$ and $d_{\omega_0} = -1$. Then, the error e is given by:

$$e = d - \mathbf{w}^T\mathbf{x} \tag{6}$$

and the neuron can be trained by the LMS algorithm. This algorithm adapts the weight vector $\hat{\mathbf{w}}$ according to:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \eta(n)\mathbf{x}(n)e(n) \tag{7}$$

For better convergence, a decreasing learning rate is used. More specifically, we use the search-then-converge schedule:

$$\eta(n) = \frac{\eta_0}{1 + n/\tau} \tag{8}$$

Monotonic classifier behavior can be obtained by using the same error measure for both training and evaluation. Since only the sign of the output is tested for classification, the corresponding activation function is given by:

$$y = \mathrm{sign(v)} \tag{9}$$

where

$$\mathrm{sign(v)} = \begin{cases} 1 & \text{for v} > 0 \\ -1 & \text{for v} \leq 0 \end{cases} \tag{10}$$

This classifier is known as Rosenblatt's perceptron; it consists of a single neuron of the McCulloch-Pitts type. For this configuration, the LMS training scheme, as given by Expressions 6 and 7, can still be used. However, in this case, the error is only nonzero for input vectors that are incorrectly classified. Rosenblatt's perceptron is proven to converge for 2 linearly separable classes. However, it will oscillate when the classes are not linearly separable, as is the case for our data set.

We indeed observed a large oscillation of the weights such that good classification performance could not be established. Using a decreasing learning rate caused the oscillations to decay. However, since the weights were adapted in opposite directions by examples from the two classes, this made the weights go to zero. Such a behavior can be avoided by a regularization condition on the weights. We have chosen to normalize the length of the weight vector to $|\mathbf{w}| = 1$ after each adaptation.

Using a normalized $\mathbf{w}$ and appropriate parameter values, for instance 10^6 epochs, $\eta_0 = 10^{-4}$ and $\tau = 10^4$, the weights converged to good classification boundaries. The results are shown in Table I. This table gives for all combinations of pixel features the optimal weight vector $\mathbf{w}$, the probability that a foreground pixel is classified as background $p(\hat{\omega}_1|\omega_0)$, the probability that a background pixel is classified as foreground $p(\hat{\omega}_0|\omega_1)$ and the probability of error p_{error} which is the average of $p(\hat{\omega}_0|\omega_1)$ and $p(\hat{\omega}_1|\omega_0)$. For each combination, the vector $\mathbf{x}$ is composed the feature values given in the first column followed by 1. The table shows that the best performance is obtained by using a combination of all three features, providing an error rate of 6.8%.

IV. Postprocessing

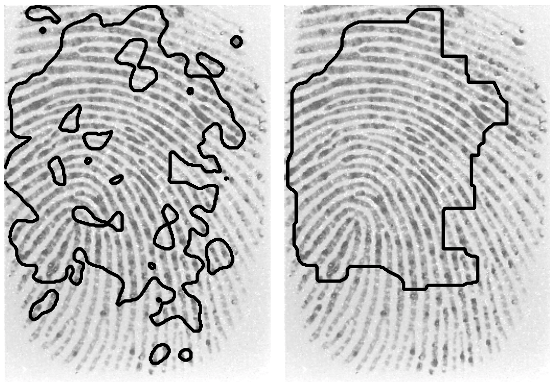
It was shown in the previous section that the classification algorithm misclassifies 6.8 % of the pixels. In some cases, this leads to a 'noisy' segmentation, where spurious small areas of one class show up inside a larger area of the other class. However, meaningful segmentation of fingerprints whould consist of compact clusters. In [8], it is suggested to use information of neighboring pixels for this purpose. This is already taken care of up to some extent by the classification algorithm since the pixel features are calculated by averaging over a spatial window W.

More compact clusters can be obtained by a number of different postprocessing methods. It is possible to use either boundary-based methods like *curve fitting* and *active contour models*, or region-based methods like *region growing* and *morphology* [11]. We have chosen to apply morphology to the classification estimate. This method 'repairs' the estimate by removing small areas, thus creating more compact clusters. It reduces the number of false classifications. First, small clusters that are incorrectly assigned to the foreground are removed by means of an *open* operation. Next, small clusters that are incorrectly assigned to the background are removed by a *close* operation.

TABLE I
RESULTS OF THE LINEAR CLASSIFIER ON DATABASE 2.

Features	$\mathbf{w}$	$p(\hat{\omega}_0\|\omega_1)$	$p(\hat{\omega}_1\|\omega_0)$	p_{error}
Coh	[0.84 -0.54]	0.205	0.185	0.195
Mean	[-0.77 0.64]	0.142	0.146	0.144
Var	[1.00 -0.017]	0.095	0.102	0.099
Coh, *Mean*	[0.13 -0.80 0.59]	0.105	0.117	0.111
Coh, *Var*	[0.0071 1.00 -0.019]	0.074	0.104	0.089
Mean, *Var*	[-0.053 1.00 0.026]	0.075	0.069	0.072
Coh, *Mean*, *Var*	[0.013 -0.05 0.99 0.015]	0.074	0.062	0.068

Fig. 3. Segmentation results of some fingerprints from Database 2.



(a) Before morphology

(b) After morphology

Fig. 4. Effect of postprocessing.

V. EXPERIMENTAL RESULTS

This section presents some experimental results of the segmentation algorithm. First, in Figure 3, segmentation results are shown for three fingerprints from FVC2000 Database 2. The segmentation algorithm has been trained on fingerprints of this database, but not on these particular fingerprints. Human inspection shows that the algorithm provides satisfactory results. The effect of the morphology is shown in Figure 4.

Apart from human inspection, there are several ways of quantitatively evaluating the results of a segmentation algorithm. For instance, the number of classification errors could be used as a performance measure. This is exactly the measure that was used during training and was found to be 6.8% for the optimal classifier. Another possibility is to evaluate a segmentation algorithm by counting the number of false and missed fingerprint features like minutiae or singular points. The results for the singular-point extraction, using the methods presented in [5], [6], are shown in Table II.

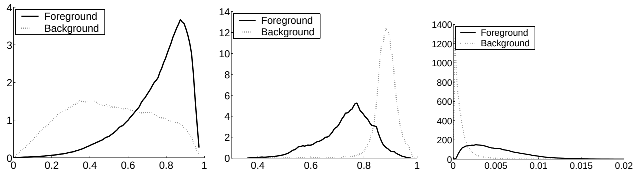
In this table, results of the singular point extraction algorithm are shown for several segmentation methods, being no segmentation, manual segmentation, segmentation that is based only on the coherence [5] and the segmentation method that is proposed in this paper. For each segmentation method, the table shows the average number of false singular points (SPs) in a fingerprint image, the ratio of the fingerprint images in which false SPs are found and the ratio of the fingerprint images in which true SPs are discarded by the segmentation algorithm.

The table shows that the proposed segmentation method rejects more false SPs than the manual method and the method that was based only on the coherence. This is caused by the fact that the proposed segmentation method is allowed to estimate holes in the foreground area at noisy areas in a fingerprint image where false SPs are likely to occur. However, this may cause true SPs to be discarded since they may also be located in these areas.

Next, the applicability of the optimal classifier to other databases has been investigated. For this purpose, FVC2000 Database 1 is used. This database is acquired from untrained volunteers using an optical

TABLE II
Results of singular point extraction from Database 2.

Segmentation method	no	manual	*Coh*	all
Average number of false SPs	15.4	0.8	0.8	0.5
Ratio of fingerprints with false SPs	0.97	0.17	0.2	0.13
Ratio of fingerprints with missed SPs	0	0	0.02	0.05



(a) Coherence (b) Mean (c) Variance

Fig. 5. Distributions of features in the foreground and background areas for Database 1.

Fig. 6. Segmentation results of some fingerprints from Database 1.

sensor. The distribution of the pixel features for the foreground and background areas are shown in Figure 5, while examples of the fingerprint images are shown in Figure 6.

Comparing Figure 5 to Figure 1 shows that the distributions of the pixel features are slightly different for both databases. Therefore, another classifier was trained on the images features of Database 1. The results are shown in Table III, which shows slightly different weight vectors and error probabilities. It can be seen that the classifier of Database 1 assigns most importance to *Mean*, while the classifier of Database 2 assigns most importance to *Var*. The segmentation results of some fingerprint images are shown in Figure 6.

The last experiment is the direct application of the optimal classifier to one database, while it is trained on another database. The results of this experiment for Databases 1 and 2 are shown in Table IV. The columns are labeled with $\mathbf{w}_i$ where i refers to the database for which the classifier has been trained.

TABLE IV
Error probabilities of the classifiers on databases that they are not trained on.

Database	$\mathbf{w}_1$	$\mathbf{w}_2$
1	0.103	0.404
2	0.143	0.068

It can be seen that the application of classifier 1 on Database 2 performs suboptimally (14.3% error instead of 6.8%), while the application of classifier 2 on Database 1 performs very badly (40.4% error instead of 10.3%). Therefore, it can be concluded that a classifier always has to be trained on that specific database that it has to be applied to.

VI. Conclusions and Recommendations

In this paper, an algorithm for the segmentation of fingerprints is presented. The method uses three pixel features, being the coherence, the mean and the variance. An optimal linear classifier has been trained for the classification per pixel, while morphology has been applied as postprocessing to obtain compact clusters and to reduce the number of classification errors.

Human inspection has shown that the proposed method provides accurate high-resolution segmentation results. Only 6.8% of the pixels is misclassified while the postprocessing further reduces this ratio. Experiments have shown that the proposed segmentation method and manual segmentation perform equally well in rejecting false fingerprint features from the noisy background.

An alternative to the morphological postprocessing that will be investigated in the near future, is the use of *hidden Markov models* (HMMs) which are widely used in speech recognition [12]. This method takes into account the context, or surroundings, for each feature vector to be classified. Using estimations of the probability of class transitions and the conditional feature distributions, the segmentation is found that maximizes the likelihood of these observations. Furthermore, the use of a third class, representing low-

TABLE III

Results of the linear classifier on Database 1.

Features	$\mathbf{w}$	$p(\hat{\omega}_0\|\omega_1)$	$p(\hat{\omega}_1\|\omega_0)$	p_{error}
Coh	[0.82 -0.57]	0.279	0.247	0.263
Mean	[-0.76 0.65]	0.110	0.115	0.113
Var	[1.00 -0.0018]	0.155	0.122	0.139
Coh, *Mean*, *Var*	[0.027 -0.77 0.061 0.63]	0.104	0.103	0.103

quality regions, is expected to improve the segmentation results.

References

[1] B.M. Mehtre, N.N. Murthy, S. Kapoor, and B. Chatterjee, "Segmentation of fingerprint images using the directional image," *Pattern Recognition*, vol. 20, no. 4, pp. 429–435, 1987.

[2] B.M. Mehtre and B. Chatterjee, "Segmentation of fingerprint images - a composite method," *Pattern Recognition*, vol. 22, no. 4, pp. 381–385, 1989.

[3] N. Ratha, S. Chen, and A. Jain, "Adaptive flow orientation based feature extraction in fingerprint images," *Pattern Recognition*, vol. 28, pp. 1657–1672, Nov. 1995.

[4] A.K. Jain and N.K. Ratha, "Object detection using gabor filters," *Pattern Recognition*, vol. 30, no. 2, pp. 295–309, Feb. 1997.

[5] A.M. Bazen and S.H. Gerez, "Directional field computation for fingerprints based on the principal component analysis of local gradients," in *Proceedings of ProRISC2000, 11th Annual Workshop on Circuits, Systems and Signal Processing*, Veldhoven, The Netherlands, Nov. 2000.

[6] A.M. Bazen and S.H. Gerez, "Systematic methods for the computation of the directional field and singular points of fingerprints," *IEEE Trans. PAMI*, 2002, To be published.

[7] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain, "FVC2000: Fingerprint verification competition," Biolab internal report, University of Bologna, Italy, Sept. 2000, available from `http://bias.csr.unibo.it/fvc2000/`.

[8] N.R. Pal and S.K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[9] A.K. Jain, R.P.W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. PAMI*, vol. 22, no. 1, Jan. 2000.

[10] S. Haykin, *Neural Networks, A Comprehensive Foundation*, Prentice Hall International, Inc., Upper Saddle River, NJ, 1999.

[11] A.K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[12] L.R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.