

Detection of Singularities in Fingerprint Images using Linear Phase Portraits

Surinder Ram, Horst Bischof and Josef Birchbauer

Abstract The performance of fingerprint recognition is heavily depending on the reliable extraction of singularities. Common algorithms are based on a Poincaré-Index estimation. These algorithms are only robust when certain heuristics and rules are applied. In this paper we present a model based approach for the detection of singular points. The presented method exploits the geometric nature of linear differential equation systems. Our method is robust against noise in the input image and is able to detect singularities even if they are partly occluded. The algorithm proceeds by fitting linear phase portraits at each location of a sliding window and then analyses its parameters. Using a well established mathematical background, our algorithm is able to decide if a singular point is existent. Furthermore, the parameters can be used to classify the type of the singular point into whorls, deltas and loops.

1 Introduction

The application of fingerprint-based personal authentication and identification has rapidly increased in the recent years. The use of this technology can be seen in forensics, commercial industry and government agencies, to mention a few. Fingerprints are attractive for identification because they can characterize an individual uniquely and their configuration does not change through the life of individuals. The processing steps required for personal verification or identification based on fingerprints

Surinder Ram and Horst Bischof
Technical University of Graz, Austria
Institute for Computer Graphics and Vision
e-mail: ram@icg.tugraz.at, e-mail: bischof@icg.tugraz.at

Josef Birchbauer
Siemens Austria
Siemens IT Solutions and Services, Biometrics Center
e-mail: josef-alois.birchbauer@siemens.com

consists of acquisition, feature extraction, matching and a final decision [7].

Large volumes of fingerprints are collected and stored everyday in a wide range of applications. Automatic fingerprint recognition requires that the input fingerprint is matched with a large number of fingerprints stored in a database. Therefore, the first step in an identification system is the classification of a given fingerprint into five categories [7]. This reduces the amount of data to be searched for matches as the database can be partitioned into subsets. These five categories are: arch, tented arch, left loop, right loop and whorl. Common algorithms extract singular points in fingerprint images and perform a classification based on the number and location of these singularities.

In order to 'match' two fingerprints it is necessary to extract minutiae, which are special points in fingerprints where ridges end or bifurcate. Two fingerprints can be reported as equal, if a certain number of minutiae positions are identical in both fingerprints. In general, matching of fingerprint images is a difficult task [5], mainly due to the large variability in different impressions of the same finger (i.e. displacement, rotation, distortion, noise, etc.). One way to relax the problem in terms of performance and runtime is to use certain "landmarks" in the image in order to apply a pose transformation. Since singular points are unique landmarks in fingerprints, they are used as reference points for matching [8].

1.1 Methods for Extraction of Singularities

Many approaches are described for singular point detection in literature. Karu and Jain [6] referred to a Poincaré-Index method. However, there are principle weaknesses adhered to this method. Many rules and heuristics have been proposed by various authors (e.g. [20]) in order to make the method robust against noise and minor occlusions. Due to its simplicity and more than adequate performance in most images, this method enjoys high popularity in fingerprint recognition systems.

Another method, described in [11] exploits the fact that partitioning the orientation image in regions, characterized by homogeneous orientations, implicitly reveals the position of singularities. The borderline between two adjacent regions is called a fault-line. By noting that fault lines converge towards loop singularities and diverge from deltas, the authors define a geometrical method for determining the convergence and divergence points.

Nilson et al. [14] identify singular points by their symmetry properties. In particular this is done using complex filters, which are applied to the orientation field in multiple resolution scales. The detection of possible singularities is done by analysing the response created by these complex filters.

1.2 Model based detection of Singularities

In [16] Rao et al. proposed a novel algorithm for singular point detection in flow fields. Their main idea is to locally approximate a flow pattern by a two-dimensional linear differential equation. This allows a parametric representation of different types of phase portraits, and their classification is possible based on the extracted parameters.

In this paper, we present a novel method for the detection of singularities based on the work of Rao et al. In comparison, our method is robust against noise in the input image and is able to detect singularities even if they are partly occluded. Additionally, we present methods for detection and recognition of all types of singularities in fingerprint images, whereas Rao et al. presented a model for vortices only. This model based attempt is new to the field of fingerprint singularity detection.

1.3 Outline

In Section 2 an explanation of linear phase portraits is given. Furthermore, the fitting of the parameters is explained in detail. We analyse the weaknesses of existing algorithms and propose a robust parameter fitting method.

In Section 3 we explain how this algorithm can be applied to fingerprint images.

Section 4 shows the conducted experiments. In the first part, we demonstrate the noise and occlusion robustness of our algorithm. Furthermore, we have tested our algorithm on 2x280 hand-labelled images in order to demonstrate the singular point detection capabilities.

Finally, in the last section a summary of the proposed method is given.

2 Two-Dimensional Linear Phase Portraits

Phase portraits are a powerful mathematical model for describing oriented textures, and therefore have been applied by many authors [3, 16, 19]. Linear phase portraits can be expressed by the following differential equation system:

$$\frac{dx}{dt} = \dot{x} = p(x,y) = cx + dy + f \quad \frac{dy}{dt} = \dot{y} = q(x,y) = ax + cy + e \quad (1)$$

By varying the parameters of these equations we can describe a set of oriented textures comprising saddles, star nodes, nodes, improper nodes, centers and spirals [9] (examples are given in Figure 1). The orientation of these fields is given by:

$$\phi(x,y) = \text{atan} \left(\frac{dy}{dx} \right) = \text{atan} \left(\frac{\dot{y}}{\dot{x}} \right) = \text{atan} \left(\frac{ax + cy + e}{cx + dy + f} \right) \quad (2)$$

The equations in (1) can further be represented in a more convenient matrix notation as:

$$\dot{\mathbf{X}} = \mathbf{A} * \mathbf{X} + \mathbf{B} \text{ where } \mathbf{X} = \begin{bmatrix} x \\ y \end{bmatrix}, \mathbf{A} = \begin{bmatrix} c & d \\ a & b \end{bmatrix}, \mathbf{B} = \begin{bmatrix} f \\ e \end{bmatrix} \quad (3)$$

\mathbf{A} is called the *characteristic matrix* of the system. A point at which \dot{x} and \dot{y} are zero is called a *critical point* (x_0, y_0) [9]. The elements of the characteristic matrix are used to determine between six flow pattern. [15]. The type of the flow pattern is determined by the eigenvalues of the characteristic matrix.

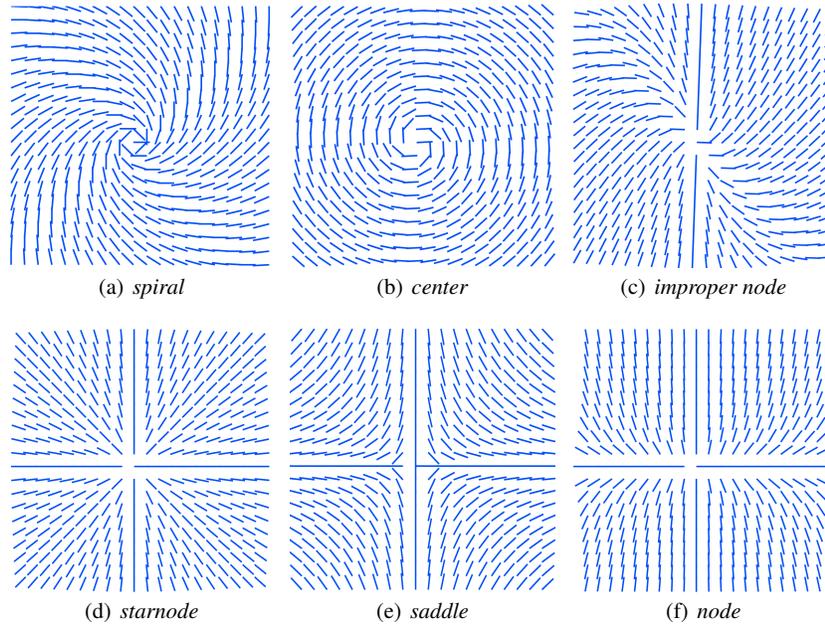


Fig. 1 A classification of different phase portraits based on the characteristic matrix \mathbf{A} [15]. Complex eigenvalues result in a spiral 1(a) or a center 1(b) type pattern, differentiated from each other only by the real part of the eigenvalues. If the eigenvalues are real and both equal, the pattern can be classified into a startnode 1(d) or into an improper node 1(c). Miscellaneous real valued eigenvalues result in a saddle 1(e) or a node 1(f), only distinguished by their signs.

2.1 Parameter estimation

Rao and Jain [15] presented an algorithm for parameter estimation of linear phase portraits. However, the non-linear least squares computation required in their original algorithm is computationally expensive and prone to local minima. In [18], Shu

et al. presented a linear formulation of an algorithm which computes the critical points and parameters for a two dimensional phase portrait. Because their approach is linear, there exists a closed form solution. Recent applications of the algorithm can be seen in [10, 21]. Throughout this paper we refer to this algorithm as the linear least squares algorithm. In the following section we give a brief introduction of the algorithm presented by Shu et al. in [18, 19].

To solve the problem one can apply a least squares algorithm. Equation (2) can be expressed as:

$$p(x_i, y_i) - \tan\phi_i * q(x_i, y_i) = 0 \quad (4)$$

We can directly estimate the parameters by using the triplet data $(x_i, y_i, \tan\phi_i)$ and (4), where (x_i, y_i) is the coordinate of a pixel and the $\tan\phi_i$ the observed data. Let $\tan\phi_i = \zeta_i$; The optimal weighted least square estimator is one that minimizes the following cost function:

$$\sum_{i=0}^n \omega_i^2 \cdot [p(x_i, y_i) - \zeta_i \cdot q(x_i, y_i)]^2, \quad (5)$$

which can be rewritten as:

$$\sum_{i=0}^n \omega_i^2 \cdot [ax_i + by_i - \zeta_i cx_i - \zeta_i dy_i + e - \zeta_i f]^2 \quad (6)$$

which is subject to the constraint: $\sqrt{a^2 + b^2 + c^2 + d^2} = 1$. Where $w_i = \cos\phi_i$, and is applied because the tangent function is not uniformly sensitive to noise, so each observed data has to be weighted by the inverse of the derivate of the tangents function. n is the total number of triplet data used to estimate the parameter set (a, b, c, d, e, f) .

Let

$$L_4 = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}, L_2 = \begin{bmatrix} e \\ f \end{bmatrix}, \Omega_2 = \begin{bmatrix} \omega_0 - \zeta_0 \omega_0 \\ \omega_1 - \zeta_1 \omega_1 \\ \omega_2 - \zeta_2 \omega_2 \\ \vdots \\ \omega_n - \zeta_n \omega_n \end{bmatrix} \quad (7)$$

and

$$\Omega_4 = \begin{bmatrix} x_0 \omega_0 & x_0 \omega_0 & \zeta_0 x_0 \omega_0 & \zeta_0 y_0 \omega_0 \\ x_1 \omega_1 & x_1 \omega_1 & \zeta_1 x_1 \omega_1 & \zeta_1 y_1 \omega_1 \\ x_2 \omega_2 & x_2 \omega_2 & \zeta_2 x_2 \omega_2 & \zeta_2 y_2 \omega_2 \\ \vdots & \vdots & \vdots & \vdots \\ x_n \omega_n & x_n \omega_n & \zeta_n x_n \omega_n & \zeta_n y_n \omega_n \end{bmatrix} \quad (8)$$

Now we can express the previous constrained optimization as minimizing the cost function:

$$C = (\Omega_4 L_4 + \Omega_2 L_2)^T (\Omega_4 L_4 + \Omega_2 L_2) + \lambda (L_4^T L_4 - 1) \quad (9)$$

Differentiating C with respect to L_4 , L_2 and to the Langrangian Multiplier λ , and setting the derivatives to zero, we obtain:

$$\frac{\partial C}{\partial L_4} = 2\Omega_4^T \Omega_4 L_4 + 2\Omega_4^T \Omega_2 L_2 + 2\lambda L_4 = 0$$

$$\frac{\partial C}{\partial L_2} = 2\Omega_2^T \Omega_2 L_2 + \Omega_2^T \Omega_4 L_4 = 0$$

$$\frac{\partial C}{\partial \lambda} = L_4^T L_4 - 1 = 0$$

which yields:

$$L_4^T L_4 = 1$$

$$L_2 = -(\Omega_2^T \Omega_2)^{-1} \Omega_2^T \Omega_4 L_4 \quad (10)$$

$$\psi L_4 = \lambda L_4 \quad (11)$$

where:

$$\psi = -\Omega_4^T \Omega_4 + \Omega_4^T \Omega_2 (\Omega_2^T \Omega_2)^{-1} (\Omega_2^T \Omega_4). \quad (12)$$

L_4 is an eigenvector of the symmetric matrix ψ and λ is its eigenvalue. Therefore, the eigenvector with the smallest absolute eigenvalue gives the best estimation of L_4 . We can further compute L_2 by using Equation (10).

2.2 Algorithm Analysis

In [19] Shu et al. presented a detailed analysis of their algorithm. From this analysis and our own experiments (see section 4) two conclusions can be drawn:

1. The presented algorithm works well in the case of Gaussian distributed noise. In the presence of occlusions, the algorithm may fail to extract the correct parameters.

2. The method has non uniform sensitivity to noise, depending on the position of the point. The sensitivity in regions close to the singular point is low, whereas the sensitivity in regions away from the singular points is increased.

2.3 RANSAC based approach

Although the roots of the linear phase portrait estimation algorithm can be tracked back to the year 1990 [18], only recently several authors applied this algorithm in their work. For example in [10], the authors applied this method in order to extract a high level description of fingerprint singularities and direction fields thereof. As mentioned above, there are conceptual weaknesses adhered to this algorithm. In order to improve the robustness of the original algorithm we propose a Random Sample Consensus (RANSAC) [2] based approach for parameter fitting. The RANSAC algorithm is an iterative method to estimate parameters of a mathematical model from a set of observed data which contains outliers. Random sampling and consensus has been applied to a wide range of problems including fundamental matrix estimation ,trifocal tensor estimation , camera pose estimation and structure from motion [4]. This algorithm has proven to give better performance than various other robust estimators.

The application of this robust algorithm to fitting parameters of linear phase portraits can be described as in the following:

1. Randomly select 6 triplet data points (x, y, ζ) from the oriented texture and compute the model parameters using the linear least squares algorithm (as described in subsection 2.1).
2. Verify the computed model by using a voting procedure. Every pixel lying within a user given threshold t , increases the vote.
3. If the vote is high enough, accept fit and exit with success.
4. Repeat 1-3 for n times

The number of iterations n can be computed using the following formula [2]:

$$n = \frac{\log(1 - z)}{\log[1 - (1 - \epsilon)^m]} \quad (13)$$

Where z is the confidence level, m is the number of parameters to be estimated and ϵ is the outlier proportion.

3 Application to Fingerprint Singularities

Most approaches classify singular points into two types, namely in cores and deltas. The approach presented in this paper distinguishes between three types of singular points: whorls, deltas and loops.

Describing whorl type singularities using linear phase portraits is straightforward. Delta and loop type singularities need a special treatment in order to be modelled using linear phase portraits. Our approach for modelling deltas is to double the orientation and then to fit a saddle type pattern to it. Loops are modelled in two parts. The upper part of a loop is modelled as a whorl. The second part of the loop consists of a homogeneous region (see figure 2(c) for an example). In order to compute the number of inliers for this region we compute the median of the doubled angle orientation. Every orientation which lies in a user given threshold t is counted as inlier.

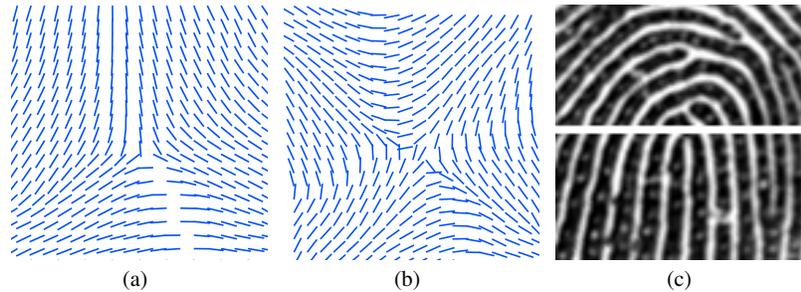


Fig. 2 The orientation fields of a delta (a). The doubled angle orientation fields can be seen in (b). The delta is modelled in the doubled angle orientation field as a saddle type singularity. Loops (c) are modelled in two parts, first the half whorl type and second a homogeneous region.

3.1 Singularity Detection using sliding Window approach

By using linear phase portraits, it is possible to model the local area around a singular point. Furthermore linear phase portraits can only describe one singular point, while a complete fingerprint impression contain a least two singular points. In order to detect multiple singular points in fingerprint images, we approach the use of a sliding window approach. The orientation in the window area is locally approximated by linear phase portraits. The size of the sliding window is depending on the sensors resolution. Values for the FVC2000 database 1 and FVC2004 database 2 are 50x50 pixels for deltas, 80x80 pixels for whorls and 100x70 pixel for loop type singularities. For the sliding step we found values of 1/3 of the window size to be sufficient to detect every singular point. Too fine steps slow down the search process. At each position of the sliding window, parameters are fitted by the RANSAC algorithm as described in section 2.3. A detection of a singular point is accepted only if the number of inliers exceeds 65%. This value is low enough in order to allow a certain number of outliers, which usually are present in noisy fingerprint images.

On the other hand it is high enough to prevent random parameters to be fitted to a region and therefore causing spurious detections.



Fig. 3 The search procedure: A window of fixed size slides over the whole image in order to search for singularities. At each position of the window, parameter of a linear phase portrait are fitted. Based on these parameters it is decided if a singularity is present. Furthermore is possible to classify the different types of singular points.

Once parameters for a given sub window have been found, these parameters must be inspected. This inspection is done using the eigenvalues of the characteristic matrix \mathbf{A} . In general, the ratio of the two eigenvalues $\frac{\lambda_1}{\lambda_2}$ is expressing the aspect ratio of an oriented pattern around a given singularity. In order to prevent physically impossible parameters to be fitted, we introduce a threshold for this ratio. In Table 1 an overview of the classification and the thresholds is given.

Also note that because of the sliding window approach in certain cases a singularity may be detected multiple times. For combining these multiple detections to one single detection, we use the mean shift algorithm as described in [1]. The mean shift algorithm is a non-parametric technique to locate density extrema or modes of a given distribution by an iterative procedure.

4 Experimental Results

4.1 Parameter Fitting Examples

In the following subsection, we give an illustrative overview of the parameter fitting capability of our algorithm. Therefore, we use synthetic orientation fields (Figure 4) and patches taken from real fingerprint images (Figure 5).

The illustration in Figure 4 shows that our robust model based algorithm can extract the correct parameters even in the case of noisy data. In Figure 5 we want

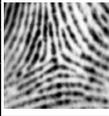
Appearance	Eigenvalues	Thresholds
Whorl 	complex eigenvalues $\lambda_1 = \Re + j\Im$ $\lambda_2 = \Re - j\Im$	$\frac{1}{3} < \frac{\lambda_1}{\lambda_2} < 3$
Delta 	real distinct eigenvalues λ_1 and λ_2 with opposite sign	$\frac{1}{4} < \frac{\lambda_1}{\lambda_2} < 4$
Loop 	upper part only: complex eigenvalues $\lambda_1 = \Re + j\Im$ $\lambda_2 = \Re - j\Im$	$\frac{1}{3} < \frac{\lambda_1}{\lambda_2} < 3$ $\Re < 0.2$

Table 1 Classification-Schema for Fingerprint Singularities: The classification is done based on the extracted parameters of the linear phase portraits. Whorls are detected in the original orientation field. Detection of deltas is done in the doubled angle orientation field. In case of the loop, first a half center is detected, followed by the detection of a homogeneous region. Thresholds are introduced in order to prevent the fitting of physically impossible parameters.

to emphasize on the uniform sensitivity of our algorithm. As the aim is to extract the singular point position as accurate as possible, this property of the algorithm is essential.

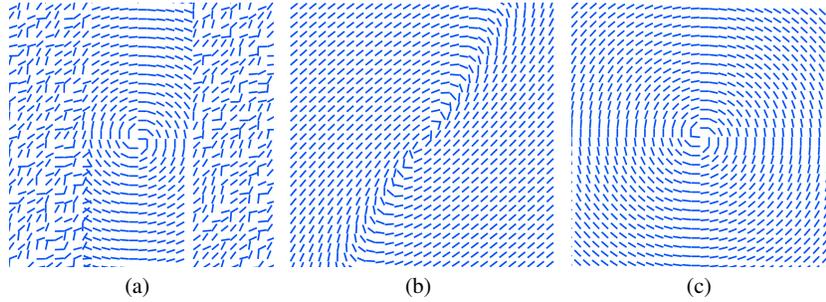


Fig. 4 Occlusion: in this example we illustrate the robustness of the two algorithms against occlusions. First an orientation field of center type has been created. On both sides of the orientation field an occlusion has been simulated by replacing the original values by random values (a). While the original algorithm of Shu et al. [18] fails (b), our new approach (c) extracts the parameters precisely.

The estimation of the orientation field is accomplished by a gradient based algorithm as described in [17].

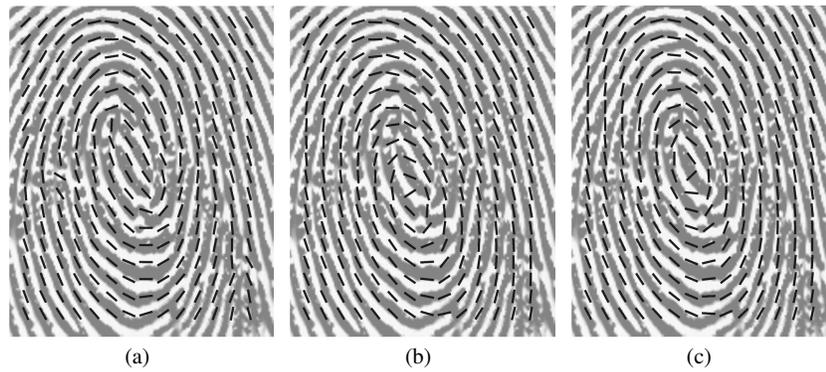


Fig. 5 Uniform sensitivity: In (a) the orientation has been extracted using a gradient based method. In figure (b) one can see the fitting result of the linear least squares algorithm as proposed by Shu et al. and applied in many recent papers. Figure (c) shows the fitting results of our proposed method. The uniform sensitivity property of our algorithm is essential for the precise localisation of singular points.

4.2 Comparison with Shu et al.

In order to compare the robustness of our method with the algorithm of Shu et al. [18], we have extracted 100 image patches from the FVC2004 database 2a [13]. These patches have a size of 150x150 pixels (as shown in Figure 6, with the singular point centred) and represent whorl type singularities. All singular point positions have been labelled manually. While the proposed algorithm classified every patch correctly, the original algorithm of Shu et al. failed in one case.

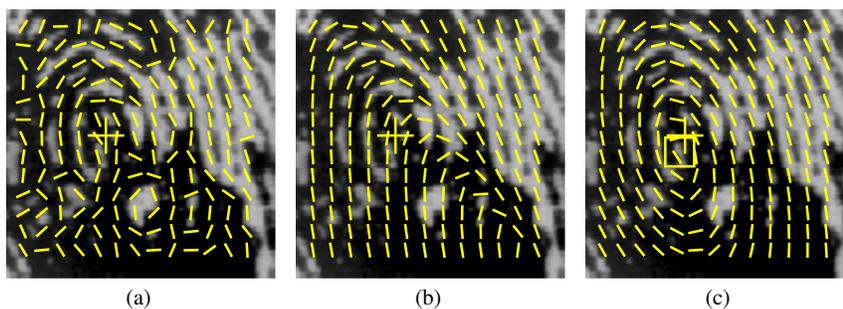


Fig. 6 A comparison of 100 images patches using our robust approach and the original algorithm of Shu et al. This experiment shows that our algorithm is not only more robust, but also more precise in locating the correct position of the singular point (manually labelled position shown as '+' and detected as '□'). In 6(c) the orientation field has been extracted using a gradient based method. Figure 6(b) shows the fitting results of the linear least square algorithm, while in 6(a) the results of our robust approach are shown.

The average distance between labelled and detected singular points is 11.3 pixels for our algorithm and 19.2 pixels for the method of Shu et al. The standard deviation for our algorithm is 4.2 pixels while being 5.6 pixels for the algorithm of Shu et al. From this experiment, it can clearly be seen that the algorithm as proposed by Shu et al. has a major drawback concerning precision. This is due to the lower sensitivity of this algorithm around the singular point and a higher sensitivity away from the singular point.

4.3 Singular Point Detection in Fingerprint Images

In this subsection we present results of the detection and recognition algorithm. For performance evaluation we use the first 280 images from the FVC2000 database 1a [12] and the FVC2004 database 2a [13]. The true positions of the singular points have been annotated manually. Using this ground truth data, the detections of the algorithm can be categorized into three types. These types are, true positive (TP), false positive (FP) and false negative (FN). The maximum distance for classifying a detection as TP was set to a value of 25 pixels. Detections which are closer than 10 pixels to the segmentation-border, are discarded. The iterations for the RANSAC procedure are 150 and the threshold for counting an orientation as inlier is 8 degrees. From the above manual classification, we can compute the following figures:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad (14)$$

For the FVC2000 database 1a, we achieve a precision of 99.4% and a recall figure of 96.0%, respectively. See Table 2 for more details on the different type of singularities. For the FVC2004 database 2a the algorithm achieves a precision rate of 95.4% and a recall rate of 96.4%. More details are given in Table 3.

type	true positive	false positive	false negative	precision	recall
delta	94	1	7	98.9%	93.0%
whorl	78	0	1	100%	98.7%
loop	193	1	7	99.5%	96.5%
total	365	2	15	99.4%	96.0%

Table 2 Results on the FVC2000 database 1a using the first 280 images (fingers 1 to 35, eight impressions per finger).

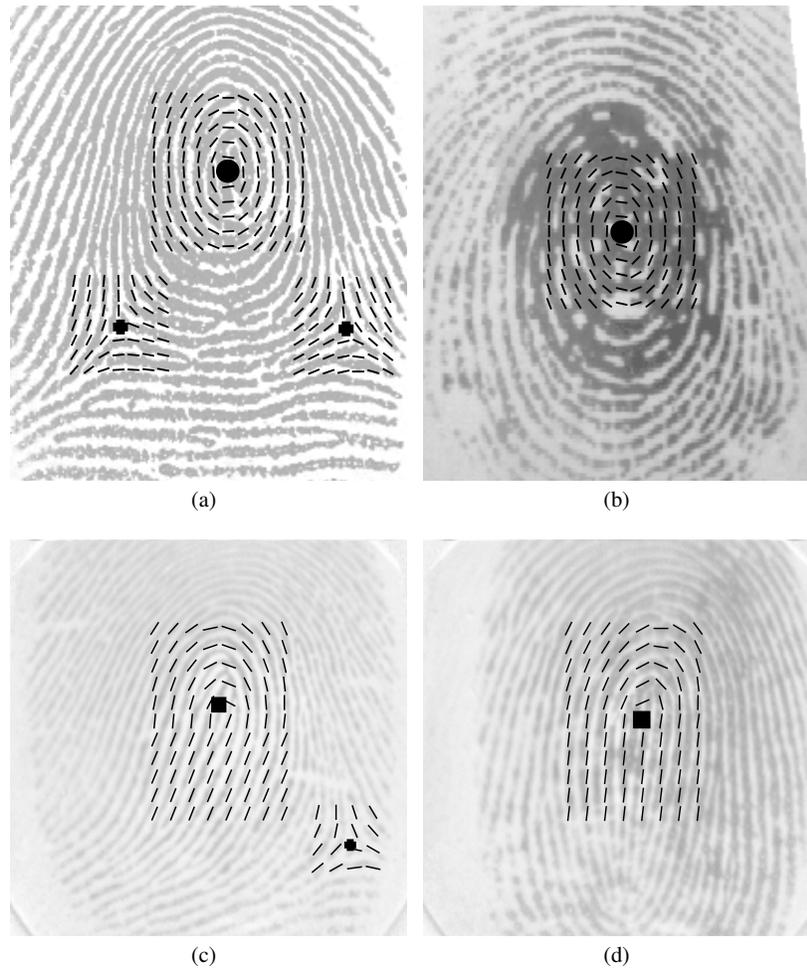


Fig. 7 This figure shows the detected singular points with the corresponding sliding windows. The orientation has been reconstructed using linear phase portraits. In Figure 7(a) a whorl type fingerprint with the detected whorl and delta type singularity is shown. A noisy whorl type can be seen in 7(b), while in 7(c) it is shown how a the sliding windows fits a loop type singularity. In Figure 7(d) a wrong detection is shown. The displayed fingerprint is of whorl type, while the algorithm detected a loop type singularity.

type	true positive	false positive	false negative	precision	recall
delta	89	1	6	98.8%	93.6%
whorl	55	7	1	88.7%	98.2%
loop	233	10	7	98.8%	93.7%
total	377	18	14	95.4%	96.4%

Table 3 Results on the FVC2004 database 2a using the first 280 images (fingers 1 to 35, eight impressions per finger)

5 Conclusion

We presented a model based method for singular point detection in fingerprint images. Our proposed method is robust to noise and occlusions in the input image. The algorithm proceeds by fitting linear phase portraits at each location of a sliding window and then analyses its parameters. Using a well established mathematical background, our algorithm is able to decide if a singular point is existent. Furthermore, the parameters can be used to classify the type of the singular point into whorls, deltas and loops.

We performed several tests on synthetic and natural images in order to point out the mentioned capabilities of our algorithm. We also showed how the algorithm is able to reconstruct the orientations near singular points. The final evaluation of our algorithm is done on a dataset of 2x280 images from publicly available fingerprint images. This evaluation attests our algorithm an excellent singular point detection capability.

Future work includes merging our model based approach with existing numerical methods.

Acknowledgments.

This work has been funded by the Biometrics Center of Siemens IT Solutions and Services, Siemens Austria.

References

1. D. Comaniciu and P. Meer. Mean shift analysis and applications. volume 2, pages 1197–1203 vol.2, 1999.
2. Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.
3. R. M. Ford and R. N. Strickland. Nonlinear phase portrait models for oriented textures. *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*, pages 644–645, 1993.
4. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.
5. A. K. Jain, Lin Hong, S. Pankanti, and R. Bolle. An identity-authentication system using fingerprints. *Proceedings of the IEEE*, 85(9):1365–1388, 1997.
6. Anil K. Jain and Kalle Karu. Learning texture discrimination masks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(2):195–205, 1996.
7. Anil K. Jain and David Maltoni. *Handbook of Fingerprint Recognition*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.
8. Xudong Jiang, M. Liu, and A. C. Kot. Reference point detection for fingerprint recognition. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 1:540–543 Vol.1, 2004.
9. Erwin Kreyszig. *Advanced Engineering Mathematics*. Wiley, 6th edition, 1988.

10. Jun Li, Wei-Yun Yau, and Han Wang. Constrained nonlinear models of fingerprint orientations with prediction. *Pattern Recognition*, 39(1):102–114, January 2006.
11. D. Maio and D. Maltoni. A structural approach to fingerprint classification. *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, 3:578–585 vol.3, 1996.
12. D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain. Fvc2002: Second fingerprint verification competition. *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, 3:811–814 vol.3, 2002.
13. Dario Maio, Davide Maltoni, Raffaele Cappelli, James L. Wayman, and Anil K. Jain. Fvc2004: Third fingerprint verification competition. In David Zhang and Anil K. Jain, editors, *ICBA*, volume 3072 of *Lecture Notes in Computer Science*, pages 1–7. Springer, 2004.
14. Kenneth Nilsson and Josef Bigun. Localization of corresponding points in fingerprints by complex filtering. *Pattern Recogn. Lett.*, 24(13):2135–2144, September 2003.
15. A. R. Rao and R. Jain. Analyzing oriented textures through phase portraits. *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, i:336–340 vol.1, 1990.
16. A. R. Rao and R. C. Jain. Computerized flow field analysis: oriented texture fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(7):693–709, 1992.
17. A. R. Rao and B. G. Schunck. Computing oriented texture fields. *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR '89., IEEE Computer Society Conference on*, pages 61–68, 1989.
18. C. F. Shu, R. Jain, and F. Quek. A linear algorithm for computing the phase portraits of oriented textures. *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on*, pages 352–357, 1991.
19. Chiao-Fe Shu and Ramesh C. Jain. Direct estimation and error analysis for oriented patterns. *CVGIP: Image Underst.*, 58(3):383–398, November 1993.
20. Shen Wei, Chen Xia, and Jun Shen. Robust detection of singular points for fingerprint recognition. *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*, 2:439–442 vol.2, 2003.
21. Wei-Yun Yau, Jun Li, and Han Wang. Nonlinear phase portrait modeling of fingerprint orientation. *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, 2:1262–1267 Vol. 2, 2004.