A Singular Point Detection Algorithm Based on the Transition Line of the Fingerprint Orientation Image

Mmamolatelo E. Mathekga, Ishmael S. Msiza

Information Security Competence: Biometrics Research Group
CSIR Modelling and Digital Science Unit, Pretoria
{dmathekga, imsiza}@csir.co.za

Abstract

A new algorithm for identifying and locating singular points on a fingerprint image is presented. This algorithm is based on properties of the fingerprint orientation image, including a feature defined as a transition line. The transition line is shown to provide an efficient means of both tracking and classifying singular points without having to search through the entire fingerprint image. The algorithm is shown to be precise and its performance is evaluated using three measures, namely, the detection accuracy (DA), the detection time (DT) and a Human-versus-Algorithm (HvA) score. Using a dataset from the Fingerprint Verification Competition 2002 (FVC 2002), the performance of the algorithm is: $DA = 89.6$ %, $DT = 0.48$ s and $HvA = 0.92$.

1. Introduction

A fingerprint is a pattern formed by the ridges and furrows on a fingertip when impressed against a smooth surface. The finger ridges are the dark lines and the furrows are the lighter lines, as depicted in figure 1. This pattern has numerous features that serve a useful purpose in fingerprint identification and/or verification. These fingerprint features can be ordered into two types, namely, the minutiae and the singular points. With reference to figure 1, the minutiae are marked by the two clear circular structures, while the singular points are marked by the filled circular and triangular structures.

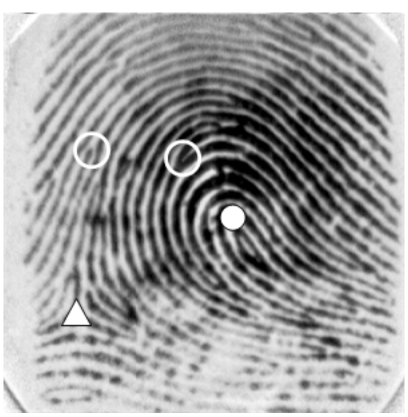

Figure 1: *Fingerprint with minutiae and singular points marked. The minutiae are marked by clear circles and the singular points are marked by the filled triangle and circle*

The minutiae occur in the form of a ridge ending and a ridge bifurcation. A ridge ending is the point where a ridge ends (or starts), while a ridge bifurcation is the point where a single ridge splits into two ridges. Singular points occur in the form of a core and a delta. A core can be defined as the turning point of the innermost loop of a fingerprint and a delta is the point where the fingerprint ridges tend to triangulate. The primary objective of this document is to present and evaluate the performance of an algorithm that can be used to detect singular points in a fingerprint image.

2. Singular Point Detection

Singular point detection can be defined as the process of identifying and locating singular points.

2.1. Singular Point Detection Methods

Even though they are all based on the computation of the fingerprint orientation image, singular point detection methods can be ordered into a set of three groups. These can be summarized as follows [1]:

- Poincaré index methods
- Methods based on the local characteristics of the orientation image
- Partition-based methods

The first group of methods classify a point as a core or delta by computing the Poincaré index along a small closed path around the point [2]. Angle differences are then summed along this closed path in order to classify the singular points as either cores or deltas. For an angle difference of 180^0, the singular point is classified as a core, while a delta is marked by an angle difference of -180^0. Works that employ the poincaré index include those of Karu and Jain [2] and Kawagoe and Tojo [3]. Bazen and Gerez [4] present an interesting version of a poincaré index method informed by Green's theorem. Rämö *et al* [5] pointed out that calculating the poincaré index is a time consuming exercise.

The second group of methods are based on the local characteristics of the fingerprint orientation image. A fingerprint orientation image is a matrix whose entries depict the local orientation of the fingerprint ridges. This orientation is measured in degrees anti-clockwise, between 0^0 and 180^0, relative to some horizontal axis. The local characteristics of the orientation image include high irregularities [1]. The irregularity operator, introduced by Cappelli *et al* [6], is used to determine the irregularity map of the fingerprint image. This map is largely white in color, and the singular points are clearly marked by black regions.

The third group of methods divide the computed pixel-wise orientation image into non-overlapping blocks and assign an average orientation to each of these blocks. The reason for this is that, the ridge orientation cannot be represented pixel-wise since the ridge is not represented by a single pixel but rather by a collection of pixels. This therefore implies that a block-wise image better represents the ridge orientation. Examples of works that used this method include Maio and Maltoni [7], Cappelli *et al* [6] and Rämö *et al* [5].

2.2. Applications of Singular Points

The two main applications of singular points are fingerprint classification and fingerprint image alignment. In fingerprint classification problems, the locations of singular points are used as part of schemes that determine the class of a fingerprint. Fingerprint classification is made possible by the sketchy nature of singular points. Examples of works that use singular points to classify fingerprints include Karu and Jain [2], Zhang and Yan [8] and Msiza *et al* [9].

During image capturing, some participants may impress their fingertip in such a way that the fingerprint is rotated, hence not in line with the image frame. The performance of some fingerprint matching algorithms may be negatively affected by images that are rotated by over 20^0 out of line with the image frame [1]. This therefore implies that image alignment can be an important preliminary step, just before fingerprint matching. Wegstein [10] executes fingerprint image alignment by making use of the core-delta segment (if the delta is present) and the average orientation of some ridges lying in the neighborhood of the core.

3. Algorithm Description

The properties of an orientation image of a fingerprint are listed below and may be verified on the fingerprints in figures 2 and 3:

- With the ridge orientation depicted as **Q** in figure 2, the range of orientation values is [0^0; 180^0]. An orientation value is either an acute or an obtuse angle.
- The difference between the obtuse and acute angled orientation values is referred to as Δ**Q**.
- A path is formed at the point where the orientation values change from acute-to-obtuse or obtuse-to-acute. This path is referred to as a *transition line*.
- The transition line is two columns wide, and the two columns in each row that form part of this line are referred to as *transition columns*.
- The transition line leads to a singular point, normally sitting at the end or beginning of the line, depending on the point of reference.
- A core singular point is either facing downwards or upwards. Downward and upward facing cores are singular points of the type shown in figure 2, top and bottom respectively.
- A transition line that leads to a delta (delta transition line) is different from the one that leads to a core (core transition line).
- For a core singular point, Δ**Q** of the transition columns is initially large and gradually decreases to a value close to zero.
- This applies to transition lines that lead to both the upward facing and the downward facing core, with the transition lines assumed to start from a smaller to a larger and larger to a smaller row value respectively.
- For a delta singular point, Δ**Q** is initially small and gradually increases to a large value, assuming that the line starts from a smaller to a larger row value.
- However, if the delta transition line is assumed to start from a large to a small row value then its properties resemble those of the core transition line.

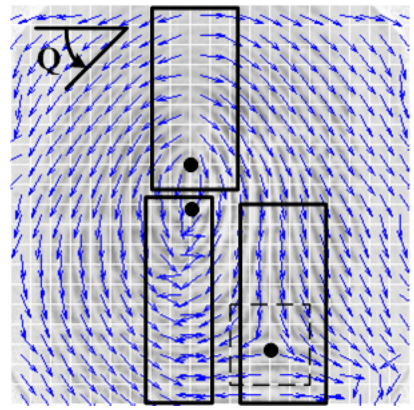

Figure 2: *Orientation image of block size 15 overlaid on the fingerprint with singular points marked and transition lines enclosed within rectangular blocks*

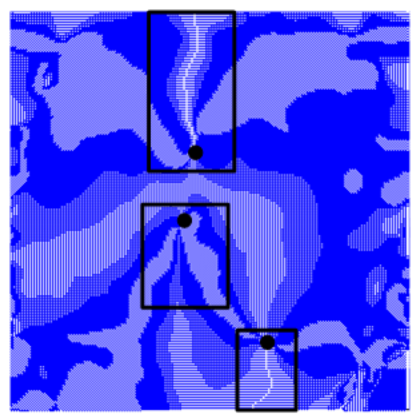

Figure 3: *Orientation image of block size 2 overlaid on the fingerprint with singular points marked and transition lines enclosed within rectangular blocks*

The location and type of singular point on a fingerprint is determined using transition lines. This is done by locating the start of a transition line and moving along the transition line across the orientation image until the singular point is reached. The type of singular point on the transition line is identified using the properties of the transition line, taking the start and end rows of the transition line into consideration.

It is evident from figures 2 and 3 that the location of singular points can be determined more precisely on a fine than a coarse orientation image - fine and coarse orientation images are orientation images computed with small and large block sizes values respectively. However, the fine orientation image is larger than the coarse orientation image and requires more processing time

A two stage tracking procedure is adopted to reduce the processing time and to ensure that the location of singular points can be determined precisely. The location of the singular point is first determined from a coarse orientation image (C) and then a region of the fingerprint image that is covered by blocks of the orientation image around the location of the singular point in the coarse orientation image is extracted. After the extraction, a fine orientation image (F) of the region is computed and used to determine a more precise location of the singular point. An example region that would be extracted in the coarse orientation image is shown in figure 2 enclosed by a dotted line. An implementation of the proposed algorithm is summarized in algorithm 1. The inputs f and O are the pixel block size used to compute F and the pixel wise orientation image, respectively. The variable I is the input orientation image, n is the number of runs and Bgn is the column where the search for the start of a transition line begins.

Algorithm 1: Singular point detection algorithm

Input : O, C and f
Output : Singular Points - **Type(s)** & **Location(s)**
Variables: F, I, n and Bgn

1 initialization: $n = 1$;
2 **while** $n <= 2$ **do**
3 $I = C$;
4 **if** $n = 1$ **then**
5 Bgn = top left hand corner of C;
6 **else**
7 Bgn = bottom left hand corner of C;
8 **end**
9 **while** *the right hand corner has not been reached* **do**
10 seek the start of the transition line;
11 **end**
12 **if** *no transition line is found* **then**
13 go to step 38;
14 **end**
15 **if** $I \neq C$ **then**
16 go to step 20;
17 **end**
18 record the starting point of the transition line;
19 determine the singular point **Type**;
20 **repeat**
21 motion along the transition line;
22 **until** *a termination condition is met* ;
23 **if** *none of the termination conditions are met* **then**
24 go to step 36;
25 **end**
26 record, on O, the point where 20 terminated;
27 **if** $I \neq C$ **then**
28 go to 35;
29 **end**
30 extract a small region of O around the point recorded in step 26;
31 compute F as specified by f;
32 $I = F$;
33 $Bgn =$ top or bottom left corner of I;
34 go to step 9;
35 compute and record the singular point **Location**;
36 $Bgn =$ point recorded in step 26;
37 go to step 9;
38 increment n;
39 **end**
40 terminate;

3.1. Start of the Transition Line

The transition line divides the orientation image into a region of obtuse and acute angled orientation values except where there is more than one transition line and there are parts where the transition lines exist over similar rows. In that case, the region across the rows has orientation values grouped into regions of acute-obtuse-acute or obtuse-acute-obtuse. The start of the transition line is, therefore, located by finding the first transition column.

Only transition columns with an acute angled orientation value on the left are detected as valid transition points when searching the orientation image from the top to the bottom row. This is to ensure that the upward and downward facing cores are identified and located using similar routines and that the transition line properties, which lead to the different singular points, may be used to differentiate them. The opposite applies when searching the orientation image from the bottom to the top row. Therefore, only downward facing core and delta singular points are identified when searching the orientation image from top to bottom and only upward facing core singular points will be identified when searching the orientation image from the bottom to the top.

The above applies when locating the start of a transition line in the coarse orientation image. It is more efficient to locate the delta in the fine orientation image by searching the image from the bottom to the top. This is because a transition column always exists on the row at the bottom of the fine orientation image and there will usually not be any transition columns in the first few rows on the top of the orientation image. Thus, the transition line for the fine orientation image of the extracted region that contains a delta singularity is assumed to start from the bottom to the top row of the orientation image and an exception has to be made as the transition columns will have acute angled orientation values on the left.

When at least one transition line has been found on the coarse orientation image, the search for subsequent transition lines must begin from the starting point of the previous transition line. This is done in order to ensure that no part of the previous transition line is identified as the starting point of a new transition line, as parts of a transition line may exist over similar rows. This may be done by recording the range of columns and rows over which the previous transition line exists and using this in subsequent searches.

3.2. Identifying Singular Points

The type of singular point that a transition line leads to, is identified by using the $\Delta\mathbf{Q}$ of the transition columns at the start of the transition line. This $\Delta\mathbf{Q}$ is smaller for a delta transition line as compared to the core transition line. This is because a delta transition line develops from parallel ridges. The decision is therefore made by comparing $\Delta\mathbf{Q}$ with a threshold value (T_1). Values greater than T_1 indicate that the transition line (if not caused by noise on the image) leads to a core, while for values less than T_1 it leads to a delta. A value of $T_1 = 35^0$ is used in the implementation of the algorithm.

3.3. Moving Through the Transition Line

Subsequent transition columns of the transition line are below (or above) each other; this could be directly below (or above) or shifted to either the left or right. For simplicity, only transition columns that are shifted by one column to the left and right are considered as being part of the transition line. Therefore, subsequent transition columns are found by searching amongst the four columns in the next row; these are the two columns directly below (or above) the current transition columns and the ones to the left and right; see figure 4.

	a	b	
c	d	e	f

Figure 4: *Transition columns (*a *and* b*) and the four columns (*c, d, e, *and* f*) that are examined for transition columns when moving along a transition line*

If the subsequent column has been shifted by more than one column to either left (or right) or a valid transition column does not exist, then it is assumed that the transition line is caused by noise on the fingerprint image.

3.4. Core Termination Condition

The core termination condition is when:

- The $\triangle\mathbf{Q}$ of one of the transition columns of the transition line is smaller than a threshold value, T_2, which was set to 25^0 for the implementation of the algorithm
- When the four columns discussed in section 3.3 all have acute or obtuse angled orientation values

The latter condition is used as a termination condition because ridges assume a similar direction after the location of the core, unless there is another core immediately afterwards, as in figures 1 and 2. Thus, the transition columns where this condition is satisfied is set as the location of the core singular point.

a and *c* or *b* and *f* in figure 4 must be comparable as ridges are expected to have almost similar orientation values after the core singular point. This is to ensure that the transition line is not caused by noise as discussed in section 3.3 and is performed by thresholding the proportion of the difference between either *a* and *c* or *b* and *f* with respect to either *a* or *b* respectively as shown in equation (1). A threshold, $T_3 = 0.6$, is used and values of the proportion difference below T_3 indicate that the transition columns where the condition is satisfied are the location of the singular point.

$$\text{Proportion Difference} = \frac{a - c}{a} \tag{1}$$

3.5. Delta Termination Condition

The delta singular point termination condition is when the $\triangle\mathbf{Q}$ of one of the transition columns of the transition line is greater than threshold value, T_4, which is set as 90^0 for the implementation of the algorithm.

3.6. Extracted Region

The region that is extracted from the fingerprint image, after locating the singular point on the coarse orientation image, is selected such that it includes the singular point. This is ensured by selecting the region such that the transition columns that are selected as the location of the singular point are in the middle of the image. The region must not be large for performance reasons.

4. Pre-processing and Post-processing

Pre-processing is performed by filtering the fingerprint image to enhance the ridges and remove the noise on the image. The filtering is performed using the technique that is proposed by Hong *et al* [11]. This technique uses the coarse orientation image (C) and ridge frequency matrix (R) for the filtering process. R is the frequency of the ridges in each of the blocks used for computing the coarse orientation image.

Post-processing is performed to remove singular points that are detected on the region of the image without ridges such as the circled point in figure 5. This is done by dividing the image into a region with and without ridges. A mask matrix (M), which is a matrix the same size as the image and whose entries are either one or zero; where one indicates that the pixel of the image is part of the area with ridges and zero that the pixel is part of the region without ridges, is computed using a procedure proposed by Thai in [12].

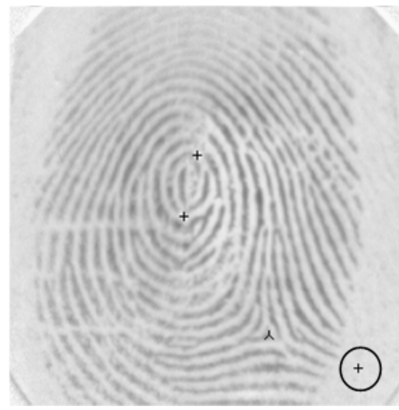

Figure 5: *Fingerprint image processed with the algorithm with the location of the singular points marked*

5. Performance Evaluation

The precision of the algorithm in locating singular points on fingerprints is shown in figure 5.

The following measures are employed to asses the performance of the algorithm:

- Detection Time (DT)
- Detection Accuracy (DA)
- Human-versus-Algorithm (HvA) Score

5.1. Detection Time (DT) and Detection Accuracy (DA)

DT is the time it takes for the algorithm to identify and locate singular points. It gives an indication of the speed with which the proposed algorithm executes its intended task. It is defined as:

$$DT = \left[\sum_{k=0}^{N-1} I_k\right] \div N, \tag{2}$$

where I_k is the $k-th$ instance of data and N is the total number of data instances in the test set.

DA is a measure of the accuracy with which the algorithm locates singular points on a fingerprint image. The algorithm is regarded as having accurately detected a singular point if the detected point is within a distance of 20 pixels when compared to the one manually identified by a human. DA is defined as:

$$DA = \frac{Cnt(\forall\, D \leq 20\, pixels)}{N} \times 100\%, \tag{3}$$

where D is the euclidean distance between the singular point detected by an expert and the one detected by the proposed algorithm. The function Cnt is the counting operation.

5.2. Human-versus-Algorithm (HvA) Scoring

This assessment provides the comparison of the singular points that are detected by the algorithm and those identified by a human being. The assessment provides a detailed picture with regard to the overall behavior of the algorithm. The information that can be extracted from this assessment includes, the number of singular points: accurately detected, inaccurately detected, missed, and falsely detected by the algorithm.

HvA is mathematically modeled as:

$$HvA = \frac{AI - FA - IL}{HI}, \tag{4}$$

where AI is the total number of singular points detected by the algorithm, FA is the number of false singular points detected by the algorithm, IL is the number of singular points detected at incorrect locations and HI is the total number of singular points detected by a human. HvAs for core and delta singular points are denoted by $C - HvA$ and $D - HvA$ respectively, and the variables in equation 4 will assume appropriate meanings. The algorithm qualifies to be labeled as accurate if the HvA score approaches unity.

6. Results and Interpretation

The algorithm is tested by making use of fingerprint images from Database Db 1_a of the year 2002 version of the Fingerprint Verification Competition (FVC2002). A total of 439 fingerprint images where used for each of the performance measures.

6.1. DT and DA Results

Because the proposed algorithm is dependent on C and R, it becomes necessary to execute a parameter study of the two pixel-block-sizes. This implies that these pixel-block-sizes have to be varied in order to establish their effect on the algorithm's DT and DA.

The process of varying these pixel-block-sizes bring into effect a total of six scenarios, with different combinations of these pixel-block-sizes. The results of this exercise are summarized in table 1. The variable $C - pbs$ is the pixel-block-size of C and, similarly, the variable $R - pbs$ is the pixel-block-size of R. From table 1 it is apparent that the pixel-block-size combination that gives the most accurate detection is that of scenario D, while scenario E gives the fastest detection. This therefore implies that, in applications where accuracy is more important than speed, scenario D is recommended. Similarly, in applications that require more speed than accuracy, scenario E is recommended.

Table 1: DT and DA Results with different combinations of the pixel block sizes

Scenario	$C - pbs$	$R - pbs$	DT	DA
A	10	36	0.48 s	88.4 %
B	10	23	0.49 s	86.8 %
C	10	10	0.49 s	86.6 %
D	08	36	0.48 s	89.6 %
E	06	36	0.46 s	83.8 %

6.2. HvA Assessment Results

The HvA assessment is executed with the assumption that accuracy is more important than speed, implying that the scenario D parameters are used. The results from the HvA assessment are summarized in table 2, and they give details about the overall behavior of the proposed algorithm. There is a subtle difference between a singular point detected at an incorrect location (misplaced) and a false singular point. A misplaced singular point is one that is located somewhere along a transition line, but not at the end. A false singular point is one that is located nowhere near a transition line. An example of such a singular point is one located somewhere in the background of the fingerprint image, that is, outside the fingerprint ridge area.

Table 2: HvA assessment results

Assessment Variable	**Value**
Human-Identified Cores (HIC)	517
Algorithm-Identified Cores (AIC)	510
Cores Missed by the Algorithm (CMA)	16
False Cores Detected by the Algorithm (FCA)	07
Cores Detected at Incorrect Location (CIL)	02
Human-Identified Deltas (HID)	227
Algorithm-Identified Deltas (AID)	197
Deltas Missed by the Algorithm (DMA)	30
False Deltas Detected by the Algorithm (FDA)	00
Deltas Detected at Incorrect Location (DIL)	00
$C - HvA$ Score	0.97
$D - HvA$ Score	0.87
Overall HvA Score	0.92

The anomalies associated with detecting the core singularity are: (i) 3.1 % of the cores are missed by the algorithm, (ii) False cores are detected by the algorithm (small number) and (iii) 0.4 % of the cores are detected at incorrect locations.

The only anomaly associated with detecting the delta singularity is that 13.2 % of the deltas are missed by the algorithm. This is the reason why the $D - HvA$ Score is slightly less than the $C - HvA$ score. The overall HvA score of the proposed tool shows that it qualifies to be labeled as accurate, and hence it is readily usable in any other modules of a biometric implementation.

7. Usability of the Algorithm

The usability of the proposed singular point detection algorithm is, beyond doubt, demonstrated by the fact that Msiza *et al* [9] successfully used it in their fingerprint classification module. The output of this algorithm served as input to the classification module which, in turn, provided the class of the fingerprint image in question. The five fingerprint classes that they considered in their study are: Central Twins (CT), Tented Arch (TA), Left Loop, Right Loop (RL) and Plain Arch (PA). The accuracy of the classification module is another way of evaluating the performance of the singular point detection algorithm.

In order to determine the accuracy of their classification module, Msiza *et al* used Database Db 1_a of the 2002 version of the Fingerprint Verification Competition (FVC2002) as the test data. They summarized their evaluation in the form of a confusion matrix, which is a widely used method for determining the accuracy of any classifier. A total of 431 data instances was presented to this classification module and, looking at table 3, a total of 360 fingerprints were correctly classified, 70 were mis-classified and only 1 was classified as unknown (UKN). The results in table 3 give a classification accuracy of 83.5%, where this accuracy is the ratio of the sum of the main diagonal over the number of data instances in the data set, expressed as a percentage.

Table 3: Five-Class experimental results tested on FVC2002 Db1_a [9]

Actual	As						
	CT	LL	PA	RL	TA	UKN	TOT
CT	**83**	05	00	04	00	01	93
LL	00	**108**	02	15	00	00	125
PA	00	00	**15**	01	00	00	16
RL	02	20	01	**96**	03	00	122
TA	01	05	04	07	**58**	00	75
			83.5%				431

A study conducted by Yager and Amin [13] revealed that most fingerprint classifiers that deal with a five-class fingerprint problem, have an accuracy that is anything between 80% and 90%. This is, therefore, indicative of the fact that the usability of the singular point detection algorithm proposed in this document is not questionable.

8. Summary and Conclusions

An algorithm for detecting singular points on a fingerprint image using the properties of the orientation image was presented. A special property of the orientation image that particularly played a useful role was identified as the transition line. Both the theoretical and algorithmic description of the proposed method clearly revealed its feasibility. A rigorous performance evaluation was executed by making use of 439 instances of data, revealing a high detection accuracy (89.6 %) and execution time (0.48 s). An innovative measure, HvA (0.92), that unpacks the overall behaviour of the algorithm is used and revealed evidence about the robust nature of the proposed algorithm.

These performance measures show that the algorithm is based on a solid theoretical foundation and hence it is readily usable. The usability of this algorithm is demonstrated by the work of other practitioners that successfully used it as a service to their fingerprint classifier.

One possible improvement of the algorithm is the introduction of some image thinning module, in order to simplify the run down the transition line. It is also necessary to execute a parameter study of the thresholds used as the terminating conditions for both the core and the delta. This will serve to determine the effect of different threshold values on the performance of the algorithm.

Acknowledgment: The support of the Department of Science and Technology (DST) is hereby acknowledged.

9. References

[1] Maltoni, D., Maio, D., Jain, A.K., and Prabhakar S., Handbook of Fingerprint Recognition, Springer, New York, 2003.

[2] Karu, K., and Jain, A.K. “Fingerprint Classification”, Patt. Recog., Vol. 29, 1996, p 389–404.

[3] Kawagoe, M., and Tojo, A. “Fingerprint Pattern Classification”, Patt. Recog., Vol. 17, 1984, p 295–303.

[4] Bazen, A.M., and Gerez, S.H. “Systematic Methods for the Computation of the Directional Fields and Singular Points of Fingerprints”, IEEE Trans. on Patt. Anal. and Mach. Intell., Vol. 24, 2002, p 905–919.

[5] Rämö, P., Tico, M., Onnia, V., and Saarinen, J., “Optimized Singular Point Detection Algorithm for Fingerprint Images”, In: Proc. of the Intl. Conf. on Image Process, (Thessaloniki, Greece), p 242–245, 2001.

[6] Cappelli, R., Lumini, A., Maio, D., and Maltoni D. “Fingerprint Classification by Directional Image Partitioning”, IEEE Trans. on Patt. Anal. and Mach. Intell., Vol. 21, 1999, p 402–421.

[7] Maio, D., and Maltoni, D., “A Structural Approach to Fingerprint Classification”, In: Proc. of the 13^{th} Intl. Conf. on Patt. Recog., (Vienna, Austria), p 578–585, 1996.

[8] Zhanga, Q., and Hong, Y. “Fingerprint Classification Based on Extraction and Analysis of Singularities and Pseudo Ridges”, Patt. Recog., Vol. 37, 2004, p 2233–2243.

[9] Msiza, I.S., Leke-Betechuoh, B., Nelwamondo, F.V., and Msimang, N., “A Fingerprint Pattern Classification Approach Based on the Coordinate Geometry of Singularities”, In: Proc. of the IEEE Intl. Conf. on Syst., Man and Cybern., (San Antonio, Texas), p 516–523, 2009.

[10] Wegstein, J.H., “An Automated Fingerprint Identification System”, U.S. Government Publication, U.S. Dept. of Commerce, National Bureau of Standards, 1982.

[11] Hong, L., Wan, Y., and Jain, A.K. “Fingerprint Image Enhancement: Algorithm and Performance Evaluation”, IEEE Trans. on Patt. Anal. and Mach. Intell., Vol. 20, 1998, p 777–789.

[12] Thai, R. “Fingerprint Image Enhancement and Minutiae Extraction”, Honors Project Report, School of Computer Science and Software Engineering, The University of Western Australia, 2003.

[13] Yager, N., and Amin, A. “Fingerprint Classification: A Review”, Patt. Anal. Appl., Vol. 7, 2004, p 77–93.