Chapter 2

Fingerprint Identification — New Directions

Group Members:

Petr Lisonek, *Simon Fraser University*
Akbar Rhemtulla, *University of Alberta*
Rolf Luchsinger, Michael Saliba, Alexandre Zagoskin, *University of British Columbia*
Dan Calistrate, Marc Paulhus, *University of Calgary*

Compiled by:

Michael Saliba
Dept. of Physics and Astronomy
University of British Columbia

25 September, 1997

1 Problem Description

A brief description of this problem as presented by Kinetic Sciences Inc. (KSI) can be found at the beginning of these proceedings. In summary, KSI have developed a small live-scan sensor (imager) that can be mounted onto a small silicon chip. KSI are developing this technology with the objective of producing a fingerprint imager that can be used to limit access to sensitive information or locations. Examples of possible applications of this technology include ATM machines, computer keyboards, and door-unlocking mechanisms.

The KSI sensor is capable of producing an image of the fingerprint that has a resolution of 850 dpi, and that provides high contrast between the ridges and valleys of the fingerprint. A sample

image of part of a fingerprint is shown in Figure 1. The finished product will be able to produce an image of a large portion of the fingerprint (refer to Figure 2). In order for this technology to be successful, KSI requires the use of a computer image-matching algorithm that can make fast and reliable comparisons, in real time, between a scanned fingerprint image and images that are contained in a digital database.

The problem that was submitted to the workshop by KSI involved finding the best model on which to base such an algorithm. The basic questions were:

1. What general features of a fingerprint image should the algorithm search for?
2. What methods should be used to locate these features?

Approach

Conventional fingerprint matching algorithms search for anomalies in the ridge flow (ridge endings and ridge bifurcations) and then, after appropriate post-processing, compare the relative locations of these anomalies to those of the reference fingerprint(s). Our approach during this workshop was to identify new features of the fingerprint that could be used for matching, and to find efficient ways to locate the various features and to store the information.

We identified the following main problems that would need to be addressed by a robust algorithm:

1. Day-to-day changes in the fingerprint (e.g. due to the weather, or due to physical condition).
2. Variations in sensor response (e.g. due to variations in finger pressure applied during scanning).
3. Variations in image location for processing (due to variations in position and/or angular orientation of the finger during scanning).

Furthermore, the following constraints also needed to be addressed:

1. Limits on database memory.
2. Limits on computational time required for matching.

A number of models for suitable algorithms were suggested and discussed at length during the workshop, and the more promising of these are described in the following pages. The various suggestions are presented in the form of a set of individual papers, and a general conclusion is given at the end of this report.

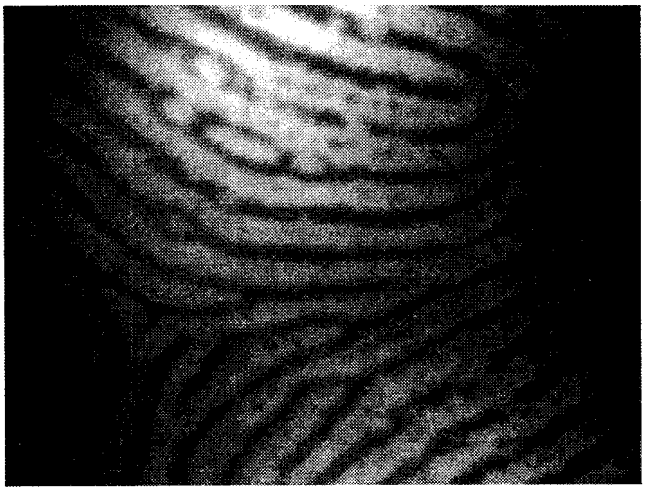

Figure 1: A sample fingerprint image, taken by the KSI live-scan sensor.

Figure 2: Ridge pattern of an inked fingerprint.

Figure 3: Locating the reference point on the image.

2 Finding a Reference Point on the Image

Akbar Rhemtulla[1] and Michael Saliba[2]

Some of the algorithms that are described on the following pages may be sensitive to changes in the positioning of the finger during scanning. Our first suggestion therefore involves an algorithm that could be used to locate a reference point on the image.

Observation of the fingerprint samples that were available during the workshop suggest that many prints have a clearly identifiable centre, or "heart", that is either the centre of a spiral ridge or the tip of an inverted U-shaped ridge. An example of the first case is shown in Figure 3. An algorithm to locate the centre would involve drawing a series of horizontal lines down the image as shown in the figure, and locating the position (end-points) of the innermost curve that crosses the line in each case. This should be done until three or more successive lines have the same innermost curve, or until this curve intersects the line at one point only. In Figure 3 this would occur at line 4. In the case of a spiral ridge, this method could also search for the position where the innermost ridge curves *below* the line instead of above it.

[1] akbar@malindi.math.ualberta.ca
[2] saliba@physics.ubc.ca

When the general location of the heart has been found, a higher resolution search can be carried out between the last three or four lines in order to obtain a more exact location of the reference point.

3 An Algorithm for Comparing Fingerprints

Dan Calistrate[3] and Marc Paulhus[4]

1 Introduction

In this report we shall describe an algorithm which assigns a sequence of integers to a fingerprint image. This sequence can then be used to decide if two fingerprints are identical. The sequence will be topologically invariant and we will describe how scratches on the finger might be ignored by the algorithm.

Section 2 will present the basic idea of the algorithm. Section 3 will address some of the obvious objections or problems that there might be. Potential solutions to these problems will also be given. Section 4 will highlight some of the advantages of using this technique.

2 Basic Idea

Our basic idea is to find a topologically invariant sequence of integers which uniquely describe a fingerprint (or at least uniquely enough). This reduces the problem of comparing two fingerprint images to the problem of comparing two integer sequences. We will call the sequence of integers which come from a fingerprint image the *image sequence*.

We now describe how to produce the image sequence. Given a fingerprint, start at the lefthand-bottom corner of the image. Choose the first valley in the print and follow along this valley using the right-hand rule of navigating[5]. As you walk down this valley, increment a counter every time you come to a fork in the road. This assigns an integer to that particular valley. Do the same for the next valley on your way up the left side of the fingerprint. You will generate an integer sequence, one integer for every valley you walk down. We will assume that we stop walking when we come to a cul–du–sac although this is not necessary.

If a longer sequence is needed you can follow the same procedure using the lefthand-rule or generate the sequence corresponding to the valleys which start on the right side of the image.

You will find an example for an actual fingerprint in Figure 4. In the example there are 28 valleys starting on the left side of the fingerprint. Below we list the integers corresponding to these valleys.

Valley	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Integer	1	2	2	3	2	3	1	4	1	1	2	0	4	2

15	16	17	18	19	20	21	22	23	24	25	26	27	28
3	8	1	1	0	1	2	3	0	0	1	1	0	2

[3] calistra@math.ucalgary.ca

[4] paulhusm@math.ucalgary.ca

[5] The right-hand rule for navigating is a well defined concept which amounts to always following the wall to your right side as you navigate a maze. One can think of putting one's right hand on the wall and never removing it as one walks.

The method described will produce a well defined integer sequence for any particular fingerprint image. Two images can then be compared by comparing their integer sequences. There are some obvious problems which we hope to address in the next section.

3 Potential Problems

- The first objection is that the starting line on the bottom could be different in two scanned images of the same fingerprint. Hence we should not ask for an exact duplication of the image sequence but instead look for one sequence as a subsequence in the other.

- Another objection is that the starting points of the valleys on the left side of the fingerprint could be different for two different images. This could affect the length of the fingerprint sequence as well as the value of the digits in the sequence. We suggest that, as long as the fingerprints are truly identical, the choice of a lefthand side starting point will only change a small number of the digits in the image sequence. Also, a comparison program which is intelligent enough to try a backtracking sequence matching algorithm with deletion, could apply a probability of match even when the sequences differ in length or when the sequences are essentially the same except for extra digits and small deviations. A cumulative sum of differences might also be useful.

  Some work has been done in the field of sequence comparison. For example, biologists who study DNA sequences commonly compute the probability that two integer sequences are the same, modulo errors in observations.

- Being topologically invariant, the sequence would be very sensitive to scratches on the fingerprint.

  If the scratch cuts across the valley you are currently following, then, while walking down this valley, the scratch will appear as an intersection of degree four. In this case ignore the intersection altogether (i.e. do not increment the counter) and continue along the middle (main) valley. Hence scratches of this type will not affect that integer in the sequence. The authors have observed that intersections of degree four rarely occur naturally in a fingerprint (in fact we do not know if they occur naturally at all). In any case, as long as degree four intersections are consistently ignored, the assigned integer will be the same.

  If the scratch runs along the valley you are currently following, there is potential for a large deviation in the assigned integer. This large deviation should only occur in one (or at most a few) digits in the sequence. An intelligent enough sequence-comparison program should be able to deal with this deviation.

- The image sequence should be fairly robust under reasonable rotations of the image. No matter which identification algorithm is used, if the user is instructed to place his finger flush against a wall beside the sensor, many problems may be avoided.

- The computational complexity of "Walking down the valley" is unknown to the authors but it does not seem like an unreasonable task.

- Depending on image quality and resolution, some form of image massage might be required to make the procedure robust. The nature of this massage can only be determined by experimentation.

- If the proper rules of navigating the valley are used, then no infinite looping can occur. Even if a navigation rule was chosen which had the potential for infinite looping, an artificial stopping criteria could be imposed (for example stop walking down a valley when your counter reaches some value).
- Is the image sequence sensitive enough to tell fingerprints apart? Only experimenting with a large database of fingerprints can answer this question. Since the sequence can be made quite long we suspect it is sensitive enough.
- It is possible that, for some fingerprints, there is a nearly vertical valley running up the sides of your image. In this case the image sequence is forced to be quite short. A number of possible solutions can be proposed. Which solution works best would have to be determined experimentally.

  One way is to translate the starting line along the fingerprint until the starting line intersects the maximum number of valleys. Then assign integers to the valleys (using the same set of rules as before) which radiate from the starting line in the left direction (and then the right direction, if a longer sequence is desired).

  Another solution would be to assign integers to valleys starting from the top and the bottom edges as well.

4 Advantages

- Since the image sequence is all that is used to compare two fingerprints, very little memory is required.
- The matching procedure could be made very fast. Also, it has many parameters that could be set by the user for desired levels of security.
- It is a topologically invariant sequence which has the potential to ignore scratches. We think this is quite powerful.

4 Fingerprint Matching - A Topological Approach

(Abstract)

Petr Lisonek
CECM, Simon Fraser University
e-mail: lisonek@cecm.sfu.ca
September 10, 1997

The geometrical approach to fingerprint matching has its drawbacks. In distance-based methods (measurement of distances between minutiae) these are, for example, effects of the ambient temperature and humidity and distortion caused by the softness of the tissue. In grid methods (considering the number of intersections, slopes, etc. relative to a given grid) we additionally encounter the problem of grid positioning whose uncertainty is much higher than the required precision (which is a fraction of the ridge/valley width) and may grossly override the fine resolution offered by the scanning technology.

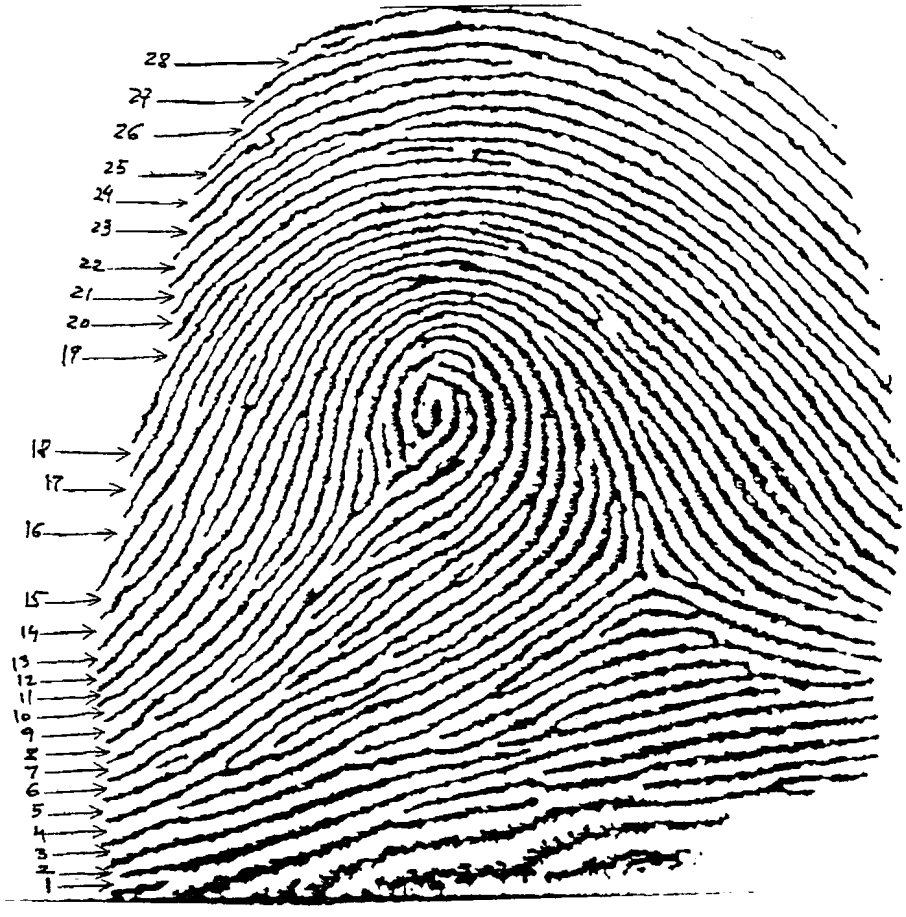

Figure 4: Fingerprint image showing the valley numbers.

Our approach proposes to encode the fingerprint using topological features found in it (spirals, bifurcations, closed ovals, etc.). These structures are (by their mathematical nature) much less sensitive to all kinds of metric distortions described above. The fingerprint is represented as a tree, whose root corresponds to the entire image, and whose branches represent the nesting/branching relations between the corresponding structures in the image. On the low levels of the tree we explicitly encode the distinguished topological features (examples above). Our method may look computationally intensive but certain patterns (spirals, ovals etc.) can be recognized in a single scanning sweep, and thus in real time, given the assumed technology of scanning that was proposed in the project description.

In the initial approximation our method is equivalent to representing the fingerprint as a planar graph (as known from Graph Theory) along with a given planar embedding, where vertices correspond to minutiae and edges to ridges between them. (This can be slightly enriched by adding an encoding for isolated closed ovals, which have no minutiae but still contain information, in the event that such features appear in fingerprints at all.) The fingerprint matching is then reduced to planar graph isomorphism (consistent with given planar embeddings) for which fast algorithms are known.

The set of distinguished topological features, some of which were suggested above, has to be determined by studying a large variety of prints and considering the complexity of their recognition in the given scanning technology (sweep).

5 Fingerprints: Global methods

Rolf Luchsinger
Dept. of Physics and Astronomy, University of British Columbia
e-mail: luchsi@theory.physics.ubc.ca

1 Introduction

Instead of looking for local features in the structure of the fingerprint picture, we propose here to map the entire picture of the fingerprint into a space where the specific features of the fingerprint might be better revealed. This is what we mean by a global method. Two possible mappings now commonly used in signal processing are the Fourier Transformation and the Wavelet Transformation. Due to personal experience with Fourier methods and a lack of experience with Wavelets, we will focus in the following on Fourier methods. However Wavelet methods might give better results for the problem at hand.

2 Method

The idea of the Fourier method is to decompose the picture (fingerprint) into a set of periodic structures. The wavelength of the periodic structures varies from the largest possible that fits into the image frame to the smallest wavelength that is comparable to the finest structures that need to be represented. The size of a Fourier coefficient at a given wavelength is a measure of the importance of the periodic structure of this wavelength in the picture. The collection of these Fourier coefficients is called the Fourier spectrum. Thus having a spectrum where only long wavelength coefficients are important, we can say that the picture has a smooth structure. If the short wavelength coefficients dominate, the picture has a rough structure. In that sense the different structure scales of a picture can be separated by the Fourier transformation.

Looking at fingerprints, we see in the first approximation a somehow regular structure of black and white lines. This regular structure will be revealed in the Fourier spectrum by a prominent peak

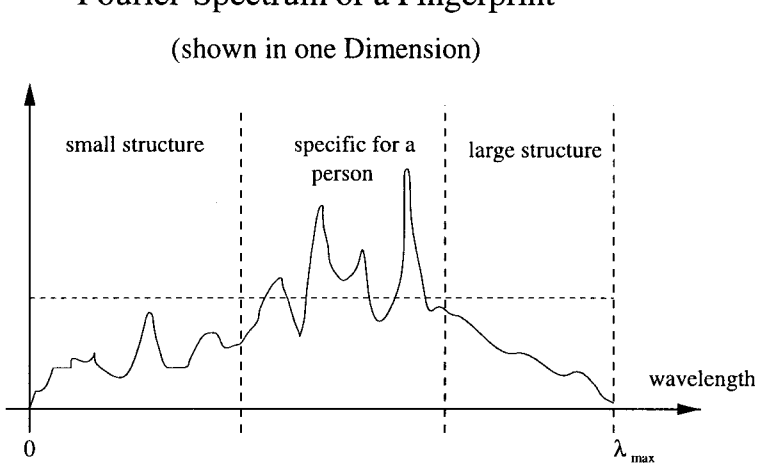

Figure 5:

and is a characteristic feature of the fingerprint. Other peaks will show up in the spectrum which will characterize the fingerprint. The short wavelength scale of the spectrum will be dominated by all kinds of noise and is not specific to a fingerprint. Since the Fourier transformation is a one to one mapping, all the information of the picture is contained in the spectrum. The person specific fingerprint information will presumably be contained in the most prominent Fourier coefficients. A possible Fourier spectrum of a fingerprint is shown in Fig. 5.

The basic idea of the Fourier method is to have a tool which can filter out the person specific information. Since Fourier methods are generally used for data compression, this approach seems to be promising. However the efficiency of the Fourier method depends on whether the characteristic information of a given fingerprint is revealed in a reasonably small number of Fourier coefficients. If this is the case, only this set of important coefficients has to be matched to the true person's set in order to classify a fingerprint. Looking at the variation of the spectrum using different fingerprints or the same fingerprint scanned several times is the only way to determine the efficiency of the method. Unfortunately this was not possible during the workshop.

3 Advantages

1. The method is fast and simple. You feed the picture of the fingerprint to a Fast Fourier Transformation algorithm, cut out the interesting region in the spectrum and compare it with your key data using a least square method with some tolerance criteria. (You don't need to follow lines, find bifurcation points etc. which is a lot of work for a computer.)

2. Fourier methods are in general well understood and used in many applications of signal processing.

3. The location of the finger on the sensor is not important. Shifting the coordinate system will just multiply a phase factor to the Fourier coefficients. This doesn't affect the absolute value of the Fourier coefficients.

4. If the high resolution of the image is useful, (that means it is not mainly noise you see on the small scale), you can make use of this information by including those small wavelength peaks.

5. The method is not sensitive to local damage to the fingerprint (cuts, dirt, etc.) .

4 Disadvantages

1. The method is a very general approach to signal processing. Whether it is reliable and optimal for the identification of fingerprints is an open question. However, fine tuning or a Wavelet approach might be possible. $->$ Check it out.

2. If it is the method of choice, it might have already been used in this context (patents).

6 Fingerprint Identification by the Measurement of the Angular Orientation of Successive Ridges within Narrow Vertical Bands

Michael Saliba
Dept. of Physics and Astronomy, University of British Columbia
e-mail: saliba@physics.ubc.ca

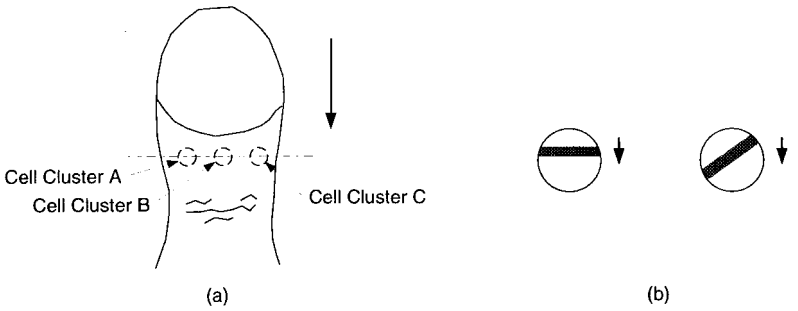

Figure 6: Location of the sensors for fingerprint scanning.

1 Introduction

This suggestion is intended to provide an alternative approach to the more conventional method of applying a pattern recognition algorithm to a static 2-D image. It attempts to take advantage of the specific characteristics of the KSI sensor, and the already selected method of scanning. It does depend rather heavily, however, on the actual capabilities of the sensor.

A more conventional variation of this approach is also suggested at the end of the next section.

2 Concept

Consider a sensor with only three active clusters of cells A, B, and C, each of about 1 mm diameter, as shown in Figure 6(a). As the finger moves down over the sensor, each cluster will be subjected to the passage of successive bright and dark fringes, corresponding to the ridges and valleys of the

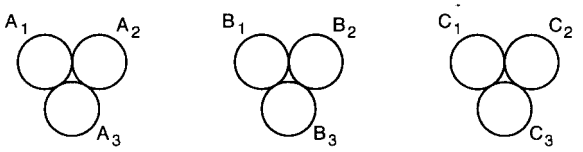

Figure 7: Utilization of sets of cell clusters for increased robustness.

fingertip topography. A horizontal fringe will cross the field of view of the cluster in a shorter time than will a fringe that has an oblique angular orientation (Figure 6(b)).

If the sensor can be made capable of registering the time that it takes for each fringe to cross over the field of view of each cluster, then each cluster can be made to output a string of numbers[6] corresponding to the times taken by successive fringes to pass across the cluster. For fingerprint identification the three sets of numbers from clusters A, B, and C can then be compared to the reference sets in the database. A suitable algorithm, with the appropriate tolerances, can be utilized to make the comparisons.

For example, it may be sufficient to search for a matching string of only six digits within the much larger string in each of the three sets; or, alternatively, the algorithm would take only (say) every third fringe into consideration. Each digit would also be allowed a certain tolerance (e.g. $\pm$ 2 bins) as determined during the testing of the prototype. As a further alternative, the algorithm could also be made to compare the *ratios* of the times between successive ridges, and/or the thickness of the fringes. If the independent matching of the three strings does not provide sufficient security, they can be tested for a correlation on the vertical (i.e. time) scale.

In the final design, in order to allow for possible variations in the lateral positioning of the finger in successive passes, each of the three clusters A, B, and C should actually be replaced by, say, three clusters that are slightly displaced from each other laterally, as shown in Figure 7. The three sets of numbers from clusters A_1, A_2, and A_3 can be compared to the "A" set in the database, and the best set can be selected (the same method is applied for sets B and C). This procedure may also correct for lateral expansion/contraction of the finger from day to day, as well as for the occasional ridge irregularity or junction that may happen to pass through the field of view of the cluster. Since the output strings of numbers will depend primarily on the *macroscopic angular orientation* of successive ridges, they are not expected to be very sensitive to ordinary day to day changes in fingertip topography, or to minor variations in the sensor response. Slight variations in the angular positioning of the finger can be made to fall within the tolerance of the comparing algorithm.

Apart from the possible constraints due to the sensor capability, a possible drawback to this concept may be due to its use of only a *part* of the fingerprint in order to make the identification (specifically, three[7] rather narrow vertical segments taken from the total 2-D image). It is however quite probable that there is in fact sufficient information within such a portion of the map. A method would also need to be devised to deal with or discard regions where the ridges may be oriented vertically. A major advantage of this overall concept is the small amount of memory needed for the database, as well as the use of relatively simple algorithms that would be computationally very fast.

A more conventional variation of this method would involve using an algorithm that simply measures the angular orientations, and possibly also the absolute and relative frequencies and thicknesses of the fringes in selected bands of the static 2-D image. An example with three bands is shown in Figure 8. The signature of a fingerprint is extracted exclusively from within these narrow regions.

[6] The measured times can be digitized into appropriately sized bins.

[7] Testing of the prototype would help to establish the optimum number of segments that should be utilized.

Figure 8: Selected vertical bands of the static image.

7 Fingerprint recognition algorithms

A. Zagoskin
Dept. of Physics and Astronomy, University of British Columbia
e-mail: zagoskin@physics.ubc.ca

- **Objective:** recognition of a sample fingerprint from a set of 100-1000 for the purposes of user access.
- **Conditions:** small computing time/memory requirements; robustness with respect to natural variations of the user's fingerprint
- **Algorithm A**

  1. **Description:** The fingerprint is coded as a square matrix of integer numbers by placing it on a rectangular $N \times N$ grid and assigning each edge of the grid a number n of its intersections by the fingerprint lines. (The actual size of the grid is variable, being determined by the size of the fingerprint, Fig 9.) Then each cell is assigned either a number $M = (n_{top} - n_{bottom}) + (n_{right} - n_{left})$ (algorithm A1), or its parity (P(M)=0 if M is even, P(M)=1 if M is odd) (algorithm A2) (refer to Fig 10). The resulting matrix is compared to the sample one stored in the system; if the number of matches is below the recognition margin, the access is denied, and vice versa.
  2. **An example:** The algorithm A2 was applied to two different fingerprints. (For the sake of simplicity we chose the initial grid 8×8; of it the central subgrid 6×6 was used in order to limit the effects of near boundary distortions.) The code matrices (6×6) are presented in Figs 11a,c. In order to model the natural variations of shape and positioning of the same fingerprint, the fingerprint (a) was also coded using a grid that was slightly distorted and displaced (by approx. 5%); the resulting matrix is presented in Fig 11b.

     The matching of code matrices is shown in Fig 12. The result suggests that there is a sufficient recognition margin that makes it possible to distinguish between natural variations of the same print (matching 66%) and different fingerprints (matching ~40%).

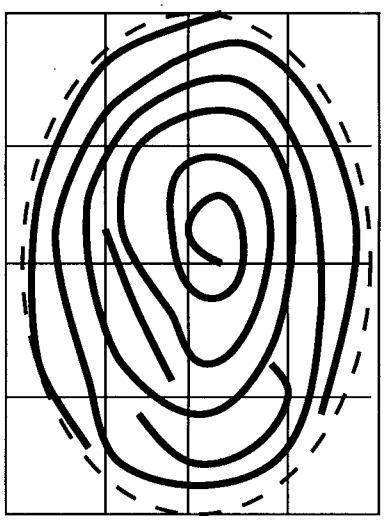

Figure 9: Grid positioning

Of course, in order to make a decisive conclusion and determine statistically significant recognition margins, a large pool of samples must be investigated.

3. **Advantages:** Simplicity; little computation time and memory required, so the check can work in real time; robustness with respect to imperfections of the print (e.g. it is not very likely that the grid line will pass exactly through an accidental gap in the ridge, thus changing the code numbers; in case of algorithm A2 accidental finger cuts will generally change the code numbers of at most 2 cells).
4. **Difficulties:** Sensitivity to the positioning of the grid; e.g., if a ridge goes through an intersection of the grid lines, or ends on the line, small variation of grid position will change the code numbers of several cells at once.
5. **Possible solutions:** The system can be programmed to perform several scans with a varied grid size, and throw away the "rogue" readings. The fringe of the grid can be discarded as in the above example; one expects the distortions to be most prominent there. The optimal grid dimension N should be determined from a large pool of fingerprints, but I estimate that the optimum will be reached at about 5 intersections per grid edge.

- **Algorithm B**

1. **Description:** This is a simpler version of the algorithm A, where instead of the $N \times N$ matrix, the fingerprint is characterized by two sequences of length $N + 1$ (Fig 13). Each term gives the number of interstections of a given *line* of the grid with the ridges. Thus a fingerprint is presented by two "scans" in perpendicular directions.
2. **Advantage:** This method is evidently simpler, faster, and more robust than the algorithm A, since the relative variation in the total number of ridges scanned across the whole grid is less than on a single edge. The shifts in the finger position are also less important, since matching of two one-dimensional sets can easily allow for such shifts.

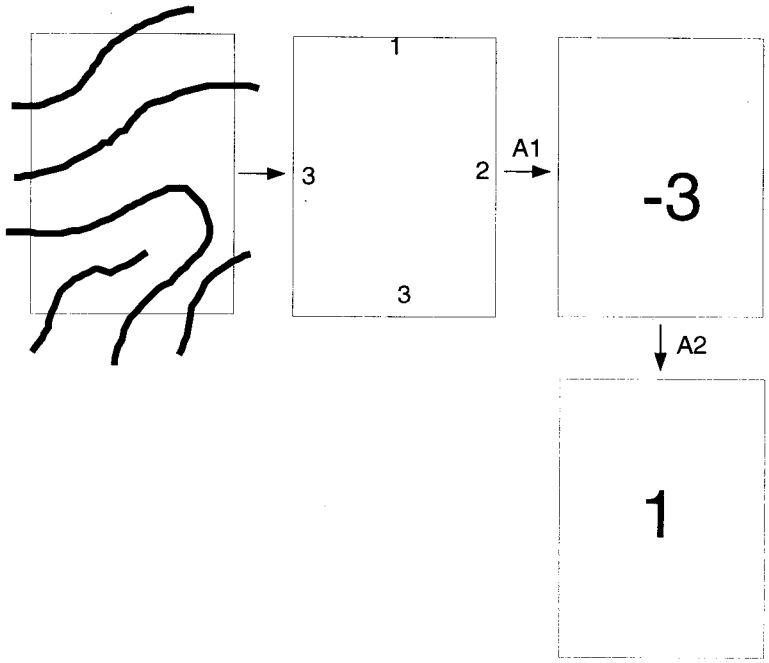

Figure 10: Algorithm A; M=(1-3)+(2-3)=-3; P(-3)=1

3. **Open question:** It is unclear *a priori* whether the codes created by this algorithm for different fingerprints can be reliably distinguished. This question can be answered only by investigating a large sample pool.

8 General Conclusion

In most of the algorithms that have been suggested in this report, the fingerprint image is reduced to a relatively short sequence of integers. This reduces the memory size required by the database. Each algorithm is intended to exploit specific properties and features of the fingerprint that vary from finger to finger, and that can be localized relatively fast using digital techniques, thus also reducing the computational time requirements to a minimum. In each case, the sensitivity of the algorithm to small variations in the image was also discussed, with the aim of reducing the False Rejection Rate, and of increasing the general robustness of the algorithm.

It is important to point out that the suggestions that have been put forward are conceptual in nature. Each model would require extensive development and testing in order to determine its feasibility, and in order to prepare for final implementation. In particular each model would have to be tested on a large number of fingerprint images, taken from a representative sample of the population. These images should preferably be taken using the KSI sensor. The final solution would possibly involve a *combination* of some of the algorithms that have been presented above.

It is hoped that many of the ideas that are contained in this report will prove useful to KSI in its development of a commercial live-scan fingerprint imager.

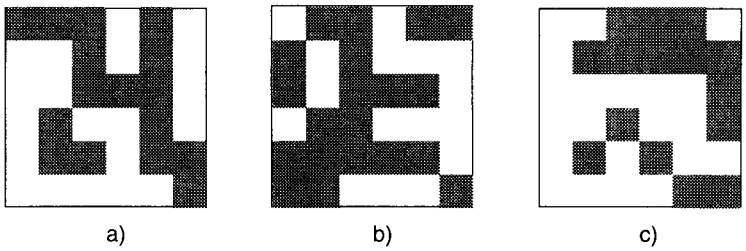

Figure 11: Code matrices (algorithm A2). Odd cells (parity 1) marked. a) Sample 1; b) Sample 1, distorted grid; c) Sample 2.

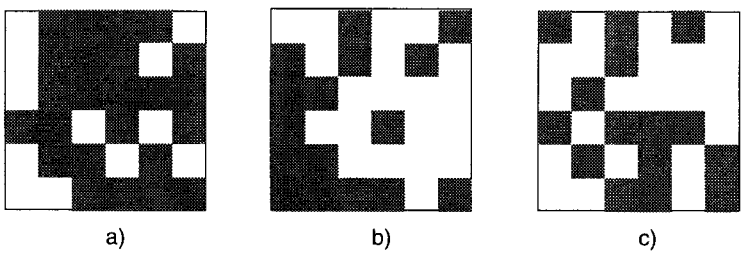

Figure 12: Matching of code matrices. Matching cells marked. a) Fig 11a vs. Fig 11b (same fingerprint): 66% match; b) Fig 11a vs. Fig 11c (different fingerprints): 44% match; c) Fig 11b vs. Fig 11c (different fingerprints); 42% match.

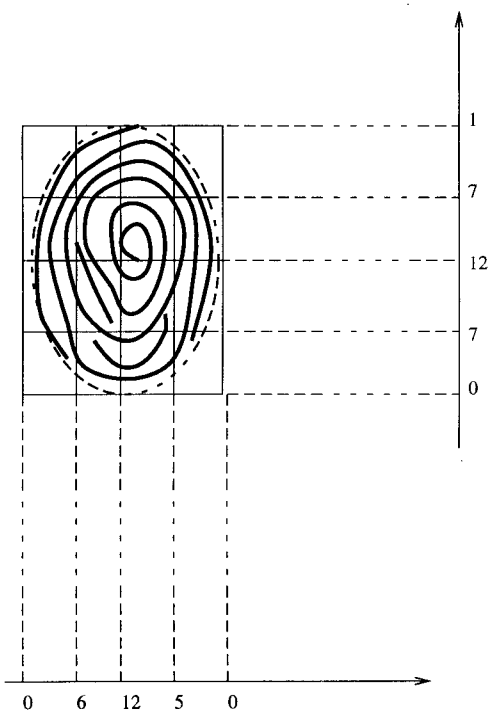

Figure 13: Code sequences (algorithm B).