U.S. Department of Commerce

National Bureau of Standards

Computer Science and Technology

NBS PUBLICATIONS

NAT'L INST. OF STAND & TECH

A11106 978112

NBS Special Publication 500-126

A Topological Approach to the Matching of Single Fingerprints: Development of Algorithms for Use on Latent Fingermarks

Malcolm K. Sparrow and Penelope J. Sparrow

QC
100
.U57
500-126
1985
c, 2

The National Bureau of Standards[1] was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, the Institute for Computer Sciences and Technology, and the Institute for Materials Science and Engineering.

The National Measurement Laboratory

Provides the national system of physical and chemical measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; provides advisory and research services to other Government agencies; conducts physical and chemical research; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

- Basic Standards[2]
- Radiation Research
- Chemical Physics
- Analytical Chemistry

The National Engineering Laboratory

Provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

- Applied Mathematics
- Electronics and Electrical Engineering[2]
- Manufacturing Engineering
- Building Technology
- Fire Research
- Chemical Engineering[2]

The Institute for Computer Sciences and Technology

Conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

- Programming Science and Technology
- Computer Systems Engineering

The Institute for Materials Science and Engineering

Conducts research and provides measurements, data, standards, reference materials, quantitative understanding and other technical information fundamental to the processing, structure, properties and performance of materials; addresses the scientific basis for new advanced materials technologies; plans research around cross-country scientific themes such as nondestructive evaluation and phase diagram development; oversees Bureau-wide technical programs in nuclear reactor radiation research and nondestructive evaluation; and broadly disseminates generic technical information resulting from its programs. The Institute consists of the following Divisions:

- Inorganic Materials
- Fracture and Deformation[3]
- Polymers
- Metallurgy
- Reactor Radiation

[1] Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Gaithersburg, MD 20899.
[2] Some divisions within the center are located at Boulder, CO 80303.
[3] Located at Boulder, CO, with some elements at Gaithersburg, MD.

NATIONAL BUREAU
OF STANDARDS
LIBRARY

Circ-NBS
QC100
.U57
no.500-126
1985
c.2

Computer Science and Technology

NBS Special Publication 500-126

A Topological Approach to the Matching of Single Fingerprints: Development of Algorithms for Use on Latent Fingermarks

Malcolm K. Sparrow
Penelope J. Sparrow

Center for Computer Systems Engineering
Institute for Computer Sciences and Technology
National Bureau of Standards
Gaithersburg, Maryland 20899

Issued October 1985

U.S. DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

National Bureau of Standards
Ernest Ambler, Director

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

Library of Congress Catalog Card Number: 85-600592

National Bureau of Standards Special Publication 500-126
Natl. Bur. Stand. (U.S.), Spec. Publ. 500-126, 67 pages (Oct. 1985)
CODEN: XNBSAV

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1985

For sale by the Suoerintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

TABLE OF CONTENTS

Abstract 1

Chapter 1. Introduction to the coding of latent marks 2

1.1 Introduction 2
1.2 Problems of interpretation and system design assumptions 2
1.3 Referencing and incompleteness problems 4
1.4 Early approaches and their drawbacks 5

1.4.1 Local characteristic codes 5
1.4.2 Series of radial lines 8

1.5 Ultimate objectives for file collection data storage 8
1.6 Sweeping-line systems 9
1.7 Radial scanning 10

Chapter 2. Early latent searching algorithms 14

2.1 Latent entry by vectors 14
2.2 Details of the latent enquiry 15
2.3 Details of the file-print coding 16
2.4 The algorithm "LATENT-MATCHER 1" (or LM1) 18
2.5 Improved latent-matching algorithms.("LM2", "LM3" and "LM4") 19
2.6 Testing algorithm performance 20
2.7 Latent enquiry by vector: shortcomings 21

Chapter 3. Latent searching: topological coordinate systems 23

3.1 Latent entry by topological coordinates 23

3.1.1 The 4th coordinate 23
3.1.2 Dispensing with boundary vectors 24
3.1.3 'Wrap around' 360° sector 24

3.2 Topological reconstruction from coordinate sets 25

3.2.1 The 'continuity' array 25
3.2.2 Opening the continuity array 27
3.2.3 Associations, entries, and discoveries in the continuity array 28
3.2.4 Properties of the completed continuity array 30
3.2.5 Final stage of topological reconstruction 30

3.3 The matching algorithm LM5 31
3.4 The vector comparison stage 31
3.5 Final score formulation 33

3.5.1 The notion of 'compatibility' 33
3.5.2 Score combination based on compatibility 34

3.5.3 Candidate promotion schemes 35

3.6 Performance of LM5 36
3.7 Computation times 37
3.8 File storage space — defaulting the 'edge topology' 38

Chapter 4. Associated applications and conclusions 39

4.1 Derivation of vectors for rolled print comparison 39
4.2 Image-retrieval systems 40
4.3 Outline of further work to be done 46
4.4 Conclusion 47

References 48
Bibliography 49
Appendices A – F
Acknowledgements

LIST OF FIGURES

Figure . . . . Page

1 Sample latent marks. 3
2 Local bifurcation-based vector coding. 6
3 'Offsetting' of generating lines. 7
4 Radial line coding scheme. 9
5 'Sweeping line' system. 10
6 Sample sector for radial scanning. 11
7 Characteristic-based radial lines. 13
8 Selected line placement on latent mark. 15
9 Boundary lines and observation point on a plain arch. 17
10 Irregularity types, and their codes. 18
11 Radial irregularity-centered lines, with the 'cut' vertically below the observation point. 26
12 Fingerprint reconstruction. 42
13 Copy of fingerprint tracing. 43
14 Latent tracing, and its reconstruction. 44
15 Reconstruction with, and without, defaulted edge toplogy. 45

A TOPOLOGICAL APPROACH TO THE MATCHING OF SINGLE FINGERPRINTS: DEVELOPMENT OF ALGORITHMS FOR USE ON LATENT FINGERMARKS

Malcolm K. Sparrow and Penelope J. Sparrow

ABSTRACT

The research described in this paper follows that reported in a previous N.B.S. Special Publication (No. 500-124), in which topological coding schemes were devised for automated comparison of rolled impressions. The contents of that publication are a pre-requisite for a proper understanding of this one. The development of topological coding schemes is here extended to cover the automated searching of fragmentary latent marks, such as would be found at the scene of a crime.

The benefits to be derived from topological descriptions of fingerprints are a direct result of their immunity to change under ordinary plastic distortion. In the case of latent marks such spatial distortions tend to be exaggerated; hence the importance of applying topology-based systems to them.

This paper describes a method of coding fingerprint patterns by a variety of 'topological coordinate schemes', with fingerprint comparison being performed on the basis of localized topological information which is extracted from the recorded coordinate sets. Such comparison is shown to offer a substantial improvement in performance over existing (spatial) techniques.

Furthermore, a method for pictorial reconstruction of a complete fingerprint, from its coordinate representation, is demonstrated.

Key words: Automated comparison; fingerprints; image-retrieval; latent-marks; minutiae; topology.

CHAPTER 1.

INTRODUCTION TO THE CODING OF LATENT MARKS.

1.1 Introduction.

A previous paper[1] has described the development of algorithms for automated comparison of rolled fingerprint impressions through the use of ordered topological descriptions. The purpose of this paper is to extend the application of topological coding and comparison to cover the automated searching of 'latent' (or 'scene-of-crime') marks against large fingerprint collections. It should be stated here that a proper understanding of this paper will only be gained when its predecessor has been read and absorbed; much material from that paper is referred to, and used, here (e.g. the methods of vector matching used in the algorithm MATCH4) without repetition of the relevant explanations.

At the commencement of the work on rolled impressions it was stated that such work could be regarded as preparatory for tackling the problems of latent marks, and that it could be expected to provide general education as to the behavior of topological codes under conditions not much worse than 'ideal'. We should consider, therefore, which of the major lessons learnt we can expect to apply to any topological coding scheme for latent searching. In fact there are just two such major lessons worth recalling at this stage :—

Firstly: that the 'placing of lines' is a neat and efficient basis for the ordering of topological information provided, of course, that sufficient global information is available to determine the 'correct' placing.

Secondly: that the greatest power of discrimination between mates and non-mates will be realized by algorithms that use a combination of topological and spatial information.

1.2 Problems of interpretation and system design assumptions.

It would also be prudent to remind ourselves of the special problems posed by latent marks. Some of those problems stem directly from the physical nature of the marks themselves — usually being chemically developed (and subsequently photographed) versions of a perspiration deposit on some object that has been handled. These are :—

(a) that the image will usually lack the clarity of an inked impression.

(b) spatial distortion will be exaggerated and unpredictable as it will be dependent on the shape of the object handled and on the direction and magnitude of pressure exerted upon it.

(c) the surface of the object itself may well give an interference pattern superimposed (rather 'sub-imposed') on the ridge detail and which needs to be filtered out from it.

(d) the fingerprint image may well be smudged if (as is usually the case) there was a degree of lateral movement at the moment the impression was left.

The sample latent marks shown in figure 1 illustrate these problems very well.

Figure 1. Sample latent marks. (Approx. 5×)

These problems sound as if they ought to be the very meat of some image enhancement process. One could expect that two-dimensional Fourier transforms would be used to remove the effects of lateral movement and to utilize the periodicity of the ridge pattern in order to separate it from the background interference.

Perhaps, at some time in the future, image-enhancement techniques may be so much improved as to render them capable of doing a reasonably good job of interpreting latent prints; for the time being, however, they are nowhere near effective enough for this application. Current research methods are, just now, bringing such processes close to the point where we can rely on them to make a fairly accurate interpretation of clearly inked rolled prints from a scanned image, and to automatically extract the positions of characteristics from that interpretation. However the degree of success with which even the most sophisticated systems can handle rolled impressions of poor quality is highly questionable — and nobody seriously expects machines to be able to read *latent* prints effectively without a great deal of human (interactive) assistance. (Some systems provide for technicians to make a tracing of the latent — the tracing then being read by automatic scanners. In this case the interpretative stage is completed in the process of making the tracing.)

Indeed the reading of latent marks requires the very highest level of interpretative filtering that the human brain can provide. The job of reading and searching latents is the most difficult task asked of the fingerprint expert and is, in many organizations, the preserve of only those technicians with the greatest amount of experience and expertise.

It is currently the case, therefore, that when minutiae data representing latent marks are fed into an automated system (for searching against a large file collection) the data are already the outcome of a human (and usually manual) interpretative process. This 'state-of-affairs' is, in fact, perfectly reasonable. A latent mark is usually found by a painstaking and thorough search of the scene of a crime by highly trained personnel. It is then developed by a variety of means (the use of laser being the most publicized recent development) but always with great care — for the information content within the mark is both scant and fragile. One could expect a similar degree of care to be exercised in entering that information into an automated system lest any of it be lost. The whole of the information-gathering process is a 'once only' process, as opposed to the comparison against file-prints, which is a repetitive process. There is therefore very little to be gained, and much that could be lost, by automating the latent entry process.

For these reasons the *fact* of manual human encoding of latent marks is an underlying assumption of this project. Consequently we should endeavor to ensure that any method devised for coding a latent mark by topological means can be carried out manually by a human technician both easily and quickly, and without requiring any detailed mathematical knowledge. This requirement is met by all the coding schemes described hereafter.

Quite a different assumption pertains to file collections — namely, that automatic file conversion (by scanners linked to processors) is a prerequisite for establishing major computerized systems. The data requirements for topological coding schemes *for the file-prints* are therefore limited to those which demand little or no advance on existing automatic-reading techniques.

1.3 Referencing and incompleteness problems.

Once the latent has been traced, or otherwise interpreted, to the best of the technician's ability some special problems remain which make significant demands on any searching algorithm :—

(a) It may not be possible to determine the 'pattern type' classification of the finger which made the mark.

(b) 'Referencing' or 'registration' of the mark to some standard orientation may not be possible as referencing features (such as cores/deltas/creases) may not be visible.

(c) Ordering of information within a latent mark according to any standardized global scheme will not be possible. Frequently one cannot tell precisely from which part of the finger the latent comes, nor can one always accurately determine its orientation.

It is clear that problems (b) and (c) above render the topological schemes used on rolled impressions (in which, for example, a line was placed through the core of loops and whorls running parallel to the crease[2]) wholly inappropriate and that either an *unordered* or a *locally ordered* information system is required as a basis for topological comparisons involving latent marks.

1.4 Early approaches and their drawbacks.

The development of systems for use on latent marks started with two simple ideas for methods of coding prints that are based on two different 'line-placement' rules. Neither of them are really satisfactory as self-contained schemes (as will be explained), but they were both important stages in the evolution of the eminently satisfactory solution to be described in chapter 3. It was the bridge built between these two ideas that pointed the way firmly towards development of a 'topological coordinate system'. The two foundation ideas are described here in turn.

1.4.1 Local characteristic codes.

A fingerprint can be coded topologically by recording an *unordered* selection of *local* topological codes. Each topological code would be a vector generated by systematic exploration from short straight lines drawn through a characteristic, and orthogonal to the local ridge flow direction. Searching a latent mark against a collection so coded would then be by extraction, from the latent, of a similar vector (or vectors), followed by vector comparison of the kind well established in the previous paper.[3] Suppose we used bifurcations alone as bases for local vector extraction — and derived eleven digit codes by allowing the line to span two ridges either side of the selected bifurcation. Such an information gathering process could be represented pictorially as in figure 2.

There are a number of adaptations to this basic idea which would help to make it compatible with those vector comparison algorithms already developed. They are :—

(a) that lines placed should be imagined to be *offset* by an infinitesimally small distance, so they pass *right by* the bifurcation rather than *through* it. The reason for this is that it gives an even number of topological exploration paths (rather than an odd number) yielding an even number of digital codes.

(b) that the order of topological exploration shall be changed to the convention *work outwards from the core, and always look left before you look right.* *

* The core itself may not be visible. There are, however, very very few latent marks where the ridge curvature does not give away a very rough location for the core (or, in

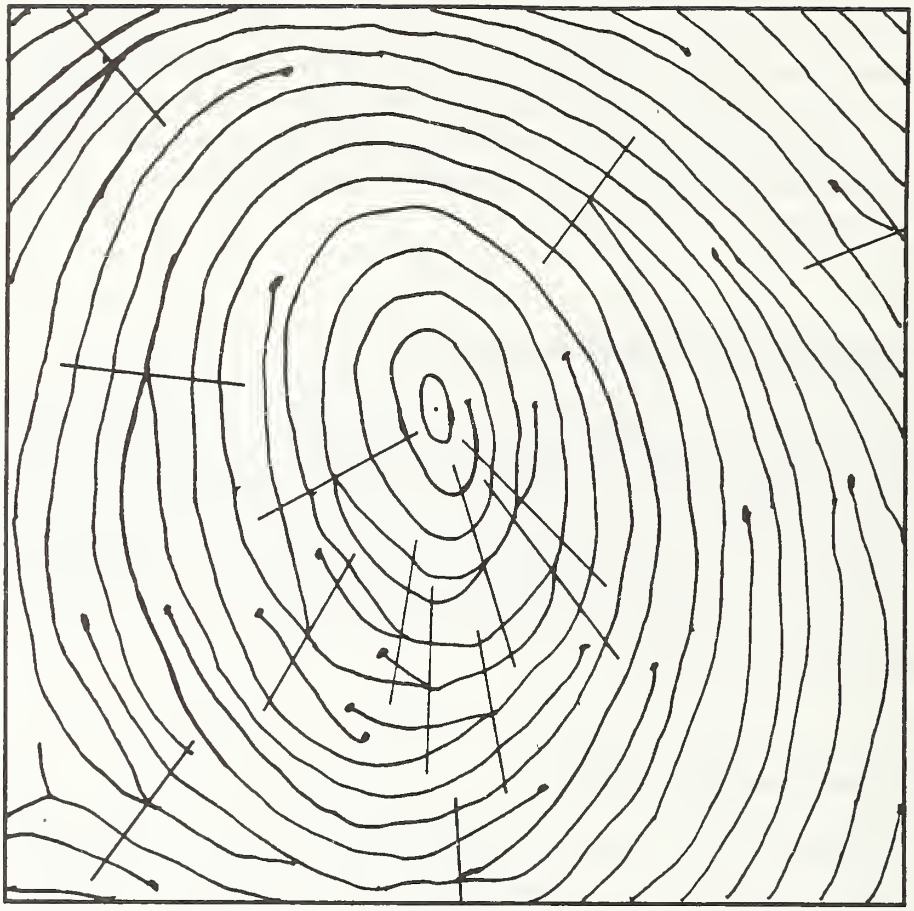


Figure 2. Local bifurcation-based vector coding (schematic).

(c) that each topological event code shall have a distance measure associated with it.

Such an updated version of local bifurcation coding would provide digital arrays compatible with the array comparison techniques incorporated into the algorithm MATCH4.[4] The new order for the ridge exploration event codes would be as shown in figure 3. Note the slightly offset line, and the fact that the 5th exploration runs immediately (i.e. at *zero* distance) into the central bifurcation where it would give digital code 7 (for 'bifurcation ahead').

If the bifurcation had faced in the opposite direction then we would choose to offset the line to the right, as before, rather than to the left. We thus add a further convention regarding the placing of characteristic-centered lines, namely that *lines based on minutiae*

---

the case of an arch, an idea of the print orientation). To order these vectors correctly in the absence of a visible core one needs only to be able to determine which is the *inside* of the mark and which is the *outside*. On those few occasions when this is not possible a double-entry facility would be needed to cover both possible interpretations.

Figure 3. 'Offsetting' of generating lines.

should be marginally offset in a clockwise direction (clockwise with respect to the assumed position of the core) for the purpose of ordering topological information, but by a negligible physical distance so as to make the distance from the characteristic to its line effectively zero.

Furthermore, in the light of previous experience with topological code vectors, the following generalizations ought to be made to this scheme :—

(a) All true characteristics should have their topological neighborhoods coded, rather than bifurcations alone. The inclusion of ridge-endings is essential in view of the increased frequency of bifurcation/ridge-ending mutations observed when dealing with latent marks whose interpretation is so difficult.

(b) Vectors should not be limited in length by the span of the generating line being set at just two ridges: rather the span should be a parameter of any comparison algorithm. [We already know that discrimination on rolled prints between mates and non-mates improves substantially with vector length up to size 30×2 (i.e. 15 ridge intersection points).[5] The assumption that vector comparison algorithms would be implemented on array processors removes any concern that there might be over increases in processing time that could result from the use of longer vectors.]

The principal drawback of this coding scheme is its data storage requirement. Having accepted the desirability of using longer vectors, let us suppose a standard span of 10 ridges was chosen: there are then 20 ridge intersection points (ten each side of the characteristic) yielding 40 topological event codes, and forty associated hexadecimal distance measures. The storage requirement for file collection prints is therefore 40 bytes *per characteristic*, which is quite unreasonable. It is particularly unreasonable when account is taken of the very high degree of redundancy that there would be in such a set of data. The relationship of one characteristic to a near neighbor would be recorded many times over.

Shortening the vectors stored (by reducing the parameter SPAN) would certainly reduce the data storage requirement but would be expected to worsen performance. Facing such trade-offs between data storage requirements and performance is a situation that we can, and should, avoid.

1.4.2 Series of radial lines.

The second fundamental approach to file-print coding for latent searches is a simple extension of the line-based coding system used in the scheme for rolled impressions.[6] One single line superimposed on the rolled print was used to generate 82-digit vectors, and the lines were placed (except in the case of arches) by reference to the central core. Topological information was thereby recorded mainly from those parts of the print close to that line, and not from the entire print.

It is essential, in any latent scheme, that information from *every* part of each fileprint be recorded in order that information from a latent mark will have some representation in the matching file-print data irrespective of which part of the finger made the latent impression.

If a whole series of lines were drawn radially from the core, as shown in figure 4, and vectors derived from each of them, then topological information would be recorded from all over the print. In figure 4 the spacing of the lines has been set at 30°. Given a latent mark one could then draw a line centrally across it at such orientation as was deemed most likely to pass through the core (assuming the core is not visible). Then one topological event code vector can be generated from that line according to established conventions (i.e. working outwards, and looking left before right).

Provided the radial lines on the matching file-print were sufficiently close together one could expect some portion of one of those file vectors to be very similar to some portion of the latent vector. The degree of similarity would depend, to a certain extent, on how lucky one was in choosing the position for the line on the latent. If it corresponded within, say, 5° of the position of one of the radial lines on the mate file-print then a very good vector comparison score would result. If the latent line fell half way between the corresponding positions of two of the file-print radial lines, and in an area of high characteristic density, then vector comparison scores would be very poor.

Use of a greater number of radial lines (e.g. with 10° spacing) would raise latent mate scores but would, once again, increase data requirements for file-prints to unacceptable levels. Moreover, use of line-placement, on a latent, that is not tied either to a core or to any visible characteristic effectively rules out the use of distance measures as a means of enhancing the performance of topological vector comparison (except, perhaps, careful use of *summed* and *differential* distance tests. These tests measure distance between two characteristics not directly associated with the line placement.[7])

1.5 Ultimate objectives for file collection data storage.

The two methods described above appear cumbersome; there is neither speed nor reliability to be obtained through their use. No substantial experiments were conducted on either of them as the data requirements (and therefore the time taken in a manual encoding process) were prohibitive — especially if any attempt was to be made to obtain

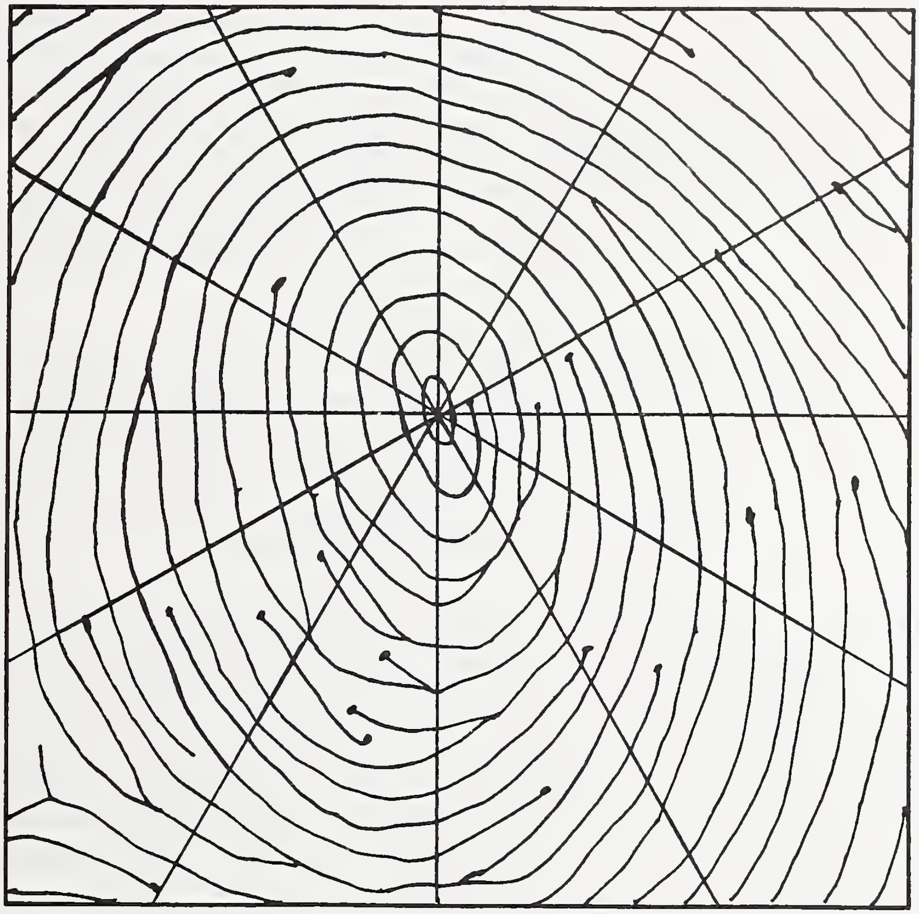


Figure 4. Radial line coding scheme.

the maximum reliability. Consideration of their use did, however, help to formulate an objective for the design of a workable topology-based latent scheme, namely that we should find :—

a method for recording a complete topological description of a print (so that the topology of any part of it can be inferred) subject to the constraint that each characteristic be recorded once, and once only.

1.6 Sweeping-line systems.

The key to attaining the objective stated above lay in the realization that characteristics could be seen as *small changes* in the otherwise laminar flow of the ridge pattern. That realization leads onto the idea that *the whole topology of a print is merely the summation of a series of small changes in an otherwise smooth ridge flow pattern.*

For the sake of a more practical understanding of this statement suppose that a

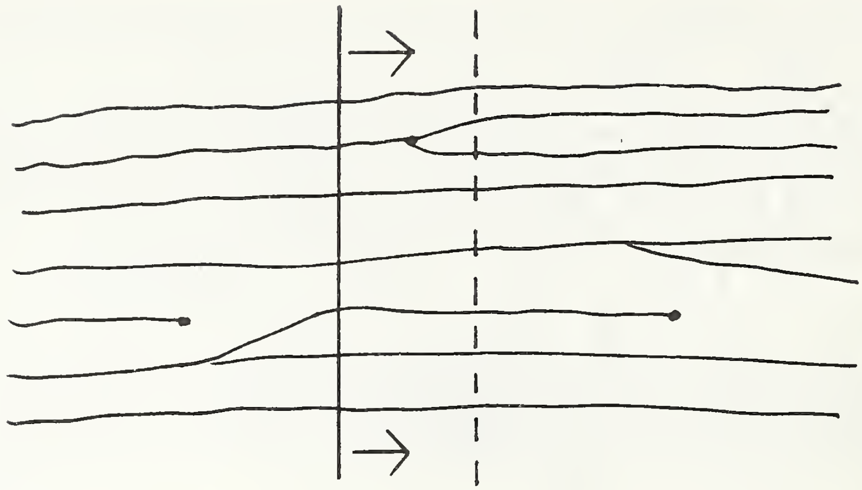


Figure 5. 'Sweeping line' system.

topological code vector had been generated by a line placed in some particular position on a print. Now suppose that the line was displaced by a small translation in the direction of the ridge flow so that it now passed the other side of one characteristic (in other words — the line *passed over* one characteristic), and a new code vector generated to represent the new line position. (See figure 5). How would the two vectors differ? Certainly they would be very similar, and the differences (which would all be local to the characteristic *passed over*) could all be deduced from certain knowledge about that one characteristic. In order to detail those changes you would need to know:—

(a) what type of characteristic was it, and which way was it facing?

(b) which ridge, or ridges, was it on?

(c) what can we now see (looking right along ridges) that we could not see before by virtue of the presence of that characteristic? (i.e. we now have new ridges to explore — two new ones in the case shown in figure 5.)

A set of rules can be built which would detail all the vector changes that are caused by each particular type of characteristic when they are *passed over* by a sweeping line.

In figure 5 the new (displaced) line vector can be seen as the original line vector 'plus' the changes caused by passing over that characteristic. Further displacement of the line (i.e. a continued *sweep*) will add further changes to the vector as other characteristics are reached and *passed over*. This is a very general introduction to the basis of what could be called 'sweeping line systems'.

1.7 Radial scanning.

The 'radial scanning' scheme is one particular case from the broader class of sweeping line systems. It provides a method for recording the *whole topology* of any sector of a fingerprint. It has two principal determining features:—

(1) that a *central observation point* on the fingerprint is selected.

(2) that the sweeping line used is a straight one, and it scans radially as if it were pivoted from the observation point.

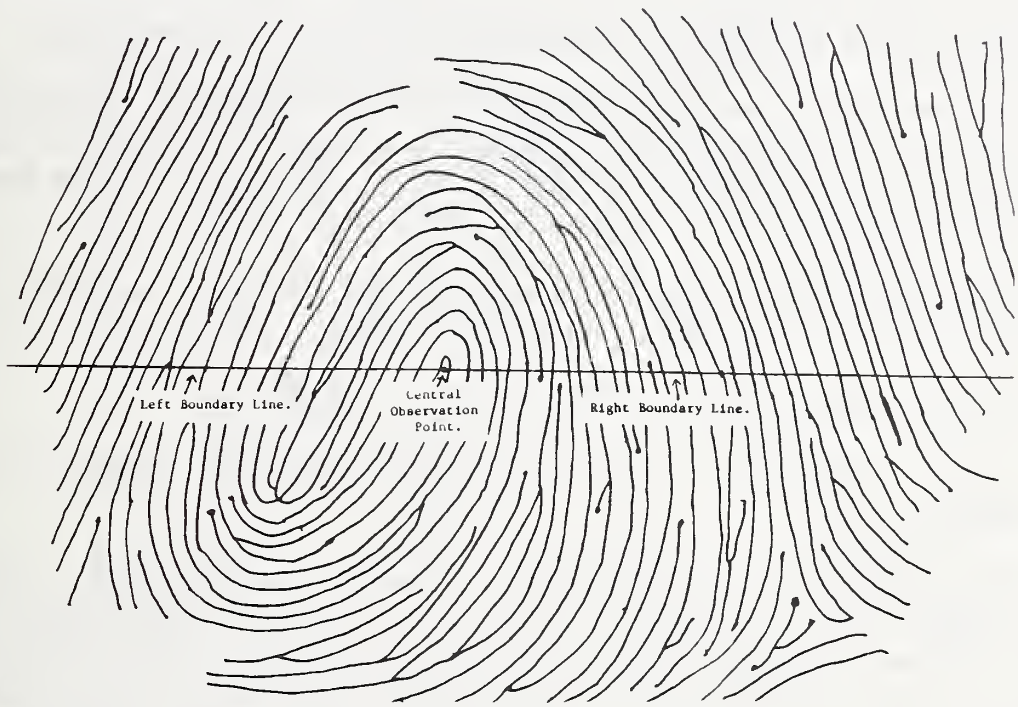


Figure 6. Sample sector for radial scanning.

The similarity of such an idea with the appearance of a radar screen is quite obvious, and may well be a helpful aid to understanding the application. To demonstrate the use of radial scanning let us consider the 180° sector of the fingerprint shown in figure 6. (In effect this means the half of the print above the horizontal line; that part of the print should be regarded, however, as a sector enclosed by two radial lines.) The topology of the whole sector can be described by recording the following information:

(a) two boundary vectors: these are the topological code vectors generated from the boundary radial lines.

(b) a complete listing of all of the characteristics, together with any other irregularities in the otherwise laminar flow, that occur within the sector. Each irregularity must be listed in a manner which shows the *nature* of the irregularity, the *order* of their appearance, and *on which ridge* each one occurs. It is important that *all* irregularities (i.e. not just those that are genuine characteristics) should be recorded; these will include ridges coming into sight, going out of sight, recurving etc.

The form of data contained in the boundary vectors can be assumed (for the purpose of this section) to be our standard format for line-generated vectors with their associated distance measures.[8] The listing of flow irregularities, however, is quite new — and takes the form of a *coordinate set.* The *coordinates* for each irregularity consist of

(i) a hexadecimal digital code (T) representing the type of the irregularity.

(ii) the angular coordinate (θ) of the irregularity. This is sufficient to specify the order in which they are *passed over* by the sweeping radial line. We will use angular measures that increase clockwise, with 0° being coincident with the left boundary line. Thus θ will range from 0° to 180°, in the case of figure 6.

(iii) the ridge-count (R) between the irregularity and the central observation point. This is sufficient to specify *on which ridge* it occurs.

A most valuable observation can now be made, namely that

a list of coordinate sets of the form (T, θ, R) *specifies the topology of a sector uniquely.*

That statement could be presented as a theorem, requiring proof — but it is hardly necessary. The best proof of the assertion that the whole ridge structure can be reconstructed unambiguously from such a set of data, is to describe the method for doing just that. In chapter 3 appears a detailed explanation of the mechanism for *topological reconstruction* from such a *topological coordinate set.* Such detailed description is not included here as the purpose of this chapter is purely to recount the evolution of ideas which led to development of *topological coordinate systems.*

In order to show just how closely related this coordinate system is to the two foundation ideas described earlier (para 1.4) let us adapt figure 4 slightly. Figure 7 shows the same print with a radial line drawn marginally offset from every visible ridge flow irregularity. The lines span the whole visible ridge structure (rather than being limited to just a few ridges), and their spacing is determined by the angular position of the irregularity (rather than by a fixed, regular interval). A set of coordinates of the form (T, θ, R) can then be seen as the most economical method of recording the sequence of changes in topological code vectors that occur between one radial line and the next. The diagram (figure 7) bears an interesting resemblance both to figure 2, and also to figure 4, and could be taken to be a hybrid of the two.

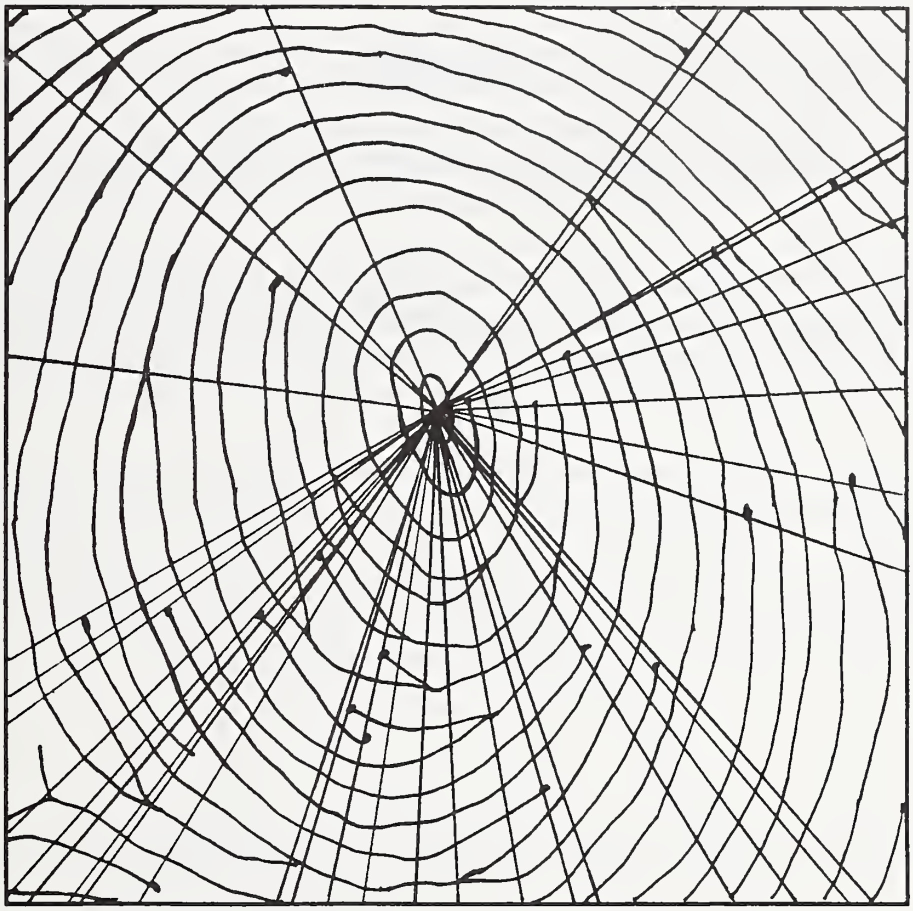


Figure 7. Characteristic-based radial lines.

CHAPTER 2

EARLY LATENT SEARCHING ALGORITHMS.

There are two different ways of describing a chicken. The first is to describe an egg in detail and then to trace all the changes that take place as it develops into a chicken. The second is to describe the fully grown chicken in detail and, perhaps, make a few brief comments about the egg just to put things in context.

In describing latent matching algorithms we shall follow the second of these two paths. Chapter 3 is the detailed account of the fully-fledged solution, and these next few paragraphs are intended merely as an overview of the early stages of development. Consequently the intricacies of these algorithms are not explained here, and there may be nagging questions in the mind of the reader as to some of the finer points of *topological reconstruction.* All those questions will be answered in due course.

2.1 Latent entry by vectors.

All of the algorithms to be mentioned in this chapter have certain basic features in common. They are: —

(a) That the entry of data from the latent mark is by way of characteristic-centered vectors which are manually encoded from a traced image of the latent.

(b) That file-print data is entered and stored in the form outlined in paragraph 1.6 (i.e. by two boundary vectors plus topological coordinates (T, θ, R) for all intervening characteristics and other ridge flow irregularities).

In order to perform a comparison each algorithm first topologically reconstructs the file-print from its coordinate set, and then automatically extracts characteristic-centered topological code vectors from its reconstruction. Vectors centered on all 'suitable looking' characteristics (i.e. characteristics of the right *type* that lie within an area of the print which is specified at the time of latent data entry) are then compared with the latent vector and a score is obtained in each case. The highest score obtained by an extracted file-print vector is taken as the score for that file-print. It is assumed to be the score from the characteristic (on the file-print) whose *topological neighborhood* most closely resembles that of the characteristic on the latent mark upon which the latent vector's generating line was centered.

The vector comparison itself is practically identical to that used on rolled impressions (i.e. as per the algorithm MATCH4).[9]

2.2 Details of the latent enquiry.

Figure 8 shows the tracing of a latent mark (at 7× magnification) with a generating line placed on it. The placing of the line requires some subjective judgement on the part of the operator. Firstly a characteristic should be chosen which is fairly central on the mark. Secondly a line should then be drawn across the ridge flow, oriented so that it points at the assumed position of the core (or actually *through* the core if it is visible on the mark), and spanning as many ridges as are considered *useful* in gathering information from the latent. The line is to be marginally offset from the central characteristic, as discussed in para 1.4.1. The topological code vector generated by this line is entered as the *latent enquiry vector*, complete with its associated distance measures (which are manually measured by the operator.)

Figure 8. Selected line placement on latent mark.

Also certain information about the selected central characteristic (hereafter referred to as the *central feature*) is entered as part of the latent enquiry. Its *type* code is required, as are angular and ridge count bounds within which it is deemed to lie with respect to the assumed core position. (These bounds are solely for the purpose of limiting the number of vector comparisons to be performed. If they cannot reasonably be specified then they are 'defaulted' so that the whole file-print sector is searched for suitable matching characteristics.) A complete latent enquiry is shown at appendix A, where the data for the latent shown in figure 8 appears on a form prepared for the purpose.

2.3 Details of the file-print coding.

The sector chosen for early experiments was a 180° sector that covered the upper half of each file-print. (This is the part of the finger that most often appears on latent marks.) Limiting the data recorded to a 180° sector was for convenience alone, due to the time consuming nature of the manual coding operation.

The *observation point* was selected to be adjacent to the core in the case of loops and whorls, and at the base of the upcurve (the point at which a 'summit line' can begin to be seen) on arches. Figure 6 shows a typical position for the *observation point* and *boundary lines* on a print with a central core, and Figure 9 shows a suitable placing for these when used on a plain arch. Notice that the observation point is always placed in a valley rather than on a ridge: this is so as to give unambiguous ridge counts in every direction.

All of the irregularities in the sector *between* (in this case *above*) the boundary lines are then recorded by sets of topological coordinates of the form (T, θ, R). The *type* of irregularity is shown by a single hexadecimal digit — and the allocation of digits is closely related to the allocation already in use for ridge-exploration events (see appendix B). The list of possible irregularities, with their hexadecimal codes is given here. The descriptions can best be understood clearly if you think of these irregularities as being *passed over* by a pivoted radial line which is sweeping in a clockwise direction.

Code 0 — ridge runs out of sight.
Code 1 — ridge comes into sight.
Code 2 — bifurcation facing counterclockwise.
Code 3 — ridge ending.
Code 4 — ridge recurves with the effect of *losing* two ridges.
Code 5 — ridge recurves with the effect of *gaining* two ridges.
Code 6 — facing ridge ending (i.e. facing in the opposite direction to a '3'.)
Code 7 — bifurcation ahead (i.e. a '2' reversed).
Code A — ridge runs into scarred tissue.
Code B — ridge runs into an unclear area.
Code C — compound characteristic (2 ridges in, and 2 ridges out).
Code D — ridge emerges from scarred tissue ('A' reversed).
Code E — ridge emerges from unclear area. ('B' reversed).

Figure 10 shows a completely artificial fingerprint pattern which just happens to have one of each type of irregularity shown on it, spaced at 25° intervals. Radial lines are

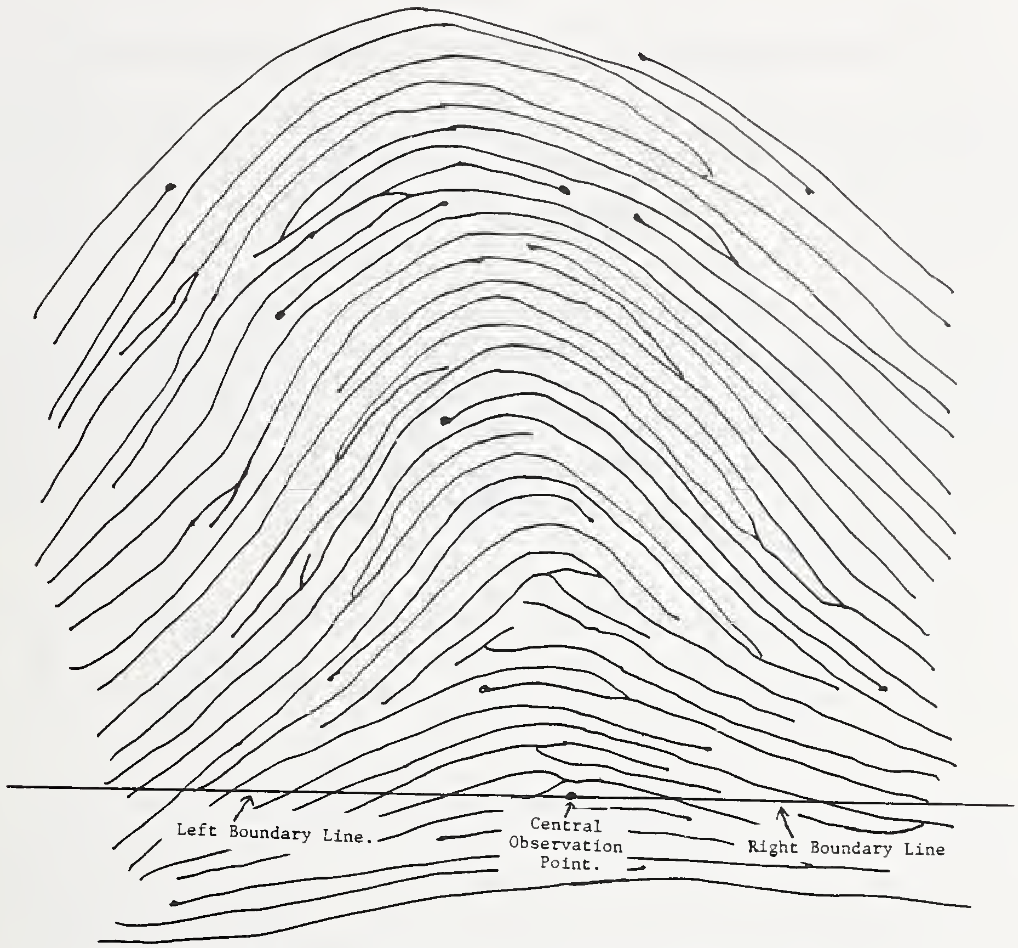


Figure 9. Boundary lines and observation point on a plain arch.

used to identify each of the irregularities with its hexadecimal code. It gives an adequate illustration of each different type.

On the print shown in figure 6 there were a total of 77 such irregularities between the boundary lines. The complete data representation of that file-print is shown in appendix C — there you will notice the inclusion of some numbers referred to as *distance conversion measures.* These give an approximate ridge spacing wavelength at four sample orientations ($0°, 60°, 120°, 180°$) which enable the comparison algorithms to convert angular information into an estimate of *ridge-traced distances* for the purposes of vector comparison. You may also observe, in appendix C, that the boundary vectors are one-sided (as opposed to the more normal double-sided form). This is because it is only necessary to provide the reconstruction algorithm with the parts of the boundary vectors that represent information from *outside* the coordinate sector. The algorithms are quite capable of working out for themselves what happens when ridges are traced *into* the sector — as this can be deduced from the coordinate information. [The reference in appendix C to units of 0.5cms is in the context of tracings done at $10\times$ enlargement.]

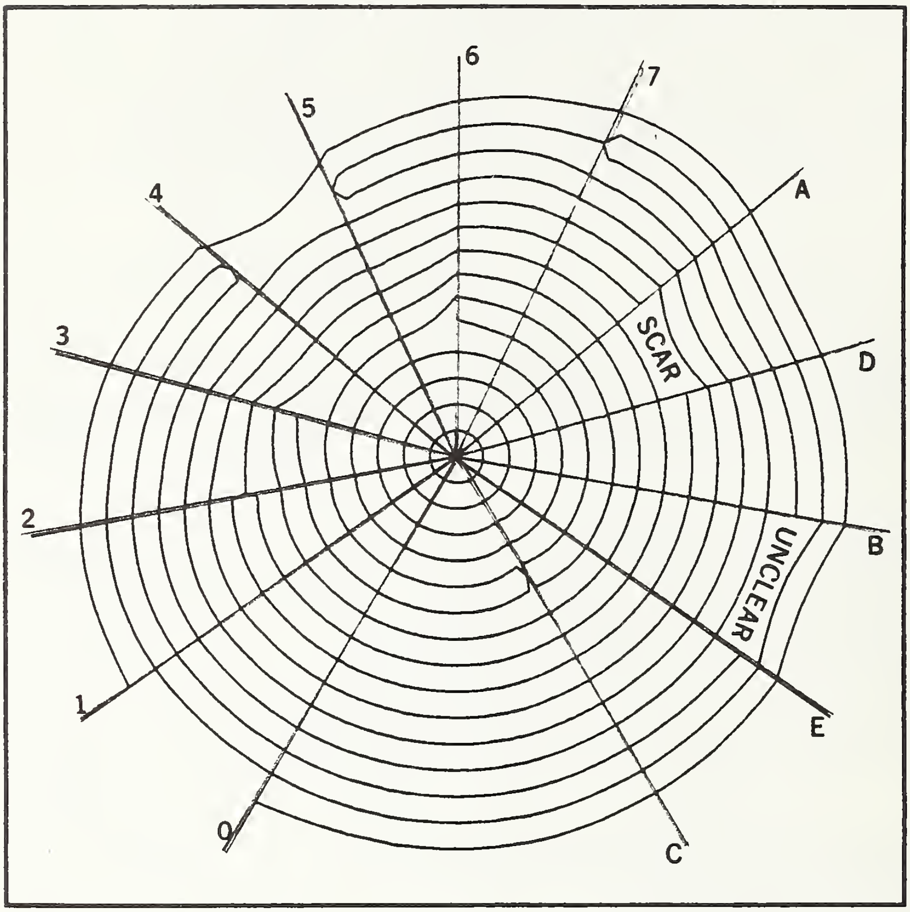


Figure 10. Irregularity types, and their codes.

2.4 The algorithm "LATENT-MATCHER 1" (or "LM1").

The first algorithm tested was an interactive one, in the sense that one vector enquiry was entered at a time and immediately searched against a prepared file collection database. It enabled experiments to be done quickly and easily to find suitable values for the many program parameters and to give an idea of what sort of *latent enquiry vectors* worked, and which ones did not.

Several valuable lessons were learnt from its use :—

(a) It rapidly became clear that entry of a *single* latent enquiry vector was a most unsatisfactory way of doing latent enquiries. Frequently the central feature upon which the vector was centered was spurious (i.e. it did not exist on the file-print, and had appeared on the latent tracing as a product of misinterpretation of the latent mark) and so no characteristic-centered vector even remotely similar could be extracted from the mate file-print data. It was found to be much more reliable to enter two or three latent vectors per latent mark, each centered on a different characteristic, and to combine their individual scores in formulating an overall score for the latent mark's comparison with each file-print.

(b) Inferred distance measures (see para 2.3) were unreliable, and demanded that distance tolerances in the vector comparison stages be set much wider than was desirable. Their use helped very little in aiding discrimination between mates and non-mates.

(c) A 180° span for the file-prints (i.e. coding the upper half only) was inadequate. There were several cases where the information available from the latent fell largely *outside* that sector, and the latent could not be identified by the fragment of information that lay *within* the sector. (Nevertheless, in the vast majority of latent marks all, or most, of the useful information lay within the sector, and usually towards the tip of the finger.)

2.5 Improved latent-matching algorithms.("LM2", "LM3" and "LM4")

In the light of these difficulties the following alterations were made to the algorithm LM1.

(a) To cover those cases where the latent mark was comparatively low on the finger, it was made permissible to enter an *approximated boundary vector*, rather than a *characteristic-centered vector*, as a latent enquiry vector. An *approximated boundary vector* was generated from a line placed at what appeared to be a horizontal orientation on the latent, and which did not need to be centered on any visible characteristic. The comparison algorithm would then recognize this vector as such, and compare it to the file-print boundary vectors rather than comparing it with any extracted characteristic-centered vectors.

(b) Facility was built into algorithms LM2, LM3 and LM4 for several latent enquiry vectors to be entered per latent mark. Each vector would then be first treated in isolation, and the best matching vector score from the file-print obtained. LM2 then simply added up the individual scores to give a combined score for the latent mark. LM3 and LM4 added the slight sophistication of combining the individual latent vector scores if, and only if, their relative angular orientation was matched (within specified angular tolerance) by the relative angular orientation of the file-print characteristics upon which the high scoring extracted vectors were centered. That procedure tended to prevent the combination of 'fluke' scores from non-mates.

(c) Distance tolerances were treated *linearly* (i.e. greater tolerance was allowed for greater distances) rather than *absolutely*.

(d) LM4 allowed a different set of distance tolerances to be used in vector comparisons involving *boundary vectors* than those used in comparing *characteristic-centered vectors*. The boundary vectors always required greater distance tolerances due to the uncertainty in the positioning of their generating lines.

Each of these modifications appeared to improve performance somewhat — and it was time to get some idea of the overall discriminatory power of the algorithm.

2.6 Testing algorithm performance.

A collection of 56 latent marks (of varying quality) was provided by the FBI. All of these were interpreted and traced using the 'Graphic Pen'.[10] Latent enquiry vectors were extracted from each tracing using a degree of subjective judgement as to selection of *central features*, and the latent enquiries formed together into a single database. The mate file-prints (rolled impressions taken from standard FBI ten-print cards) of the 56 marks, together with 44 other randomly selected prints were all traced and coded according to the scheme already described (para 2.3) to give a database of 100 file-prints.

Batch tests were then run, in which each latent search enquiry was compared with each of the 100 file-prints, and a score obtained in each case. For each latent enquiry the file-prints were then ranked according to score, and the position of the mate in the list was noted (the *mate rank*). Performance was then measured by the percentage of mates that were ranked in first place (which will be called 'MR1'). Attention was also paid to the number of mates that were ranked in the top three places ('MR3') and in the top ten places ('MR10').

As performance for latent marks is clearly very much worse than it was for clear rolled impressions, it is unnecessary to use other more sophisticated performance measures. The indicators MR1, MR3 and MR10 provide an adequate picture of comparative performance — and will continue to do so until such time as MR1 exceeds 90% .

In order to get some feeling for what levels of performance are desirable, the same set of latent marks and the same set of 100 file-prints were encoded in the traditional coordinate form for use with spatial matching algorithms. Once again the Graphic pen was used, and the data entered from the same interpretative tracings as were used for extraction of the topological information. Thus the performance of spatial matching algorithms could be measured on *precisely* the same dataset. * The best performance by the M82 matcher (which is a spatial matching algorithm developed at the National Bureau of Standards and in use at the FBI)[11] gave the following rankings :—

MR1 — 26.8%
MR3 — 37.5%
MR10 — 48.2%

* Latent marks vary so greatly in quality that it is not possible to quote meaningful performance statistics without reference to a *specific* set of latent marks. In this case, not only is the same set of prints used, but the same *interpretation* of those prints was used for the testing of both the topological and spatial matching algorithms.

A series of tests was conducted, both with LM3 and with LM4, to try to tune the various algorithm parameters. Complete tables of the test results are given in appendices D and E. The best results achieved (by LM4 in test number 39) gave the rankings :—

MR1 — 58.93%
MR3 — 67.86%
MR10 — 83.93%

This clearly represents a fairly substantial improvement on the level of performance given by the spatial approach. Special significance can be given to the raising of MR1 from 26.8% to 58.93% as it is the mates ranked in first place that tend to have scores way clear of the field and they are the only ones which would be likely to be correctly identified irrespective of the size of the file collection. Those mates that do not come in top place in a collection of size one hundred are most unlikely to come even in the top *fifty* places if the file collection were of size one million.

2.7 Latent enquiry by vector: shortcomings.

Despite its fairly impressive performance there remained something inherently objectionable about the method of latent enquiry by manual extraction of vectors. The process of selecting central features on which to base the enquiry vectors was too *subjective*: success or failure of any particular vector enquiry depended very heavily on the reliability of its central feature — and vectors based on spurious latent characteristics (those that arose from false interpretation of the mark) invariably scored abysmally against the mate file-print.

An analysis of the 23 latents (out of 56) that had their mates ranked in a position other than first (in test no. 39 on LM4) revealed the following :—

(a) in three cases — the central feature selected was spurious.

(b) in two cases — the central feature was in an unclear portion of the file print and so *apparently* did not exist.

(c) in two cases — an unclear area of the file print lay close to the central feature chosen, thus reducing vector comparison scores dramatically.

(d) in three cases — the central feature selected on the latent corresponded to a feature below the boundary lines on the mate file-print, and thus could not be correctly matched.

In at least 10 cases out of 23, therefore, the failure was directly attributable to *unlucky* (or *unwise*) central feature selections. In all of these ten cases there were other characteristics visible on the latent which would have served much better as centers for topological coding.

The sensible deduction from such observations is that it is unwise to base a complete latent search on a small number of extracted vectors. Presumably the greater the number of vectors entered, the greater the chances are of limiting the effects of unlucky central feature selection. The *ideal* policy might well be to enter *every possible* characteristic-centered vector that can be obtained from the mark; that means one vector per visible characteristic. The obvious difficulty with that proposal is the resulting complexity and tedium of the manual data extraction process.

The next step forward now becomes very clear: we must enter latent enquiry data in the highly economical *topological coordinate* form, and allow the comparison algorithm to do all the work involved in extracting the required vectors. The treatment of the latent mark data will then be virtually identical to the treatment already being given to the file-print data. Topological reconstruction of both prints (latent and file-print) becomes the essential preliminary for comparison based on characteristic-centered vectors.

CHAPTER 3.

LATENT SEARCHING: TOPOLOGICAL COORDINATE SYSTEMS.

3.1 Latent entry by topological coordinates.

The problems caused by unfortunate choice of central feature have shown the need for latent enquiry data to be less *selective* and less *subjective.* The most desirable latent data form is therefore a *complete* and *objective* description of the latent tracing. The tracing process itself still is, and always will be, substantially subjective — but it ought to be the last stage requiring subjective judgement. A set of topological coordinates of the form (T, θ, R), (showing type, angular orientation and ridge-count) provides a *complete topological* description, and it therefore becomes the basis for latent data entry. The latent mark data can then be presented in much the same form as the file print data.

The manual latent data preparation process is fairly simple: first the mark is traced (enlarged to 10× magnification). Then the position of the central observation point is *guessed* by the fingerprint expert, and its position marked on the tracing. The guessed core point position may be some way away from the 'visible' part of the latent. Then the correct orientation of the mark is estimated by the expert, and the coordinates of the characteristics, and other irregularities can then be written down.*

There are a number of very major changes in the use of topological coordinates that have to be made in order to enhance their versatility and usefulness. These changes are described in the following three sections.

3.1.1 The 4th coordinate.

Bearing in mind the unreliable nature of inferred distance measures (see para 2.4.b), and bearing in mind also that the topological coordinate scheme already records

* An extremely useful tool, for this operation, is a large board with a pin hole at its center. Around the circumference of a 7 or 8 inch circle the angular divisions are marked (i.e. much like an oversized 360° protractor). A transparent ruler is then pivoted at the pinhole in the center. When the tracing has been made it is placed over the board, the pivot pin pressed through the guessed central observation point. The tracing falls entirely *inside* the protractor markings, and the ruler is long enough to reach those markings. Radial movement of the transparent ruler (which has one central line on it) over the tracing makes it very easy both to count the ridge-counts for each irregularity, to measure radial distances (these are marked on the ruler in the appropriate units), and to read off the angular orientations from the circumference of the inscribed circle.

angular orientation of each characteristic, it would seem to be a very sound investment to include a 4th coordinate — namely a radial distance (D) measured from the central observation point. The combination of angular position and radial distance (θ, D) for each characteristic gives a complete *spatial* description of the positions of the characteristics in space. A set of coordinates of the form (T, θ, R, D) therefore gives a *complete topological and spatial* description of a print. It records everything that a comparison algorithm might need to know about the positions of the characteristics and their topological relationships to each other. The data storage requirement for such a description is a mere 4 bytes per irregularity.

We shall record radial distances in units of 0.5mm (or 0.5cm on the 10× enlargement) and round to the nearest integer. No greater accuracy is either required or useful. These distances then appear as integers in the range 0 to 50.**

3.1.2 Dispensing with boundary vectors.

Whatever the sector chosen for description by coordinates the *boundary vectors* can be made *null* by pretending that all the ridges inside the sector go 'out of sight' just before they reach the boundary lines. Thus the boundary lines cross *no ridges* and are therefore empty. The imaginary appearance of each ridge just inside the sector can then be recorded by coordinates. The resulting data is now pleasantly uniform and easier to handle. Boundary vectors, in the earlier algorithms, had been something of a nuisance.

3.1.3 'Wrap around' 360° sector.

The sector to be recorded can be enlarged at will by moving the radial boundary lines, until such time as the internal angle reaches 360°. At that stage the two boundary lines coincide and where they coincide will be called the *cut.* Provided our topological reconstruction algorithm can cope with the fact that, at the *cut*, some ridges effectively leave one end of the sector and reappear at the opposite end, then we can forget about boundary lines and boundary vectors altogether.

The reconstruction algorithm will need to be told *how many* ridges need to be connected up in this way — and that number (which is the number of ridges that cross the *cut*) will be recorded as a part of the fingerprint data. It is convenient to specify that the *cut* will be vertically below the central observation point, and that the ridges which cross it be called *moles* (as they pass *underneath* the observation point).

** The type code (T) is a hexadecimal integer, the angular orientation (θ) an integer in the range 0 – 360, and the ridge count (R) an integer in the range 0 to 50. The total storage space required for all four coordinates is, in fact, closer to 3 bytes; to be precise, it is 25 bits.

The coordinate system can now be used to describe the complete topology of a *whole fingerprint.*

3.2 Topological reconstruction from coordinate sets.

It is time to reveal the mysteries of *topological reconstruction* from a set of coordinates of the form (T, θ, R, D). The method to be described here is certainly not the only way it could be done — but this one does work very well, is probably as fast as any could be, and leads *directly* to the point at which no further work is required to be done in order to extract characteristic-centered vectors from the reconstruction. In fact all the characteristic-centered code vectors can be simply lifted out of the array formed by this method.

It will be noticed that the fourth coordinate (D) is ignored throughout this section as it plays no part in the reconstruction process. It is used in the comparison algorithms only after the topology has been restored.

Let us suppose that the print to be reconstructed has m moles and n topological irregularities, whose coordinates are the set $\{(T_i, \theta_i, R_i, D_i) : i = 1, \ldots n\}$.

3.2.1 The 'continuity' array.

This reconstruction method involves the systematic development of a large 3-dimensional array, which will be called the 'continuity' array (C) comprising elements $c(i, j, k)$. To understand the function of this array it is necessary, first, to examine figure 11: it shows a (simplified) fingerprint pattern with selected central observation point and the radial *cut* vertically downwards. A radial line from the central observation point is drawn marginally to the clockwise side of every topological irregularity in the picture (whether it be a true characteristic or not). If there are n irregularities (which we will call $\{I_1, \ldots I_n\}$, then there are $n + 1$ radial lines in total (this includes the *cut*). Calling the *cut* line l_0, and numbering the lines consecutively in a clockwise direction gives the set of lines $\{l_0, l_1, \ldots l_n\}$.

Now re-order the topological coordinate set by reference to the second coordinate (θ) — so that the coordinate set satisfies the condition :—

$$\theta_i \leq \theta_{i+1} \text{ for all } i \in \{1, 2, \ldots n - 1\}$$

There are then simple 1–1 mappings between the lines $\{l_1, \ldots l_n\}$, the irregularities $\{I_1, \ldots I_n\}$ and their coordinates $\{(T_i, \theta_i, R_i, D_i) : i = 1, \ldots n\}$.

Each of the lines $\{l_0, \ldots l_n\}$ intersect a certain number of ridges, giving an ordered sequence of *ridge intersection points.* Let the number of ridges crossed by line l_i be called r_i.

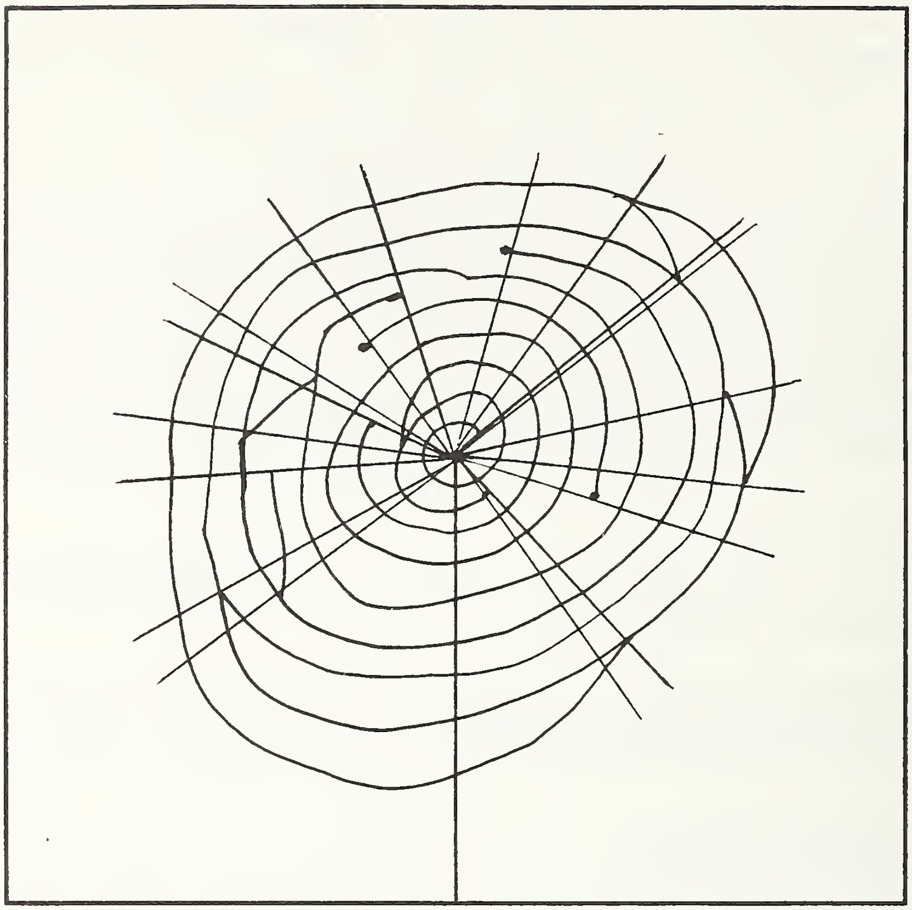


Figure 11. Radial irregularity-centered lines, with the 'cut' vertically below observation point.

Further, let the ridge intersection points on the line l_i be called points $\{p(i,j) : j = 1, \ldots r_i\}$ — point $p(i,1)$ being the closest to the central observation point and $p(i,r_i)$ being the closest to the edge of the visible print.

The continuity array C is then set up with a direct correspondence between the ridge intersection points $p(i,j)$ and the elements of C, namely $c(i,j,k)$. k takes the values 1 to 4, and thus there is a 4 to 1 mapping of the elements

$$\{c(i,j,k) : i = 0, \ldots n : j = 1, \ldots r_i : k = 1,2,3,4\}$$

onto the set of ridge intersection points

$$\{p(i,j) : i = 0, \ldots n : j = 1, \ldots r_i\}$$

The array C can therefore be used to record four separate pieces of information about each of the ridge intersection points.** The meanings assigned to each element of C are as follows :—

** The part of the matrix C which will be used for any one print is therefore irregular in its second (j) dimension.

$c(i,j,1)$ — "what is the first *event* that topological exploration from the point $p(i,j)$ in an *counterclockwise* direction will discover?"

$c(i,j,2)$ — "*which* of the irregularities $I_1, \ldots I_n$ is it that such counterclockwise exploration will discover first?"

$c(i,j,3)$ — "what is the first *event* that topological exploration from the point $p(i,j)$ in a *clockwise* direction will discover?"

$c(i,j,4)$ — "*which* of the irregularities $I_1, \ldots I_n$ is it that such clockwise exploration will discover first?"

$c(i,j,1)$ and $c(i,j,3)$ should, therefore, be ridge-tracing event codes in the normal hexadecimal integer format (not to be confused with the different set of hexadecimal codes currently being used for the irregularity type (T_i)).

$c(i,j,2)$ and $c(i,j,4)$ are integers in the range 1—n which serve as *pointers* to one of the coordinate sets. They are a kind of substitute for distance measures (being associated with $c(i,j,1)$ and $c(i,j,3)$ respectively) but they act by referring to the coordinates of the irregularity found, rather than by giving an actual distance. They will be called *irregularity indicators* in the following few sections.

3.2.2 Opening the continuity array.

To begin with, the whole of the continuity array is empty (and, in practice, all the elements are set to –1). It will be filled out successively starting from the left hand edge ($i = 0$) and working across to the right hand edge ($i = n$).

Starting with $i = 0$ (at the *cut*) we know only that $r_0 = m$ (the number of ridges crossing the *cut* is the number of *moles* recorded in the data.) Nothing is known (yet) about any of these ridges. The first set of entries in the continuity array is made by assigning a *dummy number* to every possible ridge exploration from the line l_0.

The *dummy numbers* are integers in a range which cannot be confused with real *event-codes.** Each dummy number assigned is different, and the reconstruction algorithm views them thus :

"I do not yet know what happens along this ridge — I will find out later — meanwhile I need to be able to follow the path of this ridge segment, even before I find out where it ends."

---

* In practice dummy numbers start at 100 and, whenever another one is needed, the next free integer above 100 is used. Obviously a record is kept of how many different dummy numbers have been assigned.

This first step in filling in the continuity matrix is therefore to assign *dummy numbers* to each of the elements $\{c(0, j, k) : j = 1, \ldots r_0 : k = 1 \text{ or } 3\}$.

The elements $\{c(0, j, k) : j = 1, \ldots r_0 : k = 2 \text{ or } 4\}$ are left untouched for now.

3.2.3 Associations, entries, and discoveries in the continuity array.

The next stage is to consider each of the coordinate sets $(T_i, \theta_i, R_i, D_i)$ in turn starting with $i = 1$. We know that the irregularity I_1 is the *only* change in the laminar flow between lines l_0 and l_1. We also know its *type* (T_1) and its *ridge-count* (R_1). Depending on the type T_1 there are various *associations, entries* and *discoveries* that can be made in the continuity array.

Suppose, for example, that $T_1 = 3$ (i.e. a ridge ends — according to the table of irregularity types, para 2.3). We can deduce that

$$r_1 = r_0 - 1$$

(i.e. line l_1 crosses one less ridge than line l_0), and we can make the following *associations* in the second column $(i = 1)$ of the continuity array. (*Associations* occur when one element of the array is set equal to another.)

$$\begin{aligned} c(1, j, 1) &= c(0, j, 1) \quad \text{for all } 1 \leq j \leq R_1 - 1, \\ c(1, j, 3) &= c(0, j, 3) \quad \text{for all } 1 \leq j \leq R_1. \end{aligned}$$

(i.e. ridges below the irregularity pass on unchanged) also

$$\begin{aligned} c(1, j, 1) &= c(0, j + 1, 1) \quad \text{for all } R_1 + 2 \leq j \leq r_1, \\ c(1, j, 3) &= c(0, j + 1, 3) \quad \text{for all } R_1 + 1 \leq j \leq r_1. \end{aligned}$$

(i.e. ridges above the irregularity pass on unchanged, but are displaced downwards by one ridge, due to the $R_1 + 1$'th ridge coming to an end.)

Thus many of the *dummy numbers* from the $(i = 0)$ column are copied into the $(i = 1)$ column — and their successive positions show *which ridge intersection points lie on the same ridges.*

Further information is gained from the immediate vicinity of the irregularity and this allows us to make *entries* in the array. (*Entries* result directly from the coordinate set being processed, rather than by copying from another part of the array).

$$\begin{aligned} c(1, R_1, 1) &= 8, \\ c(1, R_1, 2) &= 1, \\ c(1, R_1 + 1, 1) &= 6, \\ c(1, R_1 + 1, 2) &= 1. \end{aligned}$$

(i.e. the line l_1 is drawn marginally past the ridge-ending I_1, and so that ridge-ending appears as a facing ridge ending in counterclockwise exploration from ridge intersection points $p(1, R_1)$ and $p(1, R_1 + 1)$. The event seen, in each case, is I_1 itself.)

We also have *discovered* what happened to the ridge that passed through the point $p(0, R_1 + 1)$: it ended (code 3) at irregularity I_1. That *discovery* enables us to note the fact that the ridge exploration clockwise through point $p(0, R_1 + 1)$ ended here. The existing entry in $c(0, R_1 + 1, 3)$ is a *dummy number*, and the new found meaning for that number is recorded in the *dummy number index*. Suppose the dummy entry had been the number 107: then we store its meaning thus:

$$\text{index}(107) = (3,1)$$

Eventually all the appearances of the number 107 in the array will be replaced by '3', and, at the same time, all the associated *irregularity indicators* will be set to '1'.

Knowledge of T_1 and R_1 has therefore enabled us to make a particular set of *associations, entries* and *discoveries* — from which it has been possible to place something (either entries or dummy numbers) in all of the elements of the set

$$\{c(1, j, k) : j = 1, 2, \ldots r_1 : k = 1 \text{ or } 3\}$$

The process now begins again, with examination of irregularity I_2, followed by $I_3 \ldots I_n$. Each different possible *type* code T_i generates its own individual set of associations, entries and discoveries. Each set allows the next column of C to be filled in. ** It should be pointed out that whenever *association* is made of event codes (as distinct from dummy numbers) then association is also made of their respective *irregularity identifiers*.

After all the n coordinate sets have been processed (and entries thereby made in the whole of the continuity array) a few last associations need to be made in order to account for the fact that ridges cross the *cut*. These associations are that :—

$$\begin{array}{rl} c(0, j, 1) & \text{is equivalent to } c(n, j, 1) \text{ for all } 1 \leq j \leq r_0, \\ \text{and } c(n, j, 3) & \text{is equivalent to } c(0, j, 3) \text{ for all } 1 \leq j \leq r_0. \\ & \text{(Of course } r_0 = r_n = m) \end{array}$$

which effectively 'wrap around' the ends of the continuity array by sewing up the *cut*. As each of these elements of C already has some sort of entry in it, the mechanics of making these *associations* are more akin to the normal mechanics of *discovery*, in that they involve making entries in the *dummy number index*. They may, in fact, enter *dummy numbers* in the *dummy number index* thus indicating that two different dummy numbers are equivalent (i.e. they represent the same ridge exploration).

** Some of the *entries* may well be new (unassigned) dummy numbers. This occurs wherever new ridge segments start at the irregularity. It did not happen in the case of the ridge ending.

3.2.4 Properties of the completed continuity array.

Once this process is complete the continuity array will have acquired some very important properties:

(a) all the elements $\{c(i,j,k) : 0 \le i \le n : 1 \le j \le r_i : k = 1 \text{ or } 3\}$ contain either ridge exploration event codes (hexadecimal) or *dummy numbers* (integers over 100).

(b) wherever $c(i,j,1)$ or $c(i,j,3)$ is an event code, then the corresponding entries, $c(i,j,2)$ or $c(i,j,4)$ respectively, will contain an *irregularity identifying* number that shows *where* that ridge event occurs.

(c) all the different appearances of a particular dummy number in the continuity array reveal all the intersection points through which one continuous ridge exploration has passed. (Hence the name for the array.)

(d) a *discovery* has been made in respect of *every* dummy number that has been allocated, and there is, in the dummy number index, an equivalent event code and associated irregularity identifier waiting to be substituted for all the appearances of that dummy number. The dummy number index is therefore *complete.* This simply *must* be the case as a *discovery* has been recorded every time that a ridge ran into an irregularity. There can be no ridge explorations that do not end at one, or other, of the n irregularities — consequently there can be no outstanding 'unsolved' ridge explorations by the time all n irregularities have been dealt with.

3.2.5 Final stage of topological reconstruction.

The final stage of the reconstruction process is to sweep right through the continuity matrix replacing all the dummy numbers with their corresponding event codes from the index. The related *irregularity identifiers* are filled in at the same time, also from information held in the index. This second (and final) sweep through the elements of the continuity array leaves every element in the set

$$\{c(i,j,k) : i = 1 \ldots n : j = 1 \ldots r_i : k = 1 \text{ or } 3\}$$

as an event code, and every element of the set

$$\{c(i,j,k) : i = 1 \ldots n : j = 1 \ldots r_i : k = 2 \text{ or } 4\}$$

as an irregularity identifier.

For any particular line l_i the entries of C in the ith column correspond exactly to the elements of a topological code vector generated by that line. The only difference in appearance is that we have *irregularity identifiers* rather than *distance measures* to go with each exploration event code. The later vector comparison stages of the matching algorithm are adapted with that slight change in mind.

This completes a somewhat simplified account of a rather complex process. There are other complications which have not been explained in full — such as how the algorithm deals with sequences of dummy numbers that are all found to be equivalent, and the special treatment that ridge recurves have to receive, and how the algorithm copes with multiple irregularities showing the same angular orientation. Nevertheless this explanation serves well to demonstrate the methodical and progressive nature of this particular reconstruction process. It also makes clear that only two sweeps through the matrix are required — which is surprisingly economical considering the complexity of the operation.

3.3 The matching algorithm LM5.

The algorithm LM5 was the first to accept latent data in coordinate form, rather than by prepared vectors. Topological reconstruction was performed both on the latent mark (once only per search) and on each file print to be compared with it. The continuity matrix generated from the latent coordinate set will be called the *search continuity array*, and the continuity array generated from the file set will be the *file continuity array.*

There are two distinct phases of print comparison which take place after these topological reconstructions are complete. Firstly, the appropriate vector comparisons are performed and their scores recorded — secondly, the resulting scores are combined to give an overall total comparison score.

It is most important to realize that the *observation points* selected on two mated prints under comparison do not need to have been in the same positions. The reconstructed topology will be the same *no matter where it is viewed from.* Just as two photographs of a house, from different places, look quite different — nevertheless the house is the same. The final comparison scores will be hardly affected by misplacement of the central observation points *provided they lie in roughly the right region of the print.* The reason for *approximately* correct placement being necessary is that the orientation of the imaginary radial lines, which effectively generate the vectors after reconstruction, will depend on the position of the central observation point. The effect of misplacing that point (in a comparison of mates) is to rotate each generating line about the characteristic on which it is based. Such rotation is not important provided it does not exceed 20 or 30 degrees. Slight misplacement of the observation point is not going to materially affect the orientation of these imaginary generating lines, except those based on characteristics which are very close to it. Specifying that the central observation point should be adjacent to the core (in the case of whorls or loops) and at the base of the 'upcurve' (in the case of plain arches) is a sufficiently accurate placement rule.

3.4 The vector comparison stage.

From the *search continuity array* a vector is extracted for each true characteristic on the latent mark. Vectors are *not* extracted for the other irregularities ('ridges going out of sight', 'ridge recurves', etc.) If the latent mark shows 13 characteristics we then have 13 vectors, each vector based on an imaginary line drawn from the central observation point to one of those 13 characteristics, and passing marginally to the clockwise side of it. Let us now forget about all the other topological irregularities in the coordinate list and number the characteristics $1, 2, 3, \ldots k$. If the number of coordinate sets, in total, was n then certainly $k \leq n$. The extracted search vectors can now be called $S_1 \ldots S_k$. In a similar fashion the extracted file vectors, each based on true characteristics, can be called $F_1 \ldots F_m$.

For each search vector a subset of the file vectors is chosen for comparison. The selection is made on these bases :—

(a) that the characteristic on which the file vector is based must be of similar *type* (either an 'exact' match or a 'close' match) to the one on which the search vector is based.

(b) that the angular coordinates of the characteristic on which it is based must be within a permissible angular tolerance of the angular coordinate of the characteristic on which the search vector is based. The permissible angular tolerance is a parameter of the algorithm.

This selection essentially looks for file print characteristics that are potential *mates* for the search print characteristics. The vector comparison that follows serves to compare their neighborhoods. It is quite obvious that allowing a wide angular tolerance significantly increases the number of vector comparisons that have to be performed. If a small angular tolerance is permitted then a badly misoriented latent mark may not have the mated vectors compared at all.

The vector comparison itself is much the same as used hitherto — except that the vectors contain *irregularity identifiers* rather than *distance measures.* At the appropriate stages of the vector comparison subroutine the *actual linear distance* ('as the crow flies') from the central characteristic to the ridge-event is calculated by reference to the appropriate coordinate sets. Thus ordinary *spatial* distances can be used rather than *inferred ridge-traced* distances, and a much greater degree of reliability can therefore be attached to them.

For each search vector S_i, and candidate file vector F_j, a vector comparison score q_{ij} is obtained. For each search vector S_i a list of candidate file vectors, with their scores, can be recorded in the form of a list of pairs (j, q_{ij}). There are typically between 5 and 15 such candidates for each search vector when the angular tolerance is set at 30°. These lists of candidates can then be collected together to form a table, which will be called the *candidate minutia table.* An example of such is shown on the next page.

Each column is a list of candidates for the search vector labelled at the head of the column. In each case the first of a pair of numbers in parentheses shows *which* file vector was a candidate, and the second number is the score obtained by its vector comparison.

S_1	S_2	S_3	$\cdots\ \cdots\ \cdots$	S_k
(5,89)	(6,45)	(25,41)	$\cdots\ \cdots\ \cdots$	(15,138)
(14,29)	(10,40)	(34,12)	$\cdots\ \cdots\ \cdots$	(23,12)
(15,0)	(16,35)	(37,19)	$\cdots\ \cdots\ \cdots$	(28,65)
(52,19)	(21,92)	(41,84)	$\cdots\ \cdots\ \cdots$	(36,71)
(55,81)	(35,5)	(48,91)	$\cdots\ \cdots\ \cdots$	(37,103)
(61,34)	(36,0)	(53,101)	$\cdots\ \cdots\ \cdots$	(47,82)
(79,0)	(41,3)	(65,180)	$\cdots\ \cdots\ \cdots$	(56,41)
.	.	.	$\cdots\ \cdots\ \cdots$	.
.	.	.	$\cdots\ \cdots\ \cdots$	.
.	.	.	$\cdots\ \cdots\ \cdots$	.
.	.	.	$\cdots\ \cdots\ \cdots$	.
(0,0)	(46,85)	(0,0)	$\cdots\ \cdots\ \cdots$	(0,0)

3.5 Final score formulation.

We are now left with the problem of intelligently combining these individual candidate scores to give one overall score for the print. If the file print and latent mark are mates it would be nice to think that the highest candidate score in each column of the candidate minutia table indicated the correct matching characteristic on the file print. If that were the case then simply picking out the highest in each column, and adding them together, might serve well as a method of formulating an overall score. However that is *not* the case. Roughly 50% of true mated characteristics manage to come top (in score) of their column — the others usually come somewhere in the top five places.

3.5.1 The notion of 'compatibility'.

We learnt from earlier experiments with latent entry by vectors that combination of scores was best done subject to *conditions* — and, in that case, the condition was correct relative angular orientation (see para 2.5(b)). It will make sense, therefore, to combine the individual candidate scores when, and only when, they are *compatible*.

If (j, q_{1j}) is a candidate in the S_1 column, and (i, q_{2i}) is a candidate in the S_2 column — then there are various reasonable conditions that can be set in respect of these two candidates before we accept that they could *both* be correct. We will say that these two candidates are *compatible* if, and only if, these three conditions hold true :—

(a) i is not equal to j . (Obviously *one* file print characteristic cannot simultaneously be correctly matched to two different search print characteristics.)

(b) The distance (linear) between file print characteristics numbered i and j should be the same, within certain tolerance, as the distance between the two search print characteristics that they purport to match. That tolerance is an important program parameter.

(c) The relative angular orientation of the file print characteristics should be roughly the same as the relative angular orientation of the two search print minutiae that they purport to match. The tolerance allowed, in this instance, is the same angular tolerance that was used earlier to limit the initial field of candidate minutiae.

3.5.2 Score combination based on compatibility.

The application of the notion of compatibility in formulating a *total* score was originally planned as follows :—

Step 1: Reorder the candidates in each column by reference to their scores, putting the highest score in each column in top place.

Step 2: In each column, discard all the candidates that do not come in the top five places.

Step 3: For each remaining candidate check to see which candidates *in the other columns* are compatible with it.

Step 4: Taking at most one candidate from each column, pick out the *highest scoring mutually compatible set* that can be found. A *mutually compatible set* is a set of candidates each pair of which are compatible.

Thus a set of file print characteristics is found, each of which has similar topological neighborhood to one of the latent mark characteristics (as shown by their high vector comparison scores) and whose *spatial distribution* is very similar to that of the latent mark characteristics (as shown by their *compatibility*). Spatial considerations are therefore being used in the combination of topological scores — as is already the case at a lower level, when distance measures are used in the vector comparison process.

The algorithm LM5 was originally written to perform the steps described above. Unfortunately it ground to a halt completely when it tried to do the comparison of a *very good latent* with its mate! The reason for this is that the algorithm will examine every possible mutually compatible set in turn. Certainly non-mates have very few mutually compatible sets of any size. However if a good quality latent gives a largest compatible set of size N (i.e. N characteristics match up well with the file print) then there are $2^N - 1$ subsets of that largest set, each of which will be a mutually compatible set. The total number of such sets is therefore *at least* 2^N, and probably much greater. In some cases N exceeded 25 and, consequently, the computer did not finish the job!

Acceptable shortcuts, or approximations, to this method had to be found.

3.5.3 Candidate promotion schemes.

The following method accomplishes much the same sort of candidate selection, but very much faster, and without requiring complete mutual compatibility in the selected set. The first three steps are the same as before :—

1. Reorder the candidates in each column, by their scores.

2. Discard all candidates not ranked in the top 5 places in their column.

3. Check the compatibility of all remaining candidates with the remaining candidates in each other column.

The fourth step is calculation of what will be called a *compatible score* for each of the remaining candidates. Here are two possible alternative methods for doing this :—

(a) For each individual candidate add together all the scores of top-ranked candidates in *other columns* with which that candidate is compatible. Finally add the candidate's *own* score to the total.

(b) For each individual candidate find, in each other column, the highest scoring compatible candidate. Add together those scores (one from each column), and then add the target candidate's *own* score to the total.

On the basis of these *compatible scores*, rather than on the original vector comparison scores, reorder the remaining candidates in each column.

This 4th step can be regarded as a promotion system based on compatibility with other high-ranking candidates. The difference between options (a) and (b) is this: in rule (a) promotion depends on a candidate's compatibility with those *already in top place* (and could be called a 'bureaucratic' promotion system). With rule (b) a whole group of candidates in different columns, *none of whom are in top place* can all be promoted to the top at once by virtue of their strong compatibility with *each other* (a 'revolutionary' promotion system). Both were tried and the 'revolutionary' system was found to be the most effective.

The *promotion stage* could be repeated several times if it was considered desirable (to give the top set time to 'settle') — in practice it was found that one application was sufficient. Mate scores improved very little, if at all, when second and third stages of promotion were introduced.

After the promotion stage is complete all but the top ranked candidates in each column are discarded, and the *compatible score* for the remaining candidate in each column is then recalculated on the basis of only the other remaining candidates.

The final score is then evaluated by adding together all of these new *compatible* *scores* that exceed a given threshold. That threshold is a program parameter, and is expressed as a percentage of the 'perfect' latent self-mated score.

The use of these *compatible scores*, rather than the original vector comparison scores, in evaluating the final score has the effect of multiplying each original vector score by the number of other selected (i.e. now top-ranked) candidates with which it is compatible. The more *dense* the compatabilities of the final candidate selection, the higher the score will be.

3.6 Performance of LM5.

The latent mark data file was converted to the form of coordinate sets, and the fourth coordinate (distance) was added into the file print collection data set. A series of tests was then performed using the algorithm LM5 — and the results and parameters used are shown in full in appendix F.

The best test results obtained gave the following rankings :—

MR1	80.36%
MR3	82.14%
MR10	85.71%

These indicate a *vast* improvement over the performance of the traditional spatial methods (recall that the M82 algorithm gave test results with an MR1 value of 26.8%).

It is worth saying a few words about some of the parameter values that gave the above results : —

(a) Exact match scores were set to be 5, with close match scores (CMS) set to be 3. Thus close match scores were given a higher relative weighting than previously used in the comparison of rolled impressions (where the optimum ratio had been 5:1 [12]). The higher weighting can be attributed to a higher incidence of *topological mutation* in the interpretation of latent marks.

(b) The distance tolerances were set at 10% (of the distance being checked) with a minimum of 1. (PDT, in appendix F, stands for 'percentage distance tolerance', and MDT for 'minimum distance tolerance'.) The same distance tolerances were used in the vector comparison stage of the algorithm and in the score combination stages (where correct relative distance was one of the three conditions that needed to be satisfied for two file print minutiae to be *compatible*.)

(c) The ridge span used in vector comparison was 10 ridges — this means that vectors of a standard length of 40 digits, with 40 associated irregularity indicators, were used whenever vector comparisons were performed. The results were no worse with longer vectors, but the smaller value for SPAN gave faster comparison times on a serial machine.

(d) The minimum angular tolerance (MAT) was 20°. This is almost inconsequential as the true angular misorientation limits were set individually for each latent mark (by subjective judgement) and written as a part of the latent search data.

(e) The *candidate minutia selection depth* ('DEPTH') was 5 throughout. This means that, for each search minutia, only the top 5 candidate file print minutia would be considered. This parameter was set to 5 as a result of observation, rather than experiment (see para 3.5)

(f) The *compatible score cutoff point* ('CUTOFF') is the percentage of the latent mark's perfect self-mated score that must be attained by the final compatible score of a candidate file print minutia before it will be allowed to contribute to the final total score (see para 3.5) The best value for this parameter was found to be 15%, which is surprisingly high. The effect of this setting was to ensure that the vast majority of file print minutiae that were not true mates for search minutia contributed nothing to the score; the net effect of this was to make *most* of the mismatch comparison scores zero. In fact, for 28.6% of the latents used, the true mate was the *only* file print to score at all — the other 99 file prints all scoring zero. Of course such a stringent setting also made things tough for the mates, as shown by the fact that 7% of the mate scores were zero also. However, these 7% were mates that had not made the top ten places in any of the tests, and were therefore most unlikely to be identified anyway. It is also worth pointing out that on each occasion when one file print alone scored more than zero (i.e. exactly 99 out of the 100 in the file collection scored zero) that one was the true mate. (These are the 28.6% mentioned above.) This represents a surprisingly high level of what might reasonably be termed 'cast iron doubt-free identifications'.

3.7 Computation times.

The foregoing description of the algorithm LM5 will have made it quite clear that this is not, in its present form, a particularly fast comparison algorithm. The CPU time taken (on a general purpose computer capable of performing in the order of a million instructions per second) for the above test (5600 comparisons) was 12 hour and 11 minutes. [Hence the absence of any extensive parameter tuning.] That means an average CPU time per comparison of 7.8 seconds — which is a somewhat disconcerting figure when the acceptable matching speeds for large collections are in the order of 500 comparisons per second.

However 7.8 seconds per comparison on this machine is not quite so alarming when one considers the extensive and multi-layered parallelism of the algorithm. At the lowest level, the vector comparisons themselves are sequences of array operations. At the next level, many vector comparisons are done per print comparison. In the score combination stages calculations of compatibility and compatible scores are all simple operations repeated many many times. There is, in this algorithm, *enormous* scope for beneficial employment of modern parallel processing techniques. It is hardly appropriate to take too much notice of the CPU time in a computer in which each operation is done element by element.

Moreover, in the area of latent searching, the primary area of concern for law enforcement agencies is shifting from the issue of *speed* onto the issue of *accuracy.* The FBI, for example, is certainly prepared to obtain the necessary speed through 'hardwiring' (with its associated cost) for the sake of matching algorithms that will actually make a substantial number of identifications from latent marks.

3.8 File storage space — defaulting the 'edge topology'.

It is noticeable that the need to include *all topological irregularities*, rather than just the true characteristics, significantly enlarges the volume of the file print data. In the 100 file cards in the experimental database the average number of irregularities recorded per print was 101.35. The majority of irregularities that were not true characteristics fell at the *edge* of the print; they recorded all those places where ridges 'came into sight' or 'went out of sight'. Thus a significant proportion of the file data storage requirement is spent in describing the edge of the file print.

In practice the edge of the file print is not very important — as the latent mark invariably shows an area completely *within* the area of the rolled file print. The edge consequently plays little or no part in the print comparison process, and the edge description serves only to help the topological reconstruction process make sense of the ridge pattern.

For the sake of economy in file size, therefore, the algorithm LM6 was prepared by adapting the reconstruction stage of LM5 slightly. It is adapted in such a way that the reconstruction will *invent its own edge topology* in the absence of an edge description. The *default topology* selected is not important; it is only important that the algorithm does *something* to tie up all the loose ridges around the edge.

The file collection was then pruned substantially by elimination of all of the edge descriptions, and this reduced the average number of coordinate sets per print from 101.35 to 71.35. * The test reported above was then rerun using the algorithm LM6 and the condensed file set. The rankings obtained were *exactly the same* as before (see para 3.6) — so a saving of 30% in file data storage was achieved with absolutely no loss of resolution.

---

* The pruning operation was not performed on the latent mark data file for two reasons. Firstly, latent mark databases (where these are kept) are tiny in comparison to rolled file print collections, and so storage requirements are not a major concern. Secondly, the edge of a latent mark does play an important part in the comparison process.

CHAPTER 4.

ASSOCIATED APPLICATIONS AND CONCLUSIONS.

4.1 Derivation of vectors for rolled print comparison.

The ability to perform topological reconstruction from a set of coordinates has some rather interesting 'by-products'. The first of these relates to the fast comparison of rolled prints on the basis of a single vector.[13]

As the data format for a latent mark and a rolled impression is now identical, it would be possible to use the latent matching algorithm (LM6) to compare one rolled print with another. (One of the rolled prints would be acting as a very high quality latent.) However, to use LM6 in this way on rolled prints would be 'taking a sledge hammer to crack a nut'. We know that one single vector comparison deals with comparison of two rolled prints perfectly adequately[14] — so it would be madness to use this latent matching algorithm, with its hundreds of vector comparisons, in this application.

Nevertheless there is a significant benefit to be gained from the topological reconstruction section of the latent matching algorithm. The data-gathering requirements from the scheme for matching rolled impressions included the need to *track along ridges, in order to find the first event that happened.*[15] Although that, in itself, is not a particularly demanding programming task — the ability to reconstruct topologies from coordinates renders it unnecessary. A topological code vector representing a *horizontal* line passing through the core of a loop can be lifted out of the continuity matrix after reconstruction. The left half of it (i.e. the part that falls to the left of the core) and the right half will be extracted separately. Each half is extracted by selecting the column of the continuity matrix that corresponds with an imaginary line *just to the counterclockwise side* of horizontal. (i.e. just *below* for the left side, and just *above* for the right side). Amalgamating these two halves, reversing the 'up' and 'down' pairs from the right half, gives a single long vector of the required format.

There will be two minor differences between these extracted vectors and the design originals :—

(a) the core point, which was to be on a ridge, is replaced by the *central observation point* which is in a valley. The central observation point will, however, be only fractionally removed from the core in the case of loops and whorls.

(b) the vector has *irregularity identifiers* rather than *ridge-traced distance measures.* Consequently the vector comparison algorithm has to be adapted to refer to the appropriate coordinate sets when the time comes to apply the various distance tests.

In the case of arches the extracted vector will have to be a *vertical, straight* line as opposed to the original flexible one which followed successive ridge summits.*

In an operational system the maximum speed would be obtained by performing topological reconstruction, and vector extraction, *at the time each print is introduced to the collection.* The extracted 'long' vectors could be stored in a separate file so that they could be used for fast vector comparison without the need to perform topological reconstruction each time. That would obviously increase the data storage requirement per print by the 60 bytes required for such 'long' vectors.[16] The coordinate sets, and topological reconstruction would then only be used when a *latent search* was being conducted.

If the derived long vectors were to be made completely independent of the coordinate sets, it would be necessary to replace the *irregularity identifiers* with calculated *linear distances* at the time of vector extraction.

4.2 Image-retrieval systems.

The second by-product of the development of the latent matching algorithms is an application in image-retrieval systems. There is a significant demand for automated identification systems to be linked with an image-retrieval facility for all the prints in the file collection. The system operator obtains a list of the highest scoring candidates each time an automated search is conducted — these candidates have then to be checked visually by the fingerprint expert to determine which of them, if any, is the true mate. This visual checking can be done much more easily if the fingerprints can be displayed on a screen, rather than having to be fetched from a cabinet. Much research is currently underway with the aim of finding *economical* methods for storing the two dimensional pictures (fingerprints) in computer memory so that they can be called up and displayed on the terminal screen.

There are two distinct paths for such research. The first aims to record the original grey-scale data which is output from automatic scanners, with no interpretative algorithms ever being applied to the print (although data compaction techniques will, of course, be used). The second uses interpretative algorithms to identify the ridges and valleys within the grey-scale image, to resolve the picture into a binary (black and white) image, and then finally to reduce the thickness of each ridge to one pixel by a variety of ridge-thinning techniques. What is then stored is sufficient data to enable each thinned ridge segment to be redrawn (i.e. start position, end position, curvature etc.).

---

* The performance of vector matching algorithms on such derived vectors has not been tested. This is because of the incredibly time consuming nature of manual encoding according to the latent scheme (up to 1 hour per print for clear rolled impressions). The time for such tests will be after the development of automatic data extraction techniques, when large numbers of prints can be encoded automatically according to the latent scheme, and then have derived vectors extracted after topological reconstruction.

The data requirements per print are in the order of 2,000 to 4,000 bytes for compressed grey-scale images, and between 1,000 and 2,000 bytes for a thinned image.

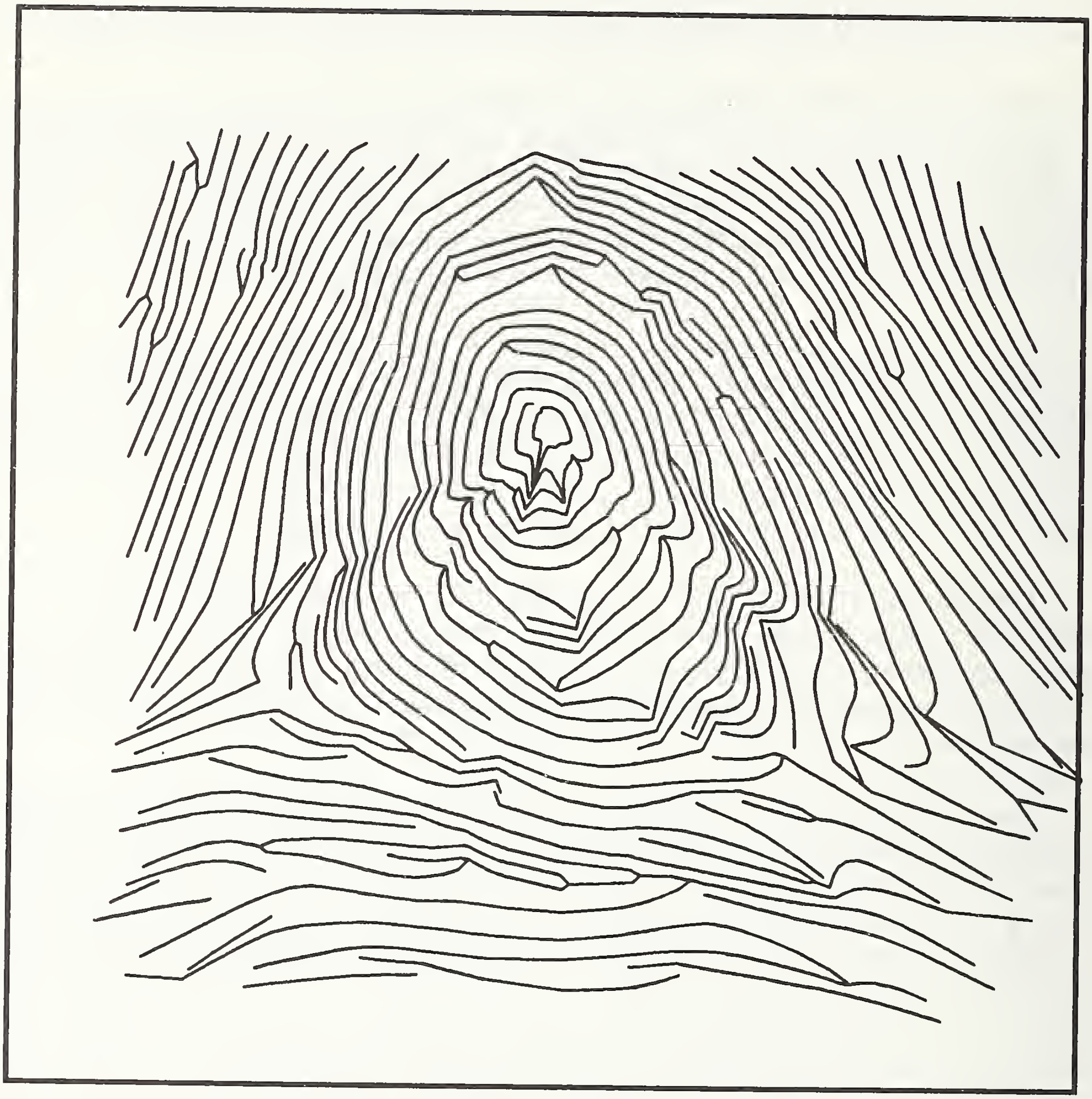
We know that the 4-coordinate system used in the latent scheme records, in between 300 and 400 bytes, a *complete topological and spatial* description of the characteristics. It should therefore be possible to *redraw* the fingerprint, in the style of a thinned image, from that data. Firstly topological reconstruction has to be performed, and then the *elastic* (topological) image has to be 'pinned down' at each characteristic, by reference to their polar coordinate positions contained in the coordinate sets.

The substantial problem in such a process is the business of generating a *smooth* ridge pattern that accommodates all the pinned points. The problems raised are not completely dissimilar to those in cartography — when a smooth contour map has to be drawn from a finite grid of discrete height (or depth) samplings.[17] [18] Certainly if a satisfactory redrawing process could be devised, the 4-coordinate system would, almost certainly, be the most economical method of image storage available.

Development of adequate smoothing algorithms was not adopted as a part of this research; it is a fairly major research problem in itself. However one fairly crude reconstruction algorithm was written, simply because generation of a *picture* from topological coordinate sets provides a most satisfying demonstration of the sufficiency of such coordinate descriptions.

The algorithm PLOT1 was written as a Fortran program: its input was the set of coordinates representing a specified print, and its output was a file of line-plotting instructions for the graphics display facility of a laser printer. The algorithm first performed topological reconstruction in the normal manner, and then assigned polar coordinates to every ridge intersection point in such a manner that all the topological irregularities were assigned their own (real) polar coordinates. A series of simple linear *smoothing* operations are applied, coupled with *untangling* and *gap-filling* procedures that make successive small adjustments to the radial distances of all the intersection points that are *not* irregularities. These processes continue until a certain standard of smoothness is attained. Finally the picture is output as a collection of straight line segments between connected ridge intersection points.

A sample reconstructed fingerprint image is shown in figure 12, together with its descriptive data. The picture is made up of 4,404 straight line segments, and it almost looks like a fingerprint! Certainly the topology is correct, and each irregularity is properly located: it is just the intervening ridge paths that have suffered some unfortunate spatial distortions. For the sake of comparison, the original print tracing from which the coordinate sets were derived is shown in figure 13 (it has been reduced from 10x to 5x magnification). Detailed comparison of figures 12 and 13 will reveal a few places where the topology appears to have been altered. In fact it has not been altered — but, at this magnification, some ridges appear to have touched when they should not. This tends to occur where the ridge flow direction is close to radial. In such places the *untangling* sub-



```
FINGERPRINT RECONSTRUCTION DATA:

Card number   6.  Finger number   8
Window size: 6"
Magnification :    5.00
Downward displacement of origin :  -700
Number of line segments drawn:  4404
Fingerprint data size : 526 bytes.
```

Figure 12. Fingerprint reconstruction.

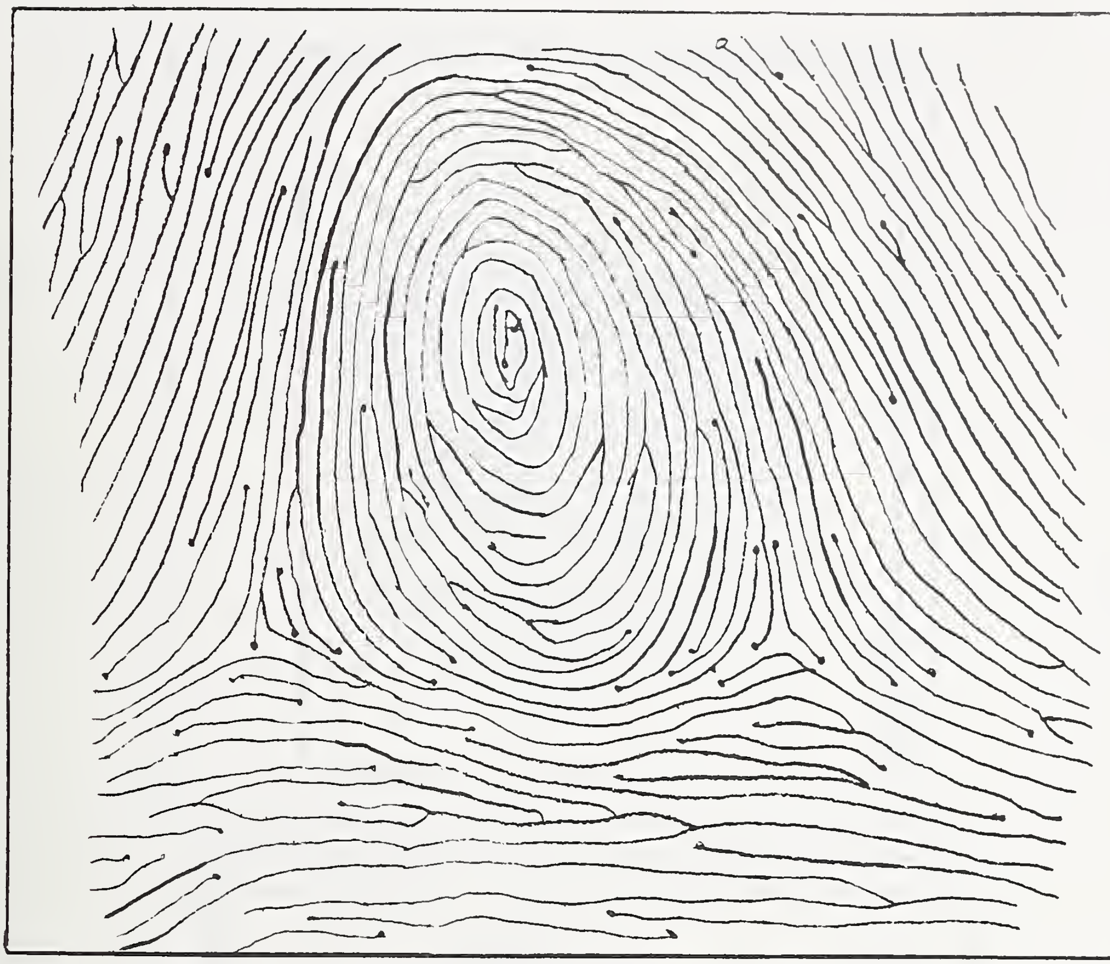


Figure 13. Copy of fingerprint tracing.

routine, which moves ridges apart when they get too close together, has not been forceful enough in separating them.

Figure 14 shows the tracing of a latent mark, together with its reconstructed picture. In this case the latent data comprised 32 coordinate sets (filling approximately 100 bytes), of which 21 make up the edge-description. There are ten genuine characteristics shown, and the remaining topological irregularity is the ridge recurve close to the core. The reconstructed image is made up from 780 straight line segments.

The facility for reconstruction also affords the opportunity to actually *see* a 'default edge-topology'. Figure 15 shows two further reconstructed images of the print in figure 12. The upper picture is the same as figure 12, except for a reduction in magnification (to 2.5×). The lower picture is a reconstruction from the condensed data set for the same print, after all the coordinate sets relating to ridges going 'out of sight' have been deleted. All the loose ends have been tied up by the reconstruction algorithm in a fairly arbitrary, but interesting, way. The lower picture does, of course, show some *false* ridge structure in areas that were 'out of sight'. However the data storage requirement for the corresponding

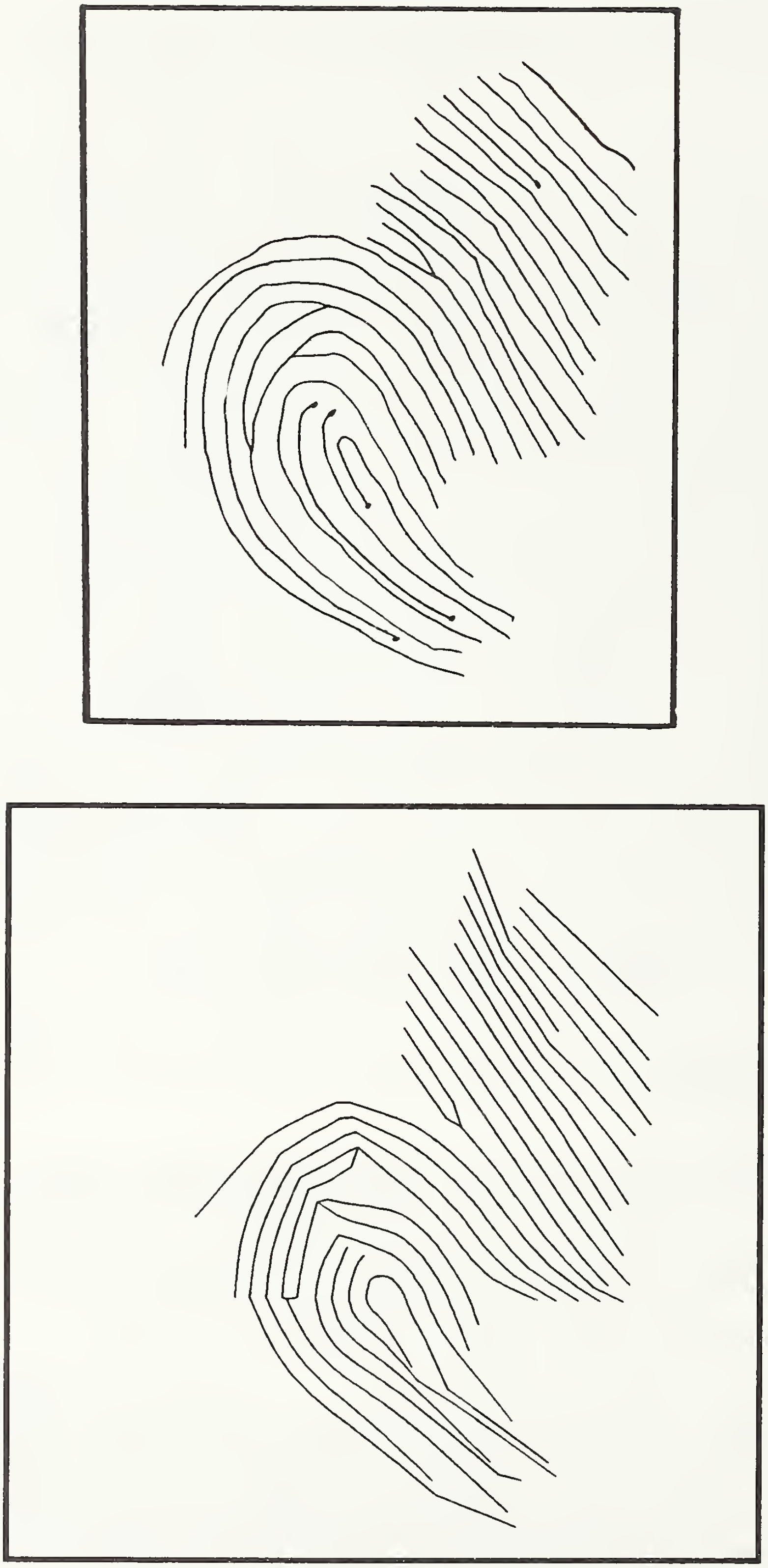


Figure 14. Latent tracing, and its reconstruction.

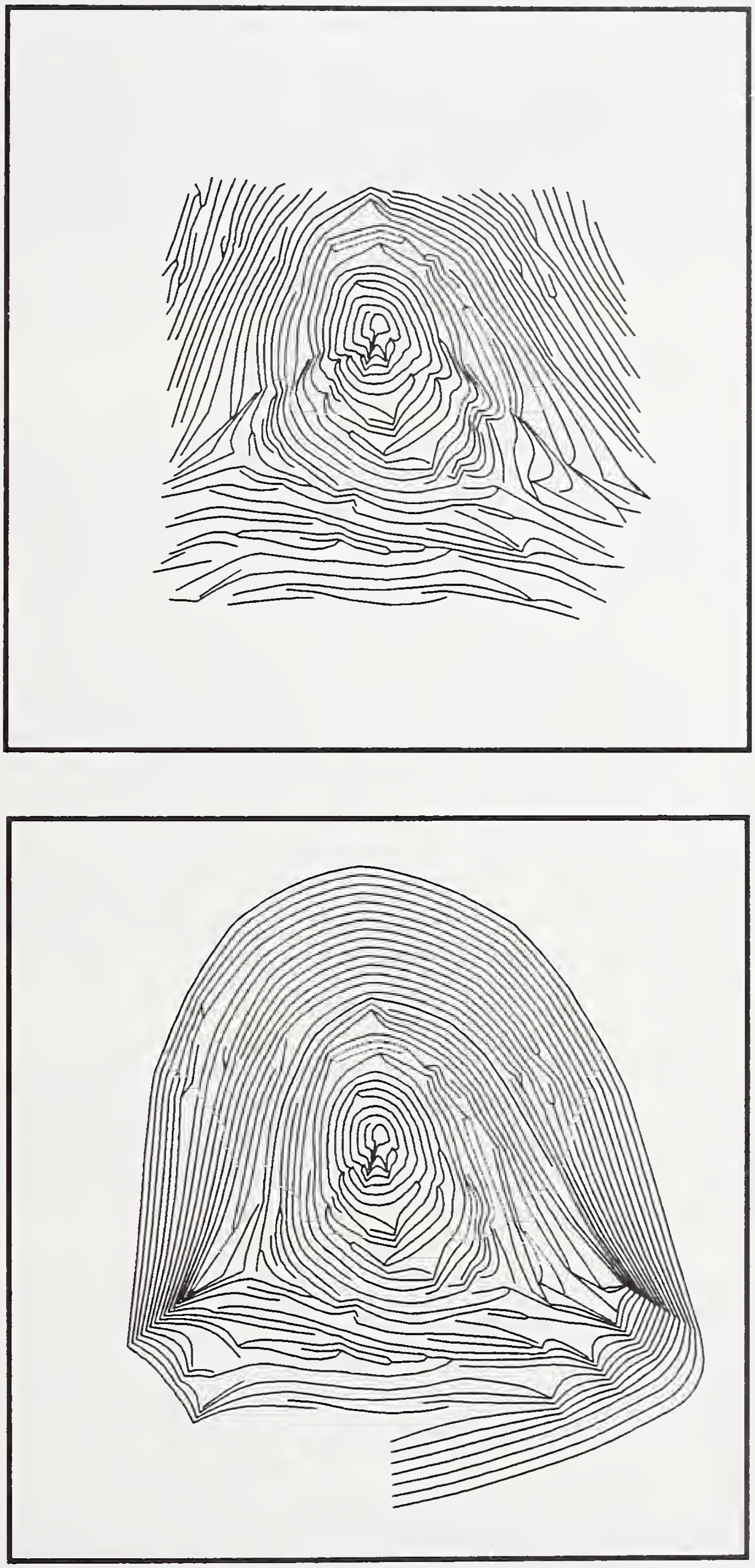


Figure 15. Reconstructions with, and without, defaulted edge-topology.

coordinate sets was only 354 bytes for the edge-free description, as opposed to 526 bytes for the original description.

From these pictures it is fairly clear that more sophisticated smoothing techniques will need to be applied before really reliable images can be retrieved. These pictures are quite sufficient nevertheless to demonstrate the potential for such a scheme. They are also a fine demonstration of the effectiveness and accuracy of the topological reconstruction algorithms. *

4.3 Outline of further work to be done.

This work outlined in this paper has lead to development of systems which could be implemented now — but which would require a *manual* file-print encoding process. It was, of course, the intention that such datafile conversion should be an automatic process; consequently development of such necessary data extraction algorithms would be desirable. A list of possible areas for further research is given here :—

(a) Automatic data gathering algorithms should be designed which are capable of extracting the required forms of data from the grey-scale output from automatic fingerprint scanners. For the reasons given in paragraph 4.2 the ability to track along ridges is *not* required. However the ability to locate every interruption of the otherwise smooth ridge flow in the print *is* needed. Moreover each interruption has to be *typed* according to the table of possibilities laid out in paragraph 2.3. 'Unclear' areas, rather than simply being rejected, must be *fenced off* — and all the places where ridges run into the *fenced area*, or emerge from it, must be recorded. This is a substantial departure from current practice; normally unclear areas would simply be rejected.

(b) Once such data-gathering algorithms have been written, and sizeable experimental databases built up — then the various parameters of the matching algorithms must be tuned finely by extensive experiments. Optimum parameter values for use on automatically read data are unlikely to be identical to their optimum values for manually prepared databases.

(c) Some investigation should be conducted in order to determine if there is any value in including a fifth coordinate, namely 'ridge direction', for each characteristic. No use of ridge direction data has been made in any of these topological schemes, even though it is the standard third coordinate for all the existing spatial methods (where (X, Y, θ) is the coordinate format for each characteristic, and θ is the ridge

---

* remember that the *path* of the ridges plays no part in the comparison algorithms LM5 and LM6; only the topology, and the positions of the characteristics are used. The defects in these pictures are not, therefore, a reflection of defects in the latent searching algorithms.

flow direction local to each particular characteristic.) There are a number of places within the various topological matching algorithms where tests on ridge direction could be applied in conjunction with consideration of angular misorientation. It is felt, however, that sufficient spatial information is already in use, and that the dividends would be too small to justify the 25% increase in data storage requirement that such a change would inevitably produce.

(d) An appropriate parallel architecture for the algorithms MATCH4 and LM6 has to be developed in conjunction with selection of the most suitable of the available parallel processors.

4.4 Conclusion.

The results obtained in these experiments show, beyond any reasonable doubt, that a topological approach to fingerprint coding offers a great deal in terms of improved accuracy and cost-effectiveness. The power of resolution between mates and non-mates given by the *combination* of topological and spatial information is vastly superior to that which can be obtained by use of spatial information alone.

The greatest benefit that has been obtained is *accuracy.* The question of *speed* has to be left open until the benefits of LM6's extensive parallelism have been realized.

REFERENCES.

[1] M.K.SPARROW and P.J.SPARROW, **"A topological approach to the matching of single fingerprints: development of algorithms for use on rolled impressions"**, *N.B.S. Special Publication 500 - 124*, May 1985.

[2] SPARROW and SPARROW, page 6.

[3] SPARROW and SPARROW.

[4] SPARROW and SPARROW, pages 21 - 55.

[5] SPARROW and SPARROW, pages 58 - 64.

[6] SPARROW and SPARROW, pages 7 - 12.

[7] SPARROW and SPARROW, pages 54 - 55.

[8] SPARROW and SPARROW, page 53.

[9] SPARROW and SPARROW, pages 21 - 55.

[10] R.T.MOORE and J.R.PARK, **"The graphic pen, an economical semi-automatic fingerprint reader"**, *1977 Carnahan Conference of Crime Countermeasures, Lexington, Kentucky*, pages 59 - 62.

[11] J.H.WEGSTEIN, **"An automated fingerprint identification system"**, *N.B.S. Special Publication 500 - 89*, February 1982.

[12] SPARROW and SPARROW, page 56.

[13] SPARROW and SPARROW, page 65.

[14] SPARROW and SPARROW, page 10.

[15] SPARROW and SPARROW, page 65.

[16] Thomas M. DAVIS and Angelo L. KONTIS, **"Spline interpolation algorithms for track-type survey data with application to the computation of mean gravity anomalies"**, (U.S. Naval Oceanographic Office 1970).

[17] R.J.VANWYCKHOUSE, **"Synthetic bathymetric profiling system (Synbaps)"**, (U.S. Naval Oceanographic Office 1973).

BIBLIOGRAPHY.

APPLEBY, Dr.R.M. **"A.V.R: a new approach to some problems of fingerprint comparison."** *Police Journal*, January 1979, pages 57 - 60.

BANNER, Conrad S. **"The FBI approach to automatic fingerprint identification."** Conrad S.Banner & Robert M.Stock. *FBI Law Enforcement Bulletin.* January 1975, pages 2 - 9 & 24.

BANNER, Conrad S. **"The state of development of the FBI's automatic fingerprint identification system."** *FBI Law Enforcement Bulletin.* June and July 1973.

CALIFANA, Anthony L. **"Simplified version of the NCIC technique for coding fingerprints."** Parts I, II, III. *Law and Order.* March, April and May 1974.

CHERNOFF, Herman. **"Some applications of a method of identifying an element of a large multidimensional population."** *Multivariate Analysis — IV, edited by P. R. Krishnaiah.* North-Holland Publishing Company, 1977.

CHERRILL, Frederick R. **"The fingerprint system at Scotland Yard."** Her Majesty's Stationary Office, London, 1954.

DAVIS, Thomas M. and KONTIS, Angelo L. **"Spline interpolation algorithms for track-type survey data with application to the computation of mean gravity anomalies."** Technical Report 226, U.S. Naval Oceanographic Office, Washington, D.C., 1970.

ERIKSSON, Sven Arne. **"The application of computerized information retrieval to fingerprint identification."** *International Criminal Police Review.* August 1976, pages 194 - 201.

FBI. **"The science of fingerprints."** U.S. Government printing office, 1979.

FLETCHER, T.A. **"Fingerprinting techniques."** *Police review.* 19/8/1977, pages 1104 - 1106.

GODSELL, J.W. **"A method for converting regional main fingerprint collections."** Home Office, Police Research and Planning Department, 1967.

GRANT, Douglas. **"Why the unknown dead?."** *Police Journal.* July 1977, pages 248 - 251.

HANKLEY, W.J. and TOU, J.T. **"Automatic fingerprint interpretation and classification via contextual analysis and topological coding: Pictorial pattern recognition."** Thompson Book Company, Washington, D.C., 1968.

KAWASHIMA, Misao and HOSHINO, Yukio. **"Automatic reading and matching method for fingerprint identification."** International Conference on Fingerprints, London. November 1984.

MACLEAN, George. **"Searching single fingerprints."** Identification Bureau, City of Glasgow Police, 1970.

MENZEL, E.R. **"Fingerprint detection with lasers."** Marcel-Dekker Inc., New York, 1980.

MILLARD, K. **"Developments on automatic fingerprint recognition."** International Carnahan Conference on Security Technology, Zurich, Switzerland, October 1983.

MOENSSENS, Andre A. **"Fingerprint techniques."** Chiltern Book Company, 1971.

MOORE, R.T. and MCCABE, R.M. **"An overview of algorithms for matching fingerprint minutiae in latent searches using relative positional information of the minutiae."** International Conference on Fingerprints, London, November 1984.

MOORE, R.T. and PARK, J.R. **"The graphic pen, an economical semiautomatic fingerprint reader."** Carnahan Conference on Crime Countermeasures, University of Kentucky, Lexington, Kentucky, 1977, pages 59 - 62.

OLSEN, Robert D. (Snr). **"Scott's fingerprint mechanics."** Charles C. Thomas, 1978.

PARZEN, Emanuel. **"On estimation of a probability density function and mode."** *Annals of Mathematical Statistics,* volume 33, Institute of Mathematical Statistics, 1962, pages 1065 - 1076.

SCHENDEL, U. **"Introduction to numerical methods for parallel computers."** Translated by B.W.CONOLLY. *Ellis Horwood Ltd., Chichester.* 1984.

SINGH, Parduman. **"Pressure distortions in fingerprinting."** *Police Journal.* February 1963, pages 79 - 82.

SNYDER, Richard E. **"Automated fingerprint identification."** *Police Chief.* October 1977, pages 58 - 61.

SPARROW, Malcolm K. **"Digital Coding of Single Fingerprints: A New Approach for the Computer Age."** *Journal of Police Science and Administration.* June 1982, pages 206 - 217.

SPARROW, Malcolm K. and SPARROW Penelope J. **"Topological Coding of Single Fingerprints for Automated Comparison."** Carnahan Conference on Security Technology, University of Kentucky, Lexington, Kentucky, 1985.

SPARROW, Malcolm K. and SPARROW Penelope J. **"A Topological Approach to the Matching of Single Fingerprints: Development of Algorithms for Use on Rolled Impressions."** N.B.S. Special Publication 500 - 124, U.S. Department of Commerce, May 1985.

SPIVA, Wayne E. **"Microfilmed fingerprints — an automated system for latent fingerprint files."** *Police Chief.* February 1971.

STANWIX, W.D. **"Fingerprint classification and identification."** *Police Surgeon.* April 1975, pages 46 - 60.

STOCK, Robert M. **"Research and development program for continued automation of the FBI's identification division."** International Conference on Fingerprints, London, November 1984.

VANWYCKHOUSE, R.J. **"Synthetic bathymetric profiling system (Synbaps)."** Technical Report, U.S. Naval Oceanographic Office, Washington, D.C., 1973

VENKAIAH, D. **"Laws of prints and impressions."** Law Book Company, Allahabad, 1979.

WEGSTEIN, Joseph H. **"An automated fingerprint identification system."** N.B.S. Special Publication 500 - 89, U.S. Department of Commerce, February 1982.

WEGSTEIN, Joseph H. **"A semi-automated single fingerprint identification system."** N.B.S. Technical Note 481, U.S. Government Printing Office, Washington D.C., 1969.

WEGSTEIN, Joseph H.**"The M40 fingerprint matcher."** N.B.S. Technical Note 878, U.S. Government Printing Office, Washington D.C., 1978.

Appendices.

APPENDIX A.

FORM FOR LATENT INFORMATION.

LATENT REF.NO: ..82..... PATTERN TYPE: .... FINGER NO: ....

NO. OF EXTRACTED VECTORS: ...1.......

CENTRAL FEATURE CODE: 6......

ANGULAR LOWER BOUND: .0°....

ANGULAR UPPER BOUND: .120°.

CENTRAL FEATURE RIDGE-COUNT LOWER BOUND: ..13....

CENTRAL FEATURE RIDGE-COUNT UPPER BOUND: ..17....

NO.OF RIDGES CROSSED BY GENERATING LINE: .21.....

NO.OF FIRST CENTRAL FEATURE RIDGE: ...12...

EVENT CODES (LEFT), 10 AT A TIME, UNIT = 0.5 cm

CODES.	B	B	B	4	2	B	3	B	B	B	B	3	B	8	6	B	B	B	7	B
DISTANCES.	6	7	8	1	1	9	7	9	10	10	10	0	10	1	1	10	11	11	2	11

CODES.	B																			
DISTANCES.	10																			

EVENT CODES (RIGHT), 10 AT A TIME, UNIT = 0.5 cm

CODES.	3	B	6	8	4	6	8	B	B	B	B	3	6	8	B	6	8	B	2	4
DISTANCES.	1	8	2	2	2	4	4	9	9	8	9	2	2	2	13	10	10	10	7	7

CODES.	7																			
DISTANCES.	1																			

APPENDIX B.

Code Description.

0. The ridge goes out of sight without meeting any characteristic.
1. Not allocated.
2. Ridge meets a bifurcation as if from left fork.
3. Ridge ends.
4. Ridge meets a bifurcation as if from right fork.
5. Ridge returns to its starting-point without any event occurring.
6. Ridge meets a new ridge starting on the left.
7. Ridge bifurcates.
8. Ridge meets a new ridge starting on the right.
9. Not allocated.

A. Ridge encounters scarred tissue.

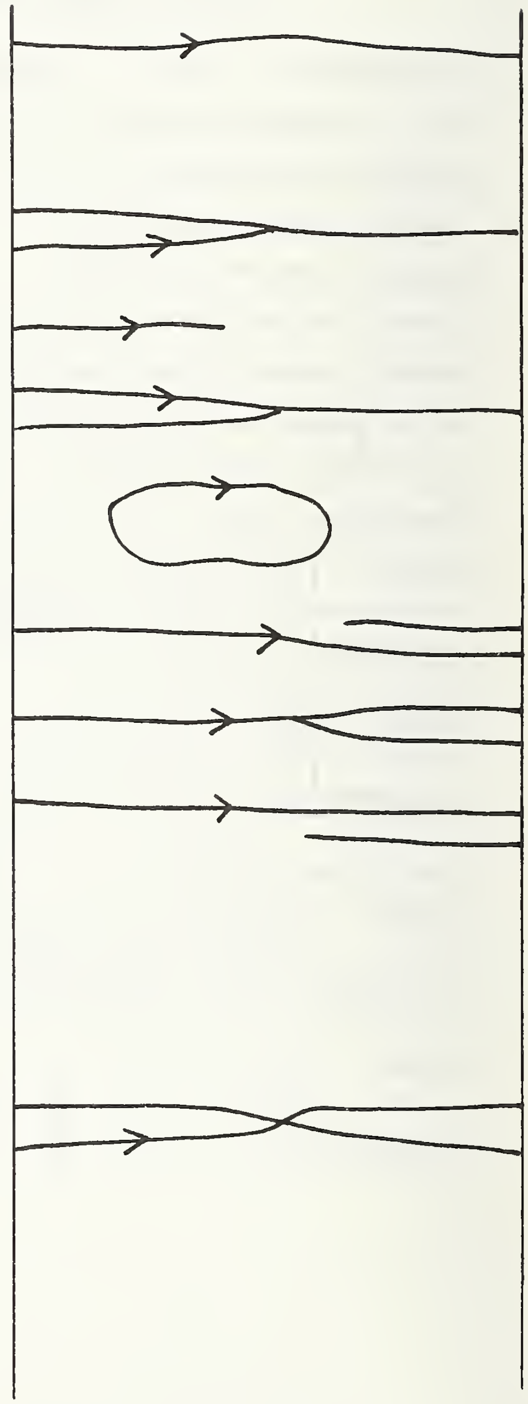
B. Ridge encounters blurred or unclear print.

C. Ridge meets a compound (e.g. a cross-over).

D. Not allocated.

E. Not allocated.

F. Used for vector padding.



APPENDIX C.

PROFORMA FOR FILE PRINT INFORMATION IN LATENT SCHEME.

CARD SET: 3333 CARD NUMBER: 82 FINGER NO: 9 PATTERN TYPE: whoH

BOUNDARY ARRAY LENGTHS: LEFT... 24 ...   RIGHT.. 33 ....

BOUNDARY ARRAY(LEFT) :   (NOTE - DISTANCE UNIT IS 0.5 cms)

CODES.	2	3	8	3	C	C	4	2	2	C	8	6	6	7	8
DISTANCES.	16	16	24	7	9	8	2	2	7	8	7	7	8	3	10
CODES.	6	8	6	0	0	0	0	0	0						
DISTANCES.	10	8	8	10	7	5	2	1	0						
CODES.															
DISTANCES.															

BOUNDARY ARRAY(RIGHT) :   (NOTE - DISTANCE UNIT IS 0.5 cms)

CODES.	4	3	7	4	6	6	3	6	8	4	7	4	3	2	6
DISTANCES.	18	22	8	34	9	6	1	1	1	24	6	20	5	20	8
CODES.	3	8	8	6	8	6	8	6	8	0	0	7	3	0	0
DISTANCES.	2	8	14	7	7	10	10	4	4	11	9	5	4	7	8
CODES.	0	0	0												
DISTANCES.	4	3	2												

DISTANCE CONVERSION MEASURES: (NOTE - DISTANCE UNIT IS 0.5 cms)

DEGREES FROM LEFT BOUNDARY:	0	60	120	180
DISTANCES MEASURED:	21.4	19.8	21.2	27.6
RIDGE COUNT COVERED:	24	20	24	33

EVENT CODES OVERLEAF.

EVENT CODES. (TOPOLOGICAL COORDINATES.)

NO.	CODE.	THETA.	RC.	NO.	CODE.	THETA.	RC.	NO.	CODE.	THETA.	RC.
1	2	1	15	26	B	83	12	51	1	134	28
2	1	2	23	27	6	84	8	52	7	136	22
3	1	5	24	28	E	87	13	53	1	138	30
4	1	8	25	29	E	90	13	54	7	140	30
5	7	11	14	30	E	90	14	55	1	140	32
6	3	18	24	31	E	92	14	56	3	143	28
7	7	19	25	32	7	95	14	57	1	144	32
8	6	22	10	33	E	97	14	58	1	146	33
9	6	40	27	34	2	98	5	59	1	147	34
10	0	44	28	35	6	103	6	60	1	147	34
11	0	47	27	36	E	104	12	61	1	149	36
12	0	48	26	37	1	106	20	62	1	150	37
13	0	49	25	38	1	109	21	63	6	150	34
14	0	52	24	39	2	113	7	64	1	151	39
15	B	53	16	40	1	114	21	65	3	152	32
16	0	55	22	41	1	117	22	66	6	154	38
17	0	56	21	42	1	119	23	67	2	156	35
18	B	59	16	43	1	124	24	68	2	163	37
19	0	62	19	44	2	125	15	69	0	165	37
20	0	64	18	45	7	129	10	70	7	169	33
21	0	67	16	46	1	129	25	71	0	170	37
22	0	68	16	47	1	130	26	72	3	171	36
23	B	73	13	48	1	133	27	73	3	172	31
24	B	79	13	49	3	134	21	74	0	176	34
25	B	80	12	50	7	134	7	75	0	178	33
								76	0	179	32
								77	6	180	14

APPENDIX D.

Table of results of tests performed using LM3.

No.	Parameters							Performance		
	BOUND	CMS	HOPS	MAXSHIFT	ADT	DDT	SDT	MR1	MR3	MR10
1	15	1	1	1	5	3	3	44.64%	62.50%	82.14%
2	5	1	1	1	4	2	2	46.43%	60.71%	82.14%
3	5	2	1	1	4	2	2	44.64%	64.29%	82.14%
4	5	3	1	1	4	2	2	46.43%	66.07%	83.93%
5	5	-1	1	1	4	2	2	42.86%	60.71%	75.00%
6	5	1	1	1	7	5	5	46.43%	53.57%	78.57%
7	5	1	1	1	10	5	5	42.86%	51.79%	75.00%
8	5	1	0	0	4	2	2	50.00%	69.64%	78.57%
9	5	1	2	2	4	2	2	44.64%	55.36%	83.93%
10	5	-1	0	0	2	2	2	42.86%	60.71%	71.43%
11	5	-1	0	0	2	1	1	44.64%	64.29%	73.21%
12	5	5	1	1	4	2	2	46.43%	60.71%	85.71%
13	5	5	1	1	7	5	5	42.86%	51.79%	80.36%
14	10	1	1	1	4	2	2	46.43%	58.93%	82.14%
15	15	0	1	1	4	2	2	41.07%	60.71%	80.36%
16	5	0	1	1	4	2	2	41.07%	60.71%	80.36%
17	5	0	0	0	4	2	2	42.86%	64.29%	76.79%
18	5	0	0	0	5	5	5	44.64%	58.93%	76.79%
19	5	1	2	2	4	2	2	44.64%	55.36%	83.93%
20	5	3	0	0	4	2	2	53.57%	67.86%	82.14%
21	5	4	0	0	4	2	2	53.57%	66.07%	80.36%
22	5	5	0	0	4	2	2	48.21%	67.86%	82.14%
23	5	2	0	0	4	2	2	53.57%	66.07%	78.57%
24	5	3	0	0	2	2	2	50.00%	67.86%	80.36%
25	5	3	0	0	3	2	2	53.57%	73.21%	82.14%
26	5	3	0	0	6	2	2	50.00%	64.29%	78.57%
27	5	3	0	0	10	2	2	42.86%	53.57%	76.79%
28	5	3	0	0	99	2	2	33.93%	53.57%	80.36%
29	5	3	0	0	3	1	1	51.79%	73.21%	80.36%
30	5	3	0	0	3	3	3	55.36%	73.21%	83.93%

Appendix D continued.

No.	Parameters							Performance		
	BOUND	CMS	HOPS	MAXSHIFT	ADT	DDT	SDT	MR1	MR3	MR10
31	5	3	0	0	3	0	0	50.00%	66.07%	83.93%
32	5	3	0	0	3	2	1	53.57%	73.21%	82.14%
33	5	3	0	0	3	1	2	51.79%	73.21%	80.36%
34	5	3	0	0	3	4	4	57.14%	71.43%	82.14%
35	5	3	0	0	3	3	2	55.36%	73.21%	83.93%
36	5	3	0	0	3	2	3	53.57%	73.21%	83.93%
37	5	3	0	0	3	1	0	48.21%	67.86%	82.14%
38	5	3	0	0	3	0	1	53.57%	71.43%	82.14%
39	5	3	0	0	3	2	0	50.00%	67.86%	82.14%
40	5	3	0	0	3	0	2	57.14%	71.43%	82.14%
41	5	3	0	0	3	5	5	57.14%	69.64%	82.14%
42	5	3	0	0	3	6	6	53.57%	69.64%	82.14%
43	5	3	0	0	3	7	7	53.57%	69.64%	82.14%
44	5	3	0	0	3	2	4	53.57%	73.21%	80.36%
45	5	3	0	0	3	3	5	57.14%	71.43%	82.14%
46	5	3	0	0	3	2	6	53.57%	73.21%	80.36%
47	5	3	0	0	3	3	6	57.14%	71.43%	82.14%
48	5	3	0	0	3	0	4	58.93%	71.43%	78.57%
49	5	3	0	0	3	1	4	51.79%	73.21%	78.57%
50	5	3	0	0	3	4	2	53.57%	71.43%	83.93%

APPENDIX E.

Table of results of tests performed using LM4.

In tests 1-24 the following parameters were fixed: BOUND=5, MAXSHIFT=0.

The following parameters were fixed for the non-boundary vectors only: CMS=3, HOPS=0, ADT=3, DDT=3, SDT=5.

Tests 1-23 were performed only on the subset of 25 latents that included at least one boundary vector. Tests 24, 25, 30-42 were performed on the whole latent set. Tests 26-29 used the subset of latents that contained no boundary vectors.

Tests 1-23 used the original 59 file prints and tests 24-42 used the expanded set of 100 file prints.

No.	Parameters					Performance		
	CMS	HOPS	ADT	DDT	SDT	MR1	MR3	MR10
1	1	1	6	4	8	44.00%	72.00%	84.00%
2	3	1	6	4	8	52.00%	72.00%	88.00%
3	3	0	6	4	8	56.00%	68.00%	80.00%
4	2	0	6	4	8	56.00%	68.00%	80.00%
5	1	0	6	4	8	52.00%	68.00%	76.00%
6	0	0	6	4	8	48.00%	60.00%	80.00%
7	-1	0	6	4	8	52.00%	60.00%	76.00%
8	1	0	4	4	8	52.00%	60.00%	68.00%
9	1	0	8	4	8	48.00%	64.00%	80.00%
10	1	0	10	4	8	48.00%	60.00%	80.00%
11	1	0	6	3	5	52.00%	68.00%	76.00%
12	1	0	3	3	5	52.00%	60.00%	84.00%
13	3	1	8	4	8	44.00%	68.00%	80.00%
14	3	1	6	3	5	52.00%	68.00%	88.00%
15	3	1	3	3	5	60.00%	68.00%	80.00%
16	3	1	6	4	6	52.00%	72.00%	88.00%
17	3	1	4	4	4	52.00%	68.00%	84.00%
18	3	1	5	3	5	48.00%	64.00%	80.00%
19	3	1	3	3	3	56.00%	68.00%	80.00%
20	3	1	2	2	2	52.00%	68.00%	84.00%
21	3	1	2	2	4	56.00%	68.00%	84.00%
22	3	1	2	3	4	52.00%	68.00%	84.00%
23	3	1	3	2	3	56.00%	68.00%	80.00%
24	3	1	3	3	5	48.21%	67.86%	80.36%

Appendix E continued.

In tests 25-42 the following parameter was fixed: BOUND=5.

The following parameters were fixed for the boundary vectors only: CMS=3, HOPS=1, ADT=3, DDT=3, SDT=5.

Tests 25, 30-42 were on the complete set of 56 latents and the 100 file prints. Tests 26-29 were on the subset of latents that contained no boundary vectors and the 100 file prints.

No.	Parameters						Performance		
	CMS	HOPS	MAXSHIFT	ADT	DDT	SDT	MR1	MR3	MR10
25	3	0	0	2	1	4	44.64%	71.43%	80.36%
26	3	0	0	3	3	5	50.00%	76.67%	83.33%
27	1	0	0	3	1	2	54.84%	74.19%	80.65%
28	3	0	0	2	1	2	54.84%	77.42%	80.65%
29	0	0	0	2	1	2	38.71%	54.84%	67.74%
30	3	0	0	2	1	2	51.79%	71.43%	80.36%
31	2	0	0	2	1	2	55.36%	71.43%	80.36%
32	4	0	0	2	1	2	51.79%	73.21%	82.14%
33	3	0	0	2	2	2	50.00%	71.43%	82.14%
34	3	0	0	1	1	1	48.21%	62.50%	82.14%
35	3	0	0	2	1	1	51.79%	73.21%	80.36%
36	3	0	0	3	1	3	51.79%	67.86%	80.36%
37	3	0	0	2	1	3	51.79%	69.64%	80.36%
38	3	1	1	2	1	2	58.93%	67.86%	83.93%
39	3	1	1	4	2	2	53.57%	66.07%	80.36%
40	3	1	1	2	1	4	51.79%	69.64%	80.36%
41	3	1	1	2	1	2	53.57%	67.86%	83.95%
42	2	1	1	2	1	1	58.93%	67.86%	85.71%

APPENDIX F.

Table of results of tests performed using LM5.

The following parameters were fixed in these tests: BOUND=5, HOPS=0, MAXSHIFT=0, MDT=1, PDT=10, DEPTH=5.

No.	Parameters				Performance		
	CMS	MAT	CUTOFF	SPAN	MR1	MR3	MR10
1	3	20	20	30	71.43%	78.57%	83.93%
2	3	20	5	30	75.00%	76.79%	80.36%
3	3	20	15	30	80.36%	82.14%	85.71%
4	3	20	13	10	78.57%	80.36%	85.71%
5	1	90	15	10	69.64%	80.36%	82.14%
6	3	20	15	10	80.36%	82.14%	85.71%

ACKNOWLEDGEMENTS.

The authors are indebted to the Institute of Computer Science and Technology at the National Bureau of Standards for its hospitality and for the use of its excellent facilities. More particularly they are most grateful to Ray Moore (NBS) for sharing the benefits of his vast experience in this field; to Mike McCabe (NBS) for his invaluable advice on computational problems and his expert assistance in negotiating the foibles of Fortran; to Bob Stock (FBI fingerprint automation and research) and his staff for their advice, their cooperation in providing the many hundreds of fingerprints required, and for allowing access to, and use of, their automatic scanning facilities.

It is almost certain that this research would never have proceeded to the experimantal stages had it not been for the provision of a Harkness Fellowship from the Commonwealth Fund of New York; it was that award which afforded the opportunity to continue this project in the U.S.A.

NBS-114A (REV. 2-80)

U.S. DEPT. OF COMM. **BIBLIOGRAPHIC DATA SHEET** *(See instructions)*	1. PUBLICATION OR REPORT NO. NBS/SP-500/126	2. Performing Organ. Report No.	3. Publication Date October 1985

4. TITLE AND SUBTITLE

Computer Science and Technology:
A Topological Approach to the Matching of Single Fingerprints: Development of Algorithms for Use on Latent Fingermarks.

5. AUTHOR(S)

Malcolm K. Sparrow and Penelope J. Sparrow

6. PERFORMING ORGANIZATION *(If joint or other than NBS, see instructions)*

National Bureau of Standards
U.S. Department of Commerce
Gaithersburg, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered

Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*

Same as item 6

10. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number: 85-600592

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)*

This paper naturally follows a previous N.B.S. Special Publication (No. 500-124), in which topological coding schemes were devised for automated comparison of rolled impressions. The contents of that paper are a prerequisite for a proper understanding of this one. Ine development of topological coding schemes is here extended to cover the automated searching of fragmentary latent marks, such as would be found at the scene of a crime. The benefits to be derived from topological descriptions of fingerprints are a direct result of their immunity to change under ordinary plastic distortion. In the case of latent marks such spatial distortions tend to be exaggerated; hence the importance of applying topology-based systems to them. This paper describes a method of coding fingerprint patterns by a variety of 'topological coordinate schemes', with fingerprint comparison being performed on the basis of localized topological information which is extracted from the recorded coordinate sets. Such comparison is shown to offer a substantial improvement in performance over existing (spatial) techniques. Furthermore, a methods for pictorial reconstruction of a complete fingerprint, from its coordinate representation, is demonstrated.

12. KEY WORDS *(Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)*

Automated comparison; fingerprints; image-retrieval; latent-marks; minutiae; topology.

13. AVAILABILITY

[X] Unlimited
☐ For Official Distribution. Do Not Release to NTIS
[X] Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

67

15. Price

USCOMM-DC 6043-P80

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SCIENCE & TECHNOLOGY

Superintendent of Documents,
Government Printing Office,
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Bureau of Standards Special Publication 500-.

Name ______________________________

Company ______________________________

Address ______________________________

City ______________ State ______________ Zip Code ______________

(Notification key N-503)

NBS Technical Publications

Periodical

Journal of Research—The Journal of Research of the National Bureau of Standards reports NBS research ana development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396).
NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Bureau of Standards
Gaithersburg, MD 20899

Official Business
Penalty for Private Use $300