# Systematic Methods for the Computation of the Directional Fields and Singular Points of Fingerprints

Asker M. Bazen and Sabih H. Gerez

**Abstract**—The first subject of this paper is the estimation of a high resolution directional field of fingerprints. Traditional methods are discussed and a new method, based on principal component analysis, is proposed. The method not only computes the direction in any pixel location, but its coherence as well. It is proven that this method provides exactly the same results as the "averaged square-gradient method" that is known from literature. Undoubtedly, the existence of a completely different equivalent solution increases the insight into the problem's nature. The second subject of this paper is singular point detection. A very efficient algorithm is proposed that extracts singular points from the high-resolution directional field. The algorithm is based on the Poincaré index and provides a consistent binary decision that is not based on postprocessing steps like applying a threshold on a continuous resemblance measure for singular points. Furthermore, a method is presented to estimate the orientation of the extracted singular points. The accuracy of the methods is illustrated by experiments on a live-scanned fingerprint database.

**Index Terms**—Image processing, fingerprint recognition, directional field, orientation estimation, singular point extraction, principal component analysis.

## 1 Introduction

Fingerprint recognition has received increasingly more attention during the last years. Since the performance of fingerprint verification systems has reached a satisfactory level for applications involving small databases, the next step is the development of algorithms for fingerprint *identification* systems that can search relatively large databases for a matching fingerprint. Although other approaches are possible, like, for instance, the hashing technique in the minutiae domain [1], the first step in an identification system is often continuous classification of fingerprints [2], [3]. This reduces the partition of the database to be searched for matches. To facilitate high-performance classification, algorithms for accurate directional field and singular-point estimation are needed.

In Fig. 1a, a fingerprint is depicted. The information carrying features in a fingerprint are the line structures, called *ridges* and *valleys*. In this figure, the ridges are black and the valleys are white. It is possible to identify two levels of detail in a fingerprint. The *directional field* (DF), shown in Fig. 1b, describes the coarse structure, or basic shape, of a fingerprint. It is defined as the local orientation of the ridge-valley structures. The *minutiae* provides the details of the ridge-valley structures, like ridge-endings and bifurcations. Minutiae are, for instance, used for fingerprint *matching*, which is a one-to-one comparison of two fingerprints.

This paper focuses on the directional field of fingerprints and matters directly related to the DF. The DF is, in principle, perpendicular to the gradients. However, the gradients are orientations at pixel scale, while the DF describes the orientation of the ridge-valley structures, which is a much coarser scale. Therefore, the DF can be derived from the gradients by performing some *averaging* operation on the gradients, involving pixels in some neighborhood [4]. This is illustrated in Fig. 2a, which shows the gradients in a part of a fingerprint, and Fig. 2b, which shows the averaged directional field. While the gradients are not all parallel because of the endpoint, the directional field is because of the averaging operator. The averaging of gradients in order to obtain the DF is the first topic of this paper.

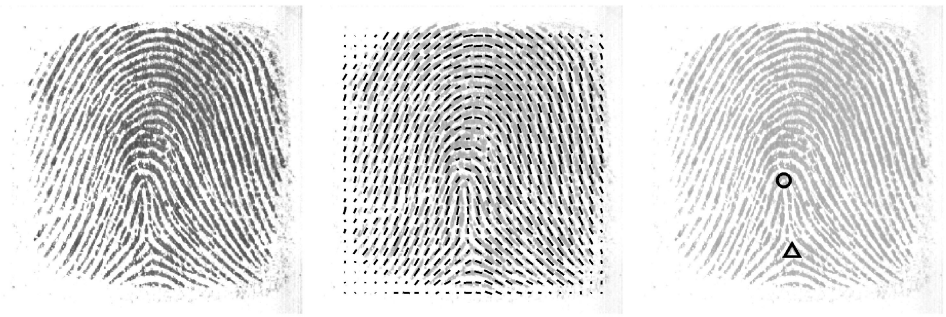
The estimation method that is described in this paper enables the application of DF-related tasks that require very high resolution and accurate DFs. Examples of these demanding techniques are, for instance, the accurate "extraction of singular points" as discussed in Section 3 and "high-performance classification." Together with the DF, the *coherence* can be estimated. The coherence is a measure that indicates how well the gradients are are pointing in the same direction. An example of its use is high-resolution segmentation [5], [6].

In the DF, *singular points* (SPs) can be identified. The extraction of those singular points is the second topic of this paper. SPs are the points in a fingerprint where the directional field is discontinuous. Henry [7] defined two types of singular points, in terms of the ridge-valley structures. The *core* is the topmost point of the innermost curving ridge and a *delta* is the center of triangular regions where three different direction flows meet. The locations of the singular points in an example fingerprint are given in Fig. 1c. Apart from its location, an SP has an *orientation*; this paper also proposes an estimation method for the orientation of SPs [8].

The most common use of SPs is *registration*, which means that they are used as references to line up two fingerprints. Another example of their use is classification of fingerprints

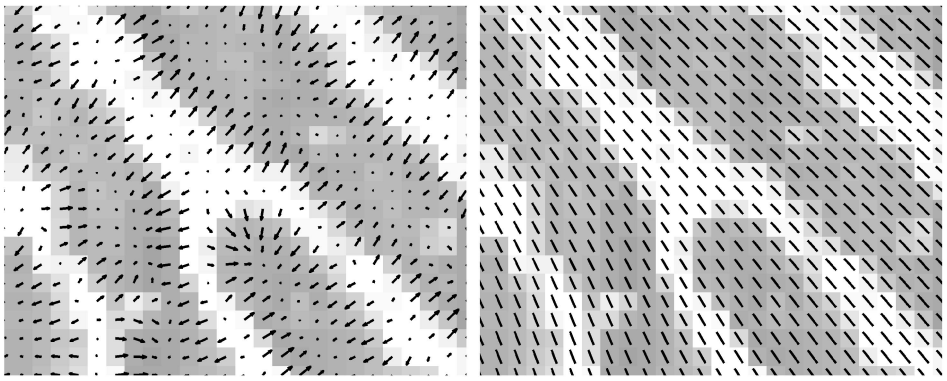
• *The authors are with the University of Twente, Department of Electrical Engineering, PO Box 217, 7500 AE, Enschede, The Netherlands. E-mail: {a.m.bazen, s.h.gerez}@el.utwente.nl.*

*Manuscript received 25 July 2000; revised 24 Apr. 2001; accepted 8 Dec. 2001.*
*Recommended for acceptance by R. Kumar.*
*For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 112593.*



(a) (b) (c)

Fig. 1. Examples of a fingerprint, its directional field and its singular points: (a) fingerprint, (b) directional field, and (c) singular points.



(a) (b)

Fig. 2. Detailed area in a fingerprint: (a) the gradient and (b) the averaged directional field.

in the Henry classes [9]. The orientation of singular points can be used for more advanced classification methods, or to initialize flow lines in the DF [9], [10], [11], [12].

This paper is organized as follows: First, in Section 2, the estimation of the DF is discussed. In Section 2.1, the traditional method of averaging squared gradients is discussed, while, in Section 2.2, a new method based on *principal component analysis* (PCA) is proposed. In Section 2.3, a proof is given that both methods are exactly equivalent and it is shown that the coherence, which is a measure for the local strength of the directional field, can be elegantly expressed in the two eigenvalues that are computed for the PCA. Then, in Section 3, SPs are discussed. Section 3.1 describes an efficient method for the extraction of SPs from the DF, while, in Section 3.2, a method is proposed for the estimation of the orientation of the SPs. In Section 4, some computational aspects of DF estimation and SP extraction are discussed. Furthermore, it is shown that the high-resolution DF can be used to obtain more accurate block-DF estimates. Finally, in Section 5, an experiment is presented where the theory is applied to fingerprints contained in one of the databases used for the *Fingerprint Verification Competition 2000* [13]. In that section, some practical aspects of the algorithms are discussed as well.

## 2 Directional Field Estimation

Various methods used to estimate the DF from a fingerprint are known from literature. They include matched-filter approaches [14], [15], [9], methods based on the high-frequency power in three dimensions [16], 2-dimensional spectral estimation methods [15], and micropatterns that can be considered binary gradients [10]. These approaches do not provide as much accuracy as gradient-based methods, mainly because of the limited number of fixed possible orientations. This is especially important when using the DF for tasks like tracing flow lines. The gradient-based method was introduced in [17] and adopted by many researchers, see, e.g., [18], [19], [20], [21].

The elementary orientations in the image are given by the gradient vector $[G_x(x,y)\ G_y(x,y)]^T$, which is defined as:

$$\begin{aligned}\begin{bmatrix} G_x(x,y) \\ G_y(x,y) \end{bmatrix} &= \operatorname{sign}(G_x)\nabla I(x,y) \\ &= \operatorname{sign}\left(\frac{\partial I(x,y)}{\partial x}\right)\begin{bmatrix} \frac{\partial I(x,y)}{\partial x} \\ \frac{\partial I(x,y)}{\partial y} \end{bmatrix},\end{aligned} \tag{1}$$

where $I(x,y)$ represents the gray-scale image. The first element of the gradient vector has been chosen to always be positive. The reason for this choice is that in the DF, which is perpendicular to the gradient, opposite directions indicate equivalent orientations. It is illustrated in Fig. 2 that some averaging operation has to be performed on the gradients in order to obtain the DF.

### 2.1 Averaging Squared Gradients

This section discusses the problems that are encountered when averaging gradients and the traditional solution of

averaging squared gradients. First, the general idea behind averaging squared gradients is presented and then, an analysis of the results of this method is given. Apart from the estimation of the DF, this section also discusses the coherence, which provides a measure for the strength of the estimated orientation.

### 2.1.1 Qualitative Analysis

Gradients cannot directly be averaged in some local neighborhood since opposite gradient vectors will then cancel each other, although they indicate the same ridge-valley orientation. This is caused by the fact that local ridge-valley structures remain unchanged when rotated over 180 degrees [21]. Since the gradient orientations are distributed in a cyclic space ranging from $0$ to $\pi$, and the average orientation has to be found, another formulation of this problem is that the "$\pi$-periodic cyclic mean" has to be computed.

In [17], a solution to this problem is proposed by doubling the angles of the gradient vectors before averaging. After doubling the angles, opposite gradient vectors will point in the same direction and, therefore, will reinforce each other, while perpendicular gradients will cancel. After averaging, the gradient vectors have to be converted back to their single-angle representation. The ridge-valley orientation is then perpendicular to the direction of the average gradient vector.

In the version of the algorithm discussed in this paper, not only the angle of the gradients is doubled, but also the length of the gradient vectors is squared, as if the gradient vectors are considered as complex numbers that are squared. This has the effect that strong orientations have a higher vote in the average orientation than weaker orientations. Furthermore, this approach results in the cleanest expressions. However, other choices, like, for instance, setting all lengths to unity [21], are found in literature as well.

In [17], a method is proposed to use the squared gradients for computation of the strength of the orientation. This measure, which is called the coherence, measures how well all squared gradient vectors share the same orientation. If they are all parallel to each other, the coherence is 1 and if they are equally distributed over all directions, the coherence is 0.

### 2.1.2 Quantitative Analysis

In this section, the qualitative analysis that was given in the previous section is made quantitative. The gradient vectors are first estimated using Cartesian coordinates, in which a gradient vector is given by $[G_x\ G_y]^T$. For doubling the angle and squaring the length, the gradient vector is converted to "polar" coordinates, in which it is given by $[G_\rho\ G_\varphi]^T$. This conversion is given by:

$$\begin{bmatrix} G_\rho \\ G_\varphi \end{bmatrix} = \begin{bmatrix} \sqrt{G_x^2 + G_y^2} \\ \tan^{-1} G_y/G_x \end{bmatrix}. \tag{2}$$

Note that $-\frac{1}{2}\pi < G_\varphi \leq \frac{1}{2}\pi$ is a direct consequence of the fact hat $G_x$ is always positive. The gradient vector is converted back to its Cartesian representation by:

$$\begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} G_\rho \cos G_\varphi \\ G_\rho \sin G_\varphi \end{bmatrix}. \tag{3}$$

Using trigonometric identities, an expression for the squared gradient vectors $[G_{s,x}, G_{s,y}]^T$ that does not refer to $G_\rho$ and $G_\varphi$, is found:

$$\begin{aligned} \begin{bmatrix} G_{s,x} \\ G_{s,y} \end{bmatrix} &= \begin{bmatrix} G_\rho^2 \cos 2G_\varphi \\ G_\rho^2 \sin 2G_\varphi \end{bmatrix} = \begin{bmatrix} G_\rho^2(\cos^2 G_\varphi - \sin^2 G_\varphi) \\ G_\rho^2(2 \sin G_\varphi \cos G_\varphi) \end{bmatrix} \\ &= \begin{bmatrix} G_x^2 - G_y^2 \\ 2G_x G_y \end{bmatrix}. \end{aligned} \tag{4}$$

This result can also be obtained directly by using the equivalence of "doubling the angle and squaring the length of a vector" to "squaring a complex number":

$$G_{s,x} + j \cdot G_{s,y} = (G_x + j \cdot G_y)^2 = (G_x^2 - G_y^2) + j \cdot (2G_x G_y). \tag{5}$$

Next, the average squared gradient $\left[\overline{G_{s,x}}\ \overline{G_{s,y}}\right]^T$ can be calculated. It is averaged in some neighborhood, using a possibly nonuniform window $W$:

$$\begin{aligned} \begin{bmatrix} \overline{G_{s,x}} \\ \overline{G_{s,y}} \end{bmatrix} &= \begin{bmatrix} \sum_W G_{s,x} \\ \sum_W G_{s,y} \end{bmatrix} \\ &= \begin{bmatrix} \sum_W G_x^2 - G_y^2 \\ \sum_W 2G_x G_y \end{bmatrix} = \begin{bmatrix} G_{xx} - G_{yy} \\ 2G_{xy} \end{bmatrix}. \end{aligned} \tag{6}$$

In this expression,

$$G_{xx} = \sum_W G_x^2 \tag{7}$$

$$G_{yy} = \sum_W G_y^2 \tag{8}$$

$$G_{xy} = \sum_W G_x G_y \tag{9}$$

are estimates for the variances and crosscovariance of $G_x$ and $G_y$, averaged over the window $W$. Now, the average gradient direction $\Phi$, with $-\frac{1}{2}\pi < \Phi \leq \frac{1}{2}\pi$, is given by:

$$\Phi = \frac{1}{2} \angle\big(G_{xx} - G_{yy}, 2G_{xy}\big), \tag{10}$$

where $\angle(x, y)$ is defined as:

$$\angle(x,y) = \begin{cases} \tan^{-1}(y/x) & x \geq 0 \\ \tan^{-1}(y/x) + \pi \quad \text{for} & x < 0 \wedge y \geq 0 \\ \tan^{-1}(y/x) - \pi & x < 0 \wedge y < 0 \end{cases} \tag{11}$$

and the average ridge-valley direction $\theta$, with $-\frac{1}{2}\pi < \theta \leq \frac{1}{2}\pi$, is perpendicular to $\Phi$:

$$\theta = \begin{cases} \Phi + \frac{1}{2}\pi \quad \text{for} & \Phi \leq 0 \\ \Phi - \frac{1}{2}\pi & \Phi > 0. \end{cases} \tag{12}$$

The coherence of the squared gradients can also be expressed using the same notations. The coherence $Coh$ is given by [17]:

$$Coh = \frac{\left|\sum_W (G_{s,x}, G_{s,y})\right|}{\sum_W \left|(G_{s,x}, G_{s,y})\right|}. \tag{13}$$

If all squared gradient vectors are pointing in exactly the same direction, the sum of the moduli of the vectors equals the modulus of the sum of the vectors, resulting in a coherence value of 1. On the other hand, if the squared gradient vectors are equally distributed in all directions, the length of the sum of the vectors will equal 0, resulting in a coherence value of 0. In between these two extreme situations, the coherence will vary between 0 and 1, thus providing the required measure.

### 2.2 Principal Component Analysis

This paper proposes a second method to estimate the directional field from the gradients, which is based on *principal component analysis* (PCA). PCA computes a new orthogonal base given a multidimensional date set such that the variance of the projection on one of the axes of this new base is maximal, while the projection on the other one is minimal. It turns out that the base is formed by the eigenvectors of the autocovariance matrix of this data set [22].

When applying PCA to the autocovariance matrix of the $[G_x \;\; G_y]^T$ gradient vectors, it provides the 2-dimensional Gaussian joint probability density function of these vectors. From this function, the main direction of the gradients can be calculated.

The estimate of the autocovariance matrix $\mathbf{C}$ of the gradient vector pairs is given by:

$$\mathbf{C} = \begin{bmatrix} G_{xx} & G_{xy} \\ G_{xy} & G_{yy} \end{bmatrix} = \sum_W \begin{bmatrix} G_x^2 & G_x G_y \\ G_x G_y & G_y^2 \end{bmatrix}. \tag{14}$$

In this estimate, the assumption is made that the gradient vectors are zero-mean, i.e.,

$$E[G_x] = E[G_y] = 0 \tag{15}$$

in a window $W$ in the given fingerprint. This is true in any window in which the fingerprint has a constant mean gray value. Then, the gradient is defined as the difference of two values that have the same expectation. Therefore, the expectation of the gradient is zero. The requirement of constant mean is reasonable in windows that contain a small number of ridge-valley transitions.

The longest axis $\mathbf{v}_1$ of the 2-dimensional joint probability density function is given by the eigenvector of the autocovariance matrix that belongs to the largest eigenvalue $\lambda_1$. This axis corresponds to the direction in which the variance of the gradients is largest, and so to the "average" gradient orientation. The ridge-valley orientations are perpendicular to this axis and, therefore, given by the shortest axis $\mathbf{v}_2$. This is the direction of the eigenvector that belongs to the smallest eigenvalue $\lambda_2$. The average ridge-valley orientation $\theta$ is given by:

$$\theta = \angle \mathbf{v}_2. \tag{16}$$

The "strength" $Str$ of the orientation can be defined as a simple function of the two eigenvalues. In order to limit the strength between 0 and 1, it is defined by:

$$Str = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}. \tag{17}$$

Again, if all gradients are pointing in the same direction, $\lambda_2 = 0$ and $Str = 1$, while, in case of a uniform distribution over all angles, $\lambda_1 = \lambda_2$ and $Str = 0$.

### 2.3 Comparison

In this section, a comparison is made between the two methods of DF estimation. A proof is given that both methods are exactly equivalent and it is shown that the coherence $Coh$ and strength $Str$ are equivalent as well. This section provides a brief description of the proofs; the mathematical details can be found in Appendix A and B.

The proof starts by deriving the average gradient, calculated by the method of averaging squared gradients as described in Section 2.1. In Appendix A, it is shown that:

$$\begin{bmatrix} \overline{G_x} \\ \overline{G_y} \end{bmatrix} = \frac{1}{c} \cdot \begin{bmatrix} \frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2} \\ G_{xy} \end{bmatrix} \tag{18}$$

with

$$c = \sqrt{\frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}}. \tag{19}$$

Next, it is shown that this vector is an eigenvector of autocovariance matrix $\mathbf{C}$, as defined in (14), which proves that both methods are equivalent.

The coherence $Coh$, calculated using the squared gradient method (see (13)) and the strength $Str$ (see (17)) are exactly equal as well. In Appendix B, it is shown that

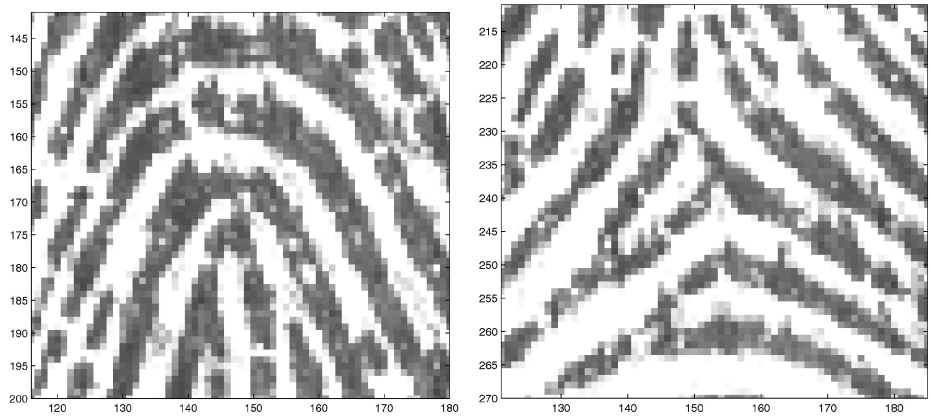
$$Str = Coh = \frac{\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}}{G_{xx} + G_{yy}}. \tag{20}$$

## 3 Singular Point Extraction

The subject of this section is extraction of the SPs, which are the points in a fingerprint where the DF is discontinuous. In Fig. 3, two segments of the fingerprint of Fig. 1 are shown, one containing a core and one containing a delta. The SPs are somewhere in the center of the segments. However, they cannot be located more accurately than within the width of one ridge-valley structure in the gray-value fingerprint, which is approximately 10 pixels for this example.

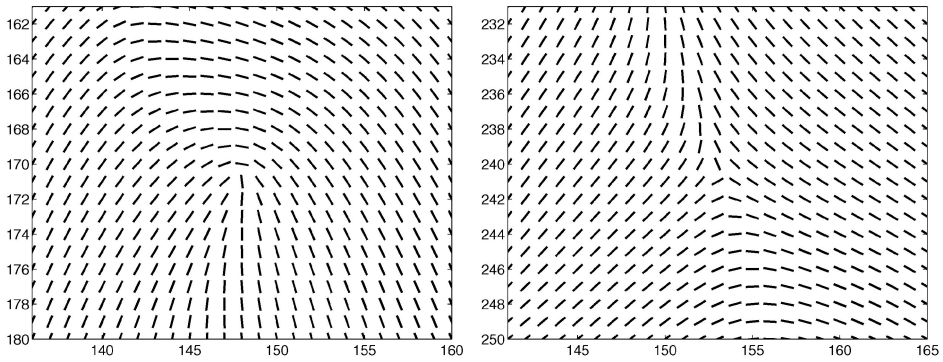
In Fig. 4, the DF of those segments is shown. From this DF, the exact SP location can be determined easily with an accuracy of only one pixel. Although it seems like a very straightforward task to extract the SPs from the DFs, many different algorithms for SP extraction are known from literature.

In [23], first areas of high curvature are identified as search areas. Then, a feature vector is estimated by taking the difference between the estimated direction and the direction of a double core (whorl) in a number of positions in a circle around a candidate area. This feature vector is classified as being core, delta, whorl, or none of these. In [24], first candidate areas of high curvature are selected, too. Then, a feature vector is constructed by taking the average directions at four positions around the candidate SP. This feature vector is classified as a core or delta. In [18], some reference models are shifted over the DF, and SPs are detected by a least-squares fit. In [21], the local energy of the DF is used as a measure for how much the local DF resembles an SP and, in [14], a neural network is slided over the DF to detect SPs. Finally, in [25], the ratio of the sines of the DFs in two adjacent regions is used as a measure to detect SPs.



(a) (b)

Fig. 3. Segments of a fingerprint that contain a singular point. (a) Core and (b) delta.



(a) (b)

Fig. 4. Directional fields. (a) Core and (b) delta.

These methods all provide somewhat unsatisfactory results since they are not capable of consistently extracting the singular points. Instead of providing a Boolean output that indicates whether an SP is present at some location or not, they produce a continuous output that indicates how much the local DF resembles a SP. Postprocessing steps, like thresholds and heuristics, are necessary to interpret the outputs of the algorithms and to make the final decisions.

The method that is presented in this section is based on the Poincaré index, which was first introduced in [10]. The Poincaré index can be explained using the DFs that are depicted in Fig. 4. Following a counterclockwise closed contour around a core in the DF and adding the differences between the subsequent angles results in a cumulative change in the orientation of $\pi$ and carrying out this procedure around a delta results in $-\pi$. However, when applied to locations that do not contain an SP, the cumulative orientation change will be zero.

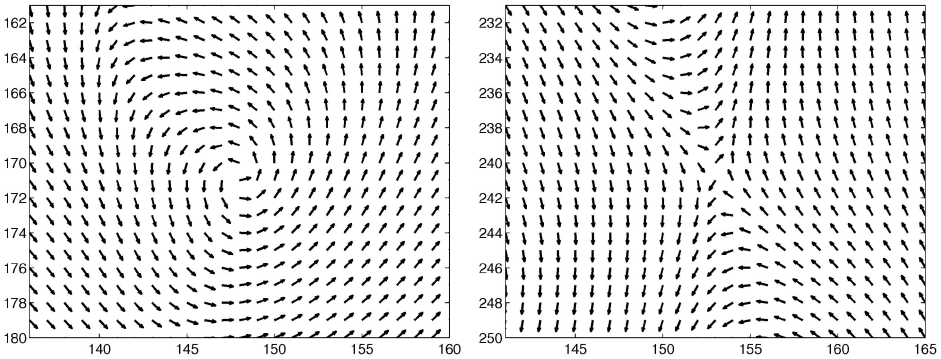
Although the Poincaré index provides the means for consistent detection of SPs, the question arises how to calculate this measure. Apart from the problem of how to calculate cumulative orientation changes over contours efficiently, a choice has to be made on the optimal size and shape of the contour. A possible implementation is described in [26]. That paper claims that a square curve with a length of 25 pixels is optimal. A smaller curve results in spurious detections, while a larger curve may ignore core-delta pairs which are close to each other. If the postprocessing step finds a connected area of more than seven pixels in which the Poincaré index is $\geq \pi$, a core or delta is detected. In the case of an area that is larger than 20 connected pixels, two cores are detected.

In Section 3.1, we propose an efficient implementation of an SP extraction algorithm that is based on the Poincaré index and makes use of small 2-dimensional filters. The algorithm extracts *all* singular points from the DF, including false SPs that are caused by an insufficiently averaged DF. Furthermore, the algorithm determines whether a core or a delta is detected.

Section 3.2 presents an algorithm for estimating the orientation of SPs. As far as we know, there exists only one earlier publication on computing the orientation of SPs [23]. That method examines the DF at a number of fixed positions in a circle around the SP and takes the position where the DF points best toward the SP as orientation of the SP. The method that is described below uses the entire neighborhood of the SP for the orientation estimate, thus providing much more accurate results.

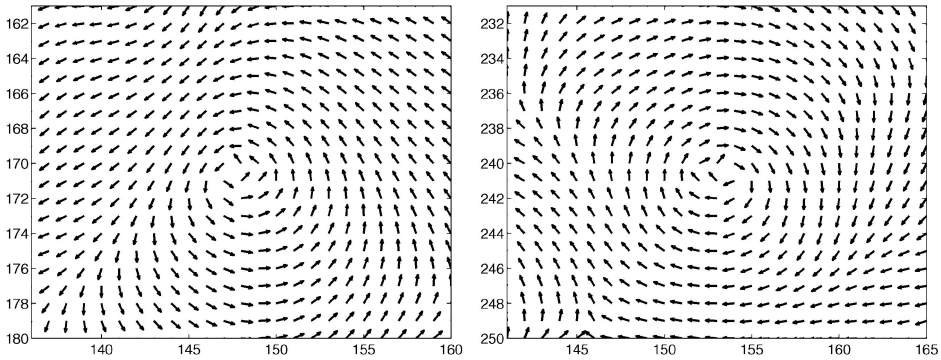
### 3.1 Extraction of Singular Points

In the implementation that is proposed in this paper, choices of the size and shape of the contour don't have to be made. Postprocessing steps are not necessary and the cumulative orientation changes over contours are implemented



(a) (b)

Fig. 5. Squared directional fields. (a) Core and (b) delta.



(a) (b)

Fig. 6. Gradient of squared directional fields. (a) Core and (b) delta.

efficiently in small 2-dimensional filters. The method computes for each individual pixel whether it is an SP, and is therefore capable of detecting SPs that are located only a few pixels apart. This property is especially useful for the extraction of SPs from *block-directional fields* (BDFs), which estimate one direction for each $n \times n$ block. Special care has to be taken that high-resolution DFs are sufficiently averaged such that spurious SPs are eliminated beforehand, as the SP-extraction algorithm will detect *all* SPs present in the DF of a given resolution.

The algorithm first takes the squared directional field (SDF). This eliminates the step of $\pi$ which is encountered in the DF between the orientations $\theta = \frac{1}{2}\pi$ and $\theta = -\frac{1}{2}\pi$. The Poincaré indexes change to $2\pi$, $-2\pi$, and 0 for, respectively, a core, a delta, and none of these. The orientation of the SDF, denoted by $2\theta$, is depicted in Fig. 5 for the areas around SPs.

Summing the changes in orientation corresponds to summing the gradients of the squared orientation. The gradient vector $J$ can be efficiently precalculated for the entire image by:

$$\begin{bmatrix} J_x(x,y) \\ J_y(x,y) \end{bmatrix} = \nabla 2\theta(x,y) = \begin{bmatrix} \frac{\partial 2\theta(x,y)}{\partial x} \\ \frac{\partial 2\theta(x,y)}{\partial y} \end{bmatrix}. \tag{21}$$

In the calculation of the discrete version of this gradient, both components of $J$ should be calculated "modulo $2\pi$," such that they are always between $-\pi$ and $\pi$. This makes the transition from $2\theta = -\pi$ to $2\theta = \pi$ continuous or, in other words, the orientation is considered to be cyclic. The gradient vectors of the squared orientation around both singular points are shown in Fig. 6.

The next step is the application of Green's Theorem, which states that a closed line-integral over a vector field can be calculated as the surface integral over the rotation of this vector field:

$$\begin{aligned} \oint_{\partial A} w_x dx + w_y dy &= \iint_A \mathrm{rot}[w_x \; w_y]^T dxdy \\ &= \iint_A \left( \frac{\partial w_y}{\partial x} - \frac{\partial w_x}{\partial y} \right) dxdy, \end{aligned} \tag{22}$$

where $x$ and $y$ define the coordinate system, $A$ is the area, and $\partial A$ is the contour around this area and $[w_x \; w_y]^T$ is the vector field. This theorem is applied to the summation of the gradients of the squared orientation over the contour:

$$\begin{aligned} Index &= \sum_{\Delta x, \Delta y \text{ along } \partial A} (J_x \cdot \Delta x + J_y \cdot \Delta y) = \sum_A \mathrm{rot}[J_x \; J_y]^T \\ &= \sum_A \left( \frac{\partial J_y}{\partial x} - \frac{\partial J_x}{\partial y} \right). \end{aligned} \tag{23}$$

Since all SPs have to be extracted from the DF, $A$ is taken as a square of one pixel. This results in a very efficient method for computation of the Poincaré index. Application of the proposed method will indeed lead to the desired SP locations. Unlike all other SP extraction methods, a core results in a Poincaré index of $2\pi$, a delta in $-2\pi$ while the

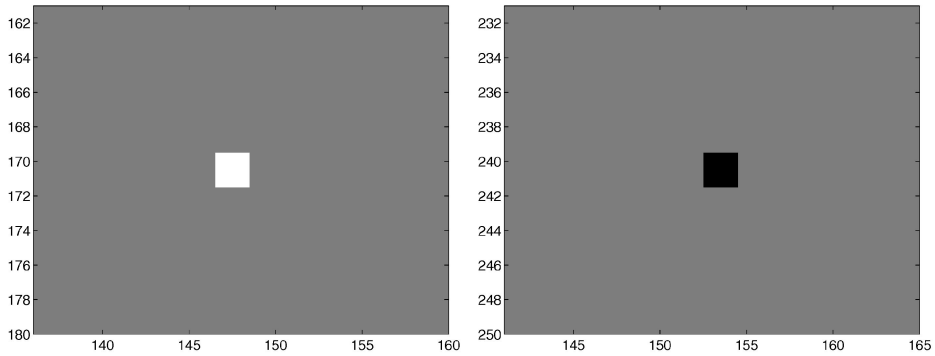



Fig. 7. Rotation of the gradient of the squared directional fields. (a) Core and (b) delta.

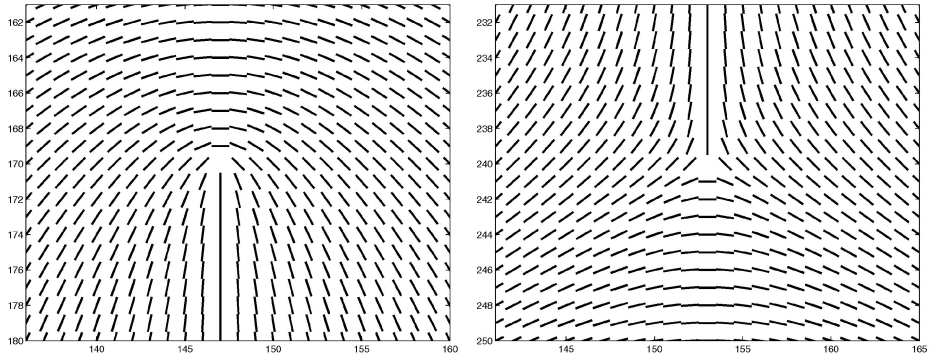



Fig. 8. Reference models of singular points. (a) Core and (b) delta.

index for all other pixels in the image is exactly equal to 0. This is illustrated in Fig. 7.

The exact locations of the SPs in the DF are just between the pixels. Our method detects an SP in all neighboring pixels of the point, because of the region of support of the gradient operator. This results in SP detections that have a size of $2 \times 2$ pixels, as can also be seen in Fig. 7.

### 3.2 Orientation of Singular Points

The last subject of this paper is the estimation of the *orientations* $\varphi$ of the extracted SPs. The method that is described here, makes use of the squared gradient vectors in the neighborhood of an SP, both for the image to be analyzed and for a reference SP. First, reference models of the DFs around standard cores and deltas are constructed. For a core at $(x, y) = (0, 0)$, the reference model that describes the SDF is given by:

$$SDF_{\text{core,ref}} = \frac{(y, -x)}{\sqrt{x^2 + y^2}} \tag{24}$$

and, for a delta at $(x, y) = (0, 0)$, it is given by:

$$SDF_{\text{delta,ref}} = \frac{(-y, -x)}{\sqrt{x^2 + y^2}}. \tag{25}$$

Note that $|SDF_{\text{core,ref}}| = |SDF_{\text{delta,ref}}| = 1$ for all $(x, y)$. The DFs that are associated with these models are shown in Fig. 8.

The SDF in the neighborhood of a core, repeated in Fig. 9a, ideally looks like the reference model in Fig. 9b. The usefulness of the squared gradients is caused by the fact that, when the gray-scale image rotates around the core, all components of the SDF rotate over the same angle, as shown in Appendix C. Therefore, the model of a core that has rotated over an angle $\varphi$, is given by a reference model with all its components multiplied by $e^{j\varphi}$.

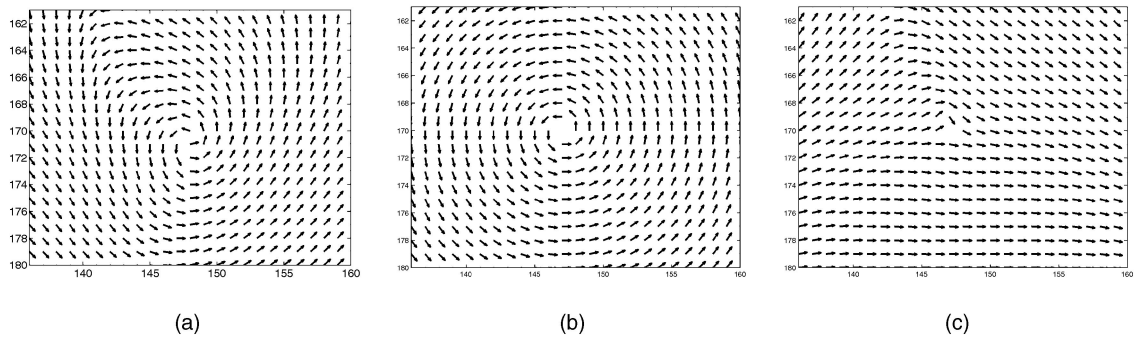
$$SDF_{\text{core},\varphi} = SDF_{\text{core,ref}} \cdot e^{j\varphi}. \tag{26}$$

This property is used for the estimation of the orientation of the core. The orientation of the core with respect to the reference model is found by taking the element-by-element product of the observed squared gradient data $SDF_{\text{core,obs}}(x, y)$ and the complex conjugated of the reference model $SDF_{\text{core,ref}}(x, y)$. This is depicted in Fig. 9c. Then, the elements are summed and the sum is divided by the number of matrix elements $N$, and the angle of the resulting vector is taken.

$$\hat{\varphi}_C = \angle \frac{1}{N} \sum_{x,y} SDF^*_{\text{core,ref}}(x, y) \cdot SDF_{\text{core,obs}}(x, y). \tag{27}$$

The relative orientation of a delta with respect to the reference model is given by one third of the angle of the element-by-element product, as also shown in Appendix C:

$$\hat{\varphi}_D = \frac{1}{3} \angle \frac{1}{N} \sum_{x,y} SDF^*_{\text{delta,ref}}(x, y) \cdot SDF_{\text{delta,obs}}(x, y). \tag{28}$$



(a) (b) (c)

Fig. 9. Processing steps in the calculation of the orientation of a core. (a) SDF around core. (b) SDF around reference core. (c) Orientation estimate.

The averaging operator provides an accurate and unbiased estimate for the orientations $\varphi_C$ and $\varphi_D$. If the observed core is exactly a rotated version of the reference core, the orientation estimate gives:

$$\begin{aligned}\hat{\varphi}_C &= \angle \frac{1}{N}\sum_{x,y} SDF^{*}_{\text{core,ref}}(x,y)\cdot SDF_{\text{core,ref}}(x,y)\cdot e^{j\varphi} \\ &= \angle \frac{1}{N}\sum_{x,y} \left|SDF_{\text{core,ref}}(x,y)\right|^2 \cdot e^{j\varphi} \\ &= \angle e^{j\varphi} = \varphi.\end{aligned} \tag{29}$$

When applying the orientation estimate to the core of Fig. 3, it is found to be rotated 4 degrees clockwise with respect to the reference core of Fig. 8, while the delta of Fig. 3 is found to be rotated 8 degrees counterclockwise with respect to the reference delta of Fig. 8. This corresponds to the estimates that were made by visual inspection.

## 4 Computational Aspects

For efficient calculation of the DF and the coherence, one should not use either of the two basic methods. Instead, first (7), (8), and (9) are used for estimation of $G_{xx}$, $G_{yy}$, and $G_{xy}$, and subsequently (51) and (58) are used for calculation of the DF and the coherence. When calculating those for all pixels in the image, the summations over $W$ reduce to linear filter operations, which can be implemented very efficiently. On a 500 MHz Pentium III computer, an efficient C++ implementation for calculation of the DF and the coherence takes approximately 300 ms of processing time for a fingerprint of 300 by 300 pixels.

For most DF-related tasks, such a high resolution estimate is not needed. In these cases, a simple block-directional field (BDF) with blocks of, for instance, $8 \times 8$ pixels provides enough accuracy. The classical way to estimate a BDF is to partition the image into blocks and estimate $G_{xx}$, $G_{yy}$, and $G_{xy}$ as the average of the block. Sometimes, overlapping blocks are used for some more noise suppression. However, averaging with a uniform window $W$ does not suppress the high-frequency noise sufficiently. Therefore, aliasing introduces artifacts in the DF, which, in turn, creates false singular points.

The cause of this problem is that the length of the averaging filter is set to the same number as the decimation rate. This can be solved by decoupling the size and the shape of the averaging filter $W$ from the subsampling rate. We propose the use of an alternative BDF calculation method that is based on the high-resolution DF. In each block, $G_{xx}$, $G_{yy}$, and $G_{xy}$ are estimated by means of *decimation* of the high-resolution DF. Scale-space theory tells that averaging with a Gaussian window $W$ minimizes the amount of artifacts that are introduced by subsampling [4]. This will reduce the number of false singular points in the DF considerably.

From multirate signal processing, it is known that the filtering and decimation steps can be implemented very efficiently using polyphase filters by interchanging the order of decimation and filtering [27]. Using this method, the calculation of a $4 \times 4$ BDF is expected to take 40 ms on a 500 MHz Pentium III, while the calculation of an $8 \times 8$ BDF is expected to take only 20 ms. Since the SP extraction algorithm makes use of small 2-dimensional filters, it takes 150 ms for a $300 \times 300$ DF. It is expected to take only 10 ms to extract the SPs from a $4 \times 4$ BDF of a fingerprint of $300 \times 300$ pixels.

## 5 Experimental Results

In this section, some experiments will be presented in which the previously derived results are applied to a large number of fingerprints. It will be shown that application of these methods enables the estimation of very accurate and high resolution DFs, accurate SP locations, and correct orientations of the singular points.

We have run our experiments on the second database of the FVC2000 contest [13]. This database contains fingerprint images that are captured by a capacitive sensor with a resolution of 500 pixels per inch. This means that two adjacent ridges are located eight to 12 pixels apart. In this database, 110 untrained individuals are enrolled, each with eight prints of the same finger.

Since there exists no ground truth for the DF of fingerprints, objective error measures cannot be constructed. Therefore, it is difficult to evaluate the quality of a DF estimate quantitatively. Alternatively, the quality of a DF estimate has to be measured indirectly. This section is organized as follows: First, in Section 5.1, the quality of the DF is assessed by means of manual inspection. Next, in Section 5.2, the number of false SPs is used as a measure for the quality of a DF estimate. However, this measure also depends on the segmentation measure used. Then, Section 5.3 presents experimental results on the orientation estimation of the SPs.

### 5.1 Directional Field Estimation

Most authors process fingerprints blockwise [9], [20]. This means that the directional field is not calculated for all pixels individually. Instead, the average DF is calculated in blocks of, for instance, 16 by 16 pixels. In this section, it will

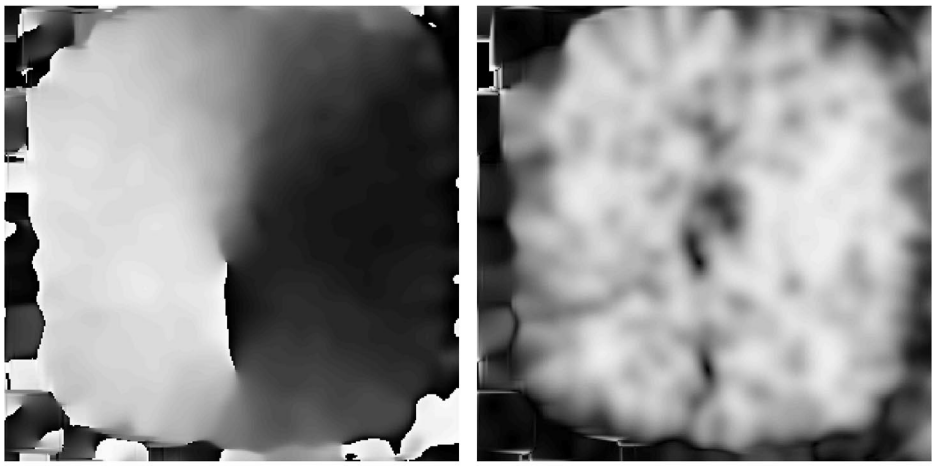



Fig. 10. Gray-scale coded directional field and coherence. (a) Directional field and (b) coherence.

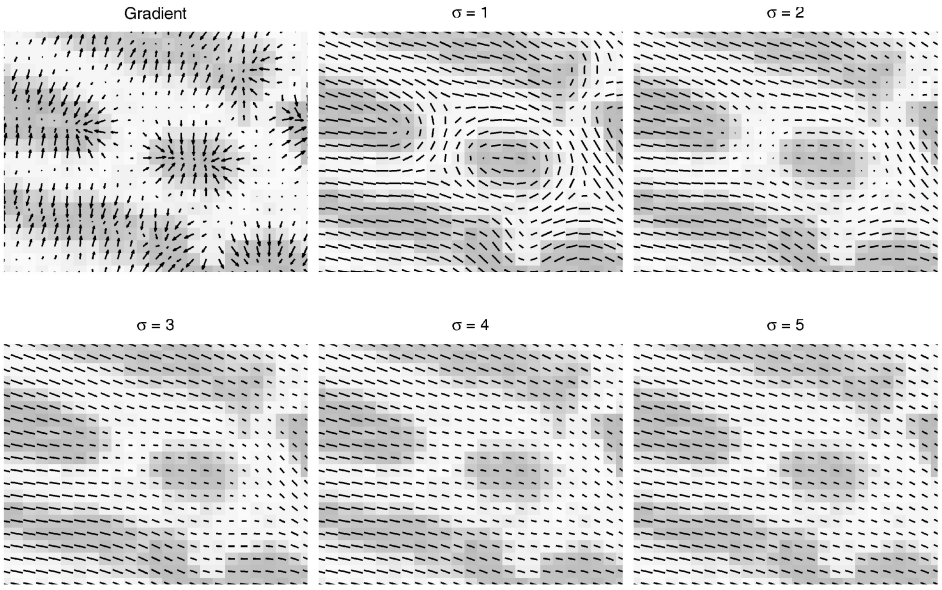



Fig. 11. Gradients and directional field for various values of $\sigma$.

be shown that the processing can be carried out pixelwise, leading to a high resolution and accurate DF estimate.

The first experiment considers the fingerprint of Fig. 1. Although the DF is only shown at discrete steps in Fig. 1b, it is estimated for each pixel. This is illustrated in the gray-scale coded Fig. 10a. In that figure, the angles in the range of $-\frac{1}{2}\pi$ to $\frac{1}{2}\pi$ have uniformly been mapped to the gray-levels from black to white. The figure is somewhat chaotic at the borders since those are areas that consist of noise. However, as shown in Fig. 10b, the coherence is very low in these noisy areas [5]. In this figure, black indicates $Coh = 0$, while white indicates maximum coherence.

Next, an experiment is carried out to illustrate the effects of the choice of the window $W$. We have chosen a Gaussian window, in accordance with the scale-space theory [4]. In Fig. 11, the DF in a small segment of $25 \times 20$ pixels is shown. This segment contains a broken ridge that is almost horizontal. In this experiment, $\sigma$ is chosen in the range from $\sigma = 1$ to $\sigma = 5$. It can be seen that the DF is very erratic for small values of $\sigma$. For higher values of $\sigma$, the DF becomes more uniform, and the lines get longer, indicating higher coherence values.

From this experiment, a window with $\sigma = 5$ seems a good choice. For this value, the DF around a broken ridge is sufficiently averaged. The window has then an effective region of support of approximately 20 pixels ($2\sigma$ on each side), which corresponds to approximately two ridge-valley structures.

### 5.2 Singular Point Extraction

In Section 3, it has been shown that the SP extraction method correctly extracts SPs from the smooth DF of Fig. 4. This was also illustrated in Fig. 1c for the fingerprint of Fig. 1a. In this section, the question will be answered about how well the method performs on a larger set of DFs that are estimated from real fingerprints.

As already mentioned in Section 3, our method extracts *all* SPs from the DF. In case the directional field is not averaged sufficiently, this may result in many false singular points. A DF that has not been averaged at all, may contain

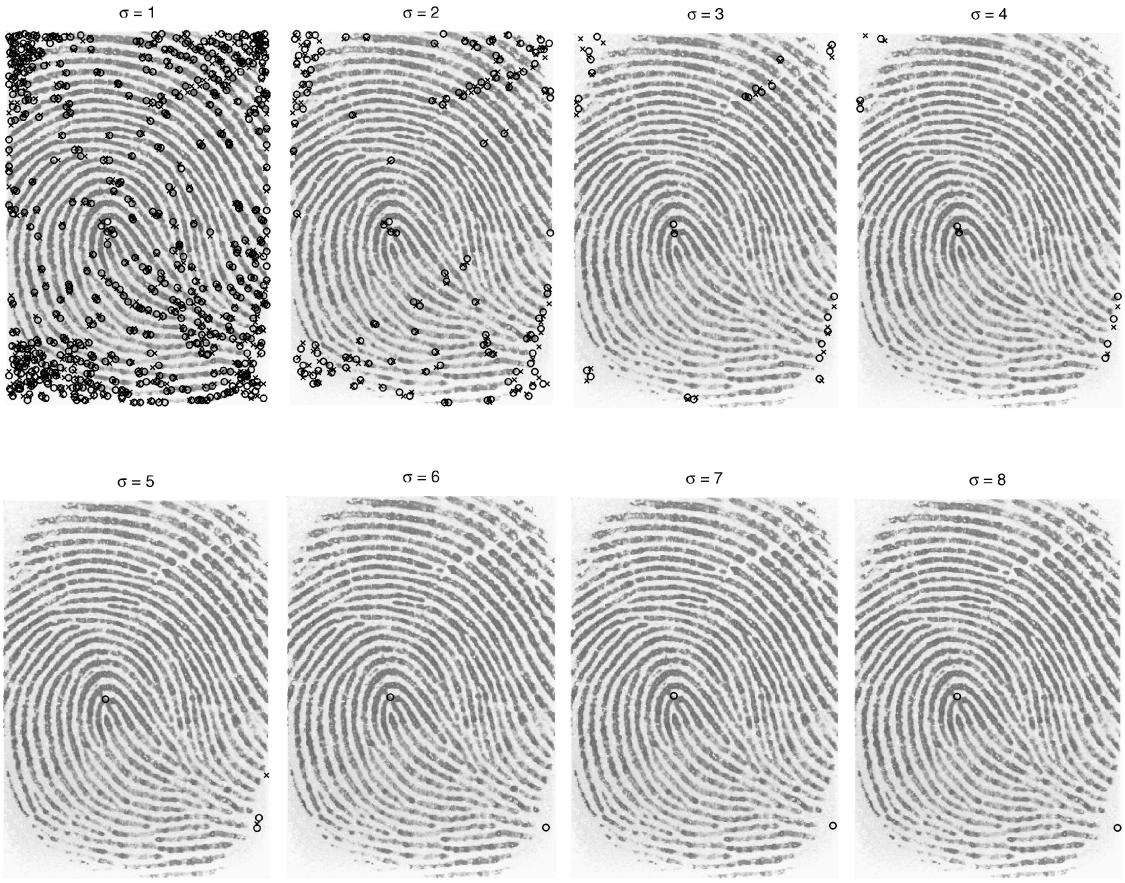



Fig. 12. Extracted singular points for various values of $\sigma$.

as many as 100 spurious core-delta pairs, especially in noisy regions like the borders of the image. When averaging the DF, these pairs either merge and disappear or float off the border of the image [21]. This is illustrated in Fig. 12, where the extracted SPs are shown for various values of $\sigma$. Another example of this behavior can be seen from Fig. 11. For $\sigma = 1$, as many as five false cores and five false deltas can be identified, which all disappear for $\sigma \geq 3$.

In fingerprint recognition, only the SPs at the ridge-valley scale are valid SPs. This means that the SPs have to be extracted from a DF that is estimated at this scale [4]. The coarse-scale directional field can be obtained by averaging it using the algorithms of Section 2. Next, the proposed SP extraction method can be applied. In fact, scale and singular point extraction are two different problems. The SP extraction method will only provide satisfactory results if the scale is chosen well by sufficient averaging. Since a fingerprint never contains more than two core-delta pairs, this might provide a check to see whether the right scale has been reached. Experiments have shown that $\sigma = 6$ is optimal for the database that is used in this section.

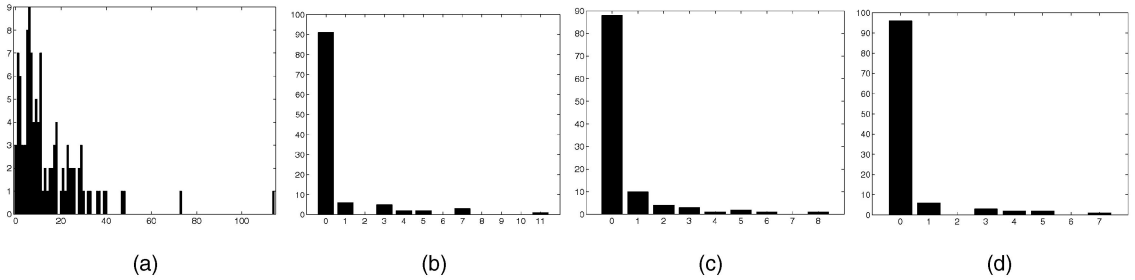
Even when the DF has been averaged sufficiently, the noisy regions outside the fingerprint area may still contain some singular points, as can also be seen from Figs. 10a and 12. More averaging in these regions of low coherence does not always solve this problem: Some false singular points will remain. This may also be the case in fingerprint regions that are very noisy.

A solution is to use *segmentation* in order to discard the false SPs. Segmentation is the partitioning of the image in a "foreground" fingerprint area and a "background" noise area. After segmentation, all SPs that are in the background can be discarded. Segmentation of inked fingerprint images is a relatively straightforward task since the background contains not much noise. Therefore, measures like the local mean gray level and the local variance of the gray level can be used [19]. However, the segmentation of live-scanned fingerprint images is much harder, since they contain much more background noise. Therefore, more advanced segmentation methods that use, for instance, the coherence as measure have to be used.

In our experiment, SPs are extracted from the first prints of all fingers of the second FVC2000 database, using the method of Section 3 and a Gaussian window with $\sigma = 6$. For the purpose of reference, the SPs in all prints were marked by human inspection. The average number of false and missed SPs are shown in Table 1, while the distribution of

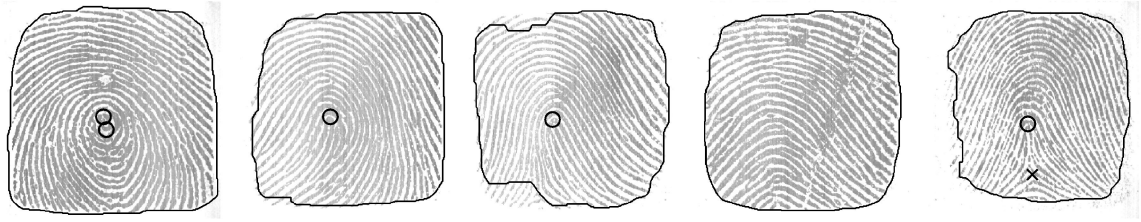
TABLE 1
Results of SP Extraction

| Segmentation method | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 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 |



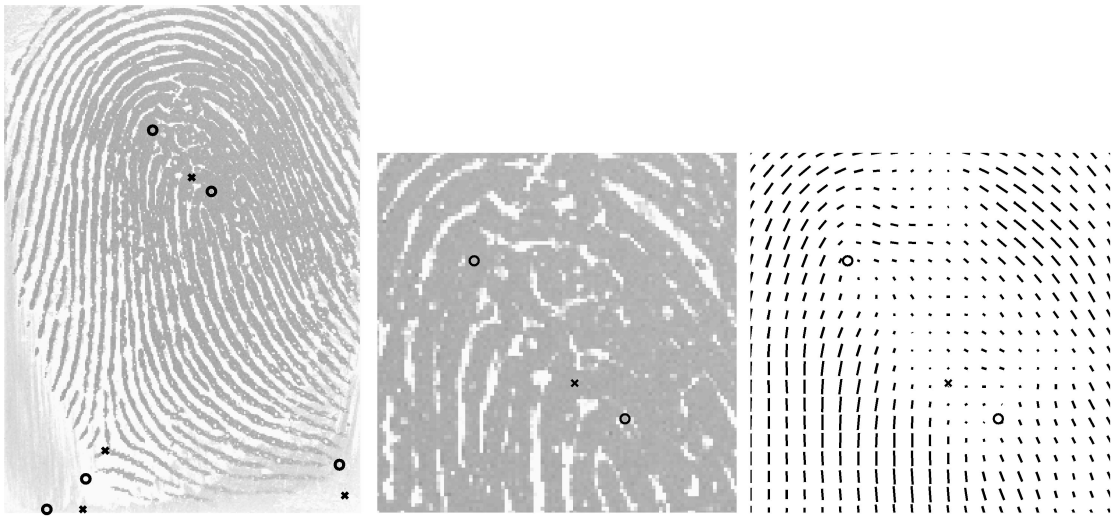
(a) (b) (c) (d)

Fig. 13. Distribution of the number of false singular points for various segmentation methods. (a) No segmentation. (b) Manual segmentation. (c) Segmentation 1. (d) Segmentation 2.



(a) (b) (c) (d) (e)

Fig. 14. SP extraction for examples from each of the five Henry classes. (a) Whorl. (b) Left loop. (c) Right loop. (d) Arch. (e) Tented arch.



(a) (b) (c)

Fig. 15. Example of extraction of spurious SPs. (a) Fingerprint. (b) Center Area. (c) DF of center area.

the numbers of false SPs is shown in Fig. 13 for four different types of segmentation:

1. No segmentation, the whole image is taken as fingerprint region.
2. Manual segmentation.
3. High resolution segmentation algorithm that uses the coherence estimate as feature and morphological operators to smooth the segmentation result [5].
4. High resolution segmentation algorithm that uses the coherence, the mean, and the variance of the fingerprint image as features and morphological operators to smooth the segmentation result [6].

In Fig. 14, the extracted SPs for fingerprints of the five Henry classes are shown. It can be seen that the SP-extraction algorithm has no difficulties in distinguishing a *tented arch*, which contains one core and one delta, from an *arch*, which contains neither of both. Furthermore, the figure shows that the delta in the *right loop* is not detected, although it is visible in the image. The segmentation boundary, which is also shown in the figure, positions this delta just outside the foreground area.

In Fig. 15, an example of the extraction of spurious SPs is shown. From the surroundings of the noisy center area, it can be concluded that this area should contain one core. However, the DF contains two cores and one delta in this area. From the short lines in the DF of Fig. 15c, it can be seen that the coherence is very low in this area. These false SPs can be eliminated by further averaging the DF, but that takes a window as large as $\sigma = 11$. In this case, it would be a better solution to develop segmentation algorithms that are capable of detecting low-quality areas and discard spurious

core-delta pairs from these areas. Furthermore, fingerprints of very bad quality, having a low coherence value in the entire print, should be rejected entirely.

### 5.3 Orientation of Singular Points

The last experiment shows the accuracy of the estimated orientation of SPs using the method of Section 3.2. In this experiment, first the orientations of all valid SPs are marked manually. Next, these orientations are determined automatically.

The distribution of the errors of the orientations, $e_\varphi = \hat{\varphi} - \varphi$, is as follows: The estimate is unbiased since the mean error is $\text{mean}[e_\varphi] = -0.012 = -0.7$ degrees. Furthermore, the variance of the estimate is $\sigma^2_{e_\varphi} = 0.044$, which means that the standard deviation is only $\sigma_{e_\varphi} = 0.21 = 12$ degrees. Therefore, we conclude that our method provides an accurate estimate of the orientations of SPs.

## 6 Conclusions

In this paper, a new PCA-based method for estimating directional fields from fingerprints is proposed. Since it is proven that this method provides exactly the same results as the traditional method, the method offers a different view and an increase of insight on the problem of estimating an "average" gradient. It is pointed out that the methods that are presented in this paper can be used either to estimate a high-resolution DF or to improve the accuracy of block directional fields.

The singular-point-extraction method that is proposed in this paper offers consistent binary decisions and can be implemented very efficiently. It is capable of high resolution SP extraction and does not need to use heuristic postprocessing. Furthermore, it is shown that a high-resolution DF can be used for the accurate estimation of the orientation of SPs. To further improve the error rates of SP extraction, accurate segmentation algorithms have to be developed that are capable of detecting low-quality areas in a fingerprint. Then, spurious core-delta pairs can be discarded from these areas.

## Acknowledgments

This research has been carried out within the Euregio Computational Intelligence Center (ECIC), subsidized by the European Commission, The Netherlands, and Nordrhein-Westfalen, in the scope of the Interreg Program.

## Appendix A

## Equivalence of DF Estimation Methods

It will be proven that the squared gradient method and the PCA-based method for the estimation of the DF are exactly equivalent. The proof starts by deriving the inverse of (4), which was given to be:

$$\begin{bmatrix} G_{s,x} \\ G_{s,y} \end{bmatrix} = \begin{bmatrix} G_x^2 - G_y^2 \\ 2G_xG_y \end{bmatrix}. \tag{30}$$

Substituting the lower part of this expression, which is given by

$$G_y = \frac{G_{s,y}}{2G_x} \tag{31}$$

into the upper part, given by

$$G_{s,x} = (G_x^2 - G_y^2) \tag{32}$$

gives

$$G_x^4 - G_{s,x}G_x^2 - \frac{1}{4}G_{s,y} = 0. \tag{33}$$

Solving this for $G_x$ gives:

$$G_x = \begin{cases} \frac{1}{2}\sqrt{2G_{s,x} + 2\sqrt{G_{s,x}^2 + G_{s,y}^2}} \\ -\frac{1}{2}\sqrt{2G_{s,x} + 2\sqrt{G_{s,x}^2 + G_{s,y}^2}} \\ \frac{1}{2}\sqrt{2G_{s,x} - 2\sqrt{G_{s,x}^2 + G_{s,y}^2}} \\ -\frac{1}{2}\sqrt{2G_{s,x} - 2\sqrt{G_{s,x}^2 + G_{s,y}^2}} \end{cases} \tag{34}$$

The second and fourth solutions can be eliminated since $G_x$ is always positive. Furthermore, since

$$\sqrt{G_{s,x}^2 + G_{s,y}^2} \geq G_{s,x},$$

the third solution results in the square root of a negative number. Therefore, only the first solution is valid:

$$G_x = \frac{1}{2}\sqrt{2G_{s,x} + 2\sqrt{G_{s,x}^2 + G_{s,y}^2}}. \tag{35}$$

The next step is to consider the squared gradients, averaged over the window $W$ and to substitute, according to (6):

$$\overline{G_{s,x}} = G_{xx} - G_{yy} \tag{36}$$

$$\overline{G_{s,y}} = 2G_{xy}. \tag{37}$$

The average gradients, derived from the averaged squared gradients, are:

$$\begin{aligned} \overline{G_x} &= \frac{1}{2}\sqrt{2\overline{G_{s,x}} + 2\sqrt{\overline{G_{s,x}}^2 + \overline{G_{s,y}}^2}} \\ &= \sqrt{\frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}} \end{aligned} \tag{38}$$

and

$$\begin{aligned} \overline{G_y} &= \frac{\overline{G_{s,y}}}{2\overline{G_x}} = \frac{G_{xy}}{\overline{G_x}} \\ &= \frac{G_{xy}}{\sqrt{\frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}}}. \end{aligned} \tag{39}$$

Now, it will be shown that the vector:

$$\begin{bmatrix} \overline{G_x} \\ \overline{G_y} \end{bmatrix} = \frac{1}{c} \cdot \begin{bmatrix} \frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2} \\ G_{xy} \end{bmatrix} \tag{40}$$

with:

$$c = \sqrt{\frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}} \tag{41}$$

is an eigenvector of autocovariance matrix $\mathbf{C}$, which is defined in (14). This will prove that both methods are

equivalent. For the eigenvectors of $\mathbf{C}$, the following expression must hold:

$$\mathbf{C} \cdot \mathbf{V} = \mathbf{V} \cdot \mathbf{\Lambda}, \tag{42}$$

where the columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{C}$ and $\mathbf{\Lambda}$ is the diagonal matrix of the corresponding eigenvalues. This expression must also hold for one eigenvector $\mathbf{v}_1$ with corresponding eigenvalue $\lambda_1$:

$$\mathbf{C} \cdot \mathbf{v}_1 = \lambda_1 \cdot \mathbf{v}_1 \tag{43}$$

In order to show this, $[\overline{G_x}\ \overline{G_y}]^T$ is substituted for $\mathbf{v}_1$

$$\mathbf{v}_1 = \begin{bmatrix} \overline{G_x} \\ \overline{G_y} \end{bmatrix} \tag{44}$$

in the left-hand side of (43). This gives:

$$\begin{aligned}
&\mathbf{C} \cdot \mathbf{v}_1 = \\
&\begin{bmatrix} G_{xx} & G_{xy} \\ G_{xy} & G_{yy} \end{bmatrix} \cdot \frac{1}{c} \cdot \\
&\begin{bmatrix} \frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2} \\ G_{xy} \end{bmatrix} = \frac{1}{c} \cdot \\
&\begin{bmatrix} G_{xx}\left(\frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}\right) + G_{xy}^2 \\ G_{xy}\left(\frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}\right) + G_{xy}G_{yy} \end{bmatrix}.
\end{aligned} \tag{45}$$

This must be equal to $\lambda_1 \cdot [\overline{G_x}, \overline{G_y}]^T$. Calculating $\lambda_1$ from the upper half of these expressions, we find:

$$\lambda_1 = \frac{G_{xx}\left(\frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}\right) + G_{xy}^2}{\frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}} \tag{46}$$

which, by multiplying numerator and denominator by

$$\frac{1}{2}(G_{xx} - G_{yy}) - \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2},$$

can be simplified to:

$$\lambda_1 = \frac{1}{2}(G_{xx} + G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}. \tag{47}$$

From the lower half of these expressions, we find:

$$\lambda_1 = \frac{G_{xy}\left(\frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}\right) + G_{xy}G_{yy}}{G_{xy}} \tag{48}$$

which can be easily simplified to:

$$\lambda_1 = \frac{1}{2}(G_{xx} + G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2} \tag{49}$$

Since both expressions give the same result for $\lambda_1$, $[\overline{G_x}, \overline{G_y}]^T$ is an eigenvector of $\mathbf{C}$. Therefore, both methods are exactly equivalent.

It is not difficult to derive the second eigenvector $\mathbf{v}_2$ and its corresponding eigenvalue $\lambda_2$:

$$\mathbf{v}_1 = \begin{bmatrix} \frac{1}{2}(G_{xx} - G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2} \\ G_{xy} \end{bmatrix} \tag{50}$$

$$\mathbf{v}_2 = \begin{bmatrix} \frac{1}{2}(G_{xx} - G_{yy}) - \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2} \\ G_{xy} \end{bmatrix} \tag{51}$$

$$\lambda_1 = \frac{1}{2}(G_{xx} + G_{yy}) + \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2} \tag{52}$$

$$\lambda_2 = \frac{1}{2}(G_{xx} + G_{yy}) - \frac{1}{2}\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2} \tag{53}$$

Note that $\lambda_1$ is always larger than or equal to $\lambda_2$ confirming that the average gradient angle is aligned with $\mathbf{v}_1$. The DF, which is perpendicular to the gradient is aligned with $\mathbf{v}_2$.

## APPENDIX B

### EQUIVALENCE OF $Coh$ AND $Str$

By substituting (52) and (53), $Str$ is given by:

$$\begin{aligned}
Str &= \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2} \\
&= \frac{\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}}{G_{xx} + G_{yy}}.
\end{aligned} \tag{54}$$

On the other hand, $Coh$ is given by:

$$Coh = \frac{\left|\sum_W (G_{s,x}, G_{s,y})\right|}{\sum_W \left|(G_{s,x}, G_{s,y})\right|}, \tag{55}$$

where, by substituting (4),

$$\begin{aligned}
\left|\sum_W (G_{s,x}, G_{s,y})\right| &= \sqrt{\left(\sum_W G_{s,x}\right)^2 + \left(\sum_W G_{s,y}\right)^2} \\
&= \sqrt{\left(\sum_W G_x^2 - G_y^2\right)^2 + \left(\sum_W 2G_xG_y\right)^2} \\
&= \sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}
\end{aligned} \tag{56}$$

and

$$\begin{aligned}
\sum_W \left|(G_{s,x}, G_{s,y})\right| &= \sum_W \sqrt{G_{s,x}^2 + G_{s,y}^2} \\
&= \sum_W \sqrt{(G_x^2 - G_y^2)^2 + (2G_xG_y)^2} \\
&= \sum_W \sqrt{G_x^4 + 2G_x^2G_y^2 + G_y^4} \\
&= \sum_W \sqrt{(G_x^2 + G_y^2)^2} \\
&= \sum_W G_x^2 + G_y^2 \\
&= G_{xx} + G_{yy}.
\end{aligned} \tag{57}$$

Therefore, the coherence of the averaging method is given by:

$$Coh = \frac{\sqrt{(G_{xx} - G_{yy})^2 + 4G_{xy}^2}}{G_{xx} + G_{yy}} \quad (58)$$

which proves the equivalence of $Coh$ and $Str$.

## APPENDIX C
## ROTATION OF SINGULAR POINTS

It can be proven that

$$SDF_{\text{core},\varphi} = SDF_{\text{core,ref}} \cdot e^{j\varphi} \quad (59)$$

by using polar notation $(\rho_s, \phi_s)$ instead of $(x, y)$ for a position in the reference model of the SPs. The orientation of the SDF is given by:

$$2\theta_{\text{core,ref}}(\rho_s, \phi_s) = \phi_s + \frac{1}{2}\pi \quad (60)$$

and the DF is given by:

$$\theta_{\text{core,ref}}(\rho_s, \phi_s) = \frac{1}{2}\phi_s + \frac{1}{4}\pi. \quad (61)$$

The problem is to determine the SDF at position $(\rho_s, \phi_s)$ after rotation of the reference model over an angle $\varphi$. The sample point at $(\rho_s, \phi_s)$ after the rotation is located at $(\rho_s, \phi_s - \varphi)$ before the rotation:

$$\theta_{\text{core,ref}}(\rho_s, \phi_s - \varphi) = \frac{1}{2}(\phi_s - \varphi) + \frac{1}{4}\pi. \quad (62)$$

The rotation adds $\varphi$ to the orientation at the sample point:

$$\theta_{\text{core},\varphi}(\rho_s, \phi_s) = \frac{1}{2}(\phi_s - \varphi) + \frac{1}{4}\pi + \varphi. \quad (63)$$

Now, the rotated DF can be converted back to the rotated SDF:

$$2\theta_{\text{core},\varphi}(\rho_s, \phi_s) = (\phi_s + \frac{1}{2}\pi) + \varphi = 2\theta_{\text{core,ref}}(\rho_s, \phi_s) + \varphi \quad (64)$$

which completes the proof. From the formula it becomes obvious that the SDF model of a core has to be rotated over $2\pi$ in order to obtain the original model.

Following the same procedure for a delta, it can be proven that

$$SDF_{\text{core},\varphi} = SDF_{\text{core,ref}} \cdot e^{j3\varphi}. \quad (65)$$

Now, the orientation of the SDF is given by:

$$2\theta_{\text{delta,ref}}(\rho_s, \phi_s) = -\phi_s + \frac{1}{2}\pi. \quad (66)$$

Following the same procedure as for the core gives:

$$\theta_{\text{delta},\varphi}(\rho_s, \phi_s) = -\frac{1}{2}(\phi_s - \varphi) + \frac{1}{4}\pi + \varphi \quad (67)$$

and:

$$2\theta_{\text{delta},\varphi}(\rho_s, \phi_s) = (-\phi_s + \frac{1}{2}\pi) + 3\varphi = 2\theta_{\text{delta,ref}}(\rho_s, \phi_s) + 3\varphi. \quad (68)$$

This corresponds to the fact that a delta has to be rotated over $\frac{2}{3}\pi$ in order to obtain the original model.

## REFERENCES


[1] R.S. Germain, A. Califano, and S. Colville, "Fingerprint Matching Using Transformation Parameter Clustering," *IEEE Computational Science and Eng.*, vol. 4, no. 4, pp. 42-49, 1997.

[2] R. Cappelli, D. Maio, and D. Maltoni, "Indexing Fingerprint Databases for Efficient 1:n Matching," *Proc. Sixth Int'l Conf. Control, Automation, Robotics and Vision (ICARCV2000)*, Dec. 2000.

[3] J. de Boer, A.M. Bazen, and S.H. Gerez, "Indexing Fingerprint Databases Based on Multiple Features," *Proc. ProRISC2001, 12th Ann. Workshop Circuits, Systems and Signal Processing*, Nov. 2001.

[4] T. Lindeberg, *Scale-Space Theory in Computer Vision.* Boston: Kluwer Academic Publishers, 1994.

[5] A.M. Bazen and S.H. Gerez, "Directional Field Computation for Fingerprints Based on the Principal Component Analysis of Local Gradients," *Proc. ProRISC2000, 11th Ann. Workshop Circuits, Systems and Signal Processing*, Nov. 2000.

[6] A.M. Bazen and S.H. Gerez, "Segmentation of Fingerprint Images," *Proc. ProRISC2001, 12th Ann. Workshop Circuits, Systems and Signal Processing*, Nov. 2001.

[7] E.R. Henry, *Classification and Uses of Finger Prints.* London: Routledge, 1900.

[8] A.M. Bazen and S.H. Gerez, "Extraction of Singular Points from Directional Fields of Fingerprints," Mobile Comm. in Perspective, CTIT Workshop Mobile Comm., Univ. of Twente, Enschede, The Netherlands, pp. 41-44, Feb. 2001.

[9] K. Karu and A.K. Jain, "Fingerprint Classification," *Pattern Recognition*, vol. 29, no. 3, pp. 389-404, 1996.

[10] M. Kawagoe and A. Tojo, "Fingerprint Pattern Classification," *Pattern Recognition*, vol. 17, no. 3, pp. 295-303, 1984.

[11] M.M.S. Chong, T.H. Ngee, and R.K.L. Gay, "Geometric Framework for Fingerprint Image Classification," *Pattern Recognition*, vol. 30, no. 9, pp. 1475-1488, 1997.

[12] A.M. Bazen and S.H. Gerez, "An Intrinsic Coordinate System for Fingerprint Matching," *Proc. Third Int'l Conf. Audio- and Video-Based Biometric Person Authentication (AVBPA 2001)*, June 2001.

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

[14] G.A. Drets and H.G. Liljenström, "Fingerprint Subclassification: A Neural Network Approach," *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, L.C. Jain, U. Halici, I. Hayashi, S.B. Lee, and S. Tsutsui, eds., pp. 109-134, Boca Raton, Fla.: CRC Press, 1999.

[15] C.L. Wilson, G.T. Candela, and C.I. Watson, "Neural Network Fingerprint Classification," *J. Artificial Neural Networks*, vol. 1, no. 2, pp. 203-228, 1994.

[16] L. O'Gorman and J.V. Nickerson, "An Approach to Fingerprint Filter Design," *Pattern Recognition*, vol. 22, no. 1, pp. 29-38, 1989.

[17] M. Kass and A. Witkin, "Analyzing Oriented Patterns," *Computer Vision, Graphics, and Image Processing*, vol. 37, no. 3, pp. 362-385, Mar. 1987.

[18] A.R. Rao and R.C. Jain, "Computerized Flow Field Analysis: Oriented Texture Fields," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 7, pp. 693-709, July 1992.

[19] 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.

[20] 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, Sept. 1997.

[21] P. Perona, "Orientation Diffusions," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 457-467, Mar. 1998.

[22] C.W. Therrien, *Discrete Random Signals and Statistical Signal Processing.* Upper Saddle River, N.J.: Prentice-Hall, 1992.

[23] O. Nakamura, K. Goto, and T. Minami, "Fingerprint Classification by Directional Distribution Patterns," *Systems, Computers, Controls*, vol. 13, no. 5, pp. 81-89, 1982.

[24] V.S. Srinivasan and N.N. Murthy, "Detection of Singular Points in Fingerprint Images," *Pattern Recognition*, vol. 25, no. 2, pp. 139-153, 1992.

[25] 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, May 2000.

[26] L. Hong and A.K. Jain, "Classification of Fingerprint Images," *Proc. 11th Scandinavian Conf. Image Analysis,* June 1999.
[27] J.G. Proakis, C.M. Rader, F. Ling, and C.L. Nikias, *Advanced Digital Signal Processing*. New York: Macmillan Publishing Company, 1992.

**Asker M. Bazen** received the MSc degree in electrical engineering from the University of Twente, The Netherlands, in 1998 for his research on high-resolution parametric radar processing, which he continued for one more year at Thomson-CSF Signal. Currently, he is finishing his PhD thesis at the Chair of Signals and Systems at the University of Twente on various topics in fingerprint recognition, including robust minutiae extraction from low-quality fingerprints, matching elastically deformed fingerprints, and indexing large fingerprint databases. Other research interests include signal and image processing, pattern recognition, and computational intelligence.

**Sabih H. Gerez** received the MSc degree in electrical engineering and the PhD degree in applied sciences from the University of Twente, The Netherlands, in 1984 and 1989, respectively. He has worked for the Department of Electrical Engineering at the Universtity of Twente as an assistant researcher (in the period 1984-1989) and as an assistant professor (since 1990). Starting from 2001, he is combining his academic activities with a position at National Semiconductor, Design Center Hengelo, The Netherlands. His research and teaching interests include design automation, VLSI design, signal processing, and computational intelligence. He is the author of the book *Algorithms for VLSI Design Automation* (Wiley, 1999).