COMPUTER ENHANCEMENT OF FINGERPRINT IMAGES

A thesis submitted for the degree of

Doctor of Philosophy

in the University of London

by

Barry Graham Sherlock

Department of Computing

Imperial College

London SW7

March 1989

ABSTRACT

This thesis presents a new technique of fingerprint enhancement, intended for use in automatic fingerprint recognition systems. The print is first smoothed using a directional Fourier filter whose orientation is everywhere matched to the local ridge orientation. It is then thresholded, yielding the enhancement.

Approaches to filtering general images whose local properties may be described by position dependent parameters are developed, and applied to fingerprints. A set of prefiltered images is produced after appropriate sampling of these parameters, and used to build up the output image. Methods are introduced for reducing the number of prefiltered images required. When applied to fingerprints, this reduces computation substantially.

The concept of the direction field is used to describe image directionality. This yields a mathematical model giving the local ridge orientation in terms of the core and delta locations. This model is applied to resolve ambiguities occurring when interpolating between sampled orientation values.

Alternative implementations of the enhancement technique using optical hardware are presented.

The application of adaptive window methods to fingerprint processing is investigated. A trainable recognition system is applied to such tasks as determining ridge orientation and locating minutiae.

TABLE OF CONTENTS

TITLE PAGE 1

ABSTRACT 2

TABLE OF CONTENTS 3

TABLE OF FIGURES 9

TABLE OF TABLES 12

DEDICATION 13

ACKNOWLEDGEMENTS 14

STATEMENT OF ORIGINALITY 16

CHAPTER 1: INTRODUCTION TO FINGERPRINT PROCESSING 19

1.1 Characteristics of fingerprints 19

1.1.1 Local ridge orientation 23

1.1.2 Local ridge spacing 24

1.1.3 Singular points (cores and deltas) 25

1.1.4 The Henry classification system 27

1.1.5 Ridge minutiae 31

1.1.6 Topology, distortion and ridge counts 34

1.2 Requirements of fingerprint processing 38

1.2.1 Areas of application of fingerprint processing 38

1.2.2 Automated police fingerprint matching systems 40

1.3 Previous work in fingerprint processing 47

1.3.1 Historical and general 47

1.3.2 Classification systems 49

1.3.3 Classification algorithms 52

1.3.4 Encoding 54

1.3.5 Enhancement 59

1.3.6 Locating singular points 61

1.3.7 Locating minutiae 62

1.3.8 Fingerprint registration 62

1.3.9 Theoretical and mathematical descriptions of fingerprints 63

1.3.10 Optical processing of fingerprints 64

1.4 An approach to fingerprint enhancement by directional filtering 65

CHAPTER 2: FREQUENCY DOMAIN PROCESSING OF FINGERPRINT IMAGES 68

2.1 Introduction to image enhancement and restoration 70

2.1.1 Image enhancement 70

2.1.2 Image restoration 74

2.2 Representation of fingerprint images in the Fourier domain 84

2.3 Position independent (stationary) linear filtering 92

2.3.1 Radial (frequency) filtering 93

2.3.1.1 The ideal filter 94

2.3.1.2 The Butterworth filter 94

2.3.1.3 The Chebyshev filter 95

2.3.1.4 The elliptic filter 96

2.3.2 Directional filtering 96

2.3.2.1 The "ideal filter" 100

2.3.2.2 The ramp filter 100

2.3.2.3 The Gaussian filter 101

2.3.2.4 The modified Gaussian filter 101

2.3.2.5 The sinusoidal filter 101

2.3.2.6 Another approach to directional filtering 102

2.3.3 Combining radial and directional filtering 103

2.4 Position dependent (nonstationary) linear filtering 104

2.4.1 Relevance of position dependence to fingerprint processing 104

2.4.2 Position dependent radial and directional filtering 104

2.4.3 Position dependent filters in terms of position dependent parameters 105

2.5 Application to fingerprints 107

CHAPTER 3: IMPLEMENTATION OF POSITION DEPENDENT FILTERS 112

3.1 Discretisation of stationary analogue filters 114

3.1.1 The discrete Fourier transform and windowing 114

3.1.2 Digital filters from analogue filters 122

3.2 Direct implementation 126

3.3 The use of moving windows 128

3.4 Obtaining a set of parameterised prefiltered images 132

3.4.1 Sampling of parameter values 135

3.4.2 Selection of the most critical parameters 136

3.5 Application to fingerprint processing 138

CHAPTER 4: DIRECTIONAL CHARACTERISTICS OF IMAGES 148

4.1 Direction and orientation 149

4.2 Directionality in images 152

4.3 Vector fields and direction fields 154

4.4 A model of local ridge orientation 160

4.5 Evaluation and extension of the LRO model 164

4.6 Direction fields of fingerprint-like patterns 168

CHAPTER 5: DETERMINING LOCAL RIDGE ORIENTATION 173

5.1 Introduction 173

5.1.1 The need for interpolation 174

5.1.2 Sampling of orientation values 176

5.2 Finding LRO at a point 177

5.2.1 The generalised operator approach 177

5.2.2 Local Fourier transforms 180

5.2.3 Local variance of directionally filtered images 182

5.2.4 Projections of local windows 184

5.2.5 Editing of orientation estimates 189

5.3 Interpolation of LRO values 192

5.3.1 Problems encountered with traditional interpolation methods 192

5.3.2 Use of the LRO model to resolve ambiguities 193

5.3.2.1 Orientation unwrapping 193

5.3.2.2 Application of the LRO model to orientation unwrapping 197

5.3.3 The orientation interpolation algorithm 200

5.3.4 Choice of classical interpolation algorithm 201

CHAPTER 6: A DIRECTIONAL APPROACH TO FINGERPRINT ENHANCEMENT 202

6.1 Directional filtering 203

6.2 Thresholding 208

6.3 Thinning 212

6.4 Results 219

6.4.1 Enhancements of rolled fingerprint images 219

6.4.2 Enhancements of scene-of-crime marks 219

6.5 Evaluation of results 231

6.5.1 Evaluation of enhancements of rolled fingerprints 231

6.5.2 Discussion of enhancements of scene-of-crime marks 237

6.6 Conclusion 239

CHAPTER 7: AN IMPLEMENTATION USING OPTICAL HARDWARE 240

7.1 Introduction 240

7.2 Previous work on optical processing of fingerprints 243

7.3 Optical implementation of the enhancement algorithm 246

7.3.1 Optical components used 246

7.3.1.1 Fourier lens 246

7.3.1.2 Beam splitter 246

7.3.1.3 Sector-detector 247

7.3.1.4 Selector 249

7.3.2 Fourier domain filtering of images using optical hardware 249

7.3.3 Implementations 250

7.3.3.1 Implementation using fixed filters 250

7.3.3.2 Implementation using servo controlled rotating filter 252

7.3.3.3 Implementation using 1-dimensional filters 255

CHAPTER 8: AN INVESTIGATION OF THE USE OF ADAPTIVE WINDOWS IN FINGERPRINT PROCESSING 259

8.1 Adaptive windows 260

8.2 The WISARD recognition device 265

8.2.1 Conceptual introduction to WISARD 265

8.2.2 The CRS WISARD system 274

8.2.3 Computer simulation of WISARD 277

8.3 Application to fingerprints 279

8.3.1 Introduction 279

8.3.2 Results 279

8.3.3 Discussion 283

CHAPTER 9: CONCLUSION 285

9.1 General summary 285

9.2 Contribution of this research 287

9.3 Further work 290

APPENDIX A: DIRECTIONALITY IN IMAGES 293

A.1 The concept of direction 293

A.2 Directionality in images 299

A.3 Differential geometry and the direction field 300

A.3.1 The manifold 300

A.3.2 Vector and direction fields 303

A.3.3 The Poincare index 304

APPENDIX B: ONE DIMENSIONAL INTERPOLATION 306

B.1 Linear interpolation 306

B.2 Cubic spline interpolation 307

B.3 Monro-Chebyshev interpolation 309

APPENDIX C: THE WALKING FFT ALGORITHM 315

C.1 Introduction 315

C.2 The Walking FFT in one dimension 316

C.3 The Walking FFT in two dimensions 319

REFERENCES 322

TABLE OF FIGURES

1.1 Physical structure of ridged skin 20

1.2 Fluid flow and ridge "flow" 20

1.3 Cores and deltas 26

1.4 The Henry classification system 29

1.5 Types of minutiae 32

1.6 Distance measures on fingerprints 32

1.7 Typical police AFR system 41

1.8 Prism apparatus for taking fingerprints 41

1.9 The basic fingerprint patterns as deformations of concentric circles 51

2.1 The histogram 72

2.2 Image degradation model 72

2.3 Region containing parallel lines, and its Fourier transform 72

2.4 Annulus containing fingerprint Fourier components 90

2.5 A fingerprint and its magnitude Fourier transform 90

2.6 "Ideal" directional filter of Ikonomopoulos & Kunt 90

3.1 Spurious discontinuities 118

3.2 Nonstationary filtering using prefiltered images 118

3.3 Enhancements using 8 and 16 filter orientations 140

3.4 Filter angular bandwidth near singular points 143

3.5 Enhancements obtained with and without varying filter angular bandwidth near singular points 143

3.6 Enhancements using wide and narrow filter radial bandwidth 146

3.7 Directional filters for fingerprint enhancement 146

4.1 Direction and orientation 150

4.2 Orientable and non-orientable fields 150

4.3 The two orientations of an orientable field 150

4.4 Types of singular points for vector fields 159

4.5 Types of singular points for direction fields 159

4.6 Orientations predicted by the LRO model 165

4.7 Illustrating that LRO can be regarded as horizontal sufficiently far from the centre of a print 165

4.8 Schematic view of ridge structure on the palm 171

5.1 Fingerprint images 179

5.2 Orientations obtained using GOP approach 179

5.3 Local Fourier transforms 183

5.4 Orientations obtained using local variance approach 185

5.5 Projections of a window of fingerprint data 187

5.6 Orientations obtained using projections 187

5.7 Ambiguities encountered during interpolation 194

5.8 Unwrapping of a 1-d orientation valued function 194

6.1 A print at various stages in the enhancement process 205

6.2 16 directionally filtered versions of a print 206

6.3 A picture element and its 8 nearest neighbours 217

6.4 Enhancements of rolled fingerprints 220

6.5 Enhancements of scene-of-crime marks 227

7.1 Optical method of detecting LRO 244

7.2 Fourier transformation using lenses 244

7.3 Beam splitter 244

7.4 The sector-detector 248

7.5 Fourier domain optical filtering 248

7.6 Apparatus for optical enhancement of fingerprints 251

7.7 Apparatus for optical enhancement of fingerprints 251

7.8 Enhancement produced by an optical filter 254

7.9 Enhancement produced by the raster-scan approach 254

8.1 Models of nerve cells 266

8.2 The WISARD system 270

8.3 Determining LRO using WISARD 281

8.4 Locating minutiae using WISARD 282

8.5 Locating cores and deltas using WISARD 282

A.1 Orientation and direction 294

A.2 The 2-sphere $\mathbf{S}^2$ and the projective plane $\mathbf{P}^2$ 294

A.3 Manifolds and tangent spaces 302

A.5 Vectors on a manifold 302

A.6 Direction field on a manifold 302

B.1 Cubic spline interpolation 308

B.2 Fourier interpolation 308

B.3 Monro-Chebyshev interpolation (first kind) 311

B.4 Monro-Chebyshev interpolation (second kind) 313

C.1 The Walking FFT in one dimension 317

TABLE OF TABLES

6.1 Results of application of Home Office minutiae detector 234

6.2 Analysis of fingerprint enhancement results by fingerprint expert 235

DEDICATION

This thesis is dedicated to my wife Jennifer.

ACKNOWLEDGEMENTS

My sincere thanks are due to the following people:

My supervisor, Dr D.M. Monro, for his help in supervising all aspects of this research, and who was the source of an endless stream of ideas.

My colleagues in the Signals and Images Laboratory, particularly David Simpson, Henry Young and Kevin O'Connor, for useful discussion and sharing of ideas.

Ken Millard, who gave me the benefit of his many years of experience as a fingerprint expert for Scotland Yard and the Home Office.

The staff of the Home Office Scientific Research and Development Branch, especially Ken Millard, James Easthope and Dr Fred Preston, with whom I spent many fruitful hours of discussion. They provided the fingerprint images used in this thesis and gave me access to the Home Office AFR development computer and a CRS WISARD.

Colin Petts of the GEC Hirst Research Centre, for his assistance with the research on optical processing of fingerprints.

Professor I. Aleksander, for discussion relating to adaptive windows and the WISARD device.

Professor G.H. Granlund and Dr Hans Knutsson of Linkoping University, for discussion regarding the GOP approach to image processing.

The Harry Crossley Foundation, the CSIR and the ORS for their invaluable financial support.

Anyone else whose help I may have inadvertently failed to mention here.

STATEMENT OF ORIGINALITY

The following aspects of the material presented in this thesis are to the best of the author's belief original, and constitute a major part of the contribution made by this research to the body of knowledge within the fields of image processing and fingerprint enhancement:

1. The fingerprint enhancement technique which is the main result of this thesis and is described in chapter 6.

2. The approach to 2-dimensional filtering which was used. This includes:

   - The application of classical 1-dimensional filtering techniques to anisotropic 2-dimensional filtering problems by expressing the filter as a product of a factor depending only upon the radial coordinate with a factor depending only on the angular coordinate, where the radial factor is given as a classical 1-dimensional filter with the desired frequency properties.

   - The expression of the position dependence of a nonstationary filter in terms of a set of position dependent parameters whose values vary according to the local nature of the image data.

- The sampling of these parameters to produce a set of stationary filters, and hence a set of prefiltered images, from which the desired filtered image may be built up.

- The methods for reducing the number of prefiltered images required, by reduction in the number of parameters and the coarse sampling of parameters.

3. The realisation that directional information within images exhibits projective geometrical properties (see appendix A), rather than the spherical geometry characteristic of the directionality associated with vectors. The consequent identification of the direction field of differential geometry as the appropriate concept for describing the behaviour of orientations within an image. In particular, the use of the 2-dimensional direction field to describe LRO in fingerprints, with the singular points of the direction field corresponding to fingerprint core and delta locations.

4. The mathematical model of fingerprint LRO, expressing the LRO at all points within the image in terms of the core and delta positions.

The following are also believed to be original, but constitute contributions of lesser importance:

1. The unwrapping technique for fingerprint LRO, making use of the model of LRO.

2. The interpolation technique for 2-dimensionally sampled orientations.

3. The methods given in sections 5.2.2 to 5.2.4 for determining LRO at a point.

4. The investigation in section 2.3.2 of various functions which could specify the angular dependence of a directional filter.

5. The investigation of the use of adaptive windows for fingerprint processing.

6. The three implementations of the enhancement algorithm presented in chapter 7.

7. The Walking FFT algorithm in one and two dimensions.

CHAPTER 1

INTRODUCTION TO FINGERPRINT PROCESSING

1.1 Characteristics of fingerprints

The hands and feet of humans have been provided by nature with a pattern of parallel ridges of skin. The ridges on the fingertips and the palms of the hands assist in the manipulation of objects by ensuring maximum friction between the skin and the object being handled. The ridges on the toes and the soles of the feet increase friction with the ground and hence help to prevent slipping when walking or running. Along the ridges at regular intervals are located sweat pores which aid in this task by keeping the skin moist. The ridged structure of the skin is also believed to increase the sensitivity of these areas of the body to touch. The physical structure of the ridged skin is illustrated in figure 1.1, which is adapted from [FBI] and [Montagna 1956].

When a surface is touched with the tips of the fingers, a small portion of the fluid excreted through the ridge pores is left behind, duplicating the ridge pattern that appears on the skin, and forming what is known as a latent print or "scene of crime mark". Typically, a latent fingerprint will weigh 0.1 milligrams, of which 99% is water which soon evaporates. The residue consists of approximately equal quantities of inorganic

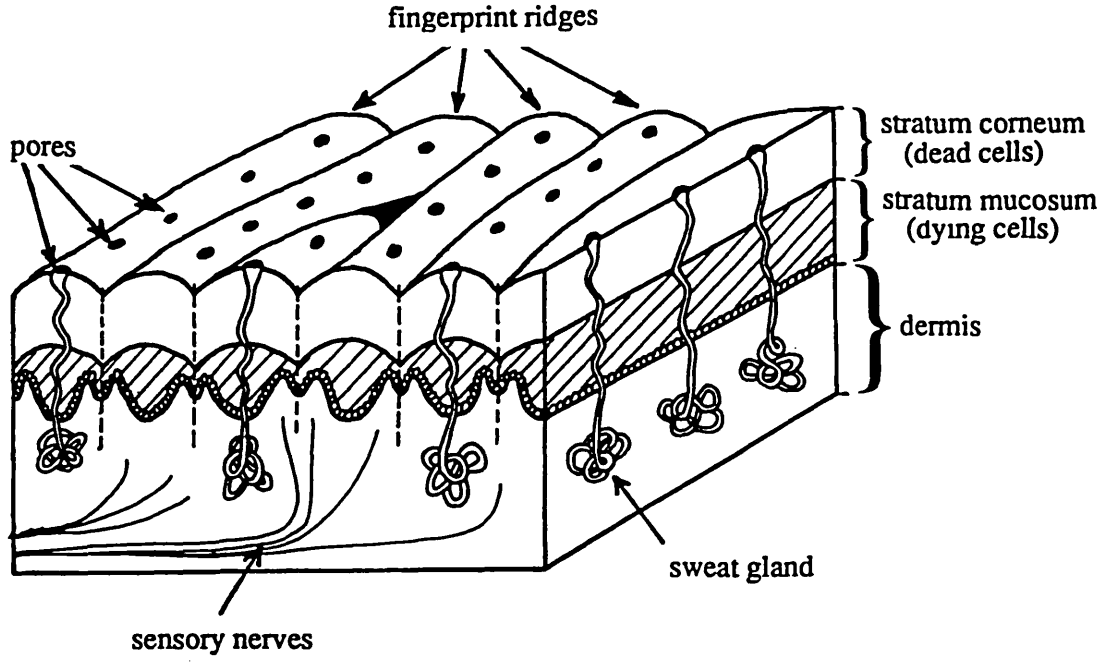
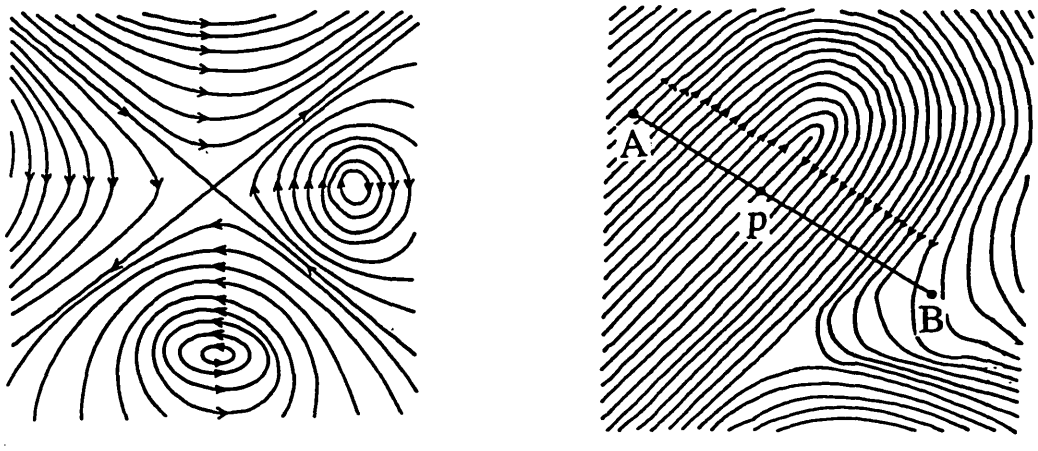


Figure 1.1. Physical structure of ridged skin



(a) (b)

Figure 1.2. Illustrating the difference between lines of flow of a fluid (a), and lines describing a fingerprint pattern (b). In (b), the lines cannot be consistently assigned a forward sense of flow (see text).

salts and organic material, and is normally not visible by simple visual inspection of the touched surface. A latent print can be rendered visible by exploiting some physical or chemical property of the residue. Dusting, which is the most frequently used method of detecting latent prints, relies for its operation upon the adherence of a fine powder to the residue. Moenssens [1971], Menzel [1980] and the Home Office [1986] describe several methods for detecting latent fingerprints.

Latent prints are usually of poor quality. Better impressions of the ridge structure may be obtained by carefully pressing the finger onto a sheet of paper after the skin has been given a thin coating of ink. An impression produced in this way is called an inked fingerprint. A rolled fingerprint is produced when the inked fingertip is rolled over the paper, leaving an impression of the entire pattern of fingertip ridges. Fingerprints for police records are produced in this way.

Latent and inked palmprints, soleprints and toeprints can of course also be produced in a similar manner. All prints have the essential property that they consist of parallel lines corresponding to the parallel ridges of skin.

Fingerprints (and less importantly, soleprints, palmprints and toeprints) can be put to an important practical use: they can serve as a means of personal identification. This is because they possess the properties of immutability and uniqueness, and these properties together enable a fingerprint to be a reliable "signature" which unambiguously identifies the individual.

Fingerprints are unique in that it is statistically extremely unlikely that two fingerprints will have identical patterns of ridges. The chances of such an occurrence are so low that it is regarded by fingerprint experts as "impossible". In section 1.1.5, it will be seen that each fingerprint has a large number of minute features whose spatial arrangement is specific to that print, and ensures that the print is very nearly unique. Stoney & Thornton [1986] present a review of mathematical analyses of fingerprint uniqueness.

However, uniqueness is not sufficient to ensure the usefulness of fingerprints for identification purposes. Suppose, for example, that humans, like snakes, shed their skins every year and grew new ones. Then a person's fingerprints could change every year and make identification and the keeping of records very difficult. Fortunately this is not the case, since fingerprints have the property of immutability, meaning that they remain the same throughout the life of the individual. A fingerprint taken from a newborn infant will consist of the same ridge pattern as at any time in later life. As time passes, the print will grow and distort slightly, but the basic layout or "topology" of the lines and their "minutiae" (see section 1.1.5) will not change. (Also, due to injuries and disease, areas of scar tissue may appear which obscure part of the print. However, such damaged regions will not consist of ridges and will therefore be easily distinguished from undamaged areas).

Cases have been recorded [Moenssens 1971] of criminals attempting to alter their fingerprints in order to avoid detection. Methods

used have included cutting off the skin of the fingertips, burning of the fingertips, application of acid to the fingertips, and grafting skin from other areas of the body onto the fingertips. The effect, however, has always been either the production of masses of scar tissue, or temporary disfigurement of the print which is soon restored to its previous state. No-one has achieved a permanent alteration of a fingerprint without destroying it completely.

The immutability and uniqueness properties and their importance to the use of fingerprints for personal identification have been known for many years, and were discussed in some detail in a 19th century book on fingerprints by Galton [1892].

1.1.1 Local ridge orientation

Upon examining any fingerprint, it will be seen that the ridge lines define everywhere a direction of "flow" of ridges. This direction of "ridge flow" at a point is the average ridge orientation within a small neighbourhood around the point and is called the local ridge orientation (LRO). The LRO is thus a continuously varying 2-d function whose value at each point is equal to the angle of ridge orientation.

It is important to note that in defining LRO, we had to talk about an average orientation of ridges within a window. This means that the LRO is a "smoothed-out" property which does not depend upon the exact detailed structure of each individual ridge. This detailed structure, consisting of many minute details or

"minutiae", is what gives the fingerprint its uniqueness and will be described in section 1.1.5. Because the LRO has smoothed out all these small details, a description of the fingerprint in terms of the LRO describes the overall collective "flow" of lines and ignores any property of single ridges.

However, it is misleading to speak of ridge "flow" in connection with fingerprints, because the local ridge orientation function does not describe a flow in the usual sense of the word. Given any two-dimensional set of flow lines describing the flow of some fluid, then not only should these lines determine a local orientation of flow, but it should in addition be possible to specify consistently a forward or reverse direction, or sense, of flow. Equivalently, it should be possible to assign consistently arrows to the lines, indicating the forward sense of flow. This can always be done to a set of lines describing the continuous flow of a fluid: see, for example, figure 1.2(a). However, examination of figure 1.2(b) shows that this is not possible in the case of lines describing a loop fingerprint. Along the path AB shown, the sense of the arrows changes abruptly at point p. The differences between LRO patterns and fluid flows is of fundamental importance and will be examined in greater detail in chapter 4.

1.1.2 Local Ridge Spacing (LRS)

Examination of a fingerprint image will show that within a neighbourhood of any point, there is a clearly-defined spacing between the centre lines of adjacent ridges, corresponding to a

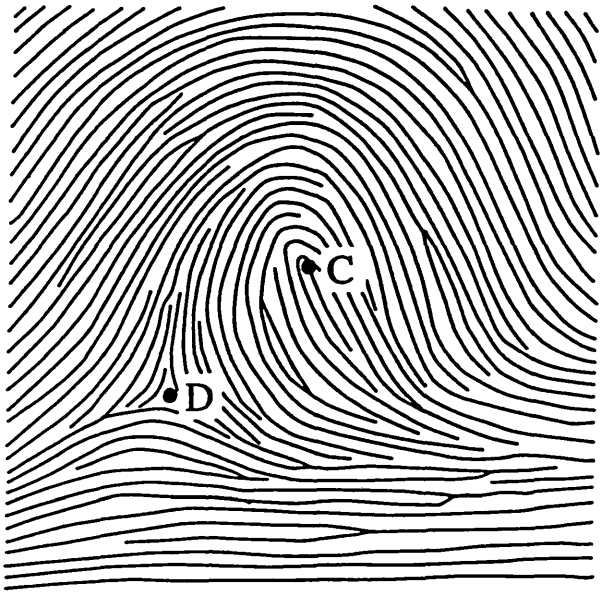
clearly-defined local spatial frequency of ridges. This spacing is called the local ridge spacing (LRS) and is a continuous function of position which is defined at all points in the print.

LRS is a slowly-varying function of position, and in addition its values are constrained within a fairly narrow range. A fingerprint expert in the Home Office [Millard 1987] provided the following figures regarding ridge spacings. Spacings may be as large as 1 ridge per mm in some men, or as small as 4.5 ridges per mm in some women and children, but are typically between 2 and 3 ridges per mm. Over all prints, LRS varies over a range of 4.5:1. Within any single print, however, the LRS will vary over a range of 3:1 at most, with 2:1 being more typical.

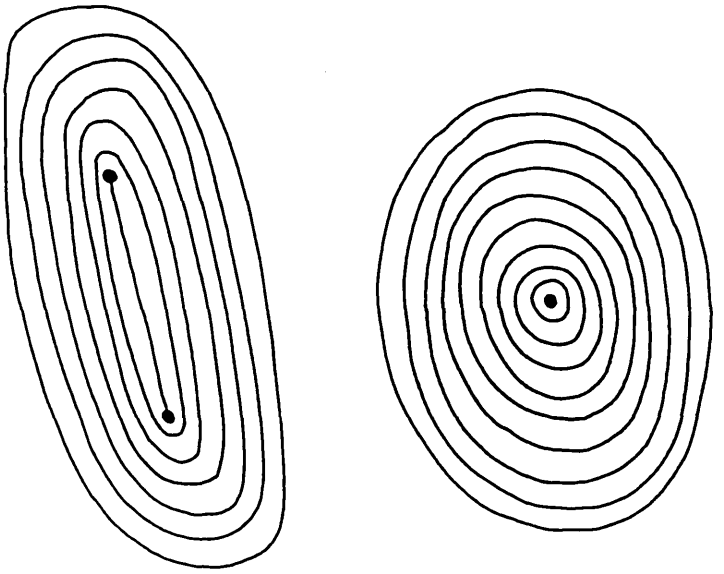
The above statistics seem to be fairly conservative. Trauring [1963], for example, assumes that LRS will not vary within a single print over a range of more than 2:1.

1.1.3 Singular points (cores and deltas)

Up to this point, LRO has been described as a continuously varying function of position. Strictly, this true only for the simplest type of fingerprint, known as the "plain arch" type. In all other classes of fingerprint, the LRO is continuous everywhere except at a small number of singular points where the local ridge orientation is not defined. There are two types of singular point: cores and deltas. The point marked C in figure 1.3a is an example of a core point. Examination of figure 1.3a shows that within any neighbourhood of C, the orientation of the ridges takes on all possible values. Therefore at C the LRO is not



(a)



(b)

(c)

Figure 1.3. (a) A fingerprint pattern, showing the singular points C (a core) and D (a delta).

(b) Centre of a whorl, showing two separate cores.

(c) Centre of a whorl, showing two coincident cores.

defined. The same can be said of point D, which is an example of a delta point. Cores and deltas form isolated singularities of the otherwise continuous LRO function.

Core points always resemble the centres of hairpin bends in the ridges, and delta points have three ridges meeting or tending to meet each other at approximately 120 degrees.

Core points should not be confused with what the literature calls the "core of the print" [Rao & Balck 1978]. The core of a print is a point which serves as a reference central point of the print and can be used for such purposes as aligning two prints with one another. This sense of the word "core" will not be used in this thesis.

All singularities in fingerprint LRO can be expressed in terms of the simple core and delta structures, which are therefore of fundamental importance in describing LRO. For example, the centre of a whorl, illustrated in figure 1.3(b) and (c) is formed from two cores of opposite hairpin orientation which are separated in (b) and coincident in (c).

There are simple rules governing core and delta numbers and positions and their relationship to the behaviour of the LRO function, and these are described in chapter 4.

1.1.4 The Henry classification system

Fingerprints may be divided into various types or classes, in such a way that any particular fingerprint will be a member of one of

these classes. By far the most frequently used classification system is the one developed in 1890 by E.R. Henry and published ten years later [Henry 1900], and it is now in almost universal use by police forces around the world. The Henry system is described further here; other classification systems are considered in section 1.3.2.

The Henry system, as adapted for use by the FBI, divides fingerprints into eight basic types. These types are usually easily distinguished from one another visually, but there are borderline cases which could be construed to belong to more than one class. Detailed definitions of each type are given in the FBI manual [FBI], but for our purposes the following descriptions should be adequate:

- **Plain arch** (see figure 1.4a)

The ridges run across the print without turning back on themselves. There are no cores or deltas, i.e. the local ridge orientation is continuous everywhere.

- **Tented arch** (see figure 1.4b)

The tented arch is similar to the plain arch except that some of the ridges near the centre have a vertical orientation. The LRO of a tented arch has two singularities: a core and a delta, with the core situated approximately vertically above the delta.

- **Radial Loop** and **Ulnar loop** (see figure 1.4c and 1.4d)

In the loop, the ridges near the centre of the print double back



(a) Plain arch



(b) Tented arch



(c) Radial loop



(d) Ulnar loop



(e) Central pocket loop



(f) Plain whorl



(g) Double loop



(h) Accidental

Figure 1.4. The Henry classification system

on themselves. There are two types of loop: **radial**, in which the hairpin bend formed by the loop slopes towards the little finger, and **ulnar**, in which it slopes away from the little finger. Loops have one core and one delta. The delta is lower than the core and to the side of it.

- **Central pocket loop** (see figure 1.4e)

This pattern is similar to the loop except that a circular or vortex pattern is enclosed within the centre of the loop pattern. There are two cores within the circular enclosure, and a delta to either side of it, with one of the deltas much closer to the cores than the other.

- **Plain Whorl** (see figure 1.4f)

The ridges near the centre form a circular or vortex pattern. Whorls have two cores (which may coincide if the central ridges are truly circular) and two deltas, below and to either side of the pair of cores.

- **Double loop (twin loop)** (see figure 1.4g)

Two loop structures, originating from opposite sides of the fingerprint, are wrapped around each other near the centre. Double loops have two cores (one at the centre of each loop) and two deltas, below and to either side of the cores.

- **Accidentals** (see figure 1.4h)

Accidentals are fingerprints which do not fall into any of the above classes. The number of cores is in all cases equal to the

number of deltas, however, as shown in chapter 4.

The plain whorl, central pocket loop and twin loop classes are collectively referred to as whorl patterns. In fingerprints from people of Caucasian origin, loops constitute 65% of all prints, whorls 30%, arches 5% and accidentals 1% [Hankley & Tou 1968]. Bates [1978] presents a more detailed statistical analysis of the distribution of the various classes.

1.1.5 **Ridge Minutiae**.

We have seen that fingerprints consist of locally parallel curved lines or ridges. If we follow the course of an individual ridge, we may find that at some point it comes to an end, or splits into two ridges. Such tiny irregularities in the pattern of ridges may be used to describe the detailed structure of individual ridges and are called minutiae (or Galton details, features or characteristics). Because of their tiny size, the position of each minutia is precisely defined. The following types of minutiae, which are illustrated in figure 1.5, may be encountered:

Ridge ending. See figure 1.5a.

Ridge bifurcation (fork). This occurs when a ridge splits into two ridges. See figure 1.5b.

Isolated dot. See figure 1.5c.

Isolated short ridge. See figure 1.5d.

Ridge break. This occurs when the path of a ridge is interrupted by a short break, as shown in figure 1.5e. Genuine

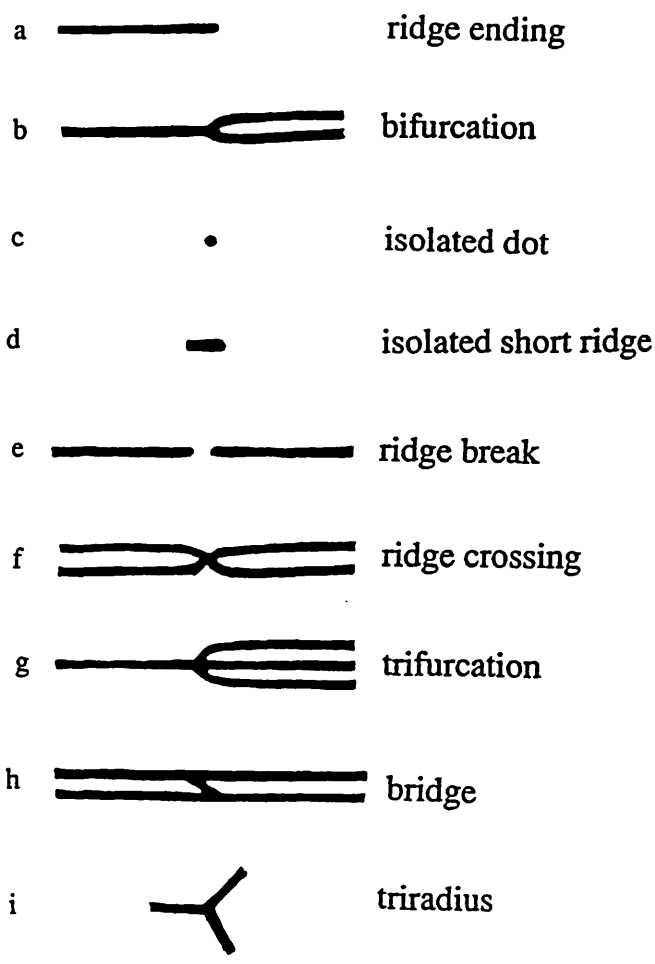


Figure 1.5. Types of minutiae

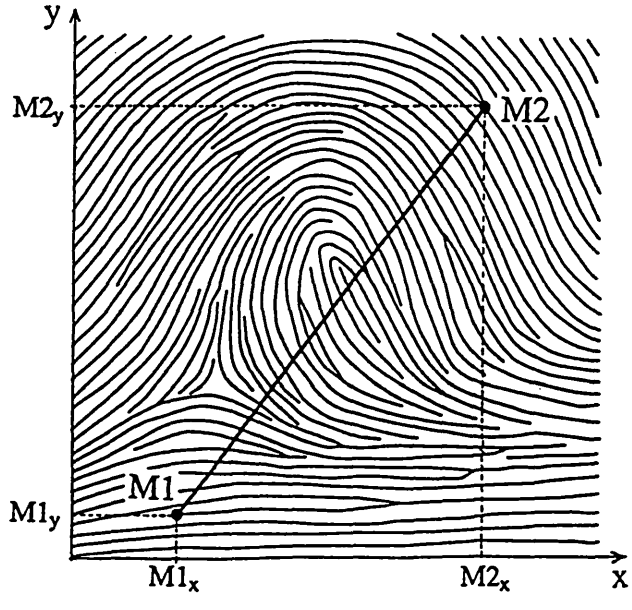


Figure 1.6. Distance measures on fingerprints. The Euclidean distance between the minutiae M1 and M2 is the length of the line joining them. The ridge count between them is the number of ridges crossing this line, and is 39 for this example.

breaks are often difficult to distinguish from spurious ones caused by imperfections in the image.

Ridge crossing. This occurs when two ridges cross over each other, as shown in figure 1.5f.

Ridge trifurcation. This occurs when a ridge splits up into three ridges at a single point, as illustrated in figure 1.5g. Trifurcations may be treated as a pair of coincident bifurcations.

Bridge. A bridge occurs when a short ridge joins one ridge to its parallel neighbour, as shown in figure 1.5h. Bridges are of little importance in automated fingerprint processing since they often cannot be distinguished from similar effects caused by noise.

Triradius. A triradius occurs where three ridge segments join together at a point at angles of approximately 120 degrees. See figure 1.5i. Triradii are often encountered at deltas.

By far the most frequently occurring minutiae are the bifurcations and ridge endings, which are consequently also the most significant in computer fingerprint processing.

A minutia at a given position (x,y) may be assigned an orientation equal to the local ridge orientation at (x,y).

Minutiae are of paramount importance in the use of fingerprints for criminological and other identification purposes, because it is precisely this pattern of small irregularities in the ridges

that ensures the uniqueness of a fingerprint. This fact was first noted in 1892 by Galton [1892]. A typical rolled fingerprint will have in excess of 100 minutiae, and these minutiae will be laid out in a pattern that is unique to that fingerprint. Patterns of minutiae are very strongly individual, and a matching of a very small number of minutiae between a scene-of-crime mark and a print is regarded as proof of identity in a court of law. In the UK, for example, 16 matching minutiae are required, whereas in many other nations 12 are required [Sparrow 1986, page 10].

A simplification is usually introduced in practice, which is to ignore minutiae types and consider only their position. This is done because misclassifications of minutiae frequently occur in practice due to "topological mutations" [Sparrow 1986, page 24]. Due to an indistinct image or differing pressures being applied, a minutia that appears to be a bifurcation in one print may look more like an ending in a subsequently produced print.

1.1.6 Topology, distortion and ridge counts

The skin of the fingertips is flexible and stretches and compresses easily. Consequently, two fingerprints taken from the same finger will differ from each other to some extent, since different amounts of stretching of the skin will occur each time a print is formed.

This distortion of fingerprints is not an isotropic process: greater distortion tends to occur perpendicular to the ridges than along the ridges. The skin stretches more easily perpendicular

to the ridges, because its thickness varies, enabling the ridges to "concertina" together or apart. Along the ridges, however, the skin is stronger because its thickness does not vary, and therefore it is less prone to stretching.

Another type of distortion occurs as the result of growth of the fingers during the interval between taking the two prints. Normal growth of the skin causes an increase in the size of the fingerprint.

Although two different prints from the same finger will be distorted versions of one another, they will nevertheless be easily recognisable as having come from the same finger. The distortion of prints caused by skin stretching, compression and growth is not sufficiently extensive to affect the basic properties of the ridges and minutiae and their relationships to each other. For example, if a given ridge segment starts with a bifurcation and terminates in a ridge ending, it will also do so in any subsequent print of the same finger. Also, those ridges which are adjacent to a given ridge will still be adjacent to it in a distorted version of the print.

The properties of a fingerprint which are invariant (or nearly invariant) under the types of distortion which occur when prints are made are referred to in the fingerprint processing literature as the "topological properties", or the "topology", of the print. In fingerprint processing, the words "topological" and "distortion-invariant" may be regarded as synonymous.

It is important to note that the word "topology" is not being used

here in its correct mathematical sense; its use is, however, based upon an analogy with ideas from the branch of mathematics known as topology. Strictly, any property of a topological space is said to be a topological property of the space if it is invariant under a class of distortions called the homeomorphic transformations of that space. (A homeomorphism is a continuous one-to-one onto mapping whose inverse is also continuous). Although the analogy is useful, it must be remembered that "topological" properties of prints are not necessarily mathematically topological as well.

The ridge count, which will now be introduced, is an example of a topological measure of distance within fingerprints. Consider figure 1.6 in which two minutiae M1 and M2 have been located as shown. Suppose that it is desired to obtain a measure of the "distance" between the two minutiae. The plane Euclidean distance

$$d(M1,M2) = \sqrt{(M1_x - M2_x)^2 + (M1_y - M2_y)^2}$$

could be used, but unfortunately it is not a topological measure since d(M1,M2) will vary as M1 and M2 move due to the effects of distortion.

The ridge count is a topological measure of the "distance" between minutiae. The ridge count between two minutiae is given by the number of ridges crossing a straight line segment joining the two minutiae, as illustrated in figure 1.6. Clearly, the ridge count is much less sensitive to the effects of distortion than the Euclidean distance measure. Ridge counts are used almost

universally in the encoding of fingerprints because their relative distortion-invariance ensures repeatability from print to print.

Another topological distance measure can be defined for points that happen to lie on the same ridge. Since distortion is much less pronounced along the ridges than perpendicular to them, the distance can be given by the length of the ridge segment joining the two points. This measure is not often used, however, because it is not defined for points on different ridges, and ignores distortions caused by growth of the entire print.

1.2 Requirements of fingerprint processing

The type of processing that will be required depends, not surprisingly, upon the application envisaged. The three major areas of application of fingerprint processing are outlined in the next subsection. Comparisons between the requirements of these different applications are given by Swonger [1973] and in the popular article by Eleccion [1973].

1.2.1 Areas of application of fingerprint processing

1. **Criminology**. This is the most widely used application of fingerprint processing and is the one to which this thesis will devote the most attention. In criminological systems large data bases of fingerprints are stored and attempts are made to find a match between an unknown fingerprint, usually obtained from the scene of a crime, and one of the fingerprints in the data base. The data bases are often of enormous size, and large numbers of enquiries are received per day. For example, the FBI's system [Swonger 1973; McGinty 1974] uses a data base of some 200 million fingerprints, and must handle up to 30 000 search requests per day. It is therefore clear that the use of computers to perform automatic fingerprint recognition (AFR) is necessary. Matching must be fast in order that the large number of enquiries can all be handled, and it must also be precise so that the system can distinguish between the several million different fingerprints within the data base.

2. **Access control**. A high-security access control system could be based upon the use of fingerprints to identify persons

authorized to enter a sensitive area. An access control sytem has much in common with a criminological AFR system, since its task is also to match unknown fingerprints against a data base of fingerprints of persons for whom access is permitted. However, different design techniques are needed for access control systems because they must satisfy different requirements. In access control systems, the main difficulty is to ensure fast response, preferably real-time response, to enquiries. The data bases involved are several orders of magnitude smaller than criminological data bases, containing tens, hundreds or at most thousands of fingerprints. Not only does this mean that far shorter searches are possible, but also that it is not necessary to match prints to as great an accuracy. The number of enquiries per unit time is much lower than for criminological systems.

3. **Genetics and Medicine**. Dermatoglyphics is a branch of genetics that deals with the correlation between different patterns of dermal ridges on the fingers, palms and soles and the likelihood of contracting or inheriting certain genetic diseases. The standard textbook on dermatoglyphics is by Cummins and Midlo [1943]. The work of Penrose [1963, 1965a, 1965b] is often quoted; extensive lists of further references may be found in [Penrose 1963], [Bates 1978] and a special issue of the American Journal of Anthropology devoted to dermatoglyphics [Mavalwala 1975]. Researchers in dermatoglyphics have developed their own classification systems appropriate to their aims [Grasselli 1968; Levi & Sirovich 1972].

The criminological application is by far the most frequently

encountered one, and it is therefore discussed in more detail in the next section.

1.2.2 Automated police fingerprint matching systems

Many large police forces have computer based systems to perform automated fingerprint matching. The largest system in the world is in use by the FBI in the United States. The algorithm used in the FBI system is presented in detail in [Wegstein 1982]. In the United Kingdom, the Home Office SRDB has developed an advanced system which is in use by Scotland Yard. The activities involved in a large police fingerprint matching system are illustrated in figure 1.7, which is based on [Swonger 1973]. There may be minor variations to this pattern in certain cases, usually involving the omission of one of the stages.

Central to all automated fingerprint recognition (AFR) systems is the existence of a large data base of fingerprint data against which unknown fingerprints can be matched.

The various stages shown in figure 1.7 are discussed below.

Sensing and digitisation

This is the process of converting the fingerprint into digital form so that it may be processed by a computer.

Before digitisation, the image may be in one of several forms, for example

1. An inked fingerprint on a standard police ten-fingerprint card.

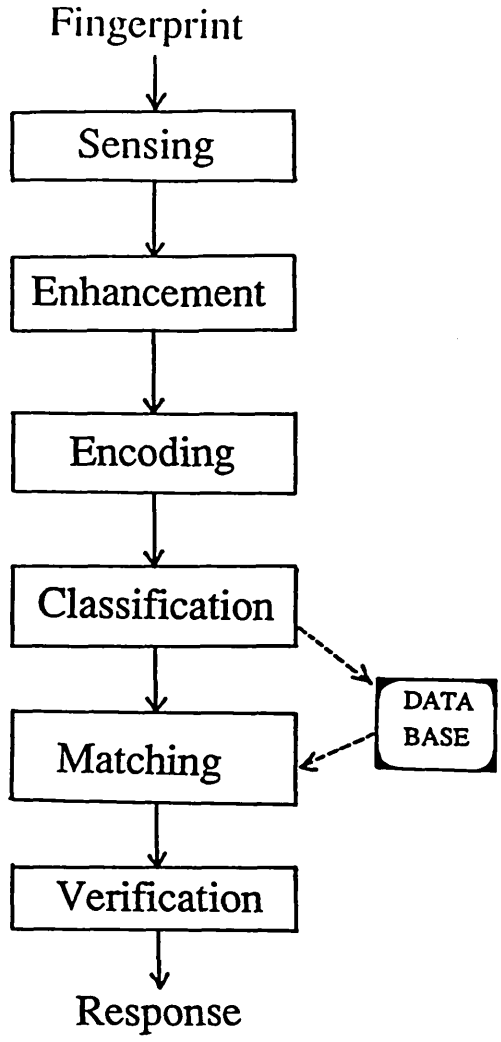


Figure 1.7. Diagram of the activities involved in a large police fingerprint matching system.

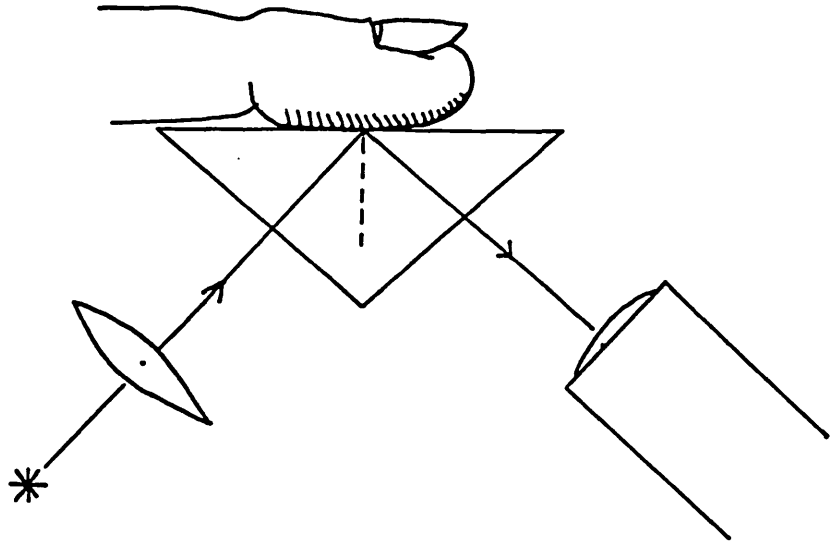


Figure 1.8. Optical apparatus for producing high quality fingerprint images without the use of ink.

2. A detected scene-of-crime mark.

3. An image obtained directly from the finger using an apparatus like the one shown in figure 1.8 [Kawagoe & Tojo 1984]. A prism is set up so that light from the source is totally internally reflected from the base of the prism into the camera lens. At those points where the skin of the finger is in contact with the glass, i.e. along the ridges, the total internal reflection does not occur, and therefore a clear image of the fingerprint is received by the camera.

Digitisation involves sampling the picture spatially to form a square array of intensity values. The intensity values are also quantised, so that the image becomes represented in the computer as a 2-dimensional array of numbers representing the quantised intensity at each picture element (pixel). Typically, a digitised image may have a size of 512 by 512 pixels, and each pixel will be quantised to one of 256 grey levels so that one byte of storage is needed per pixel. Such an image occupies 256 kilobytes of storage. The automatic fingerprint recognition system developed by the Home Office uses an image size of 480 by 480 one-byte pixels.

Enhancement

The enhancement stage is the one which transforms the digitised image into an "improved" form suitable for further processing by the encoding stage.

The enhancement stage of the processing is the main subject of this thesis. A serious obstacle to automatic fingerprint processing is the variability of print quality, even among good quality rolled fingerprints, and enhancement serves to reduce the effect of this variability. The most common defects are ridge gaps (usually caused by folds in the skin), joining together of adjacent ridges (caused by spreading of ink due to high finger pressure, by dirt on the skin or by excessive inking), smudging and smearing. In addition, light or underinked prints, thin ridges and dotted ridges (due to unusually large pores) are often encountered.

However, enhancement consists not only of the removal of defects such as these. It is also a process of removal of any information that is not needed by the encoding stage. For example, a typical coding system would completely disregard the presence and positioning of the sweat-pores within the ridges, and therefore a good enhancement algorithm should remove all pores from the image. It is clear, therefore, that what one means by enhancement depends upon what information one intends to extract from the image.

When a trained fingerprint officer examines a print, he or she is interested only in the ridge lines and where irregularities in them form minutiae. One can imagine that in the mind's eye an "ideal" enhanced image is perceived containing only this information. The techniques developed in this thesis aim to convert a raw fingerprint into such an "idealised" enhanced image. The enhanced image is binary in nature, with black indicating the

presence of a ridge and white indicating the absence of a ridge. Any breaks in the ridges, smudges and other defects caused by the fingerprint rolling process should be removed, leaving only a clear image showing each ridge as a smooth black line.

The output of the enhancement stage is the only information which the later stages of classification and coding have to work with, and it is therefore of vital importance that good quality enhancement be achieved in order to ensure successful completion of the later stages of processing. If syntactic methods of coding and classification such as those in sections 1.3.3 and 1.3.4 are used, the importance of good enhancement becomes even greater, since the syntactic methods tend to be sensitive to the presence of noise within the image.

Encoding

In the encoding stage, the information which is to be stored in the data base (ie the encoded form of the fingerprint) is extracted from the enhanced fingerprint image.

Usually, a fingerprint will be stored in the data base in an encoded form rather than simply as a digitised form of the fingerprint image. The reasons for this are twofold:

1. The amount of memory required to store digitised images is very large, typically about a quarter of a Megabyte per image. In a large system such as that used by the FBI, there may be up to 200 million prints, which would imply a data base size of about 50,000 Gigabytes. Clearly, it could be important to reduce

the amount of data to be stored, and this is done by encoding the fingerprint images.

2. In addition to having to store data on millions of prints, large AFR systems must also be capable of handling large numbers of requests for matching of unknown prints against the data base. The FBI system, as mentioned earlier, must deal with up to 30,000 requests per day. It is imperative, therefore, that the data be encoded into a form which is suited to being rapidly dealt with during the matching process.

In designing an encoding technique, one may encounter a trade-off between the reduction of data storage per fingerprint and the rapidity of matching or search possible. In such a trade-off, matching speed is usually regarded as more critical, especially in recent years as the cost of bulk memory has dropped dramatically.

Classification

Fingerprints may be classified into different types or classes according to some system of classification such as the Henry system. The classification stage determines which class the print belongs to. Knowledge of the print's class reduces the amount of searching required, since only entries of the correct class need be considered. Various classification systems are described in section 1.3.2, and classification algorithms are described in section 1.3.3.

Data base searching

In this stage, a list of potential candidates for matching is formed. The class of the print and any other information that is known, such as the person's age or sex, is used to reduce the number of candidates.

Matching

The matching process involves comparison of the encoded print with that of prints in the data base. In large police systems, the encoding is done by storing in some form or other the minutiae locations and orientations. Matching is handled by an algorithm which determines the degree of correlation between the minutiae locations and orientations in the two prints.

The main difficulties encountered in the matching process are caused by distortion, and for this reason there has been much interest in distortion-invariant techniques of coding.

Verification

The AFR system, after completing the searching and matching operations, will have produced a short-list of the most likely matches found. This list will be short enough (say, between one and ten candidates) for verification and choice of the correct match to be performed manually. Even if the system were good enough always to find the correct match, manual verification by trained fingerprint officers would always be necessary to satisfy legal requirements before a suspect can be convicted.

1.3 Previous work in fingerprint processing

In this section, an overview of the fingerprint processing literature is presented. Each aspect of fingerprint processing is given under a different subsection heading.

1.3.1 Historical and general.

The earliest known western writing on fingerprints is a report given in the Philosophical Transactions of the Royal Society in 1684 by Dr Nehemiah Grew [Grew 1684]. This article gives a description of the dermal ridges and sweat pores on the hands and feet. Grew noted that the ridges are everywhere equally spaced and run parallel to one another, but did not attempt to classify the patterns.

The earliest attempt at classification of fingerprint patterns was given in the **Commentatio de Examine Physiologico Organi Visus et Systematis Cutanei** (commentary on the physiological examination of the organs of vision and the cutaneous system) of Johannes Evangelist Purkinje, delivered at the University of Breslau in 1823. A translation of this rare document is included in chapter 5 of [Galton 1892].

In 1880, Henry Faulds suggested [Faulds 1880] that the printed impressions from bloody fingers could lead to the scientific identification of criminals, and also proposed that fingerprints be investigated as a possible means of proving that two persons are related to one another. In the same year Sir William Herschel wrote [Herschel 1880] of his successful use of

fingerprints in India as a means of positive identification in pension offices, registration offices and jails, and suggested that their use be extended to the army and police force.

In 1892, Francis Galton produced a book [Galton 1892] on the subject of fingerprints, which described in detail various methods of taking prints, and presented results of research proving both the uniqueness and immutability properties of fingerprints. Various applications of the use of fingerprints as a means of personal identification were discussed. A classification system based on loops, arches and whorls was developed, and techniques of indexing and storing fingerprints were suggested. An investigation of the hereditary properties, and the dependence of characteristics upon social and anthropological factors, was presented.

E.R. Henry in 1900 published his famous book [Henry 1900] on the classification and uses of fingerprints, in which the well-known Henry classification system was presented.

General treatments of fingerprinting for police use were given in the books by Bridges [1963] and Cherrill [1954], and the FBI's manual "The Science of Fingerprints" [FBI]. Conferences devoted solely to fingerprint processing were held by the Home Office, London in 1974 [Home Office 1974] and 1984 [Home Office 1984].

In his surveys of pattern recognition [Kanal 1974] and classification systems [Kanal 1972], Kanal showed how some of the fingerprint processing techniques of the early 1970's fit into these more general fields.

Popular general introductions to fingerprint processing were given by Eleccion [1973] and Lerner [1983]. Riganati [1977] gave an overview of the algorithms used in automated fingerprint processing.

1.3.2 Classification systems.

Development of a system of classification involves dividing the entire range of possible fingerprints into various types or classes, such that any given fingerprint will fit into one of these classes. It of great advantage to a classification system for the classes to be balanced, i.e. to have as nearly equal a population as possible. In a perfectly balanced system, the average amount of information given by a knowledge of the class to which an individual fingerprint belongs will be maximised. This corresponds in an AFR system to minimising the average number of prints that need be considered during the matching stage.

By far the most frequently used classification system is the Henry system described earlier. Although the Henry system produces classes that are not well balanced, none of the alternative systems have gained any wide acceptance. The Henry system retains the all-important advantage that it is universally understood by workers in all applications of fingerprint processing. A brief description of other classification systems in the literature will now be given.

Because the requirements of fingerprint processing for medical and anthropological purposes are fundamentally different from those

of police systems, it is natural that such applications may need different classification systems. Levi and Sirovich [1972] put it as follows:

> A classification scheme to be used for genetical and anthropological studies should satisfy different requirements [to one for police purposes]. It should take into account only the global structure of the print, disregarding minute features (ridge endings and branchings), which are valuable only when personal identification is required. In addition it should have some degree of "continuity" over the spectrum of all possible figures, thus leading to a natural classification of the transitional figures.

One system which satisfies these requirements was developed by Clerici [1979], who states that any pattern of fingerprint ridges can be obtained by a suitable deformation and windowing of the set of concentric circles shown in figure 1.9a. Figures 1.9b,c and d illustrate how the basic arch, loop and whorl patterns could be produced by suitable deformations. The deformed lines are then regarded as the contour lines of a mathematical surface. It is suggested that this surface could be described in closed form in terms of a suitable number of parameters so that, by varying the parameters, the contour lines obtainable may describe any given fingerprint. The class to which a print belongs is determined by the parameter values necessary to describe it.

Moayer and Fu [1973, 1975, 1976a, 1976b] developed a system of classification intended to correct the drawback of the Henry system that there are too few classes and that the classes are imbalanced. The system involved obtaining a two-dimensional array of LRO values sampled over the image, and then parsing this array using a suitable grammar and syntactical analyser. The string

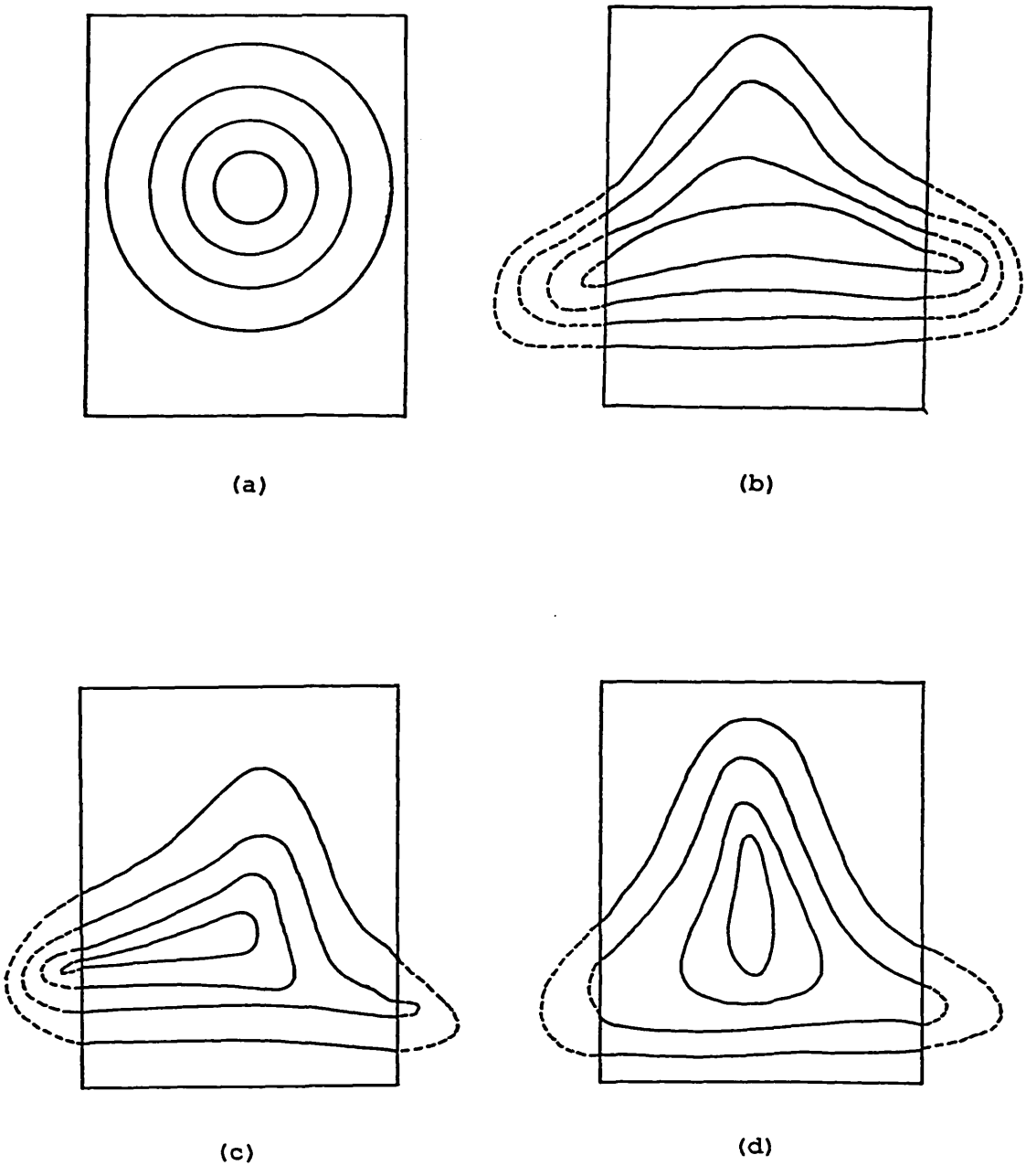


Figure 1.9. Deformation of a set of concentric circles (a) to produce arch (b), loop (c) and whorl (d) patterns.

obtained is then analysed further to give the class to which the fingerprint belongs. This system has the advantage that it can be implemented as a multi-level classifier, so that the classification can be as detailed or coarse as is desired.

Shelman [1967] developed a classification system giving a large number of classes with a well-balanced distribution of prints within these classes. As with the previous system, a two-dimensional array of LRO values is obtained, but here these LRO values are placed into a large vector describing the LRO information within the print. A metric function is defined to give a measure of the distance between two vectors corresponding to two prints. A class is given by a set of prints whose vectors are sufficiently close to one another according to this metric.

Shelman and Hodges [1973a, 1973b] present three classification systems which give a better balanced distribution than the Henry system by means of a further subdivision of the overcrowded classes within the Henry system. Solosko and Paley [1973] subdivide the overcrowded classes by including a set of five fingerprint descriptors based upon the positions of the singular points and the crease line at the base of the print.

1.3.3 **Classification algorithms**.

Once a classification system has been chosen, it is necessary to find a classification algorithm which, when presented with a print, will determine the class to which it belongs. Several researchers have described classification algorithms. Not surprisingly, most of these algorithms are based upon the use of

the Henry classification system. A few of these are listed below.

Marom [1967] investigated the appearance of the magnitude Fourier transforms of fingerprints, as produced optically using Fourier transform lenses. The claim is that the basic Henry classes can easily be distinguished from one another in the Fourier domain when the transforms of the images are rotationally averaged around the DC component. A finer classification can be obtained thereafter by taking into account the angular distribution of the Fourier components.

Nakamura et al [1982] determine the Henry class of a fingerprint by locating the core and delta points and determining their "directional distribution patterns" by observing how the local ridge orientation varies around these points.

Kawagoe and Tojo [1984] describe a two-stage technique of determining the Henry classification of a given print. The first stage finds the LRO at an array of points and analyses this to find the location and types of all delta and core points. It is interesting to note that they evaluate the Poincare index about each singular point to confirm its nature (see chapter 4 where a detailed account of the use of Poincare indices is given). The second stage involves "flow tracing" to give a finer classification, and can discriminate, for example, between twin loops and whorls.

Grasselli [1968] classifies prints by smoothing and then linguistically analysing a matrix of sampled local ridge orientation values.

The topological coding method of Hankley and Tou [1968] yields the Henry classification of a print during the final stage of the coding process. See also section 1.3.4.

Rao and Balck [1976] determine a set of "ridge flow lines" from the ridges. Each "flow line" is syntactically analysed into a string of symbols, each symbol marking a change of direction of the line. Appropriate analysis of the strings then determines the Henry class of the print.

Rabinowitz [1980] discusses a technique using a decision tree algorithm employing information relating to singularities as well as global "ridge flow" and gives an evaluation of its performance and reliability.

T.C.M. Rao [1976] developed a technique which works from an enhanced and thinned image. By locating "features" (minutiae) and tracing the ridges between them, it can distinguish between whorls, loops and arches. Incomplete results and limitations to this approach were pointed out by C.V.K. Rao [1978].

1.3.4 Encoding.

Many different types of encoding are possible, and much research remains to be done in this field. Some techniques from the literature will be discussed briefly here.

1. The Home Office AFR Encoding technique [Millard 1974, 1983]

This was developed by the Scientific Research and Development

Branch (SRDB) of the Home Office and is currently in use at Scotland Yard as part of a very successful AFR system.

Since it is the presence and relative positioning of the minutiae that ensures the uniqueness of a fingerprint, it is this information that is encoded by the Home Office system. The algorithm, in simplified form, is as follows:

a. Locate all minutiae on the fingerprint image. Minutiae will not be distinguished according to type because, as we have seen, minutia type cannot always be determined reliably and repeatably. All that is used is the presence and position of the minutiae.

b. For each minutia M on the image :

   i. Determine which five minutiae M1..M5 to the right of it are the closest to M.

   ii. Determine the ridge counts RC1..RC5 from M to these minutiae.

c. As the encoding, store the following information:

   - identification numbers for M and M1..M5.
   - absolute (x,y) coordinates for M and M1..M5.
   - ridge counts between M and M1..M5.

2. The topological coding method of Sparrow

This was developed [Sparrow & Sparrow 1985a, 1985b; Sparrow 1986] in an attempt to produce an approach to coding that is distortion

invariant. Ideally, topological coding takes no account of any aspects of the fingerprint which can be expected to change with distortion, such as minutiae position and ridge orientation.

A four bit code is given to the various occurrences that can happen to a ridge as it is followed (such as termination, bifurcation, going off the edge of the print, etc.), and codes are stored for each ridge segment.

A disadvantage of this technique is that despite the aims behind its design, it still exhibits sensitivity to distortion of the image. The type of distortion to which it is most sensitive is that which causes minutiae to change type when a print is taken twice. If a minutia represents an ending in the first print and a bifurcation in the second, the four-bit codes assigned will change, causing the coding technique to give different results for two copies of the same print. The technique is also sensitive to noise in the print.

Another problem encountered was that a non-topological and hence distortion-sensitive technique had to be used to determine starting points for ridge tracing in the algorithm used.

3. Contextual analysis and topological coding

Hankley and Tou [1968] developed a coding technique which is essentially a syntactic analysis of the topology of the fingerprint ridges. The analysis produces a string of text as the encoding of the print. An advantage of this method is that the encoded string is meaningful and understandable to humans as a

description of the ridge structure.

Again this technique has the disadvantage that it is sensitive to imperfections in the input image. A change of minutia type for only one of the minutiae could cause the output string to change drastically. This would also be the case if a small part of the print were missing or of poor quality.

Another disadvantage is that, just as in Sparrow's method above, a non-topological method is used to determine start points for ridge tracing.

4. Other encoding methods

De Bruyne and Orum [1982] developed a technique for reducing the data storage requirements of fingerprint images. The low spatial frequency components are removed from the image, which can then be stored using fewer bits per pixel. The picture is then further compressed using a syntactic algorithm based upon a "four-tuple phrase-structured context-free" grammar. The net effect is a data compression of between 5.6 and 13 times, while still allowing reconstruction of a good-quality image. It should be noted that this compression technique would be useful for data storage or transmission, but not as a coding system for an AFR data base, since the encoded form of the data is not suited to being rapidly searched through.

Freeman [1974] describes several coding schemes for line-drawing images. These could be of use in compressing the data obtained by thinning a binary (thresholded) version of a fingerprint, such

as the one produced as the output of the enhancement algorithm of chapter 6.

Isenor and Zaky [1986] developed a topological encoding method for thinned binary fingerprint images. All of the minutiae are located, and the ridge segments joining pairs of minutiae are each uniquely numbered and their lengths in pixels measured. the ridges are then assigned level numbers as if they were the contour lines of a topographical map. Having identified all ridges and their parameters, a directed graph representing and describing the structure of the print is generated. Each node in the graph represents one ridge segment, and the graph contains all information needed in the matching process.

Nakamura et al [1981] produce a tree structure from a fingerprint image by labelling the minutiae as nodes and joining them together appropriately by straight line segments. This tree structure is then subjected to a rule-based reduction process yielding the "minimal spanning tree". Information describing the tree structure plus the absolute positional coordinates of each node are stored as the encoded form of the print.

Liu and Shelton [1970] describe a simple computer-assisted fingerprint encoding system requiring an operator to point out the singular points and minutiae. The computer then stores these data after transformation to standard coordinates.

The syntactic classification systems of Moayer and Fu which were described in section 1.3.2 can also be used as encoding systems simply by storing the outputs of the analysis as the encoded data.

1.3.5 Enhancement

Several methods of fingerprint enhancement have been proposed. Those that take into account the local ridge orientation almost always produce superior results.

Scherlowski [1974] attempted to enhance fingerprint images by filtering them in the Fourier domain using optical hardware. Unfortunately, his results were inconclusive, because defects within the hardware produced blurring effects which masked any improvements in the images. Several types of rotationally symmetric filter were used, but directional filtering was not tried. The use of optical hardware to enhance fingerprints is discussed in chapter 7.

Moayer and Fu [1975] enhanced fingerprints by thresholding each pixel against the mean value of a 12 by 12 pixel neighbourhood of it.

The method of T.C.M. Rao [1976] enhances a fingerprint by first thresholding it and then smoothing it using an algorithm which changes the value at a point depending upon the value of a boolean function of the pixels withion a 7 by 7 neighbourhood of it. This approach is not expected to work well for any but the best quality raw images.

Martelli and Montanari [1971] developed a smoothing technique for two dimensional data, which they applied to the problem of smoothing a matrix of sampled LRO values. Smoothness requirements are translated into a global cost function whose least squares

minimization gives an optimal smoothing.

The preprocessing stage of the Home Office AFR system [Millard 1983] performs enhancement of the raw image. The image is partitioned into an array of 32 by 32 pixel squares, and the average ridge orientation is determined within each square. It is assumed that the ridge orientation is constant within each of the squares. A simple nonlinear directional smoothing operator is applied to the pixels within each square, in order to smooth it in the orientation of the ridges. This approach works well except in regions of high ridge curvature, for which the assumption of piecewise constant ridge orientation is not valid. In regions of high curvature, therefore, the directional smoothing is omitted. Once the image has been smoothed, it is thresholded to yield a binary image as the enhancement.

Chiralo and Berdan [1978] propose a simple technique for the enhancement of latent prints. The print is partitioned into square subblocks of a given size, and the contents of each subblock is thresholded against the mean pixel value within the subblock.

The image restoration techniques due to Knutsson, Wilson and Granlund [1983] have been efficiently implemented in a special purpose image processing hardware system known as GOP [Granlund & Arvidsson 1983]. These techniques are described in more detail in chapter 2, but it should be mentioned here that the GOP has been used successfully to produce enhancements of fingerprint images.

1.3.6 Locating singular points

Chang [1980] defines two quantities for closed curves surrounding various numbers of cores and deltas:

i. the net angle change (NAC), which is the net change of LRO as the curve is traversed, and

ii. the maximum angle span (MAS), which is the total range over which the LRO varies around the curve.

He tabulates results for the NAC (which agree with the results given by the mathematical model for LRO which will be derived in chapter 4), but gives no theoretical justification for his results. By analysing NAC and MAS around small loops in the sampled LRO matrix, he is able to locate the individual cores and deltas.

Kawagoe and Tojo [1984] note that four different types of LRO pattern are possible around a point, corresponding to whorl, core, delta and nonsingular points. Two "whorlness" parameters are calculated for the points in a 3 by 3 neighbourhood of sampled LRO values around the given point. These are used to generate a string called the "flow code" which describes how LRO changes as one travels around the point. The flow code is simplified using rewriting rules until it resembles the flow code for one of the four basic LRO pattern types.

Nakamura et al [1982] use the "directional distribution pattern" (DDP) of the local ridge orientation around a point to locate cores and deltas by comparing them with standard DDPs for cores

and deltas.

1.3.7 Locating minutiae

Most techniques for locating minutiae are simple and involve tracing ridges on a (preferably thinned) binary image, often followed by a stage to eliminate spurious minutiae. The method of Singh et al [1977] is an example.

A different approach is that of Freedman and Hietanen [1967], who demonstrate how ridge bifurcations and endings can be detected by performing simple sequences of shifts and basic logic operations to the image. This approach, called "parallel neighbourhood logic", is shown to be related to similar operations performed by the human visual system. It is argued that it could be implemented efficiently in optics taking advantage of its inherent parallelism.

Shelman [1968] proposed a technique for locating ridge endings within a noisy fingerprint. The image is scanned both vertically and horizontally, building up two tables of candidate ridge endings. False endings due to noise are then detected and eliminated from the tables.

1.3.8 Fingerprint registration

If a fingerprint image is to be manually or automatically compared with another, it is necessary to align the two images so that they correspond as closely as possible before the actual matching is performed. This alignment process is called registration.

Pernus et al [1980] give a summary of registration techniques and present their own algorithm which aligns two prints given the coordinates and orientations of the minutiae. Their method is insensitive to rotation, translation and local stretching.

Rao and Balck [1978] present an algorithm for locating the "core" point of a print. Knowledge of the location of this point can assist during registration.

1.3.9 Theoretical and mathematical descriptions of fingerprints

Penrose [1965b] presented without proof several geometrical relationships which are valid for patterns of curve segments which may bifurcate or terminate. These results concern the relationships between the numbers of endings, cusps, etc that occur within specified areas of the pattern.

The same author [Penrose 1965a] presented an equation relating the numbers of cores and deltas that are encountered on the palms of the hand and the soles of the feet. No mathematical justification for this result was given, although a clear heuristic explanation was provided.

The mathematical theory of regular curve families describes families of curves which are locally parallel except at isolated singular points, and could therefore potentially be applied to fingerprints. This theory was developed by Whitney [1933] and Kaplan [1942, 1955] and generalized by Boothby [1951a, 1951b]. See also Beck [1974]. In these presentations, however, the singularities were always required to be of the multiple saddle

point type found in the level curves of harmonic functions. This ensured that the theory could not be applied to fingerprints since core and delta singularities are not of this type. In this regard, Bewsher [1967] wrote:

> In extending this work [to fingerprints], one of the first things to be noted is that singularities in fingerprints involve one or three ridges, whereas singularities in harmonic functions always involve an even number of lines.

Zucker [1985, 1986] divided directional structures within images into two types. Any one-dimensional structure such as a contour line, the outline of an object or an individual fingerprint ridge is of Type I. Any two-dimensional structure, such as an area of oriented texture or the entire surface of a fingerprint is of Type II. Zucker described both types of structure in terms of vector fields. For a Type I structure the vector field is defined on a curve within the plane, and for a Type II structure it is defined on a region of the plane.

In chapter 4, it will be seen that the "direction field" from differential geometry provides a correct mathematical description of directionality within fingerprints and other images. This concept will be used in the derivation of an original mathematical model of fingerprint LRO.

1.3.10 **Optical processing of fingerprints** - See chapter 7.

1.4 An approach to fingerprint enhancement by directional filtering

The purpose of this section is to introduce and summarize the fingerprint enhancement technique which is the main result of this thesis. This section is presented in general terms and is intended simply to help the reader keep in view the overall plan as the material is covered in detail in chapters 2 to 8.

Fingerprint images have an exceptionally large amount of noise and unwanted information, as was mentioned in the discussion of enhancement in section 1.2.2. In order to produce a good enhancement, it is necessary to remove as much of this as possible. Fourier domain techniques are particularly well suited to this purpose.

Because the ridge spacing remains within a relatively narrow range of values, the meaningful data in the Fourier transform of the print is contained within an annular region around the origin. All components not in this annulus can be filtered out.

The ridge orientation, on the other hand, varies widely, which means that Fourier components at all orientations within this annulus are of significance. However, near to any one point in the image the orientation does not vary much and the components corresponding to useful information for that neighbourhood only come from a small range of orientations within the annulus. We can therefore use narrower filters and remove more unwanted components if we allow the filter to be non-stationary, i.e. vary from point to point within the image, adapting itself to filter in

the same orientation as the ridges.

Instead of scanning the image while filtering directionally, one could produce a series of "directional components" of the image, each component being a version of the image that has been filtered in a different direction. A knowledge of the actual ridge orientation at each point then enables one to build up a smoothed directionally filtered version of the print by selecting at each point the pixel value from the component whose orientation is the LRO at that point.

This filtered image is then suitably thresholded to give the binary enhanced version of the print. This is, if necessary, thinned to reduce the ridges to single-pixel width.

Chapter 2 develops the filters used and chapter 3 discusses reducing the computational cost so that they can be implemented on a general-purpose computer.

Chapter 4 introduces a mathematical model of LRO. Chapter 5 describes methods of determining LRO at a point, and applies the LRO model to develop a technique to interpolate between sampled values of LRO so that the orientation will be known at all image points.

Chapter 6 brings together the filtering results of chapters 2 and 3, and the orientation results of chapters 4 and 5, to present and evaluate the full enhancement algorithm. Several prints and their enhancements are included to demonstrate the technique.

Chapter 7 discusses alternative implementations of the enhancement

technique using optical hardware.

Chapter 8 investigates the use of an adaptive window based hardware system to process fingerprint images.

CHAPTER 2

FREQUENCY DOMAIN PROCESSING OF FINGERPRINT IMAGES

This chapter develops filters which are useful in the process of enhancing fingerprints. Section 2.1 examines image enhancement and the related topic of image restoration. In section 2.2 the properties of fingerprints which make Fourier domain processing particularly appropriate are discussed, and in the rest of the chapter the filters are developed.

An image may be defined as a positive real-valued function of two spatial variables. These variables may be of finite or infinite extent. The image is said to be analogue if the two variables take on a continuum of values, or digital if the variables are sampled so that they are restricted to certain discrete values. Usually the sampling of the spatial variables will be done at equally-spaced intervals, resulting in an image sampled on a square grid.

Because of the finite amount of storage space available on a general-purpose digital computer, an analogue image must first be sampled in order to convert it into a digital image of finite extent before it can be stored and then processed. In this digital image, each position is called a picture element (pixel) and the image value at that point is called the grey level or

pixel value.

In this chapter it will be assumed that the image is expressed as an analogue signal of infinite extent, rather than as a digital signal, except where digital signals are explicitly referred to. This approach is taken because existing one-dimensional filtering techniques for infinite-extent analogue signals are most simply and easily generalised to perform the sort of two-dimensional operations that will be required.

In implementing the filters in optical hardware as described in chapter 8, their analogue nature poses no problems since the optical hardware is also analogue. However, if the filters are to be implemented using a general-purpose computer, they must be transformed in some way into digital filters, since in this case the images to be filtered will be stored in a digital form. It will be seen in the next chapter that if a sufficiently fine sampling grid is used, the required digital filter can be obtained simply by sampling the corresponding analogue filter.

The techniques and properties which are described in sections 2.1 and 2.2 for analogue images are, unless specifically stated to the contrary, equally applicable to digital images.

2.1 Introduction to image enhancement and restoration

General introductions to the topics of image enhancement and image restoration are given in the books by Gonzalez & Wintz [1977], Rosenfeld & Kak [1976], Pratt [1978] and Hall [1979], and in the papers by Huang et al [1971], Andrews [1974] and Hunt [1975]. The book edited by Young & Fu [1986] is one of the most recent and has contributions by several of the leading researchers in image processing.

2.1.1 Image enhancement

Image enhancement is defined by Gonzalez and Wintz [1977] as the processing of a given image so that the result is more suitable than the original for a specific application. Enhancement techniques are therefore strongly problem-dependent, since a result which is ideal for one application may be entirely unacceptable for another.

In terms of this very broad definition of enhancement, image restoration (see section 2.1.2) and other image processing operations such as segmentation and thresholding could be regarded as examples of image enhancement, provided that the result is more suitable than the original for the application.

In practice, however, image enhancement is usually regarded as referring to those ad hoc or basically heuristic procedures which have been designed to manipulate an image in order to make it more pleasing in appearance to the viewer, by taking advantage of psychophysical aspects of the human visual system.

A brief outline of some classical enhancement techniques is given below. These techniques will be described for digital images of finite extent.

Histogram modification.

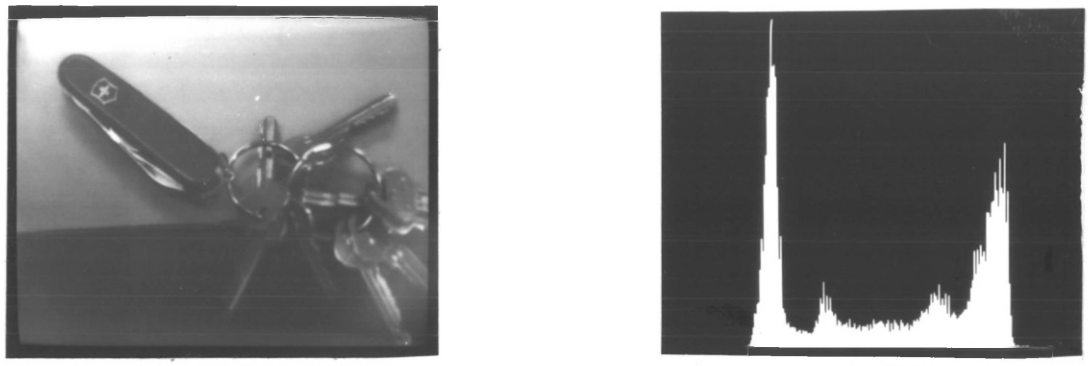
The histogram of an image is a graph with the various grey levels g on the abscissa and the number of pixels n(g) having the grey level g on the ordinate. The histogram, when suitably scaled, becomes the probability density function [Papoulis 1965] of the grey levels within the image. As an example, figure 2.1 shows a picture together with its histogram. In this example, part of the background is dark and the rest is bright. The pixels of the background are therefore represented in the histogram by two peaks, one near each end of the grey scale.

Histogram modification techniques involve altering the histogram in a desired way by transforming the grey levels using a function of the form

$$g' = T(g)$$

which maps the grey level g of each pixel to a new grey level g'. T is a monotonically increasing function so that the order from black to white is preserved.

In histogram equalisation T is chosen such that the resulting histogram is flat or as nearly flat as possible. In histogram specification T is chosen so that the histogram approximates some specified function.



(a) (b)

Figure 2.1. The histogram. (b) shows the histogram of the image (a). Note that the dark and light regions of the background give rise to the two peaks in the histogram.

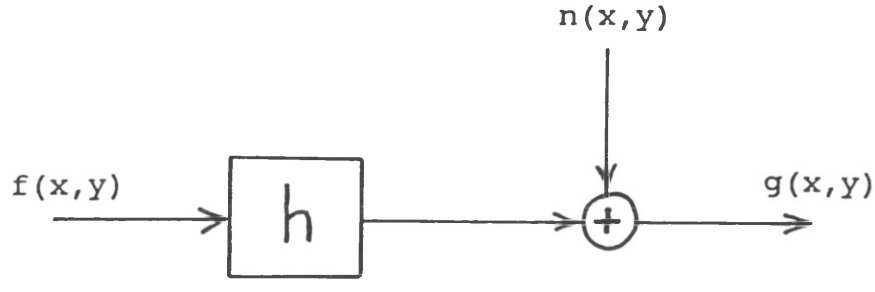
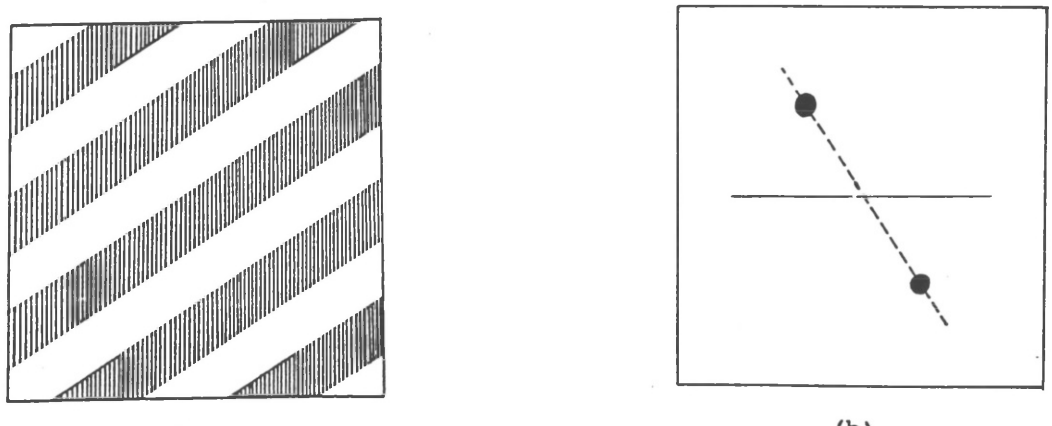


Figure 2.2. Image degradation model. The image f(x,y) is degraded by the operator h and additive noise n(x,y) to produce the degraded image g(x,y).



(a) (b)

Figure 2.3. An idealised region (a) of a fingerprint, and (b) the magnitude Fourier transform of this region, showing a concentration of energy around two radially opposite points.

Image smoothing.

Smoothing techniques are used to remove local spurious effects such as single-pixel noise, which may be caused by poor sampling systems or transmission channels. Two spatial smoothing techniques are neighbourhood averaging and median filtering, in which each pixel is replaced by the average (or, respectively, the median) value of the pixels within a given sized neighbourhood of it. Low pass filtering is a Fourier domain method of smoothing images.

Coherent averaging

If several noisy copies of an image are available and the noise is uncorrelated with the image and has zero mean, then an improved image can be produced as the average of several noisy images. The more images are averaged together, the less noisy will be the resulting average.

Image sharpening

Sharpening techniques can be used to emphasize the edges and lines within an image. This can be done spatially by performing a differentiation-like operation upon the image (e.g. the Laplacian edge detection algorithm), or in the Fourier domain using a high pass filter.

Pseudo-colour

The human eye can distinguish more easily between different colours than between adjacent shades of grey. Pseudo-colour techniques take advantage of this by assigning a colour to each pixel value so that differences in greyscale become much more apparent, thereby increasing the contrast between the different pixel values. The colours used do not usually have any significance in themselves, since they are being used solely as an aid to distinguishing different pixel values.

2.1.2 Image restoration

In image restoration, as in image enhancement, the aim is also to improve an image in some sense. However, the approach is somewhat different. In image restoration techniques, the image is regarded as having been degraded in quality by some process such as noise, blurring or distortion. The aim is then to recover the original image by reversing the degradation process, using a priori knowledge or reasonable assumptions about the nature of the degradation process. Of course, it is usually not possible to recover the original image exactly: instead one must strive to obtain as good an estimate as possible of the desired image. Therefore, it is necessary to define a criterion of goodness in terms of which an optimal estimate can be obtained.

It will now be argued that improving the quality of a fingerprint in order to make it more suitable for applications involving personal identification is a process of enhancement rather than restoration. It was explained in chapter 1 that the desired image

must contain only information useful to personal identification, i.e. the ridge and minutiae structure. All other information, such as pores and breaks caused by folds in the skin, is irrelevant to the application and must be removed. Although a fingerprint will usually have suffered from the effects of degradation, removal of these effects will not in itself produce the desired image, since the irrelevant information will not be suppressed. Therefore the desired image is an enhancement, not purely a restoration.

Nonetheless, restoration techniques which have been presented in the literature have been of particular interest during the development of the fingerprint enhancement technique which is presented in this thesis. Some recent advances in image restoration have involved research into local image properties such as directionality, and the development of anisotropic and nonstationary filters. These results are all of interest to fingerprint enhancement.

Image restoration techniques will now be discussed in more detail. The model of image degradation used in most approaches to restoration is the one in figure 2.2. The signals are regarded as analogue and infinite in extent.

f(x,y) is the original (undegraded) image, which is operated upon by a degradation operator h, and then an additive noise term n(x,y) is included to yield the degraded image g(x,y). The restoration problem involves finding an estimate f^(x,y) of f(x,y), given knowledge of the degraded image g(x,y), the

degradation operator h and the statistics of the noise n(x,y). The estimate should be optimal with respect to the error criterion that is being used.

The model in figure 2.2 is described by the system equation

$$g(x,y) = h(f(x,y)) + n(x,y)$$

The degradation operator h is <u>linear</u> if

$$h(k_1 f_1(x,y) + k_2 f_2(x,y)) = k_1 h(f_1(x,y)) + k_2 h(f_2(x,y))$$

and the system equation can then be given in the form

$$g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x,y;x',y') f(x',y') dx' dy' + n(x,y)$$

where the degradation is now represented by the (position-dependent) <u>point spread function</u> h(x,y; x',y').

h is <u>stationary</u> (<u>position-independent</u>) if

$$h(f(x-\alpha,y-\beta))|_{xo,yo} = h(f(x,y))|_{xo-\alpha,yo-\beta}$$

in which case the system equation becomes

$$g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x-x',y-y') f(x',y') dx' dy' + n(x,y)$$

$$= h(x,y) \otimes f(x,y) + n(x,y),$$

where $\otimes$ is the convolution operator. In the Fourier domain, this equation becomes

$$G(u,v) = H(u,v)F(u,v) + N(u,v).$$

Because statistical information is all that is known about the signals $f(x,y)$, $\hat{f}(x,y)$, $n(x,y)$ and $g(x,y)$, these signals are best dealt with as being examples of statistical random fields [Rosenfeld & Kak 1976]. The concepts of autocorrelation and spectral density of random fields will now be introduced.

The autocorrelation of a random field $f(x,y)$ is defined by

$$R_{ff}(x_1,y_1;\ x_2,y_2) = E\{f(x_1,y_1)\ f(x_2,y_2)\}$$

where $E\{\bullet\}$ is the expected value operator [Papoulis 1965].

If the statistics of f are independent of position, then f is said to be stationary, and the autocorrelation can be expressed more simply as a function of two variables [Rosenfeld & Kak 1976] :

$$R_{ff}(\alpha,\beta) = E\{f(x+\alpha,y+\beta)\ f(x,y)\}.$$

The spectral density $S_{ff}(u,v)$ of a stationary random field $f(x,y)$ is the 2-d Fourier transform of the autocorrelation function, i.e.

$$S_{ff}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} R_{ff}(\alpha,\beta)\ \exp(-j2\pi(\alpha u + \beta v))\,d\alpha d\beta$$

The classical solution to the restoration problem is the stationary Wiener (or least squares) filter, of which several variants exist [Helstrom 1967; Slepian 1967; Rino 1969; Horner 1969; Sondhi 1972; Pratt 1972; Hunt 1973]. In this solution the assumption is made that the distortion operator is both linear and stationary, and that the noise is stationary. The error criterion used is the mathematically simplest one: least squares.

That is, it is sought to minimise the quantity

$$\varepsilon^2 = E\{ (f(x,y) - \hat{f}(x,y))^2 \}$$

The result obtained under these assumptions is [Gonzalez & Wintz 1977 page 209] :

$$\hat{F}(u,v) = M(u,v)G(u,v),$$

$$\text{where } M(u,v) = \frac{1}{H(u,v)} \cdot \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_{nn}(u,v)/S_{ff}(u,v)}$$

where $S_{nn}(u,v)$ and $S_{ff}(u,v)$ are the spectral densities of $n(x,y)$ and $f(x,y)$ respectively.

For simplicity, it is often assumed that the degradation is caused entirely by the noise. Then the degradation operator h becomes the identity operator, and therefore

$$M(u,v) = \frac{S_{ff}(u,v)}{S_{ff}(u,v) + S_{nn}(u,v)}$$

If in addition the noise $n(x,y)$ is stationary white noise of zero mean and variance σ_n^2, then

$$M(u,v) = \frac{S_{ff}(u,v)}{S_{ff}(u,v) + \sigma_n^2} .$$

It is almost always the case that the noise is of wider bandwidth than the signal. This is certainly true for white noise, which has flat spectral density, whereas the signal will have its energy concentrated in the frequency components near the origin. Therefore, the signal to noise ratio $S_{ff}(u,v)/S_{nn}(u,v)$ should be

large for frequencies for which $\rho = \sqrt{(u^2 + v^2)}$ is small, but decrease to zero as $\rho \rightarrow \infty$. Consequently, $M(u,v) \rightarrow 0$ as $\rho \rightarrow \infty$.

Therefore, the Wiener filter M(u,v) has a low-pass character. This has proved to be a major disadvantage in practice, since the low-pass filtering effect often leads to unacceptable blurring of lines and edges within the picture.

Several attempts to overcome this problem have taken advantage of a psychophysical effect, known as the masking effect [Abramatic & Silverman 1979], whereby noise is subjectively less noticeable in regions of higher edge content. In these approaches, a nonstationary filter is developed whose filtering effect is different for regions of high and low edge content.

An early example of such a filter is that developed by Anderson and Netravali [1976]. However, their approach is very expensive computationally, since the entire filter optimisation process must be performed for each image point.

Abramatic and Silverman [1979] developed nonstationary filters which take account of the masking effect, but are much simpler computationally. A visibility function $\alpha(x,y)$ with $0 <= \alpha(x,y) <= 1$ is introduced to express the masking effect as a function of position within the picture. For points whose immediate neighbourhood contains no edge information, α takes the value 1. As the local edge content increases, α decreases monotonically to zero. $\alpha(x,y)$ can be estimated from the image using an edge detection algorithm.

Abramatic and Silverman proposed two different filters based on the use of this visibility function. The first of these is given by

$$M(u,v;\ x,y) = \frac{S_{ff}(u,v)}{S_{ff}(u,v) + \alpha(x,y) \cdot \sigma_n^2}$$

This filter varies between the identity mapping and the pure Wiener filter as α increases from 0 to 1. This ensures that the edge information is not blurred since $\alpha \approx 0$ in the vicinity of edges.

This filter has the advantage over the approach of Anderson and Netravali that it does not need to be optimised at each point; however, it still varies from point to point in a computationally inconvenient way. The second filter proposed by Abramatic and Silverman is expressed as a position dependent linear combination of the identity filter and the Wiener filter:

$$M(u,v;\ x,y) = \alpha(x,y)\frac{S_{ff}(u,v)}{S_{ff}(u,v) + \sigma_n^2} + \beta(x,y)$$

where $\beta(x,y) = 1 - \alpha(x,y)$.

This filter retains the edge-preserving properties of the previous one.

Granlund [1978] proposed a "generalised operator" approach to image processing in which it was hypothesized that edge information within the neighbourhood of a point is (when present) predominantly linear, i.e. can be described by a well-defined local orientation of image anisotropy. This approach led to the

development of an image restoration technique by Granlund, Knutsson and Wilson, elements of which are described in several sources [Knutsson & Granlund 1980, 1983; Knutsson 1982; Knutsson et al 1981, 1983a, 1983b; Granlund et al 1983; Wilson et al 1983, 1985].

In the above work, the conclusion is drawn that the masking effect upon noise in the vicinity of an anisotropy (linear feature) depends not only upon the magnitude of the local edge information but also upon the orientation of the noise relative to the orientation of the edge. Noise perpendicular to an edge has the greatest degradational effect, with the effect of the noise being proportional to the square of the cosine of the difference between the orientation of the edge and the orientation of the noise.

Based upon the above considerations, and the assumption that the signal spectral density $S_{ff}(u,v)$ of the signal is isotropic, the following "equivalent signal" spectral density function (see [Knutsson et al 1983a]) is proposed:

$$S_{f',f'}(\rho,\phi;\ x,y) = S_{ff}(\rho) + \beta(x,y)\,(1-\gamma(x,y))\cos^2(\theta(x,y) - \phi)\sigma_n^2 + \beta(x,y)\gamma(x,y)\sigma_n^2$$

where $\rho = \sqrt{(u^2 + v^2)}$

$\phi = \arctan(v/u)$

$\beta(x,y)$ is a measure of local edge content, as used earlier

$\theta(x,y)$ is perpendicular to the orientation of the local

anisotropy

$\gamma(x,y)$ varies between 0 and 1 giving a measure of the curvature, or rate of change of $\theta(x,y)$.

The resulting restoration filter is

$$M(\rho,\phi;\ x,y) = \frac{S_{f',f'}(\rho,\phi;\ x,y)}{S_{ff}(\rho) + \sigma_n^2}$$

$$= \alpha(x,y)H_i(\rho) +$$
$$\beta(x,y)(1-\gamma(x,y))H_a(\rho,\phi;\ \theta(x,y)) +$$
$$\beta(x,y)\gamma(x,y)H_i'(\rho)$$

$$H_i(\rho) = \frac{S_{ff}(\rho)}{S_{ff}(\rho) + \sigma_n^2}$$

where $$H_a(\rho,\phi;\ \theta(x,y)) = \frac{\sigma_n^2.\cos^2(\theta(x,y) - \phi)}{S_{ff}(\rho) + \sigma_n^2}$$

$$H_i'(\rho) = \frac{\sigma_n^2}{S_{ff}(\rho) + \sigma_n^2}$$

This filter consists of a linear combination of three parts:

(1) The stationary Wiener filter $H_i(\bullet)$. This is active when $\beta<<1$ (and therefore $\alpha \approx 1$), i.e. in the absence of edge information.

(2) An anisotropic (directional) high-pass filter $H_a(\bullet,\bullet;\theta)$. This depends upon the orientation of local anisotropy and is active when $\beta \approx 1$ and $\gamma<<1$, i.e. when edges of low

curvature are present.

(3) An isotropic high-pass filter $H_i'(\bullet)$. This is active when $\beta \approx 1$ and $\gamma \approx 1$, i.e. when edges of high curvature are present.

To implement this filter, it is necessary to obtain estimates for the local orientation $\theta(x,y)$, the edge content or "first bias image" $\beta(x,y)$ and the curvature or "second bias image" $\gamma(x,y)$.

This filter has been applied with success to a wide variety of images, including fingerprint images, and it is implemented in hardware on the powerful GOP image processing system [Granlund 1981; Granlund & Arvidsson 1983].

The above approach is of interest as background to this thesis in the following respects:

(1) It deals with the concepts involved in directional filtering, use of which will be needed in the development of the filters later in this chapter.

(2) The problems involved in estimating the local direction $\theta(x,y)$ and the bias images $\beta(x,y)$ and $\gamma(x,y)$, are similar to those encountered when trying to estimate fingerprint local ridge orientation and curvature. See chapter 5.

2.2 Representation of fingerprint images in the Fourier domain

The generalised operator approach of Granlund [1978] assumes that spatial orientations and frequencies defined within a neighbourhood of an image point are localised not only in the spatial domain, but also in the Fourier domain. This section seeks to demonstrate the validity of this assumption when applied to the particular case of fingerprint images, by demonstrating that the LRS and LRO parameters are localised in both domains.

Because of the particular importance of LRO and LRS in describing fingerprints (see chapter 1), this result will imply that the Fourier domain provides a natural way of representing the information contained within fingerprint images.

In addition, a knowledge of LRS and LRO at each point and of the manner in which they are localised in the Fourier domain will make it possible to filter out much of the unwanted noise using narrow bandwidth Fourier filters. The noise content of the image is greatly reduced by such filters because the Fourier components corresponding to the noise are not localised.

Firstly, however, it is necessary to look at the Fourier domain in two dimensions, to establish the exact nature of the relationship between the various Fourier components and their corresponding spatial equivalents. Also, it is important to establish clearly what similarities and differences there are between the one- and two-dimensional Fourier domains, i.e. to establish in what ways the 2-d domain can be regarded as a generalisation of the 1-d domain.

For a real-valued spatial signal f(x,y) the Fourier transform has Hermitian symmetry, i.e.

$$F(u,v) = F^{*}(-u,-v), \text{ for all u and v,}$$

where u and v are Cartesian coordinates with the DC frequency at (0,0). (This is also true for digital signals if the discrete Fourier transform is used).

Therefore, the simplest Fourier domain signal (corresponding to a real spatial image) that we can investigate consists of a single component or impulse at (u_o,v_o), and a corresponding impulse at $(-u_o,-v_o)$ as required for Hermitian symmetry, i.e.

$$F(u,v) = \delta(u - u_o, v - v_o) + \delta(u + u_o, v + v_o)$$

or, in polar coordinates,

$$F(\rho,\phi) = \delta(\rho - \rho_o, \phi - \phi_o) + \delta(\rho - \rho_o, \phi - \phi_o + \pi)$$

where

$$\rho_o = \sqrt{(u_o^2 + v_o^2)}$$

$$\phi_o = \arctan v_o/u_o$$

The inverse Fourier transform of this is

$$f(x,y) = \underline{F}^{-1}\{ F(u,v)\}$$

$$= \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(j(ux + vy))\, F(u,v)\, du\, dv$$

$$= \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(j(ux + vy))\, [\, \delta(u-u_o, v-v_o) + \delta(u+u_o, v+v_o)\,]\, du\, dv$$

$$= \frac{1}{4\pi^2} \exp(j(u_o x + v_o y)) + \exp(-j(u_o x + v_o y))$$

$$= \frac{1}{2\pi^2} \cos\ (u_o x + v_o y)$$

$$= \frac{1}{2\pi^2} \cos\ (\underline{k} \bullet \underline{r})$$

where

$$\underline{k} = (u_o, v_o)$$

$$\underline{r} = (x, y)$$

$f(x,y)$ is therefore the equation of a plane sinewave of infinite spatial extent. This sinewave has wavelength

$$\lambda_o = 2\pi/|\underline{k}| = 2\pi/\rho_o$$

and the orientation of its troughs and crests is

$$\theta_o = \phi_o + \pi/2.$$

Notice that the wavelength λ_o of this plane wave is independent of the Fourier domain coordinate ϕ_o, and that the orientation θ_o of the wave is independent of the Fourier domain coordinate ρ_o. We conclude that the Fourier domain radial coordinate ρ determines the wavelength (or frequency) of the corresponding spatial wave, and the Fourier domain angular coordinate ϕ_o determines the orientation of this wave. This implies that there is a natural separation or independence between the two polar coordinates ρ and ϕ in the Fourier domain, and therefore that polar coordinates reflect more clearly the physical significance of Fourier components than do the Cartesian coordinates u and v.

Thus we see that it is the ρ coordinate that is analogous to the 1-d frequency ω, because it is ρ that determines the physical wavelength of the signal, independent of the direction ϕ. Of course, the direction coordinate ϕ does not have an analogue in 1-d since there is no concept of directions for 1-d signals.

It was seen above that a spatial plane wave of infinite extent corresponds to a single pair of Fourier components. Now let us see what the Fourier representation is of a plane wave of finite extent. This is of particular interest to fingerprint processing since, locally, a fingerprint can be approximated by a plane wave of wavelength LRS(x,y) and orientation LRO(x,y). An entire fingerprint could be visualised as being built up as a sum of such localised plane waves, and, by superposition, the Fourier transform of the entire fingerprint would be the sum of the Fourier transforms of the individual localised plane waves used to build up the spatial fingerprint image.

Suppose that the plane wave, localised near the point (x_o, y_o), is defined as the product of an infinite plane wave and a Gaussian function, i.e.

$$f(x,y) = \cos(u_o x + v_o y) \; . \; \exp[-\alpha((x-x_o)^2 + (y-yo)^2)]$$

The Fourier transform of this product is the convolution of the transforms of the two terms, i.e.

$$F(u,v) = 2\{[\delta(u-u_o, v-v_o) + \delta(u+u_o, v+v_o)] \otimes \exp[-(u^2+v^2)/4\alpha]\}$$

(where $\otimes$ is the convolution operator)

$$= 2\ \exp\{-j[(u-u_o)x_o + (v-v_o)y_o]\}.\exp\{-[(u-u_o)^2 + (v-v_o)^2]/4\alpha\}$$

$$+ 2\ \exp\{-j[(u+u_o)x_o + (v+v_o)y_o]\}.\exp\{-[(u+u_o)^2 + (v+v_o)^2]/4\alpha\}$$

Therefore, the effect of limiting the spatial extent of the plane wave is to broaden the Fourier domain impulses representing the wave into Gaussian peaks, still centred at (u_o,v_o) and $(-u_o,-v_o)$. The more tightly restricted the wave is spatially, the more pronounced is this broadening effect. This is a direct consequence of the uncertainty principle in image processing [Wilson & Granlund 1984]. The spatial location (x_o,y_o) of the localised plane wave is reflected only in the phase of the Fourier transform and does not affect its magnitude.

The above observations will now be applied to fingerprint images. Figure 2.3 is an idealised illustration of a small region of a fingerprint together with its Fourier transform. The size of this region has been chosen small enough that the ridges within it are approximately equally-spaced and the ridge orientation approximately constant, i.e. the values of LRS and LRO may be regarded as fixed. Therefore, this image may be approximated within the window by a localised plane wave of wavelength

λ = LRS, and orientation

θ = LRO.

Therefore by our earlier discussion, the Fourier transform of this image will consist of two radially-opposite peaks at distance

$\rho = 2\pi/\text{LRS}$

from the origin, and with the line joining them oriented at angle

$$\phi = LRO + \pi/2.$$

The width of the peaks will be inversely related to the size of the region within which the wave is localised. In addition, in a real fingerprint, the assumption of constant LRO and LRS within the region will not be satisfied exactly. The peaks will therefore be further broadened by the following effects:

- The ridge curvature (rate of change of LRO with position). The more strongly the ridges are curved, the greater will be the variation of ridge orientation within the region. Variation in LRO causes the Fourier components to spread in the ϕ direction.

- The ridge expansion (rate of change of LRS with position). The greater the ridge expansion, the wider will be the range of spacings included within the window. Variation in LRS causes the components to spread in the ρ direction. This effect is small because of the slowly varying nature of LRS.

Suppose now that the small region is scanned over the print. Since the LRS does not vary widely (see chapter 1), it is clear that the distance ρ from the origin of the pair of peaks will not vary widely during this process. However, since the ridge orientation can (and in most cases does) take on all values from 0 to π, the orientation ϕ of the peaks will vary widely.

We have seen that the fingerprint can be regarded as the superposition of local plane waves. Figure 2.4 illustrates that the components of interest within the Fourier transform of a fingerprint image are contained within an annular region centred

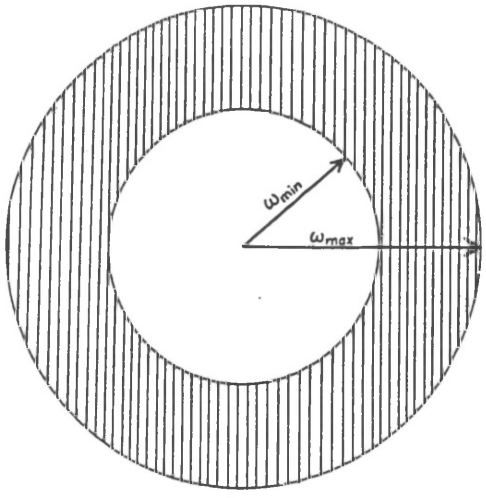
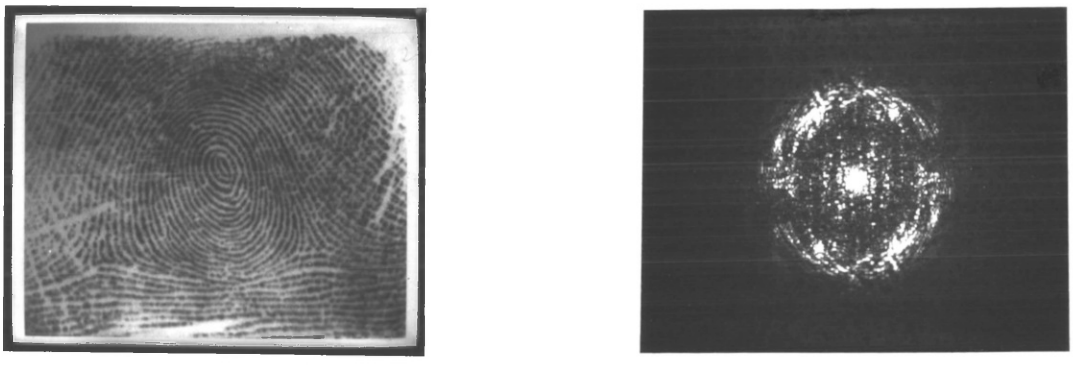


Figure 2.4. The annular region in the Fourier domain within which the components contributing to the pattern of parallel ridges are located.



(a) (b)

Figure 2.5. A fingerprint image (a) and its magnitude discrete Fourier transform (b).

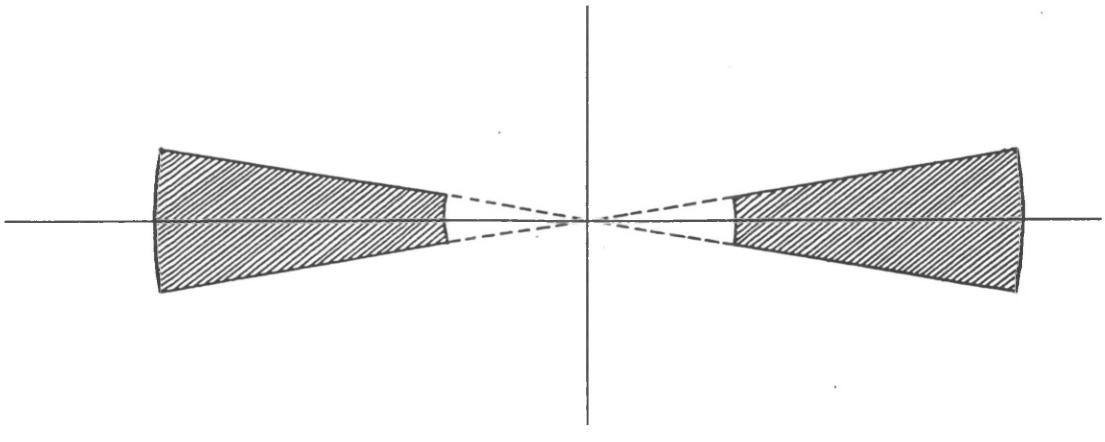


Figure 2.6. The "ideal" directional bandpass filters proposed by Ikonomopoulos & Kunt.

about the frequency (0,0). This reflects the fact that LRS (corresponding to the ρ coordinate) is constrained within a range of values, whereas the LRO (corresponding to the ϕ coordinate) may take on any value.

The photographs of Figure 2.5 show a fingerprint and its magnitude discrete Fourier transform. The annular region is clearly visible.

2.3 Position independent (stationary) linear filtering

Stationary Fourier filtering of one-dimensional continuous or sampled signals is a topic which has been very extensively researched and many excellent 1-d filtering techniques have been developed [Van Valkenburg 1982; Oppenheim & Schafer 1975; Rabiner & Gold 1975], some of which will be discussed in this section. It would clearly be advantageous to use this extensive base of knowledge when designing two-dimensional filters. Therefore, the problem arises of how best to generalise the theory of 1-d filters to two dimensions.

We saw in section 2.2 that in the 2-d Fourier plane it is the radial coordinate ρ that is analogous to the 1-d frequency ω, because it is ρ that determines the physical wavelength of the signal, independent of the orientation ϕ.

This natural splitting of the Fourier plane into ρ and ϕ coordinates suggests that it would be natural to allow the filter function $H(\rho,\phi)$ to be separable in ρ and ϕ, i.e.

$$H(\rho,\phi) = H_{radial}(\rho).H_{angle}(\phi)$$

so that its radial frequency response and directional properties can be independently manipulated.

The following two subsections will therefore consider purely radial and purely directional filters separately. A filter with any desired radial and directional properties can then be produced as the product of two filters, one of each type.

2.3.1 Radial (frequency) filters

Performing a 1-d stationary filtering operation in the Fourier domain involves multiplying the Fourier transform $F(\omega)$ of the signal $f(x)$ by a filter transfer function $H(\omega)$ and inverse transforming the product to give the filtered signal $g(x)$. The filter function $H(\omega)$ can be regarded as a weighting function which selectively emphasises or de-emphasises the various frequency components in the input signal.

In two dimensions we also wish to perform such a weighting operation upon components of different frequency. Given a 1-d filter $H_{1-d}(\omega)$, a 2-d filter which affects the radial frequency components in an identical manner can be produced simply by replacing the 1-d frequency component ω by its 2-d analogue ρ.

This yields $H_{2-d}(\rho,\phi) = H_{1-d}(\rho)$ where $\rho = \sqrt{(u^2 + v^2)}$.

Such a 2-d filter will have "similar" properties to the corresponding 1-d filter, in the sense that all its radial frequency properties (such as its centre and cut-off frequencies, its rate of roll-off and degree of ripple) will be identical to the frequency properties of the 1-d filter.

Clearly $H_{2-d}(\rho,\phi)$ will be a circularly-symmetric function, i.e one whose effects depend only upon the wavelengths of the components of the signal and not upon their orientations.

The rest of this section describes examples of classical analogue 1-d filters. Fingerprint signals have their significant components concentrated within a band of radial frequencies

described by the annulus of the previous section, and bandpass filtering is therefore appropriate to such signals. Classical filters are usually specified in terms of their normalised low-pass prototypes: to convert to other filters the following transformations [Van Valkenburg 1982] may be applied:

- To low-pass: Replace ω by ω/ω_c
- To band-pass: Replace ω by $\dfrac{\omega^2 - \omega_o^2}{\omega\ \omega_{BW}}$
- To high-pass: Replace ω by ω_c/ω

where ω_c, ω_{BW}, and ω_o are the required cut-off frequency, bandwidth and centre frequency respectively.

2.3.1.1 The ideal filter

The ideal bandpass filter passes unchanged all frequency components within the passband and rejects all other components. Therefore it is described by:

$$H(\omega) = \begin{cases} 1 & \text{for } \omega_o - \omega_{BW}/2 < \omega < \omega_o + \omega_{BW}/2 \\ 0 & \text{for other } \omega \end{cases}$$

The ideal filter is a discontinuous function, and this leads to oscillation or ringing in the spatial domain which is often unacceptable.

2.3.1.2 The Butterworth filter

The Butterworth filter is a continuous approximation to the ideal

filter which has the property [Oppenheim & Schafer 1975, page 211] that the amplitude response is maximally flat in the pass-band.

The nth-order prototype low-pass Butterworth filter is defined by:

$$|H(\omega)|^2 = \frac{1}{1 + \omega^{2n}}.$$

Transforming this to a band-pass filter yields:

$$|H(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega^2 - \omega_o^2}{\omega\ \omega_{BW}}\right)^{2n}}$$

$$= \frac{(\omega\ \omega_{BW})^{2n}}{(\omega\ \omega_{BW})^{2n} + (\omega^2 - \omega_o^2)^{2n}}$$

2.3.1.3 **The Chebyshev filter**

The Chebyshev filter achieves a more rapid cut-off between passband and stopband than the Butterworth filter, at the expense of the presence of ripple within either the passband or the stopband.

The nth order prototype lowpass Chebyshev filtered is defined by:

$$|H(\omega)|^2 = \frac{1}{1 + \varepsilon^2 [C_n(\omega)]^{2n}}$$

where $C_n(\omega) = \cos(n \arccos(\omega))$ is the nth order Chebyshev polynomial, and ε^2 is a constant indicating the magnitude of passband ripple.

The equivalent bandpass filter is

$$|H(\omega)|^2 = \frac{1}{1 + \varepsilon^2 \left[C_n \left(\frac{\omega^2 - \omega_o^2}{\omega \, \omega_{BW}} \right) \right]^{2n}}$$

2.3.1.4 **The elliptic filter**

The elliptic filter achieves an even faster rate of cutoff than the Chebyshev filter, at the expense of allowing ripple both within the passband and the stopband. Its nth order lowpass prototype is:

$$|H(\omega)|^2 = \frac{1}{1 + \varepsilon^2 [U_n(\omega)]^2}$$

where $U_n(\omega)$ is a Jacobian elliptic function, and ε^2 a measure of the ripple.

Any one of the above filters would produce good results. The Butterworth filter was chosen because the explicit mathematical formula for its bandpass version is simpler than that of the Chebyshev or elliptic filter, especially if it is desired to vary the filter order n. A value of n=2 was found to work well and was used throughout.

2.3.2 **Directional filtering**

The concept of direction will be discussed in detail in chapter 4 and appendix A, where it will be seen that there are two distinct types of direction. In this thesis, the words "direction" and

"orientation" will be used to distinguish between the two types. The essential difference for directionality defined on two-dimensional surfaces (such as those occuring within an image) is that "directions" are described by an angle in the range 0 to 2π radians, whereas "orientations" are described by an angle in the range 0 to π.

In the case of images in general, and fingerprint images in particular, it will be seen the concept of orientation rather than direction is the more appropriate, because one cannot distinguish in images between the orientations θ and $\theta+\pi$. That is to say, orientation is a quantity which is periodic, with period π.

Because of this periodicity, it is clear that there is no natural or preferred choice of ϕ corresponding to an origin or zero orientation. (This is in marked contrast to ρ, for which the DC frequency forms a natural clearly-defined origin). It follows that there is no such thing as a "high" or "low" orientation. It is therefore apparent that there can be no such concept as a low-pass or high-pass directional filter. A "directional band-pass filter" is a filter which passes a single range of orientations while removing components at orientations outside this range. Such a filter will produce as output an image in which only the information near a given orientation has been allowed to pass through.

Knowledge of 1-d filtering techniques cannot be directly applied to directional filtering because there is no 1-d analogue to direction. However, it is possible to lay down certain general requirements that a good directional filter would have to satisfy,

as well as defining directional analogues to the centre frequency and bandwidth of a 1-d bandpass filter.

A directional bandpass filter is a filter that passes components having orientations within a certain range. Clearly then, the analogue to the centre frequency is the orientation ϕ_o of the filter. This is the value of ϕ for which the filter response is maximum. We shall define the angular bandwidth as the difference in angle between the cut-off points ϕ_1 and ϕ_2 to either side of ϕ_o for which the filter response $F(\phi)$ has dropped to half its value at the centre, i.e.

$$F(\phi_1) = F(\phi_2) = 1/2\ F(\phi_o)$$

The fact that ϕ is periodic has two important consequences:

- Because all orientations are equivalent, it is reasonable to require that $F(\phi)$ take the same form whatever orientation ϕ_o is chosen for the filter. That is to say, given a filter $F_1(\phi)$ centred at ϕ_{o1}, and an equivalent filter $F_2(\phi)$ centred at ϕ_{o2}, the two will simply be shifted versions of each other, i.e.

$$F_2(\phi) = F_1(\phi - (\phi_{o2} - \phi_{o1})).$$

  (Note that this requirement could not be specified for 1-d filters because of the clearly defined zero frequency).

  It will be assumed in further discussion of filter design that the filter orientation is zero. Any desired orientation ϕ_o can then be obtained by shifting:

$$F_{desired}(\phi) = F(\phi - \phi_o)$$

- There is no preferred sense of "increasing" orientation. It is therefore reasonable to require that the filter be symmetrical about its orientation, i.e. that

$$F(\phi_o - \phi) = F(\phi_o + \phi)$$

(Again, this requirement could not be specified for 1-d frequency filters, for which a clearly defined sense of increasing frequency does exist).

The following additional requirements should help ensure a well-designed directional filter:

1. The filter function $F(\phi)$ should be as "smooth" as possible as a periodic function of ϕ. That is to say, $F(\phi)$ should be continuous in as many orders of derivative as possible. A minimum requirement is that $F(\phi)$ be continuous everywhere, because in general discontinuities in filters lead to undesirable ringing in their spatial domain responses (cf the 1-d "ideal" filters, which are discontinuous).

2. Using an appropriate set of directional filters, an image can be split up into a set of "directional components" which sum together to give the original image. For a filter $F(\phi)$ to be suitable for use in forming components, it is sufficient that it satisfy the following superposition property. If two filters, whose orientations ϕ_{o1} and ϕ_{o2} are exactly ϕ_{BW} apart, are summed, then the resulting function should be unity at all points between the two filter orientations, i.e.

$$f_1(\phi) + f_2(\phi) = 1 \text{ for } \phi_{o1} <= \phi <= \phi_{o2}.$$

Satisfaction of the superposition property is desirable even when directional components are not explicitly required, because a set of directional components formed using such a filter will contain sufficient information for the original to be exactly reconstructed, while having as narrow a range of orientations as possible within each component.

Possible choices of directional filter are given below. Assume for convenience that ϕ takes on values in the interval $[-\pi/2,\pi/2)$ and that the filter orientation is $\phi_o = 0$.

2.3.2.1 **The "ideal filter"**

This filter passes all orientations falling within the angular bandwidth and rejects all orientations outside it.

$$F_{ideal}(\phi) = \begin{cases} 1 & \text{if } |\phi| <= \phi_{BW}/2 \\ 0 & \text{otherwise} \end{cases}$$

This is the simplest filter but it is discontinuous, and this would give rise to ringing in the spatial domain. It does, however, satisfy the superposition property.

2.3.2.2 **The ramp filter**

$$F_{ramp}(\phi) = \begin{cases} 1 - |\phi|/\phi_{BW} & \text{for } |\phi| < \phi_{BW} \\ 0 & \text{otherwise} \end{cases}$$

This filter also satisfies the superposition property, and it is

continuous. However, its derivative is discontinuous at $\phi = -\phi_{BW}$, 0 and ϕ_{BW}.

2.3.2.3 **The Gaussian filter**

$$F_{gauss}(\phi) = \exp\left[- \frac{\ln 2}{(\phi_{BW}/2)^2} \phi^2 \right]$$

This filter is continuous, but its derivative is discontinuous at the point $\phi = -\pi/2$. This filter does not satisfy the superposition property.

2.3.2.4 **The modified Gaussian filter**

The modified Gaussian filter is an adaptation of the Gaussian filter, designed to produce a filter which is smoother than the Gaussian filter. It is C^{∞} everywhere. Its equation is

$$F_{ModGauss}(\phi) = \exp\left[- \frac{\ln 2}{\tan^2(\phi_{BW}/2)} \tan^2 \phi \right]$$

Again, however, the superposition property is not satisfied. Another difficulty is the mathematical complexity of the function.

2.3.2.5 **The sinusoidal filter**.

$$F_{sinus}(\phi) = \begin{cases} \cos^2 \frac{\pi}{2} \cdot \frac{\phi}{\phi_{BW}} & \text{for } |\phi| < \phi_{BW} \\ 0 & \text{otherwise} \end{cases}$$

This function is the one used by Knutsson et al [1983a] in their development of anisotropic "kernels" for the GOP image processor

system.

It is continuous and differentiable everywhere, and satisfies the superposition property. It also has the conceptual advantage that it seems most natural to use trigonometric functions when dealing with angular measures. This is supported by the fact that it was found necessary in 2.3.2.4 above to use the trigonometric tangent function to improve the performance of the Gaussian filter.

On the basis of the above, it was decided to use the sinusoidal filter, since it is the only filter which both is C^1 everywhere and satisfies the superposition property.

2.3.2.6 **Another approach to directional filtering**.

An alternative approach to directional filtering has been developed [Ikonomopoulos & Kunt 1985; Kunt et al 1985; Kunt 1986; Ikonomopoulos & Unser 1984]. The filter used is a 2-d function which is not separable in ρ and ϕ, nor is it analytically expressible in terms of elementary functions. The filter is obtained by Gaussian smoothing the "ideal" 2-d filter response shown in figure 2.6. This smoothing is performed in order to prevent the spatial domain ringing effects that would be caused by the use of this discontinuous function in its unsmoothed form.

This produces a filter whose radial response is high-pass and whose angular response is band-pass, but whose design is not based upon classical filter design techniques. It does not satisfy the superposition property. Although this approach has been successfully used in the production of directionally filtered

images similar to directional components for use in image coding applications, it was rejected as unsuited to our purposes. Its major disadvantage is that the filter is not separable in the polar coordinates, and therefore its radial and angular properties cannot be independently controlled in a simple way.

2.3.3 Combining radial and directional filtering

Once the radial and directional responses of the filter have been designed, these are multiplied together:

$$H(\rho,\phi) = H_{radial}(\rho) \cdot H_{angle}(\phi)$$

to give the polar-coordinate separable 2-d filter.

The radial part of the filter is the Butterworth bandpass filter:

$$H_{radial}(\rho) = \sqrt{\frac{(\rho\ \rho_{BW})^{2n}}{(\rho\ \rho_{BW})^{2n} + (\rho^2-\rho_o^2)^{2n}}}$$

and the directional part is the sinusoidal filter:

$$H_{angle}(\phi) = \begin{cases} \cos^2 \dfrac{\pi(\phi - \phi_o)}{2\ \phi_{BW}} & \text{for } |\phi - \phi_o| < \phi_{BW} \\ 0 & \text{otherwise} \end{cases}$$

2.4 Position dependent (nonstationary) linear filtering

2.4.1 Relevance of position dependence to fingerprint processing

In the process of forming an enhanced fingerprint image, the image must be filtered directionally in order to pass information only at orientations near to the local ridge orientation. Information at other orientations is not of interest. However, the LRO varies widely within the image, and therefore the behaviour of the filter will also have to vary as a function of position. Any filter whose behaviour depends upon position is known as a position dependent or nonstationary filter.

Position dependent filters are therefore required because the LRO and other local properties of fingerprints are position dependent. It is fortunate that fingerprints have such a highly-structured nature, for few classes of images are as structured. It is important to use whatever opportunities are available to take advantage of our knowledge of this structure, and position dependent Fourier filtering techniques allow this to be done.

2.4.2 Position dependent radial and directional filtering

Section 2.3 showed that a polar-separable stationary filter can be produced as the product of a directional and radial filter response:

$$H(\rho,\phi) = H_{radial}(\rho) \cdot H_{angle}(\phi)$$

This approach is not limited to the design of stationary filters: it can be generalised to the design of position dependent filters

simply by varying $H(\rho,\phi)$ within the image so that at each point (x,y) in the image we use some filter $H_{x,y}(\rho,\phi)$. Such a filter is usually denoted

$H(x,y;\ \rho,\phi)$, in polar coordinates,

or $H(x,y;\ u,v)$, in Cartesian coordinates.

2.4.3 Position-dependent filters in terms of position-dependent parameters.

The usual reason for wishing to use a position dependent filter is that there is some parameter or parameters of the image that vary from point to point, and it is decided that the filter should vary to adapt itself to the particular local values of these parameters. In other words, the filter function depends directly upon the parameters which in their turn depend upon position. The implementation of a position dependent filter can be simplified if the filter is expressed in terms of its parameter dependence rather than its (less direct) spatial dependence. The present section describes this in general, and the next section discusses the particular case of fingerprint images.

Suppose that there are m parameters describing local image properties, and which are going to be used to vary the filter function. These parameters may be given as

$p_i(x,y)$, $i=1..m$.

Rewriting the filter function H, dependent upon x and y, as a new function H', dependent upon the parameters, yields:

$$H(x,y;\ \rho,\phi) = H'(p_1(x,y),\ldots,p_m(x,y);\ \rho,\phi)$$

where H and H′ are the same function in the sense that they take on the same values at each x and y; the only difference is that the dependence on x and y is explicit for H and implicit for H′.

The design route for a position dependent filter expressed in terms of parameters is as follows:

1 (a) For the class of images to which the filtering operation is to be applied, identify the parameters which describe those local spatial properties which are relevant to the desired type of filtering.

(b) If, as is usually the case, the dependence of the parameters $p_1,\ldots,p_m$ upon position varies from image to image, determine methods for evaluating $p_1(x,y),\ldots,p_m(x,y)$ from any given image.

2. Design a filter function $H'(p_1,\ldots,p_m;\ \rho,\phi)$ which performs the required operation for any combination of the parameter values.

3. At each point (x,y) in the image, filter using the function

$$H(x,y;\ \rho,\phi) = H'(p_1(x,y),\ldots,p_m(x,y);\ \rho,\phi)$$

using the parameters determined for the point (x,y) from 1(b).

Details of the implementation of position dependent filters are given in the next chapter.

2.5 Application to fingerprints

The four local position-dependent parameters which describe the important local properties of fingerprints are:

1. The local ridge orientation $p_1(x,y) = LRO(x,y)$. This is the most important local property of fingerprints, and any filtering technique which does not take it into account will necessarily be at a severe disadvantage.

2. The local ridge curvature $p_2(x,y) = LRC(x,y)$. The local ridge curvature describes how rapidly the local ridge orientation is changing at a given point. It can be expressed in vector form as the gradient of the LRO:

   $$LRC(x,y) = \text{grad } LRO(x,y),$$

   or as a scalar as the magnitude of the gradient:

   $$LRC(x,y) = \| \text{grad } LRO(x,y) \|.$$

   LRC increases without bound as a core or delta is approached, and decreases asymptotically to zero with increasing distance from delta/core points. It is of greatest importance where its magnitude is largest (i.e. near the cores and deltas).

3. The local ridge spacing $p_3(x,y) = LRS(x,y)$. This, as we have seen, is defined and continuous everywhere, but is of considerably less importance than the first two parameters because it is slowly varying and constrained within a relatively narrow range of values.

4. The local ridge expansion $p_4(x,y) = LRE(x,y)$. This parameter

describes how rapidly the local ridge spacing is changing at a given point. As in the case of the LRC, it can be expressed either as a vector quantity

$$LRE(x,y) = \text{grad } LRS(x,y),$$

or as the scalar magnitude of this quantity.

In section 2.3.3 an expression was given for the stationary linear anisotropic bandpass filter

$$H(\rho,\phi) = H_{radial}(\rho) \cdot H_{angle}(\phi)$$

It is straightforward to extend this filter to a position dependent filter in terms of the four parameters identified above. First, rewrite the equation for $H(\rho,\phi)$, including the filter orientation, angular bandwidth, radial bandwidth and radial centre frequency as arguments (i.e. where these were formerly regarded as constants, they are now variables):

$$H(\rho_c,\rho_{BW},\phi_o,\phi_{BW};\ \rho,\phi) = H_{radial}(\rho_o,\rho_{BW};\ \rho) \cdot H_{angle}(\phi_o,\phi_{BW};\ \phi)$$

where H_{radial} and H_{angle} are as given in section 2.3.3.

The above equation could be interpreted as a position dependent filter if the the four new variables ρ_o, ρ_{BW}, ϕ_o and ϕ_{BW} are allowed to be functions of position. That these variables are indeed functions of position will now be shown, by expressing each of them as a function of one of the four position dependent parameters LRO, LRC, LRS and LRE. The relationships are:

1. <u>The filter orientation</u> ϕ_o. It is clear that the filter should

be oriented so that it passes information at the local ridge orientation, and blocks components at other orientations. Therefore, identify

$$\phi_o = \text{LRO} + \pi/2 \ .$$

The $\pi/2$ is needed because the line joining a conjugate pair of Fourier components is always perpendicular to the corresponding spatial wave orientation.

2. The filter angular bandwidth ϕ_{BW}. Where the ridge curvature is high, it will be necessary to increase the angular bandwidth of the filter to allow a greater range of orientations to be passed, for the following reasons:

a. In regions of high curvature, a wider range of orientations is present within a given size of window than in low-curvature regions. Where the curvature tends to infinity (the core and delta points) all orientations are present. Where the curvature is zero, only one orientation is present.

b. The practical algorithms used to measure and interpolate LRO can be expected to give less reliable results in higher curvature areas, and therefore the angular bandwidth must be large enough for it to be tolerant to a certain amount of error in the LRO estimate near the cores and deltas.

Thus the angular bandwidth used will be a function of the LRC. The exact functional dependence $\phi_{BW}(\text{LRC})$ has to be determined empirically, and this will be discussed in the next chapter. However, it is clear that ϕ_{BW} must be a monotonically increasing

continuous function of LRC taking on its minimum value for LRC=0 and increasing to π for LRC=∞.

3. The centre frequency ρ_o. Clearly, one would wish for the filter centre frequency to be the same as the local frequency of the fingerprint ridges, so that this frequency is passed more strongly than others. Therefore, identify

$$\rho_o = \text{local ridge frequency} = 2\pi/\text{LRS}.$$

(This becomes $\rho_o = 2\pi N/\text{LRS}$ in the case of the discrete transform of an image sampled on an N by N grid with LRS measured in pixel units).

4. The radial bandwidth ρ_{BW}. In regions where the LRS is changing rapidly, there will be a wider range of radial frequency components present than in regions where the LRS does not change much. Therefore the bandwidth must be increased with increasing LRE. The exact dependence of ρ_{BW} upon LRE must again be determined experimentally.

Including these functional dependences, the filter function of section 2.3.3 becomes

$$H(\text{LRO},\text{LRC},\text{LRS},\text{LRE};\ \rho,\phi) = H_{\text{radial}}(\text{LRS},\text{LRE};\ \rho)\ H_{\text{angle}}(\text{LRO},\text{LRC};\ \phi)$$

where

$$H_{\text{radial}}(\text{LRS},\text{LRE};\ \rho) = \sqrt{\frac{(\rho\ \rho_{BW}(\text{LRE}))^{2n}}{(\rho\ \rho_{BW}(\text{LRE}))^{2n} + (\rho^2 - (2\pi/\text{LRS})^2)^{2n}}}$$

and

$$\text{Hangle (LRO,LRC; } \phi) = \begin{cases} \cos^2 \dfrac{\pi(\phi - \text{LRO} - \pi/2)}{2\ \phi_{BW}(\text{LRC})} & \text{if } |\phi - \text{LRO} - \pi/2)| < \phi_{BW} \\ 0 & \text{otherwise} \end{cases}$$

Note that H is still separable into two groups of variables, the radial group {LRS,LRE,ρ} and the angular group {LRO,LRC,ϕ}.

CHAPTER 3

IMPLEMENTATION OF POSITION DEPENDENT FILTERS

Chapter 2 discussed the design of position dependent filters in terms of a number of position dependent parameters, and in particular produced a position dependent anisotropic filter function intended for use in enhancing raw fingerprint images.

This chapter investigates how such filters could be efficiently implemented on a general-purpose computer. It will be shown that a direct implementation of the filter would be very computationally intensive and that a considerable amount of work can be saved by introducing a few physically reasonable simplifications. In the case of filtering of fingerprints, it will be seen that without the use of such simplifications, solution of the problem would not be practicable.

It is important to note that this chapter deals only with implementation on general-purpose computing systems. In systems which have specialised or non-conventional hardware, the implementation considerations will usually be different, in which case different approximations and assumptions may be necessary. An example of implementation on an alternative hardware system is dealt with in chapter 7, which discusses the implementation of the enhancement algorithm in optical hardware.

For the purposes of this chapter it is assumed that the values of the m position-dependent parameters introduced in the previous chapter are known at all points within the image. In the case of fingerprints this will mean that the values of LRO, LRS, LRC and LRE can be found at all points. The separate problem of determining the actual values of these parameters will be dealt with in chapter 5. (In fact, it will be argued later in this chapter that values at each position are required for LRO only, and the other parameters can be approximated to varying degrees.)

3.1 Discretisation of stationary analogue filters

3.1.1 The discrete Fourier transform and windowing

The filters developed in chapter 2 were analogue in nature and infinite in extent; that is to say, the signals under consideration were assumed to be functions of continuous spatial coordinates x and y of infinite extent. If the signal is an image which is stored in a digital computer, it will have been sampled and stored as a function of discrete coordinates x and y of finite extent. (For example, the images used in this research were sampled on a 512 by 512 spatial grid.)

For discrete data, the continuous Fourier transform cannot be found, and it is necessary to use a discrete transform instead. A transform known as the Discrete Fourier Transform will now be developed from the continuous transform.

The continuous Fourier transform of a 1-d analogue signal $f_a(t)$ is defined as

$$F(\omega) = \int_{-\infty}^{\infty} f_a(t) \exp(-j\omega t)\, dt$$

with inverse

$$f_a(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \exp(j\omega t)\, d\omega \ .$$

This transform exists provided that $f_a(t)$ satisfies fairly mild integrability constraints. However, if $f_a(t)$ is a periodic signal, i.e.

$$f_a(t) = f_a(t+iT) \quad \text{for all integer } i,$$

where T is the period, then the above integrals will not converge except in trivial cases. Instead, the integral is taken over a single period of the signal, which leads to the Fourier Series, defined as:

$$F(u) = \frac{1}{T} \int_0^T f_a(t) \exp(-j2\pi ut/T)\, dt \quad \text{for all integer } u,$$

with inverse

$$f_a(t) = \sum_{u=-\infty}^{\infty} F(u) \exp(j2\pi tu/T) \quad \text{for all real } t.$$

If in addition the signal is band-limited to a frequency u_o, so that F(u) is zero for $|u| > u_o$, then by the Nyquist-Shannon sampling theorem [Jerri 1977] the signal can be exactly reconstructed from a set of N samples equally spaced over one period, i.e. from the samples

$$f(x) = f_a(xT/N) \qquad x=0..N-1,$$

provided that $N >= 2u_o$.

The Discrete Fourier Transform (DFT), which will be defined shortly, can be shown to have the following relationship to the Fourier Series. The DFT of the set $\{f(x) | x=0..N-1\}$ of N sampled points is identical (up to a constant scaling factor) to the Fourier Series of the periodic bandlimited signal $f_a(t)$ over the frequency range $|u| < u_o$.

Given N values f(x), x=0..N-1, representing one period of the above data, the most usual definition for the DFT is:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp(-2\pi jux/N) \quad \text{for } u=0..N-1$$

The inverse DFT is then given by

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp(2\pi jxu/N) \quad \text{for } x=0..N-1$$

If we assume that the above two equations hold for all integers u and x, then both f(x) and F(u) must be periodic:

$$f(x) = f(x+iN) \quad \text{and} \quad F(u) = F(u+iN) \quad \text{for any integer } i.$$

f(x) then represents a sampled version of the entire periodic signal $f_a(t)$ rather than of just one period of it.

For a 2-dimensional signal f(x,y) having period N in both the x and y directions, i.e.

$$f(x,y) = f(x+iN, y+jN) \quad \text{for all integer } i \text{ and } j,$$

the DFT relationship is:

$$F(u,v) = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \exp[-2\pi j(ux + vy)/N] \quad \text{for } u,v=0..N-1$$

with inverse

$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) \exp[2\pi j(ux + vy)/N] \quad \text{for } x,y=0..N-1$$

Again, both the signal and its transform exhibit the same periodicities.

For signals which are bandlimited as above and exhibit the above perodicities, the DFT will behave in a manner which is similar to the behaviour of the continuous Fourier transform. In particular, the considerations in chapter 2 about directionality, use of polar coordinates, meaning of a pair of components, etc., will then be equally valid if the discrete instead of the continuous transform is used.

In practice, however, these spatial periodicity properties are not usually satisfied, since there is seldom any continuity relationship between the first and last points in a 1-d signal, nor is there between the top and bottom or left and right of a 2-d image. Nevertheless, the DFT will treat such data as if it were periodically repeated according to the relationships given above. Therefore, it is very likely that there will be spurious discontinuities at either end of 1-d data, or along the edges of 2-d data. (See figure 3.1). These discontinuites will result in a broadening in the width of spectral components within the transform. Spatial ringing effects near the edges of filtered signals may also occur [Gonzalez & Wintz 1977].

The unwanted edge effects due to the periodic nature of the DFT can be reduced by a technique known as **windowing**. Windowing will first be described for 1-d signals, and then generalized to two dimensions.

The windowing process modifies the signal, producing a new version

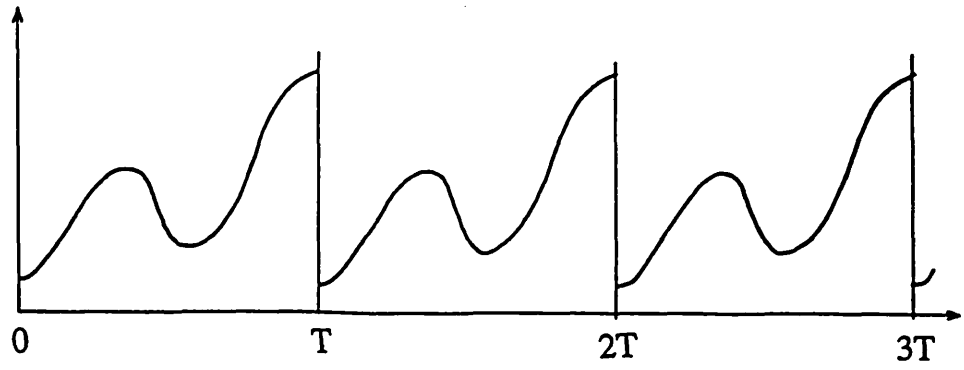


Figure 3.1. Spurious discontinuities exhibited by 1-dimensional data when it is treated as periodic data. These discontinuities occur at the points t=nT for integer n, where T is the length of the data.

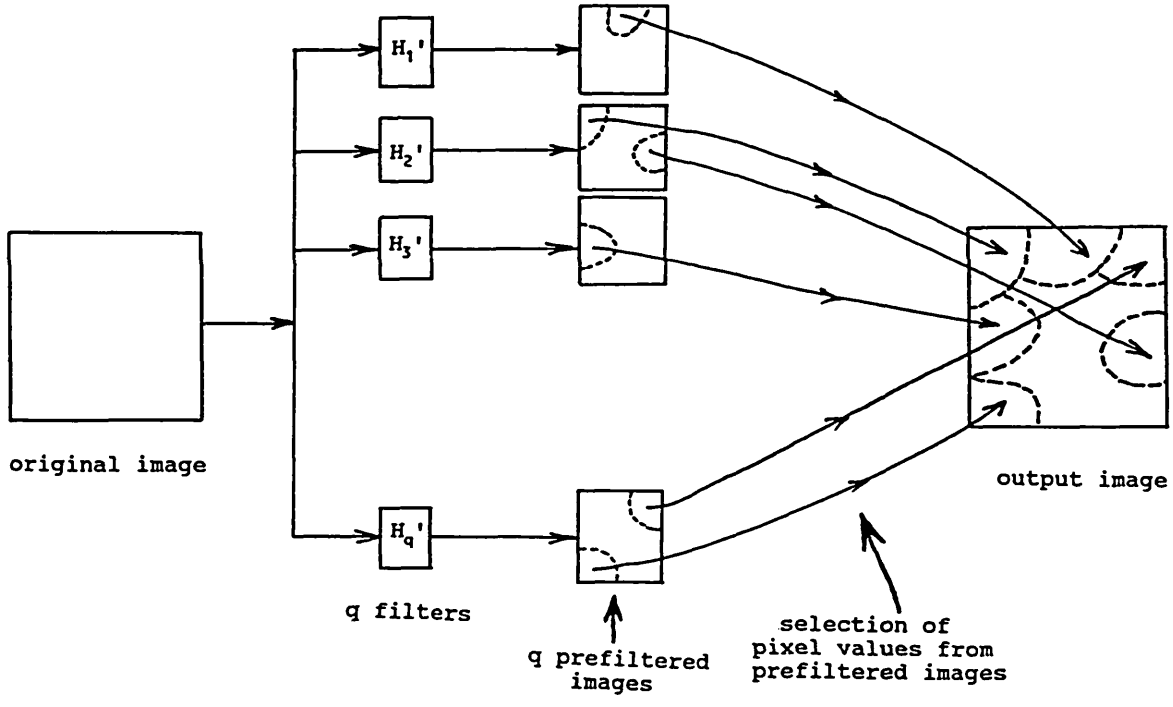


Figure 3.2. Nonstationary filtering implemented by selection of each output pixel from one of a set of prefiltered images.

of it known as the **windowed signal**, in order to make or tend to make it continuous at the end-points when viewed as a periodically repeated signal. This is done by multiplying the signal f(x) by an appropriate windowing function w(x):

$$f_w(x) = f(x)w(x) \quad \text{for } x \text{ in the real interval } [0,N),$$

where the window w(x) can be normalized to take on values between 0 and 1. The filtering operation is then performed using the DFT of samples of the windowed rather than the original signal, i.e. using the values

$$f_w(n) = f(n)w(n) \quad \text{for } n=0,\ldots,N-1.$$

If no windowing is explicitly performed, this is equivalent to using the **boxcar** window, w(x)=1 for all x. Of course, the boxcar window has no effect on the discontinuities at the endpoints.

Several alternative windowing functions are described and compared (in the context of spectral estimation applications) in the papers by Harris [1978] and Geckinli & Yavuz [1978]. All of these windows are symmetric about the midpoint at x=N/2, where they take on the value 1. With the exception of the boxcar window which is constant, they decrease monotonically and continuously towards the endpoints at x=0 and x=N, where they usually take on the value 0. In this way, the windowed signal is forced to be continuous. It is generally desirable that a window function exhibit a high order of continuity.

For the present application, the window has to be chosen according to a trade-off between two conflicting requirements:

i. In order to ensure that the effect of the discontinuity is minimized, the window should not rise too rapidly as one moves away from the endpoints.

ii. In order to ensure that as little data as possible is lost by being multipled by window values near zero, the window must rise rapidly towards unity away from the endpoints.

The **Tukey** (or **split Hanning** or **cosine taper**) window, which was originally presented in [Tukey 1967], allows the trade-off between the above two requirements to be controlled in a particularly simple way. The middle section of this window is the constant value 1, and it tapers down to 0 at the endpoints according to a cosine function, as follows:

$$w(x) \begin{cases} = \frac{1}{2}\left(1 - \cos\frac{2\pi x}{\alpha N}\right) & \text{for } 0 \le x \le \alpha N/2 \\ = 1 & \text{for } \alpha N/2 < x < N(1-\alpha/2) \\ = \frac{1}{2}\left(1 - \cos\frac{2\pi(N-x)}{\alpha N}\right) & \text{for } N(1-\alpha/2) \le x < N \end{cases}$$

where α is a fraction which when multiplied by 100 gives the percentage of the window length which the tapered portion occupies. $w(x)$ is C^1 continuous.

Huang [1972] proposed a method of design of windows for 2-d signals. He showed that, given any 1-d window $w_{1-d}(x)$ of length L and symmetric about its midpoint, a 2-d window of dimension L by L

and having similar properties can be produced by rotating the 1-d window about its midpoint. Let the window length be L, and for notational convenience assume that the x axis is shifted such that the window extends from -L/2 to L/2 with x=0 as its midpoint and centre of symmetry. Then the equivalent 2-d window is expressed by:

$$w(x,y) = \begin{cases} w_{1-d}\left(\sqrt{(x^2+y^2)}\right) & \text{if } \sqrt{(x^2+y^2)} < L/2 \\ 0 & \text{otherwise} \end{cases}$$

The disadvantage of this approach is that all the data outside the circle $\sqrt{(x^2+y^2)} = L/2$ is set to zero. This is unavoidable because of the circularly symmetric nature of the window.

It is not necessary, however, to lose so much data in the process of converting a 1-d window to a 2-d equivalent window. An alternative approach is to form the 2-d window as a separable product of two 1-d windows [Merserau 1981], as follows:

$$w(x,y) = w_{1-d}(x)w_{1-d}(y).$$

w(x,y) will exhibit similar continuity properties at its edges as $w_{1-d}(\bullet)$ has at its endpoints. Also, the width of the tapered region around the edge of the signal will be more uniform than is possible using a circularly symmetric window, and therefore less data will be lost during windowing.

Because w(x,y) is separable in the spatial coordinates, its transform W(u,v) is also separable:

$$W(u,v) = W_{1-d}(u)W_{1-d}(v),$$

where $W_{1-d}(\bullet)$ is the 1-d Fourier transform of $w_{1-d}(x)$. It is clear that the Fourier properties of the 2-d window are both similar and simply related to those of the 1-d window.

In forming the windowed version of a signal, it is important to deal with the DC component correctly. If this is not done, spurious low-frequency components will be introduced into the transform of the windowed signal, because the windowed signal, when viewed as a periodic signal, will have been forced to a value near zero at the edges of each period.

To avoid this, the DC component must first be removed before windowing, and then replaced either immediately or after the transform domain processing is complete and the inverse transform has been taken. Alternatively and equivalently, the removal and replacement of the DC component could be regarded as an integral part of the windowing process. This yields the following expression for the windowed signal in terms of the original:

$$f_w(x,y) = (f(x,y) - f_{ave})w(x,y) + f_{ave}$$

$$= f(x,y)w(x,y) + (1-w(x,y))f_{ave}.$$

where f_{ave} is the DC component of the signal.

3.1.2 Digital filters from analogue filters

A vast literature exists on the design of filters for discrete signals. Many of the available techniques are purely "digital" in nature, but there are also several approaches to developing a digital filter having similar properties to a given analogue

filter. One of these is described below.

The **bilinear transform** approach [Oppenheim & Schafer 1975, page 206] maps the infinite analogue frequency range $0 <= \Omega < \infty$ to the finite frequency range $0 <= \omega < \omega_{max}$ encountered in discrete signals, using the function

$$\omega = 2 \arctan \frac{\Omega T}{2},$$

where T is the sampling interval, and we have taken $\omega_{max}=\pi$ which is the usual convention.

Thus an analogue filter of frequency response $H_a(\Omega)$ would map to the equivalent digital filter of frequency response

$$H(\omega) = H_a(\Omega)$$

$$= H_a\left(\frac{2}{T} \tan \frac{\omega}{2}\right)$$

Note that since ω is a non-linear function of Ω, there is a distortion along the frequency scale, which will cause centre frequencies and cut-off frequencies of the filter to be changed by the mapping. This effect can be prevented by using "prewarping" to shift these significant frequencies <u>before</u> applying the transform so that they will end up in the desired place. For a frequency which is desired to be at ω_d, it must be "prewarped" to be at

$$\Omega_{pw} = \frac{2}{T} \tan \frac{\omega_d}{2},$$

so that $H(\omega_d) = H_a(\Omega_{pw})$ as required.

A useful simplification may however be introduced if the signals to be filtered are sufficiently finely sampled that the frequency components of interest are all much lower than the sampling frequency. In this case, the filter will have its significant (cutoff and centre) frequencies Ω_s such that

$$\Omega_s << 2\pi/T$$

corresponding to the digital frequencies

$$\omega_s << \pi.$$

This leads to the conclusion that prewarping is not necessary, since

$$\Omega_{pw} \approx \omega_d/T, \quad \text{since } \tan x \approx x \text{ for } x<<1.$$

Another consequence of keeping the significant filter frequencies well below the sampling rate is that the filter function $H(\Omega)$ will then only vary significantly for $\Omega << 2\pi/T$. For other frequencies, $H_a(\Omega)$ will closely approximate zero for a bandpass or lowpass filter, or one for a highpass filter. That is, H_a is approximately constant for such frequencies.

For frequencies $\Omega<<2\pi/T$, we have $\tan(\omega/2)<<\pi$ and hence $\omega << \pi$. Therefore $H(\omega) \approx H_a(\omega/T)$. This is the linear region of the bilinear frequency transformation.

For other Ω, $H(\omega) \approx H_a(\omega/T)$ is still a good approximation since the significant frequencies are such that H_a is roughly constant

for such Ω.

Therefore, if the significant frequencies are low enough, a digital filter which closely approximates a bilinear transform/prewarping design can be obtained simply by sampling the corresponding analogue filter response, i.e.

$$H(\omega) = H_a(\omega/T).$$

This approach was used in converting the filters of chapter 2 to digital filters. A centre frequency of 50 pixel units with a bandwidth of 80 units were used.

3.2 Direct implementation

It is required to implement the position-dependent filter

$$F(x,y;\ u,v) = F'(p_1(x,y),\ldots,p_m(x,y);\ u,v)$$

meaning that at each point (x_o,y_o), the spatial domain image must be filtered using the function

$$H(u,v) = F(x_o,y_o;\ u,v).$$

Before the filtering is done, it is necessary to window the image in order to ensure its continuity as a periodic function. A Tukey window with a 10% taper (adapted for 2-d using the spatially separable product method) was chosen as a compromise between the two requirements of section 3.1.1.

The direct implementation of this for filtering an image **in**(x,y) to give the image **out**(x,y) is:

(1) Window the image in preparation for taking the transform:

$$\mathbf{in}(x,y) = \text{window}(\mathbf{in}(x,y))$$

(2) Take the discrete Fourier transform of the windowed image, to get

$$\mathbf{IN}(u,v) = \text{DFT}(\mathbf{in}(x,y))$$

(3) For each pixel (x,y) in the image do steps (4),(5) and (6):

(4) Evaluate the filter function for the point x,y as

$$H(u,v) = F(x,y;\ u,v)$$

(5) Multiply the filter function by the image function to get

$$TMP(u,v) = H(u,v) * \mathbf{IN}(u,v)$$

(6) Compute the (x,y) pixel of the output image as the (x,y) pixel of the inverse DFT of this result:

$$\mathbf{OUT}(x,y) = IDFT_{x,y}\ (TMP(u,v))$$

The following pseudo-code illustrates how this algorithm could be encoded on a computer (assuming the use of N by N pixel images). The step numbers used above are included in comment braces.

```
{1}     window(in)
{2}     DFT(in)
{3}     for x=0 to N-1 do
          for y=0 to N-1 do

{4,5}       for u=0 to N-1 do
              for v=0 to N-1 do
                TMP(u,v) = IN(u,v) * F(x,y; u,v)
              endfor
            endfor

{6}                        N-1 N-1
            out(x,y) =      Σ   Σ  TMP(u,v)exp[2πj(ux+vy)/N]
                           u=0 v=0
          endfor
        endfor
```

The problem with this approach is that the entire image must be multiplied by a filter N^2 times (once for each pixel) and one point of the inverse transform of the product must be evaluated N^2 times. This amounts to $O(N^4)$ operations, which is a prohibitively large amount of computation.

3.3 The use of moving windows

This approach involves choosing a square window of appropriate size, and moving it so that it becomes centred at each point within the image (except at those points in a border region round the edge, where the window centre cannot reach). For each position visited by the window, data contained within it is modified by an appropriate window function. The 2-d Fourier transform of the region is then taken and multiplied by the filter function appropriate to the position of the centre of the window. The centre pixel of the inverse transform of this product is the contribution to the filtered image at the point.

Alternatively, the filtering could be done in the spatial domain as a convolution between the contents of the window and the inverse transform of the filter function. This approach has the advantage that the convolution used can be linear rather than the circular convolution implicit in the use of the DFT, and therefore the complications involved in windowing the data can be avoided. This advantage is gained at a computational expense, as will be seen shortly.

Clearly, the value of the filtered image at any point (x,y) can only be influenced by the values of the input at points within the window which was used to evaluate the output at that point. Thus, by saying that we may filter within a window of size k by k, we are saying that the filter impulse response or "point spread function" is of finite extent. The maximum impulse response size is equal to the window size used. Such filters are often termed FIR (finite-extent impulse response) filters [Rabiner et al 1972].

The window must be chosen large enough so that the characteristics of the image which are described by the position dependent parameters used in the filter are adequately represented in the window. For example, in the case of a fingerprint, the window must be large enough to include at least two ridge wavelengths so that the parameters LRO and LRS have meaning within the window. For small window sizes, the effect of noise and imperfections within the window area will be more pronounced, since the filter will have fewer pixels and hence less information to work with. On the other hand, for larger window sizes, more computational effort will be required to perform the filtering. Also, the greater variation of LRS and LRO within larger windows will adversely affect filter selectivity.

One of the disadvantages of any moving window image processing technique is that special arrangements have to be made to deal with the points near the edge of the image, because the window cannot be centred about such border pixels. For larger windows this problem is more serious because a larger area of the image will be affected. Several approaches to dealing with the border pixels in moving window techniques have been proposed, and some of these are discussed in [Fernando 1985].

To evaluate the amount of computation involved in filtering the N by N pixel image using a k by k pixel filter, note that for each of the N^2 image pixels it is necessary to window the data, forward transform it, multiply this by the filter function and then inverse transform the result. The number of operations required

for the forward transform is of order $k^2 \log_2 k$ if the FFT algorithm is used, for the windowing and multiplication k^2, and for the evaluation of the inverse transform at a single point k^2. Therefore the total number of operations required will be of order

$$N^2 . k^2 \log_2 k$$

which is considerably less than the N^4 required by the direct method, provided $k<<N$.

(If the filtering is performed by spatial convolutions, then the overall computational order is

$$N^2 k^4,$$

since direct evaluation of the convolution has a computational order of k^4).

If no windowing is done (i.e. the boxcar window is chosen), then the computation order can be reduced by a factor of $\log_2 k$, to $N^2.k^2$, by using the "Walking FFT" algorithm of Appendix C to evaluate the forward transforms. This reduces the computation order from $k^2 \log_2 k$ to k^2 per transform. Unfortunately, the walking FFT will only produce good results for large window sizes. As the window size is reduced, the ratio of window edge length to window area increases, and consequently the importance of edge effects will also increase. Therefore, for small windows, it becomes essential to use a good windowing function in order to reduce these edge effects as far as possible, and the boxcar window implicit in the walking FFT approach is not good enough.

The moving window approach was rejected in favour of the method of

section 3.4 because of the difficulties and complications that are encountered in dealing with windowing and border areas.

3.4 Obtaining a set of parameterised pre-filtered images

It was seen in chapter 2 that the usual reason for filtering an image in a position-dependent way is to allow the filter to adapt to the local properties of the image as it is filtered at each point. The positional dependence of the filter is indirect and consequently the exact dependence will change from image to image. The filter's direct dependence is upon the data itself, via a set $p_1,\ldots,p_m$ of m parameters describing the local properties of the image as functions of position.

The direct approach of section 3.2 does not take regard of any parameter values, but instead considers the filter to be an explicit function of x and y. The result of this approach is that there will be as many different filters as there are (x,y) positions, i.e. N^2 for N by N pixel images. If N is typically 512, then the image has to be filtered using 262,144 filters, which as we have seen is computationally prohibitive.

A method is developed here which seeks to take advantage of the parameter-dependence of the filter function in order to reduce the number of distinct filters used from N^2 down to some more manageable number q (say, q = 30 or less: in the case of filtering for fingerprint enhancement, it will be seen that good results can be achieved with q as small as 8).

The approach is illustrated conceptually in figure 3.2, in which the q filters are applied to the original image to form q filtered images. The output image is built up by appropriate selection of pixel values from these filtered images, as will be described

shortly.

Denote the q filters by

$$H'_i(u,v), \quad i=0,\ldots,q-1.$$

The q filters should "span" the space occupied by the N^2 filters in the sense that, given any one of the N^2 filters $H(x,y;\ u,v)$, one of the q filters $H'_i(u,v)$ can be found that is "sufficiently close" to the given filter to use instead of it. By sufficiently close is meant that the use of the $H'_i(u,v)$ instead of the $H(x,y;\ u,v)$ will not degrade the resulting filtered image unacceptably according to whatever criteria are being used to evaluate the quality of the filtered picture.

Given a point (x,y), the filter $H_i(u,v)$ which must be used at that point is the one that is closest to the correct filter $H(x,y;u,v)$. The selection of the filter will be based upon the consideration of the values of the parameters at (x,y). That is, the ith filter will be selected where i can be determined from the parameter values:

$$i(x,y) = i(p_1,\ldots,p_m).$$

The filter $H'_{i(x,y)}(u,v)$ selected in this manner will be "sufficiently close" to the correct filter $H(x,y;\ u,v)$.

We shall now discuss how to implement an efficient position-dependent filtering algorithm assuming that q filters $H'_i(u,v)$ have already been obtained and i(x,y) can be determined for any x and y.

First, each of the q filters $H'_i(u,v)$ must be applied to the input image as a stationary filter, thereby yielding k pre-computed filtered images, as illustrated in figure 3.2. These will be called the prefiltered images. This is done as follows:

1. Take the DFT of the input image f(x,y), after it has been suitably windowed:

```
window (f(x,y))
F(u,v) = FFT{f(x,y)}
```

2. Obtain q Fourier filtered images $filt_i(x,y)$ by filtering the input image using each of the q filters:

```
for i = 1 to q do
  Multiply H'i(u,v) by F(u,v) to give FILTi(u,v).
  Inverse transform FILTi(u,v) to give the ith filtered
  image filti(x,y).
endfor
```

This is all the filtering that is necessary and amounts to the performance of one forward FFT and q inverse FFTs, giving a computational order of

$$q.\ N^2 \log_2 N.$$

Thus it is clear that this technique can be vastly more efficient than the direct approach, which required $O(N^4)$ operations.

Once the q prefiltered images have been obtained, it simply remains to build up the final output image g(x,y) by scanning the input image and selecting at each point which one of the prefiltered images $filt_i(x,y)$ to use:

```
for x=0 to N-1 do
  for y=0 to N-1 do
    Determine the values of the parameters p1...pm at
    the point (x,y).
    Use these parameter values to get i = i(p1...pm).
    Set g(x,y) to filti(x,y).
  endfor
endfor
```

The next two sections discuss two techniques which can be used to obtain the q prefiltered images.

3.4.1 Discretisation of parameter values

Consider any one of the q parameters, say the ith parameter p_i. Let p_i be sampled so that it is restricted to a finite set of n_i discrete values p_{ij}. The sampled values p_{ij} must be chosen such that, given any value of p_i, the use of the nearest sampled value p_{ij} instead of the correct value p_i in the filter will not degrade the output image unacceptably.

Suppose that all of the m parameters $p_1, p_2, \ldots, p_m$ have been sampled to $n_1, n_2, \ldots, n_m$ values each, then the total number of different filters that can be generated is equal to the number of possible combinations of sampled parameter values, i.e.

$$q = n_1 n_2 \ldots n_m.$$

The number of prefiltered images therefore depends strongly upon how finely the parameters are sampled.

A more coarse sampling can be done, thereby reducing q, if it is acceptable to interpolate between pixel values from prefiltered images rather than simply selecting the one generated by the filter having parameters closest to the correct values.

Interpolation in as many dimensions as there are coarsely-sampled parameters must be performed. Therefore this is most feasible when only one or two of the parameters is so coarsely sampled. In the case of filtering for fingerprint enhancement, for instance, the LRO parameter is sampled in this way and one-dimensional linear interpolation is used to interpolate between pixel values.

3.4.2 Selection of the most critical parameters

We saw in the previous section that the number of prefiltered images needed is

$$q = n_1.n_2...n_m$$

and investigated how to reduce q by reducing the value of the n_i's.

This section shows how to reduce q by reducing the number m of parameters involved in the computation of prefiltered images.

If a parameter p_i depends only weakly upon position, it may in certain circumstances be assumed to be constant over the entire image. If this assumption is made, then that parameter can be omitted from the parameter list. Often, a small amount of variation in a parameter can be allowed for in the design of the filter by assuming the parameter to be constant, but increasing the filter's tolerance to variations in its value by, for example, increasing the bandwidth of a filter. This results in a filtered image which is slightly noisier, but may be preferable to calculating many more prefiltered images. We shall see that the

LRS parameter can be omitted in fingerprint filtering if the filter is made less sensitive to ridge spacing by increasing its radial bandwidth.

The effect of a parameter can sometimes be omitted from the calculation of the prefiltered images even if it does vary over a wide range of values. Suppose, for instance, that the value of p_1 affects the centre frequency of a filter, and p_2 affects the bandwidth of the filter. It may then be possible to ignore p_2 in calculating the prefiltered images, because a filter of wide bandwidth can be simulated by combining the results of filters of narrow bandwidth but differing centre frequencies. Such filters would be generated by the variation of p_1. It is of course still necessary to evaluate p_2 at each point, in order to know how many prefiltered images to combine to get the required effective bandwidth. This technique is used to remove the LRC parameter from the computation of prefiltered images.

3.5 Application to fingerprint processing

In chapter 2, the following filter function intended for fingerprint preprocessing was derived:

$$H(\rho,\phi;\ LRO,LRC,LRS,LRE) =$$

$$H_{radial}(\rho;\ \rho_o(LRS),\rho_{BW}(LRE)).H_{angle}(\phi;\ \phi_o(LRO),\phi_{BW}(LRC))$$

This filter involves the four position-dependent parameters LRO, LRS, LRC and LRE. To form a set of as few prefiltered images as possible, the values of these parameters must be sampled, and where possible the number of parameters involved must be reduced. We shall now consider each parameter in turn.

• Local ridge orientation

The LRO determines the orientation (or "centre angular frequency") of the filter and takes on values in the range $[0..\pi)$. It is the most important of the four parameters and must be carefully dealt with if high quality filtering is to be achieved. Since all orientations are equivalent, it is sensible and convenient to sample this parameter at an integral number n of evenly-spaced intervals between 0 and π, i.e. to sample at the values

$$\phi_i = i\pi/n \quad \text{for } i = 0..n-1.$$

(It will turn out that the number q of prefiltered images will equal n, since the other three parameters can be eliminated and will therefore not require sampling).

In chapter 5 it will be argued that the LRO need not be known to any greater precision than to the nearest $\pi/8$, provided that the

LRO estimates vary smoothly as a function of position. Indeed, the technique which is used to evaluate the LRO at a point, as described in chapter 5, gives the LRO as one of eight discrete values:

$$\theta_i = i\pi/8, \quad i=0..7.$$

This implies that the filters should have a directional bandwidth not narrower than $\pi/8$ radians, so that a filter centred at ϕ_c will pass components with orientations in the range $\phi_c - \pi/8$ to $\phi_c + \pi/8$. A narrower bandwidth would result in excessive sensitivity to orientation and therefore the possibility of filtering out the desired components. On the other hand, a wider bandwidth is also undesirable as the filter would then be too insensitive to orientation.

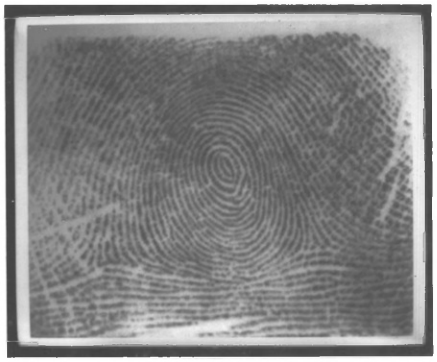
Satisfaction of the superposition property using filters of bandwidth $\pi/8$ implies that the number of images per complete set of directional components is eight. Therefore, the total number n of orientations used will be a multiple of eight.

If a single set of directional components is used, then n=8 and the orientations of the filters are

$$\theta_i = i\pi/8 \quad \text{for } i=0..7.$$

If two sets of components are used, they should be oriented at $\pi i/8$ and $\pi(i+0.5)/8$ (for $i = 0..7$), giving a total of 16 equally spaced orientations.

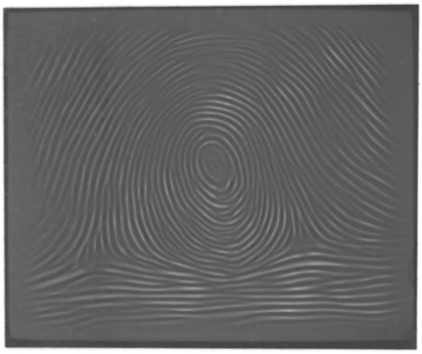
The example of figure 3.3 compares the enhancement effect of using



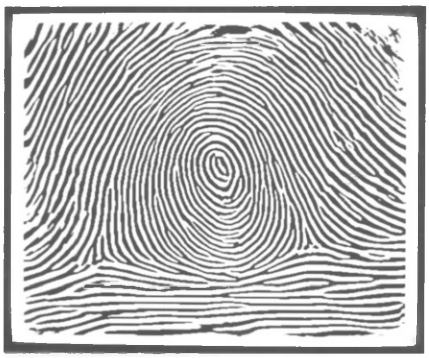
(a)



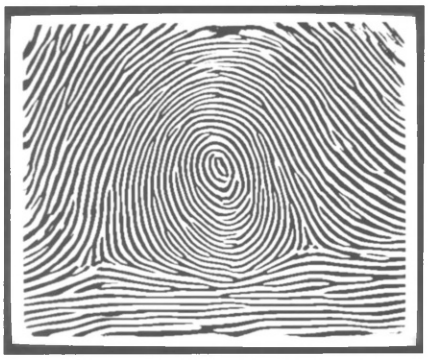
(b)



(c)



(d)



(e)

Figure 3.3. Enhancements using 8 and 16 filter orientations. The results of filtering the fingerprint (a) using 8 and 16 orientations are shown in (b) and (c) respectively. (d) and (e) are thresholded versions of (b) and (c) respectively.

eight and sixteen orientations in forming prefiltered images. Although the (greyscale) filtered image using 16 orientations appears slightly clearer visually than that using 8 orientations, very little difference is discernible between the thresholded versions which constitute the final enhancements. Therefore, the use of eight orientations rather than sixteen is adequate.

The LSI-11 computer which was used to perform the image processing was severely limited at many times in its available disk storage capacity, and it was not always possible to store prefiltered images for sixteen orientations. For this reason, some of the enhancement results presented in this thesis used 8 orientations rather than 16.

In all cases, interpolation between prefiltered image pixel values was performed, i.e. the coarse parameter sampling technique of section 3.4.1 was applied. Linear interpolation was used as this is sufficient to ensure a continuous change in output pixel value as filter orientation is changed.

• **Local ridge curvature**

The LRC affects the angular bandwidth of the filter as discussed in section 2.5, where it was argued that the angular bandwidth must be a monotonically increasing function of LRC taking its minimum value $\pi/8$ for LRC=0 and its maximum π for LRC=∞.

An estimate of the LRC can be obtained by using the gradient of the LRO estimate determined using the mathematical model of LRO as presented in chapter 4. Alternatively, noting that LRC

increases towards infinity as the singular points are approached and decreases asymptotically to zero with distance from the nearest singular point, an empirical relationship sufficiently accurate for our purposes can be obtained, giving the required angular bandwidth as a function of the distance to the nearest singular point.

This latter approach was used, and the angular bandwidth used as a function of the distance r to the nearest singular point is shown in figure 3.4. ϕ_{BW} takes on its minimum value if r is greater than some value r_2. If r is less than r_1, then the full bandwidth of π is used, and if $r_1<r<r_2$ then the bandwidth used decreases linearly with r. Values for r_1 and r_2 were determined empirically. Good results were obtained using $r_1=0$ and $r_2=20$ pixels, and these values were used in all of the enhancements presented in this thesis.

For a given value of bandwidth $\phi_{BW} > \pi/8$, the effect of increased bandwidth was simulated by forming an appropriately weighted linear combination of prefiltered image values of adjacent filter orientation, using the technique described at the end of section 3.4.2.

Figure 3.5a shows a region of a fingerprint containing singular points. Figure 3.5b is the enhancement produced using the above method of varying the angular bandwidth near to the singular points. On the other hand, if no regard is taken of the higher curvature near the singular points, i.e. the angular bandwidth is kept at $\pi/8$ throughout, then the enhancement of figure 3.5c is obtained. Comparison of these two enhancements shows that the

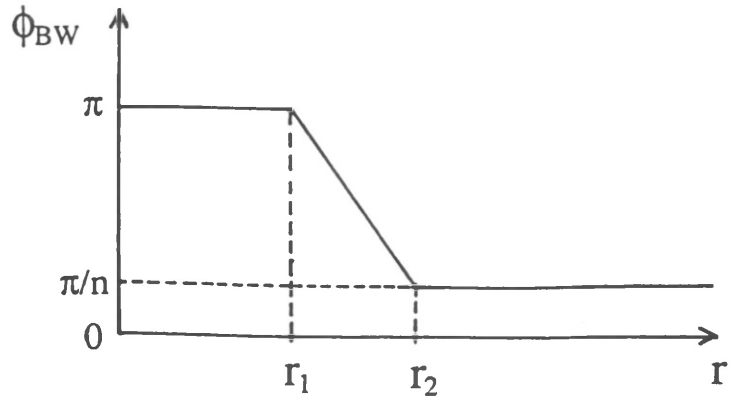
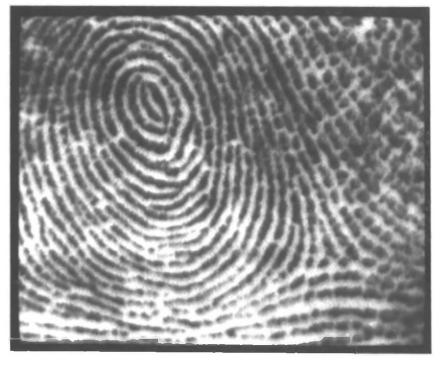
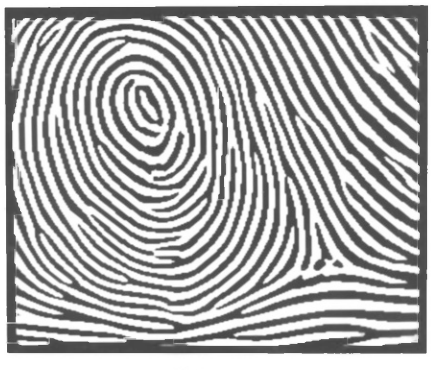


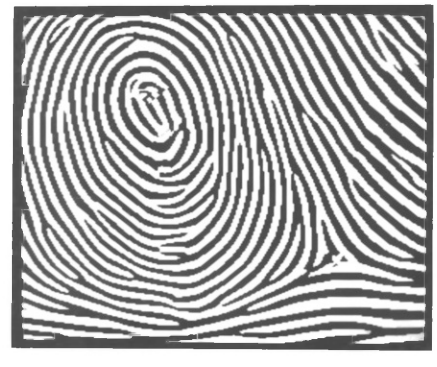
Figure 3.4. Filter angular bandwidths near singular points. The graph shows the angular bandwidths used, as a function of distance from the nearest core or delta point.



(a)



(b)



(c)

Figure 3.5. (a) A region of a fingerprint, containing singular points.

(b) The enhancement obtained when the filter's angular bandwidth is increased near the singular points, in accordance with the graph of figure 3.4.

(c) The enhancement obtained when the filter's angular bandwidth is kept constant everywhere.

ridge structure in the neighbourhood of the singular points is correctly represented in figure 3.5b, but distorted in figure 3.5c. This demonstrates both the necessity and the effectiveness of the method proposed.

• **Local ridge spacing and local ridge expansion**

These two parameters determine the radial centre frequency and radial bandwidth of the filter respectively.

The approach adopted here is a simplified one. As explained in the first chapter, the LRS does not vary by more than a factor of 4.5 overall (or 3 within any individual print), and additionally it it a slowly-varying function, so that the LRE is small everywhere. The present technique as implemented takes advantage of this by keeping the radial centre frequency at a constant value corresponding to the average LRS, and by using a wide enough radial bandwidth to pass all values of LRS that could occur.

This approach has the advantage that it is simple to implement, since there is no need to have estimates for LRS and LRE as functions of position. Although good enhancement is obtained using a relatively wide radial bandwidth, it is true that noticeably better results can be achieved if the radial bandwidth is reduced and the radial centre frequency varied according to the LRS, for with narrower bandwidth filters less of the unwanted noise would be allowed through. This was not attempted in this work primarily because it was not expected that the likely benefits would justify the extra time that would have to be spent

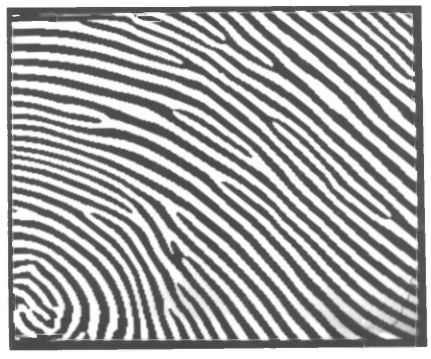
in implementing this approach. Additionally, it would have necessitated the storage of more pre-filtered images.

Nevertheless, a limited investigation of the effect of using narrower-bandwidth filters of the correct centre frequency was undertaken. The fingerprint of figure 3.6a was used in this investigation because it is of poor quality and therefore the relatively large amount of noise present could be expected to adversely affect the enhancement using the standard wide-bandwidth filters. This enhancement is shown in figure 3.6b, and a certain amount of noise can be distinguished visually in this image. In particular, the presence of pores on the ridges has led to unwanted frequency components at twice the ridge frequency. These components have been passed by the filters, resulting in jagged edges and small holes in the ridges.

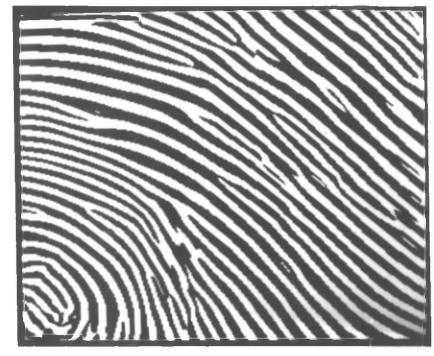
The bandwidth used to produce the enhancement of figure 3.6b was wide enough to pass the full range of ridge spacings for all fingerprints. However, within a single print, the range of spacings that occur is smaller. Therefore, the bandwidth was reduced to the narrowest value consistent with passing all the desired components within the print of figure 3.6a, and the enhancement operation was performed again, yielding the image of figure 3.6c. This image has less noise than figure 3.6b, because the unwanted components at twice the ridge frequency have been filtered out. This shows that improved enhancement can be obtained by reducing the radial bandwidth of the filters. Indeed, the quality of enhancement attainable using accurate LRS estimates (and hence larger numbers of prefiltered images) should be better



(a)



(c)

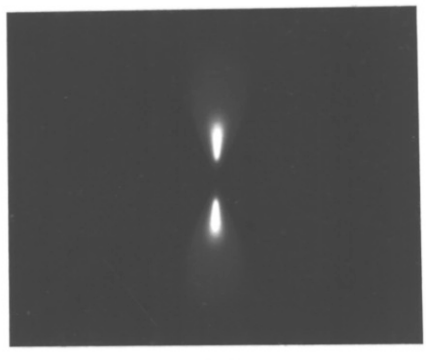


(b)

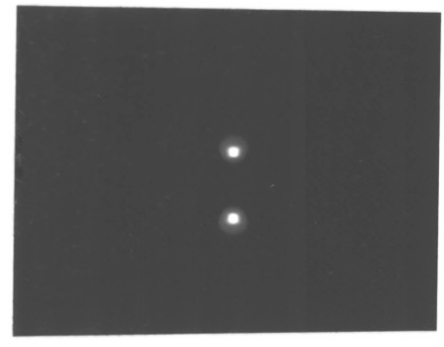
Figure 3.6. (a) A poor quality region of a fingerprint.

(b) Enhancement obtained using filters of wide radial bandwidth.

(c) Enhancement obtained using filters of narrow radial bandwidth.



(a)



(b)

Figure 3.7. Directional filters.

(a) The filter used to produce the image in figure 3.6b and the results in chapter 6.

(b) The narrow bandwidth filter used to produce the image of figure 3.6c.

still, since the bandwidth can then be reduced even further.

Figures 3.7a and 3.7b show the wide and narrow radial bandwidth filters which were used to form the horizontal components from which the enhancements of figure 3.6 were constructed.

CHAPTER 4

DIRECTIONAL CHARACTERISTICS OF IMAGES

This chapter, together with Appendix A, examines the concept of directionality in images. This is done because local ridge orientation is the most important parameter describing the structure of a fingerprint image, and is an example of a type of direction. As a result of this investigation a mathematical model of local ridge orientation in fingerprints will be obtained which is a function of singular point types and positions. This model will prove useful in resolving ambiguities due to the periodic nature of orientation, in an algorithm that will be developed in chapter 5 to interpolate between sampled values of local ridge orientation.

4.1 Direction and orientation

Two distinct types of direction can be defined. One of these, which we shall call simply "direction", is easily described by the elementary mathematics of vectors. The other type, called "orientation", is the one that turns out to be applicable to directionality in images, and requires for its description the concept of the "direction field" from a branch of mathematics known as differential geometry.

There are fundamental differences between directions and orientations. Fortunately, however, these differences become comparatively trivial if the discussion is restricted to two dimensions. Discussion of directionality in higher dimensional images requires the introduction of extra concepts and notation. These extensions are presented separately in appendix A.

The two types of directionality will now be introduced. Consider the plane and suppose that an arrow or vector has been drawn with its tail at the origin, as shown in figure 4.1a. This vector will form some angle θ with the Cartesian x-axis, where θ can take on any value in the range 0 to 2π radians. θ is called the **direction** of the vector. Direction is a periodic measure, with a period of 2π, since any angle $\theta+2n\pi$ corresponds to the same vector as the angle θ, for any integer n. For every direction θ there is a corresponding point on the unit circle S^1, and therefore S^1, which has circumference 2π, can be used to represent the periodic or circular interval $[0,2\pi)$. Directions of vectors can therefore be naturally represented as elements of the set S^1.

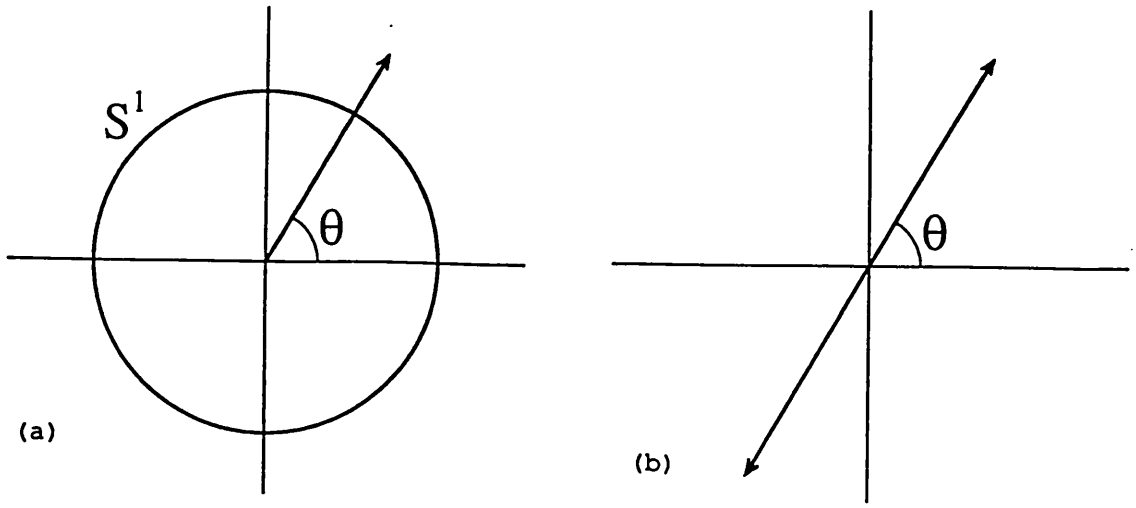


Figure 4.1. (a) direction of a vector
(b) orientation of a line

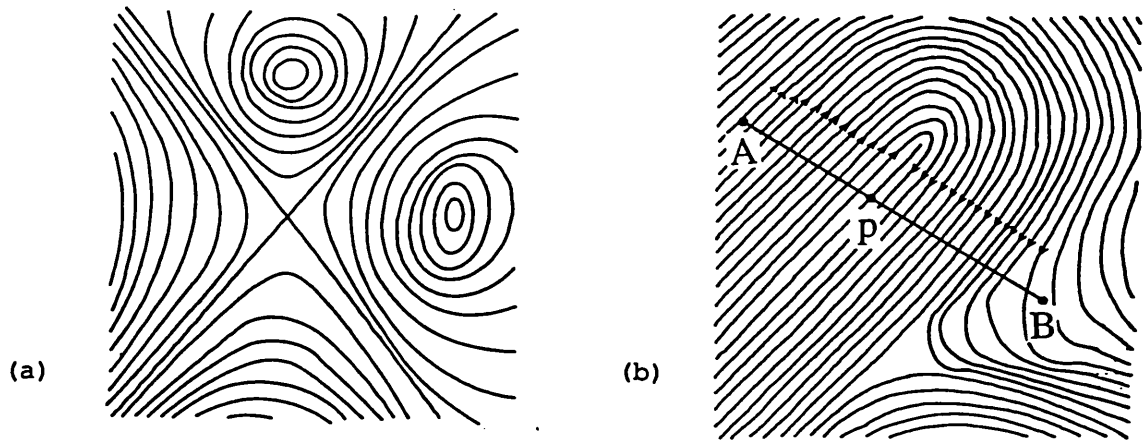


Figure 4.2. Orientable (a) and non-orientable (b) sets of curves.

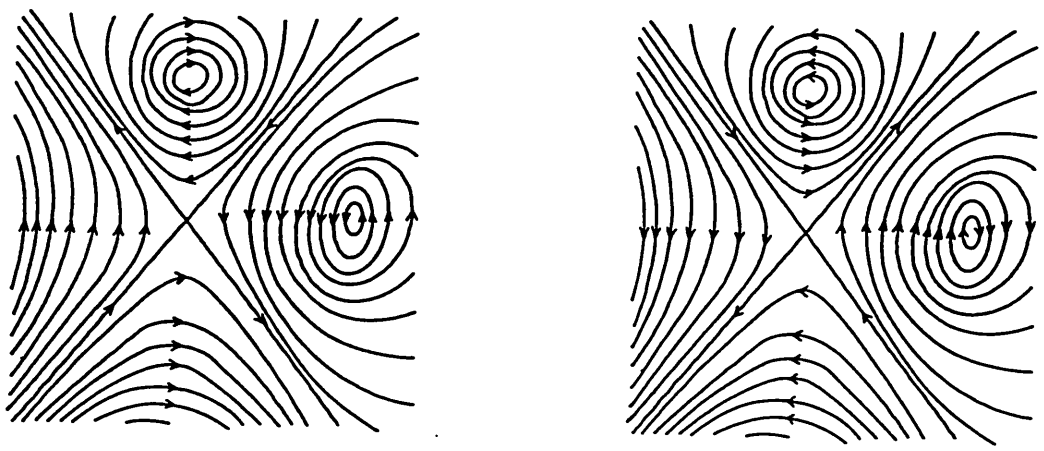


Figure 4.3. The two orientations of the curves of figure 4.2a.

Suppose now that instead of a vector, a straight line passing through the origin is drawn, as illustrated in figure 4.1b. This line forms some angle θ with the x-axis; however, this time the angle is in the range $[0,\pi)$ rather than $[0,2\pi)$ because rotating the line through $n\pi$ restores it to its original position, for any integer n. The angle θ is called the **orientation** of the line. The periodic interval $[0,\pi)$, to which orientations belong, will be represented by the symbol P^1 and may be thought of as a circle with circumference π, and radius 1/2.

4.2 Directionality in images

Within a small neighbourhood of a point in an image, it is often true that the information is not isotropic. In this case there is usually a predominant direction in which the information is orientated. (For example, the anisotropic image restoration techniques of Knutsson, Granlund and Wilson are based upon the assumption of the existence of a predominant direction of local anisotropy). Whether or not this is true for any given point may be determined by examining the Fourier transform of a small window of data around the point. If the energy in the magnitude transform is mainly concentrated near a line through the origin, then the local orientation of data in the spatial domain at the point is perpendicular to this line.

For example, various textures in images, such as grass or water ripples in photographs, clearly do have predominant local orientations. There may be regions where the local orientation is defined, and regions where it is not. The local orientation, where it is defined, need not be continuous as a function of position. For example, a photograph of an object textured at one orientation set against a background textured at another orientation, will determine an orientation function which is discontinuous at the boundary between object and background.

Fingerprints, however, are a nearly unique class of images in terms of their directional properties. A local orientation is both defined and continuous at all points within a print except for the few isolated singular points corresponding to the cores and deltas. The methods developed in this thesis take full

advantage of this property of fingerprint images.

With very few exceptions, directionality within images is better described in terms of orientation rather than direction. Given a small neighbourhood of some point, the local orientation of information can be found by examination of the data within the neighbourhood. However, a local direction can in general not be determined by such an examination since there will not be any way of assigning a forward sense to the orientation thereby yielding a direction. (The only exceptions occur when the image happens also to represent a scene whose directionality is "orientable" globally (see section 4.3), and even then additional a priori information will be required to identify the positive sense of local orientation). It is clear, then, that the natural directionality of images is the directionality of lines, i.e. orientation.

The appropriateness of the concept of orientation rather than direction to image directionality is confirmed by the Hermite symmetry exhibited by the Fourier transform of any real-valued function $f(x,y)$ (such as an image). Expressed in polar coordinates (ρ,ϕ), this symmetry is:

$$F(\rho,\phi) = F^{*}(\rho,\phi+\pi) \quad \text{for all } \rho \text{ and } \phi.$$

Therefore there is a two-fold redundancy of information in the Fourier domain. Points radially opposite each other in the Fourier domain are complex conjugates of each other, which indicates that information in the direction ϕ cannot be distinguished from information in the direction $\phi+\pi$, which corresponds to the same orientation as ϕ.

4.3 Vector fields and direction fields.

Direction fields, which will be introduced in this section, are not usually covered in detail in differential geometry textbooks, presumably because not many applications for them have been found. The most comprehensive treatment encountered during the course of this research was [Godbillon 1983]. Spivak [1979, Vol 3 Chap 4 Addendum 2] introduces direction fields in his 5-volume work on differential geometry and provides examples of various types of singular points. Markus [1955] proved theorems concerning the singularities and orientability of direction fields. Others mention direction fields in passing [Arnold 1981 page 7, 1973 page 12, 1983 page 2; Arnold et al 1985 page 21; Dodson 1980 page 176; Hawking & Ellis 1973 page 39; Reid 1971 page 9; Stoker 1969 page 244].

A **vector field** V on the plane is defined by specifying in a continuous manner an arrow or vector V_p at each point p in the plane, except for isolated points known as the singular points of V. The magnitude of the vector is a nonnegative real number indicating the length of the arrow and the direction of the vector is the direction of this arrow, which will be an element of S^1, the circular interval $[0,2\pi)$.

This discussion will only be concerned with the directional properties of vector fields, and therefore the magnitude will be assumed to be unity everywhere. Under this restriction, the direction is continuous everywhere except at the singular points of V.

A **direction field** (or **line element field**) E on the plane is defined by specifying in a continuous manner a line E_p at each point p, except for isolated points called the singular points of E. The orientation of the field at p is the orientation of the line E_p, and so a direction field may be regarded as a 2-d function taking on values in the set P^1, the circular interval $[0,\pi)$. This function will be continuous everywhere except at the singular points of E.

It is perhaps unfortunate that direction fields were not named "orientation fields" instead, since they are fields of orientations rather than of directions. A similar remark applies to the term "orientable" introduced later; "directable" or some similar term would be less confusing.

An **integral curve** C of a vector field V is a curve to which the vector field is tangential everywhere: given any point p on C, the tangent vector to C at P is V_p. If the vector field is taken to represent the flow of some fluid, then the integral curves represent the paths of travel of the individual fluid particles.

For direction fields, integral curves are defined analogously. An integral curve of a direction field E is a curve C to which the direction field is everywhere tangential. Given any point p on C, the line E_p is tangential to C at p.

Consider the set of all integral curves of a direction or vector field. This set of curves is called **orientable** if it is possible to assign a forward sense to the curves by putting arrows on them, in such a way that the sense of these arrows is consistent and

continuous everywhere excepting the singular points of the field. Figure 4.2 shows an example of an orientable and a non-orientable set of curves. In the non-orientable example, a discontinuous change in the direction of the arrows occurs at the point p along the path AB shown.

A vector or direction field will be called **orientable** if its set of integral curves is orientable.

Vector fields are always orientable. Because their integral curves are everywhere tangential to vectors, the forward sense of the curves can be equated to the sense of the tangent vectors, and the continuity of the vector field will ensure consistency in this process.

Direction fields, on the other hand, are not necessarily orientable. In this sense, direction fields are a generalization of vector fields. Note the similarity between the non-orientable example of figure 4.2 and the ridge pattern of a loop fingerprint.

The following two results [Godbillon 1983] concerning the relationship between the integral curves of direction and vector fields are relevant:

i. Any vector field V induces an orientable direction field E having the same integral curves as V. For each p, E_p is given by a line at orientation $\theta \bmod \pi$, where θ is the direction of V at p.

ii. If a direction field E is orientable, then it induces two vector fields having the same integral curves as E. Each one

is called an **orientation** of E. This is illustrated in figure 4.3.

In the discussion that follows on Poincare indices of fields, the word "curve" should be taken to mean a curve not passing though any of the singularities of the field in question.

Given a vector field V and any closed curve C on the plane the **Poincare index** i(C) of V along C is defined as the "algebraic number of turns" which the direction of V makes as C is traversed in the anticlockwise sense. (By a "turn" is meant a turn through 2π radians).

The Poincare index is defined analogously for a direction field as the number of "turns" (still through 2π) made by the orientation of E as the curve is traversed in the anticlockwise sense.

Any two closed curves C_1 and C_2 continuously deformable into one another without passing through singularities of the field satisfy

$$i(C_1) = i(C_2),$$

and therefore the index ix(p) of any point p with respect to the field can be defined to be the Poincare index of any closed curve surrounding p and no other singular point.

The Poincare index of a field along a closed curve C is the sum of the indices of all the singular points surrounded by C.

For any vector field, the index of any point is always an integer. If this were not so, the continuity of the direction of the field would be violated along any curve around the point. The

following, which are illustrated in figure 4.4, are frequently encountered:

- ix(p) = 0 for all non-singular points (no net direction change).
- ix(p) = -1 for a saddle point (direction change of -2π).
- ix(p) = +1 for a centre (direction change of 2π).
- ix(p) = +1 for a source or sink (direction change of 2π).

For direction fields, on the other hand, the index of a point must be an integer or half-integer. Each orientation change of π changes the index by 1/2, and therefore continuity of the orientation is still consistent with half-integer indices. However, if the direction field is orientable, then the indices of all points are integers, since the indices of the direction field must be the same as those of the two corresponding vector fields.

Figure 4.5 shows patterns of lines resembling the areas around core and delta points in fingerprints, showing that the index of a core is +1/2 and the index of a delta is -1/2, corresponding to changes in LRO of π and $-\pi$ respectively as the point is circumnavigated in an anticlockwise direction. The centre of a circular whorl has an index of +1 and can be regarded as two superimposed cores each contributing +1/2 to the index. These values of Poincare indices for the various fingerprint singular points were observed by Kawagoe and Tojo [1984], but without explanation or justification.

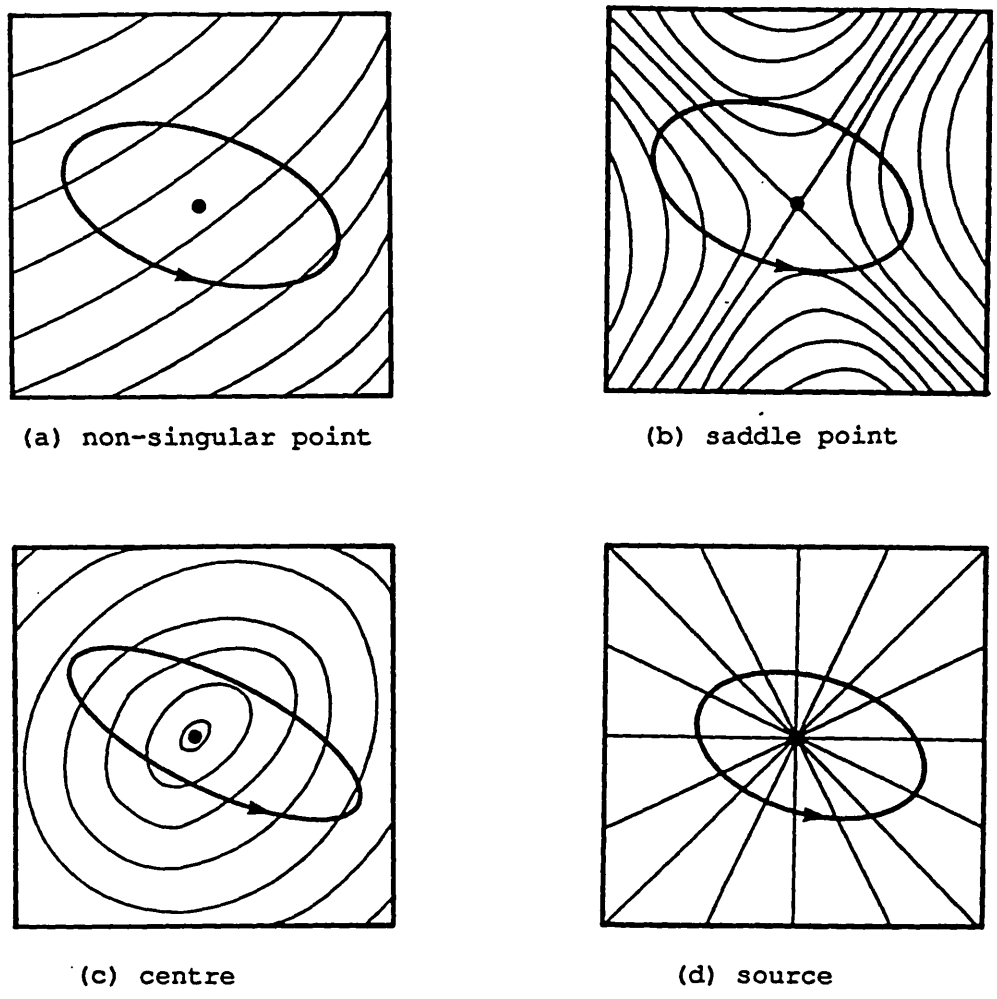


Figure 4.4. Singular and non-singular points of vector fields.

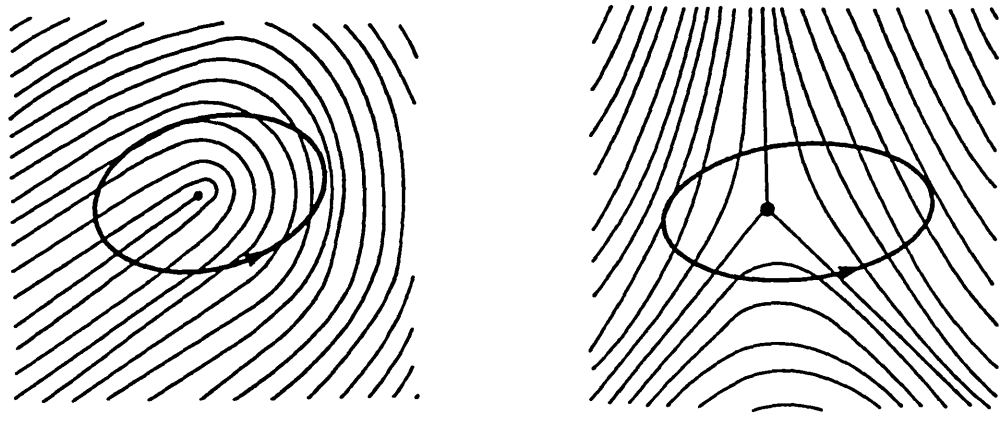


Figure 4.5. Types of singular points for direction fields. Note that these resemble the core and delta structures of fingerprints.

4.4 A model of local ridge orientation

It should now be clear that the local ridge orientation function LRO(x,y) should be modelled as a direction field having singular points $d_1 \ldots d_k$ of index -1/2 at the delta positions and singular points $c_1 \ldots c_m$ of index 1/2 at the core positions.

Since vector fields cannot have half-integer indices, LRO cannot in general be modelled as the direction of any vector field. This rules out several alternatives which might have been considered in developing a mathematical model of ridge orientation and the corresponding "ridge flow lines":

i. They cannot be represented as the lines of flow of a fluid.

ii. They cannot be represented as the contour or gradient lines of a 2-d function.

iii. They cannot be represented as the integral curves of any other vector field.

All of the above are expressible as integral curves of some vector field, and therefore cannot exhibit the non-orientability and Poincare index values characteristic of fingerprints.

The major factor influencing the behaviour of the LRO function is the number and position of the cores and deltas. The model of LRO which is developed here is the simplest possible model of LRO which accounts correctly for its behaviour around the singular points.

Of course the function $\theta(x,y)$ obtained in this way cannot be

expected to predict the true value of LRO accurately, because the positioning of the singular points, although of greatest influence, is not the only influence upon LRO. That is to say, it is easy to imagine two fingerprints with significantly different LRO but having identical singular points. (As an example, note that all plain arch fingerprints are devoid of singular points and would be represented by the same function in our model). Nonetheless, any two prints with the same singular points will exhibit identical indices everywhere, and these indices will be the same as those predicted by the model.

Therefore, the $\theta(x,y)$ which will be obtained from the model should be regarded as the LRO of some "ideal" fingerprint having the given core and delta positions. Its relation to the actual LRO of a real fingerprint is discussed briefly in section 4.6.

The model is easily derived. First, regard the image plane as the complex plane **C**, and consider the rational polynomial function

$$q(z) = \frac{(z-z_1)(z-z_2)..(z-z_k)}{(z-p_1)(z-p_2)..(z-p_m)}$$

with first-order poles and zeros $p_1,...,p_k$ and $z_1,...,z_m$. It is well known that the argument of $q(z)$ satisfies:

1. The Poincare index of each pole is -1; and
2. The Poincare index of each zero is +1.

Higher-order poles and zeros can be accommodated by repeating the factor $(z-z_i)$ or $(z-p_i)$ according to the order of the pole or zero.

Noting that cores and deltas have Poincare indices of 1/2 and -1/2, define

$$p(z) = \sqrt{\frac{(z-z_{c1})(z-z_{c2})..(z-z_{ck})}{(z-z_{d1})(z-z_{d2})..(z-z_{dm})}}$$

where $z_{c1},\ldots,z_{ck}$ are the locations of the k cores, and

$z_{d1},\ldots,z_{dm}$ are the locations of the m deltas,

which has the required Poincare indices, since taking the square root of a complex number halves its argument.

The centre of a circular whorl can, as we have seen, be modelled as two superimposed cores. In such a case, the factor $(z-z_c)$ would have to be included twice.

Thus the model is

$$\theta(z) = (\mathrm{Arg}(p(z))) \bmod \pi$$

A slight extension to the model was obtained by using the result given in section 4.6, that it is consistent to assume that far from the centre of the image, the ridge orientation tends towards a constant value, say θ_∞. It is also shown there that this implies that the number of cores is equal to the number of deltas, i.e. that k=m for any fingerprint.

Consequently, p(z) can be modified to

$$p(z) = \sqrt{\exp(j2\theta_\infty).\frac{(z-z_{c1})(z-z_{c2})\ldots(z-z_{ck})}{(z-z_{d1})(z-z_{d2})\ldots(z-z_{dk})}}$$

where θ_∞ is the value of the ridge slope at infinity. The image

is usually aligned so that $\theta_\infty = 0$ (horizontal ridges at infinity).

4.5 Evaluation and extension of the LRO model

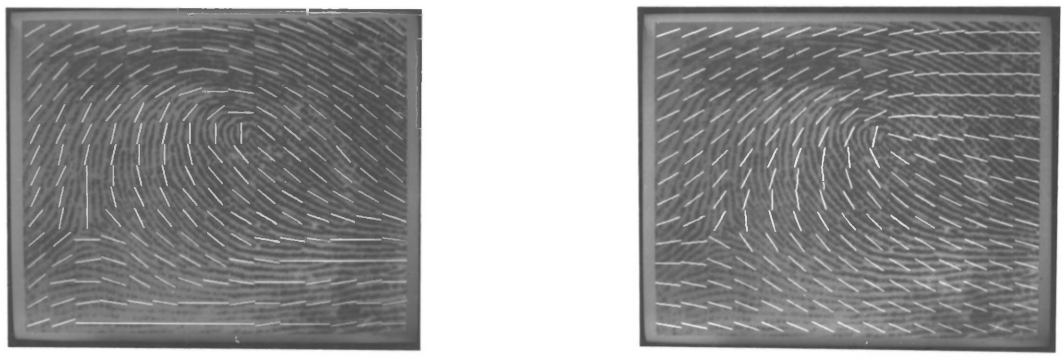
Figure 4.6a shows a fingerprint with line segments superimposed to indicate sampled values of the LRO. In figure 4.6b, the superimposed lines indicate the orientations predicted by the model. As expected, the individual values of LRO predicted by the model are not correct everywhere. However, the overall behaviour of the modelled LRO is correct, because it can be seen to "flow" around the cores and deltas in the correct manner, i.e. it exhibits the correct Poincare indices everywhere.

The actual LRO of a real fingerprint is a direction field with singular points at the cores and deltas. The modelled orientation function $\theta(x,y)$ is a direction field having identical singular point positions and types. Therefore, the following equation will be satisfied:

$$LRO(x,y) = \theta(x,y) + \varepsilon(x,y)$$

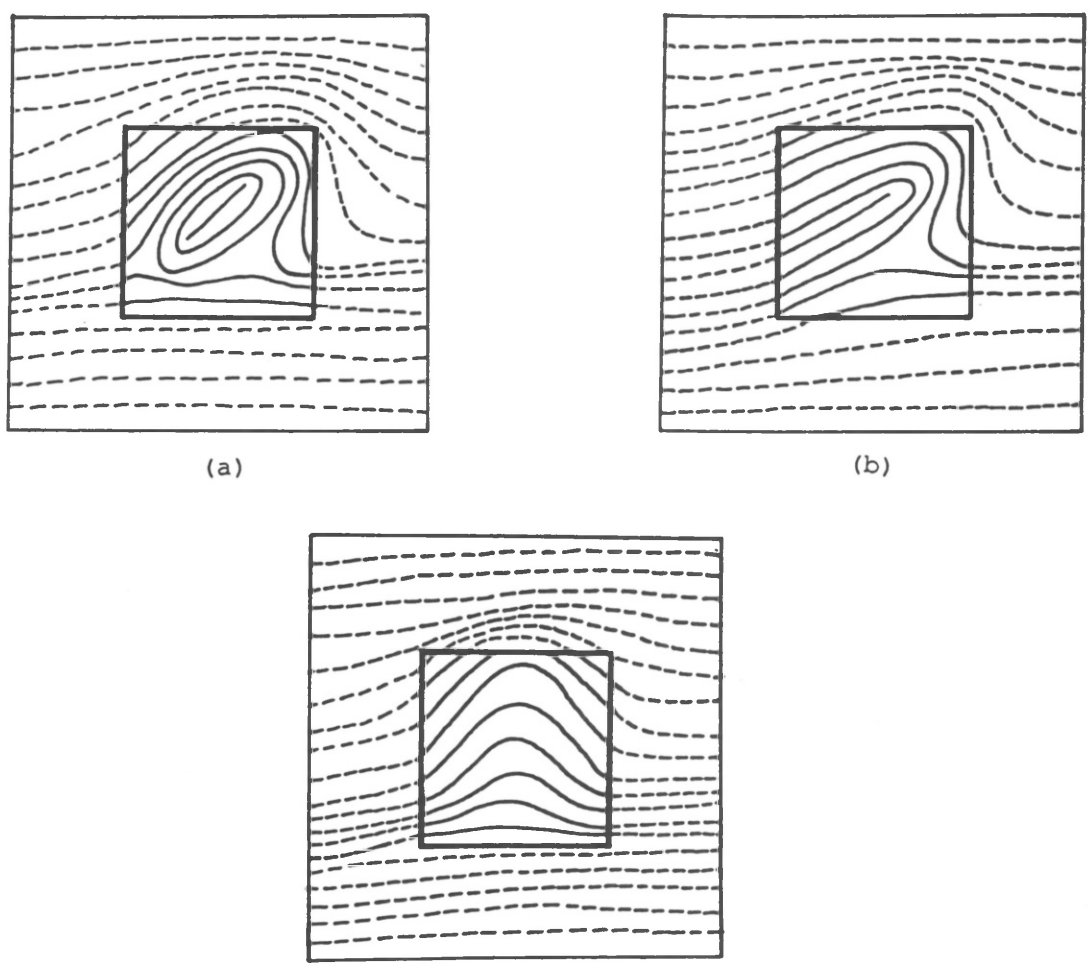
where the error $\varepsilon(x,y)$ is continuous as an orientation-valued function, i.e. is a direction field, and has indices of value zero at all the singular points of $LRO(x,y)$.

The application for which the model was developed, i.e. the resolution of ambiguities which occur during interpolation between sampled LRO values, does not require the model to provide an accurate value of LRO at each point. The LRO is, after all, already known at a grid of sample points. Instead it is sufficient for this application that, given two points near to each other, the model provide an indication of the rate of change of LRO as the line joining the two points is followed. It is



(a) (b)

Figure 4.6. Orientations predicted by the LRO model. The correct values of the orientations are shown in (a), and the values given by the model in (b).



(a) (b)

(c)

Figure 4.7. Illustrating that LRO can be regarded as tending towards zero (horizontal) sufficiently far from the centre of the print. this is illustrated for whorl (a), loop (b) and arch (c) patterns.

shown in chapter 5 that an estimate of the local ridge curvature given by grad $\theta(x,y)$ is adequate for this purpose.

In this research, therefore, it was not necessary to improve upon the value $\theta(x,y)$ given by the model. However, if a more accurate expression for LRO were required, this could be obtained by obtaining an estimate of the error $\varepsilon(x,y)$. $\varepsilon(x,y)$ is continuous and will usually be slowly varying except in the vicinity of the singular points, since both LRO and θ have this property.

Suppose that some accurate, but possibly expensive, method is known which yields the true value of LRO at a point. Let the LRO be known at the point (x_p,y_p). If this point is not close to singular points, then the error $\varepsilon(x,y)$ can be assumed to be approximately constant within some neighbourhood of the point. Within this neighbourhood, the ridge orientation can be approximated by

$$LRO(x,y) \approx \theta(x,y) - \varepsilon(x_p,y_p)$$

where $\varepsilon(x_p,y_p) = LRO(x_p,y_p) - \theta(x_p,y_p)$.

If LRO is known accurately at a few widely-spaced points $\{p_i\}$ in a region which is not close to singular points, then an estimate $\hat{\varepsilon}(x,y)$ of the error could be obtained by interpolation from its sampled values $\{\varepsilon_i\}$, yielding

$$LRO(x,y) \approx \theta(x,y) - \hat{\varepsilon}(x,y)$$

valid within the above region.

The above technique is essentially a means of interpolation

between sampled LRO values, guided by the model $\theta(x,y)$, rather than a more accurate model of LRO. If the sampled values of LRO vary within a range greater than $\pi/2$, then problems may arise in interpolating between the ε_i. The function $\varepsilon(x,y)$ is continuous as an orientation-valued function but discontinuous as a real-valued function, and standard interpolation techniques will not work on discontinuous functions. The problems associated with interpolating between sampled LRO values are discussed in detail in chapter 5.

4.6 Direction fields of fingerprint-like patterns

Fingerprint-like patterns may be defined as patterns of locally parallel curves within a simply-connected region of the plane, which are such that the orientation of the curves is defined and continuous everywhere except at a finite number of singular points. This ensures that such patterns consist of the integral curves of direction fields, and therefore puts certain constraints upon the range of patterns that are possible. One of these constraints is the law stating that the Poincare index of a curve is equal to the sum of the indices of the enclosed singular points.

Dermal ridge patterns upon the palm and the sole of the foot are examples of fingerprint-like patterns. Other examples also occur in nature, such as the stripes on a zebra, and the pattern of sand-dune ridges on the surface of the earth in desert areas. In this section, two results of Penrose [1965a] concerning core and delta numbers within dermal ridge patterns will be confirmed using the model of LRO.

Consider first the fingerprint itself. After examination of a large number of prints it was concluded that it is always consistent to assume that the LRO tends to a constant value, say θ_∞, as one moves far enough out from the centre of the print. This idea is illustrated in figure 4.7 for a few types of print. No reference could be found to support this conclusion, although a fingerprint expert at the Home Office verified personally that it is indeed true [Millard 1987].

Therefore, a circle C of sufficiently large radius around the print will have a Poincare index of zero, so that

$$ix(C) = 0.$$

Therefore

$$\sum_{i=1}^{k} ix(z_{ci}) + \sum_{i=1}^{m} ix(z_{di}) = 0$$

if the print has k cores and m deltas.

However, each core has index 1/2 and each delta has index -1/2, and therefore

$$k = m,$$

i.e. the number of cores is equal to the number of deltas. Examination of a few fingerprints should satisfy the reader that this is true in practice.

Let us now examine palmprints. Cummins and Midlo [1943] describe print patterns on palms and soles in detail. The research of Penrose [1963, 1965a] showed that the following results are true for all palmprints, even when the hand is deformed or has extra or missing fingers:

- Where a finger is attached to the palm, the ridges run across the finger rather than along it.
- The same is true for the region where the arm attaches to the palm. The ridges run across the arm rather than along its orientation.

- In parts of the edge of the palm where no limb or finger is attached, i.e. the gaps between the fingers or between finger and limb, the ridges are perpendicular to the edge of the palm.

(These results also apply to footprints if toes are substituted for fingers and legs for arms in the above discussion).

A schematic view of a palm is given in figure 4.8, showing the pattern of dermal ridges around the edge of the palm, and illustrating the above three results.

As the edge of the palm is traversed in an anticlockwise direction, we see that the LRO rotates clockwise through π whenever the join between a limb or finger and the palm is traversed. In addition, there is an overall anticlockwise rotation through 2π which is contributed by the fact that the orientation of the edge of the palm changes through -2π as it is followed round.

We can therefore rather whimsically propose a "generalised palm" which has n_f fingers and/or toes and n_l limbs attached to it, and satisfies the properties given above for palms. Then, as the edge is traversed, the net rotation of LRO is

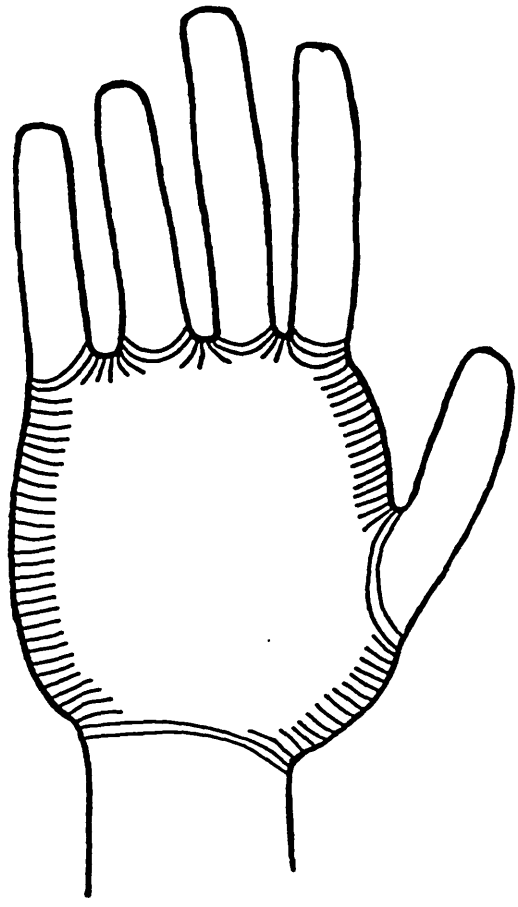
$$-2\pi + (n_f+n_l)\pi,$$

so that the index of the curve C around the edge is

$$ix(C) = 1/2(-2 + n_f + n_l).$$

If the palm has n_c cores ands n_d deltas on it, then we know that

(a)



(b)

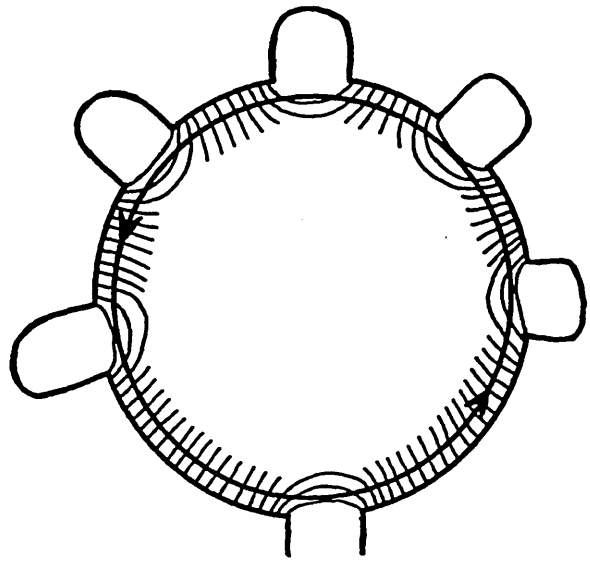


Figure 4.8. Ridge structure around the edge of the palm. (b) shows the palm edge represented schematically in circular form.

$$ix(C) = \sum_{i=1}^{n_c} (-1/2) + \sum_{i=1}^{n_d} (1/2)$$

$$= 1/2\ (n_d - n_c).$$

Therefore $n_d - n_c = -2 + n_f + n_l$.

For the particular case of a normal hand or foot, $n_f=5$ and $n_l=1$ so that $n_d - n_c = 4$, meaning that any palm always has four more deltas than cores.

CHAPTER 5

DETERMINING LOCAL RIDGE ORIENTATION

5.1 Introduction

The enhancement algorithm which will be presented in Chapter 6 makes use of the filtering techniques which were developed in Chapters 2 and 3. Because of the nature of the filters, i.e. their spatial dependence upon the local ridge orientation, it is necessary when implementing them to have a means of determining the local ridge orientation at each point within the image. Solutions to the problem of determining local ridge orientation are developed in this chapter.

During the research leading to this thesis, the development and evaluation of high quality filtering techniques for enhancement was the primary objective. Development of ways to determine the values of the parameters (such as LRO), though of course necessary as part of any implementation of such techniques, was regarded as a digression from the main theme of investigation, and consequently of relatively lower importance. The approach taken was therefore to develop techniques which performed satisfactorily, without expending an excessive amount of effort upon optimizing these techniques and making them fully robust, since this effort could be more profitably directed towards the main areas of research. In particular, methods of smoothing

sampled estimates of LRO were not investigated.

5.1.1 The need for interpolation

The enhancement algorithm of chapter 6 requires a value for LRO at each point within the image. It is important that the filtering stage of this algorithm produce a filtered image which is a continuously varying function of position, otherwise unwanted artefacts could occur near the points of discontinuity during the thresholding which is performed after the filtering stage. In order for the filtering stage to produce a filtered image that is continuous, it should be presented with values of LRO that vary in a continuous manner from pixel to pixel.

Because slight inaccuracies in orientation values can if necessary be catered for by a small increase in the angular bandwidth of the filters used, high precision in the orientation function values (though desirable) is of less importance than spatial continuity of this function.

Obtaining a value for LRO at each pixel is believed to be a new approach to fingerprint enhancement on general purpose computers. In previous algorithms, such as the Home Office AFR algorithm, the print has been divided into a grid of rectangular sub-images, and an average value for the LRO determined within each. The image has then been processed assuming that the LRO is constant within each sub-image. This assumption is made primarily because evaluation of the LRO on a general purpose computer usually involves considerable computation, and it would therefore be too

slow and inefficient to evaluate LRO directly at all points within the image. In any case, LRO varies rapidly only in the regions of high curvature near the cores and deltas, and therefore the grid of values is sufficiently accurate over most of the image.

For the algorithm proposed in this thesis, however, this assumption of a piecewise constant LRO function is not adequate, because of the importance of continuity of LRO as a 2-d function.

Values of LRO satisfying the requirement for continuity can be obtained by evaluating the LRO directly at each point in the image, using one of the methods that will be presented in section 5.2. However, this approach would be far too slow unless special hardware is available to perform the ridge orientation detection algorithm. For example, evaluation of the LRO at a point using the method of section 5.2.4 (which was the one chosen) involves rotating a window through eight orientations, forming and filtering eight projections, and evaluating a variation measure eight times. This would prove excessive if repeated for each of 262144 pixels in a 512 by 512 pixel image.

An alternative is to evaluate the LRO directly at a grid of points and then obtain values at intermediate points by interpolating between the samples. In this work, the LRO was evaluated on a grid of 30 by 30 positions, two consecutive sampling points being separated by 16 pixels. This reduces the number of times that the LRO must be evaluated by a factor of about $16^2 = 256$, and leads to a correspondingly large saving in processing time.

In interpolating between orientation-valued samples, problems

caused by the multivalued nature of orientation may be encountered. These problems and their solution are discussed in section 5.3.

5.1.2 Sampling of orientation values

In the technique of section 5.2.1 for evaluating LRO at a point, the LRO estimate obtained can take on any value in the range $[0,\pi)$ of possible orientations. However, in the other techniques of section 5.2, the estimate is obtained as one of a number n of discrete equally-spaced orientation values θ_i such that

$$\theta_i = i\pi/n, \quad \text{for } i=0..n-1.$$

The estimate θ_i is then the nearest of the θ's to the correct value of LRO at the point.

In such methods, a value must be chosen for the number n of distinct orientations that can be determined. For larger values of n, the LRO can be specified to greater precision, but more computational effort will usually be required to obtain the estimate. Above a minimum of about n=6, any value for n could be chosen, since any slight imprecision introduced by a low value can be compensated for by a small increase in the angular bandwidth of the filters used.

Other researchers have used values of n=4 [Moayer & Fu 1975] and n=8 [Grasselli 1968]. Using n=8, an angle can be expressed to a precision of +/- $\pi/16$. It was not expected that the methods of section 5.2 could reliably determine LRO to any greater precision, and a value of n=8 was therefore chosen.

5.2 Finding LRO at a point

Various alternative approaches to determining the LRO at a point in a fingerprint image are described in sections 5.2.1 to 5.2.4. Section 5.2.5 describes the use of an "orientation editor" for the correction of inaccurate estimates.

5.2.1 The generalised operator approach

As we have seen, Granlund [1978] introduced the Generalised Operator (GOP) approach to image restoration. In this approach, local anisotropy within images is regarded as one of the most important properties of images, and it is assumed that at each point within the image there is a well defined local orientation of the anisotropy. It is postulated that the local anisotropy can be described in terms of lines and edges in the image, and therefore to detect the orientation of the anisotropy, spatial convolution masks based upon Fourier domain line and edge detectors are used [Knutsson et al 1983b].

A set of line and edge extraction filters $E_i(x,y)$ and $L_i(x,y)$ oriented in the four orientations

$$\theta_i = (i-1)\pi/4 \qquad i=1..4$$

are used. Descriptions of these filters are given in [Knutsson & Granlund 1980]. The image $F(x,y)$ is convolved with these filters, to give:

$$\begin{aligned} S_i(x,y) &= F(x,y) * E_i(x,y) \\ C_i(x,y) &= F(x,y) * L_i(x,y) \end{aligned} \qquad i=1..4$$

S_i and C_i are measures of the edge and line information near the point (x,y).

Then a magnitude $B_i(x,y)$ of line/edge information in each of the four orientations is calculated using

$$B_i(x,y) = \frac{\sqrt{S_i^2(x,y) + C_i^2(x,y)}}{V(x,y)}$$

where

$$V(x,y) = [\sum_{i=1}^{4} \{S_i^2(x,y) + C_i^2(x,y)\}]^{\beta}$$

where $\beta=0.5$.

The local orientation of anisotropy $\theta(x,y)$ is calculated using

$$\sin 2\theta(x,y) = \frac{B_2(x,y) - B_4(x,y)}{B(x,y)}$$

$$\cos 2\theta(x,y) = \frac{B_1(x,y) - B_3(x,y)}{B(x,y)}$$

where

$$B(x,y) = \sqrt{[B_1(x,y) - B_3(x,y)]^2 + [B_2(x,y) - B_4(x,y)]^2}$$

This gives an orientation $\theta(x,y)$ in the range 0 to π radians.

This technique was implemented, using a window size of 15 by 15 pixels, and tested on the two fingerprint images of figure 5.1. Figure 5.2 shows the results obtained, both for a sampled grid of values and for direct evaluation at each pixel in the image. It can be seen from these results that individual applications of the

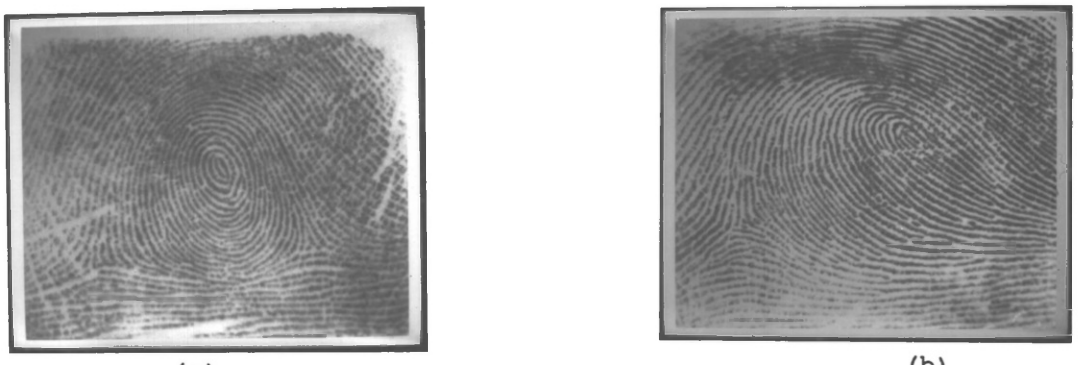


Figure 5.1. Two fingerprints which will be used to demonstrate the various methods of determining LRO. These will be referred to as fingerprints 5.1a and 5.1b.

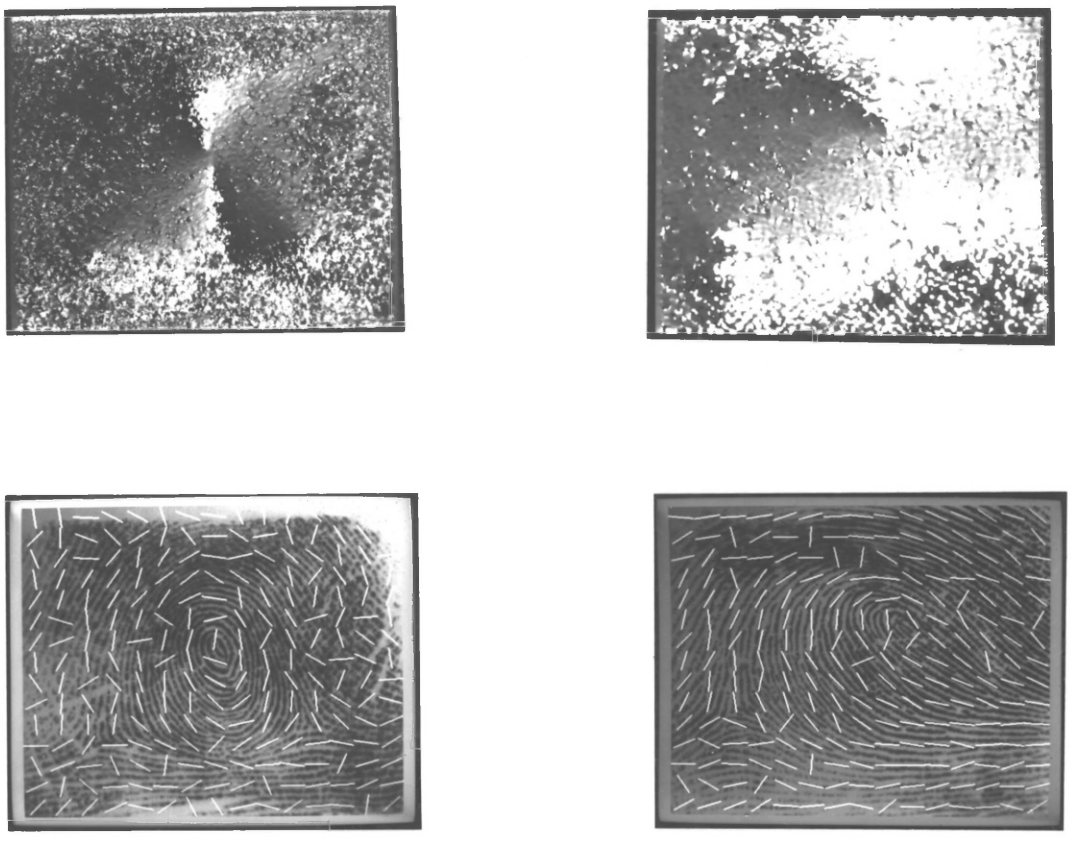


Figure 5.2. Orientations obtained using the approach of section 5.2.1. The images on the left correspond to fingerprint 5.1a and those on the right to fingerprint 5.1b. The top two show the orientations obtained when the algorithm is applied at every pixel position. Orientations from 0 to π are represented by greyscale values from black to white. The bottom two images show the orientations obtained at a grid of points.

filter did not give reliable results. In order for this technique to work well it is necessary to form an orientation estimate for each pixel in the image, and then smooth the results obtained. This approach was not used because it was found to take an excessive amount of computer time to evaluate the LRO at every pixel in the image. However, this technique has been incorporated successfully into the purpose built GOP image processing system [Granlund 1981] which has hardware to perform convolutions rapidly.

Because this technique is designed for detection of edge and line orientation in general images, it cannot be expected to perform as well as a technique which has been carefully optimised for the particular class of images consisting of fingerprints. It is therefore not surprising that other techniques were found to give better results.

5.2.2 Local Fourier Transforms

This technique determines a value for the ridge orientation from the magnitude Fourier transform of an appropriately-sized window of data around the point. The window must be at least as wide as twice the ridge spacing, but not so wide that too large a range of ridge orientations is present. Since the window (ideally) consists of a set of parallel straight lines, its magnitude Fourier transform should have a concentration of energy within a pair of radially-opposite peaks. This pair of peaks will correspond to the LRS and the LRO, and therefore the LRO can be obtained by determining the orientation of this pair of peaks.

An advantage of this approach is that it can in principle yield not only the LRO but also the other three parameters should they be required. Having located the position of peak energy in the Fourier domain, the LRO is given by the orientation of this peak, and a measure of the LRC is given by the angular width of the peak, measured in the direction of the θ coordinate. The LRS is inversely proportional to the distance of the peak from the origin and the LRE is given by the radial width of the peak, measured in the direction of the r coordinate.

It is important that a good windowing function be applied before the transform is taken, particularly if the parameters based upon peak width (LRC and LRE) are to be measured. For this application, the circularly symmetric 2-d windowing technique of section 3.1.1 was used, in order that every orientation of the data be treated equivalently. A Tukey window with a 30 percent taper was chosen as the underlying 1-d window.

A disadvantage of this method, which caused it to be rejected in favour of the method described in section 5.2.4, is the trade-off that occurs between peak definition, noise and orientation resolution. If the window is small, then the range of orientations included within it is small and so the peak is well-defined. However, the peak will be close to the origin since the ridge frequency will be relatively lower than for a large window, and this will decrease the resolution with which the orientation of the peak can be determined. The LRO reading obtained is unreliable because of the large effects that noise has within small windows. For a larger window, the effect of noise

is less disturbing, but the position of the peak is less clearly defined because of the larger range of directions included within the larger window.

These considerations are illustrated in figure 5.3 which shows the local Fourier transforms of the fingerprints in figure 5.1 for various window sizes. These transforms have been rotated through 90 degrees so that the principal pair of components is aligned along the ridge orientation.

5.2.3 Local variance of directionally filtered images

Suppose that n radially bandpass-filtered directional components of the fingerprint image have been formed. (These are the "prefiltered images" described in chapter 3).

Any one of the n components will be a smoothed version of the original print in those regions where the direction of filtering corresponds to the actual ridge orientation. In other regions, where the ridge orientation is significantly different from the filter orientation, the prefiltered image will consist almost entirely of the unwanted noise at the filter orientation, within the radial bandwidth of the filters.

If it is assumed that the signal in the correct direction is greater that the noise in any of the incorrect directions (within the radial bandwidth of the filters), then the LRO can be estimated by examination of the prefiltered images, as follows. Suppose the n prefiltered images are $f_1(x,y),\ldots,f_n(x,y)$ and the LRO at point $p=(x_o,y_o)$ is to be found. Then, in each image

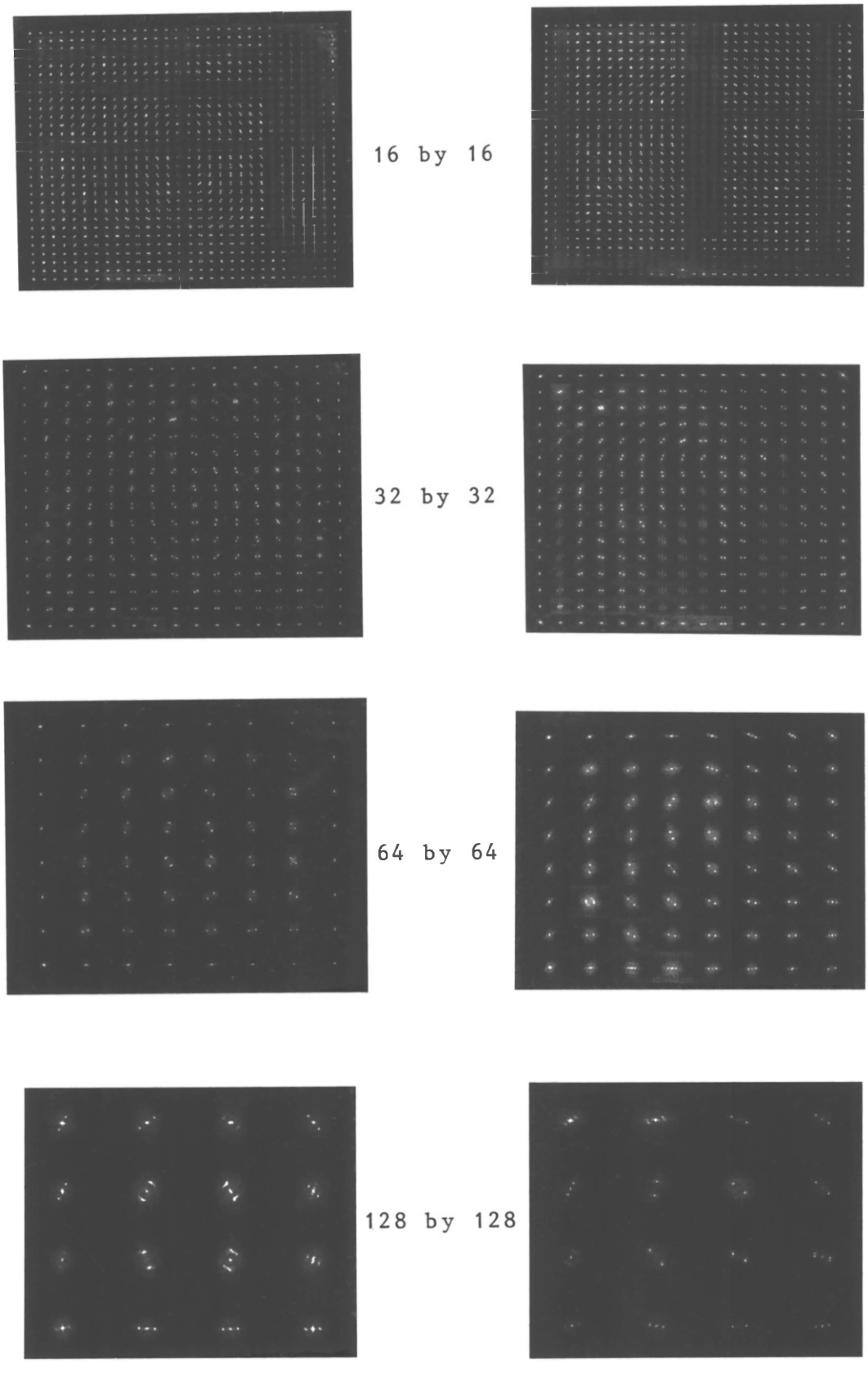


Figure 5.3. Local magnitude Fourier transforms of fingerprints 5.1a (on the left) and 5.1b (on the right). The transforms have been rotated through 90 degrees. The window size used is indicated near each pair of images.

$f_i(x,y)$, form a window W_i centred at (x_o,y_o). Again, the window size should be a small number of ridge widths. Then, the window corresponding to the correct LRO will be the one that contains the largest signal, and this may be determined by evaluating the variances within each of the windows and maximising.

Unfortunately, it is not always valid to assume that the signal is larger than the noise within a window of the size in question. If, for example, there is a pronounced break in the ridges perpendicular to the ridge orientation (caused perhaps by a cut in the skin), then the noise in the direction of the cut may dominate and lead to the wrong value of LRO being estimated.

The method was implemented for the case n=8. Sample results for various choices of window size are shown in figure 5.4. Note that the smaller the window, the more pronounced is the effect of the local noise. For a window size of 32 by 32 pixels, the results can be seen to be the most accurate for the examples shown. This was found to be the case for each of the five prints on which the method was tested.

5.2.4 Projections of local windows

This is the method that was chosen for use in the enhancement algorithm which will be presented in Chapter 6.

In this technique, a window of size 32 by 32 pixels and centred at the point at which the LRO is to be determined, is oriented at

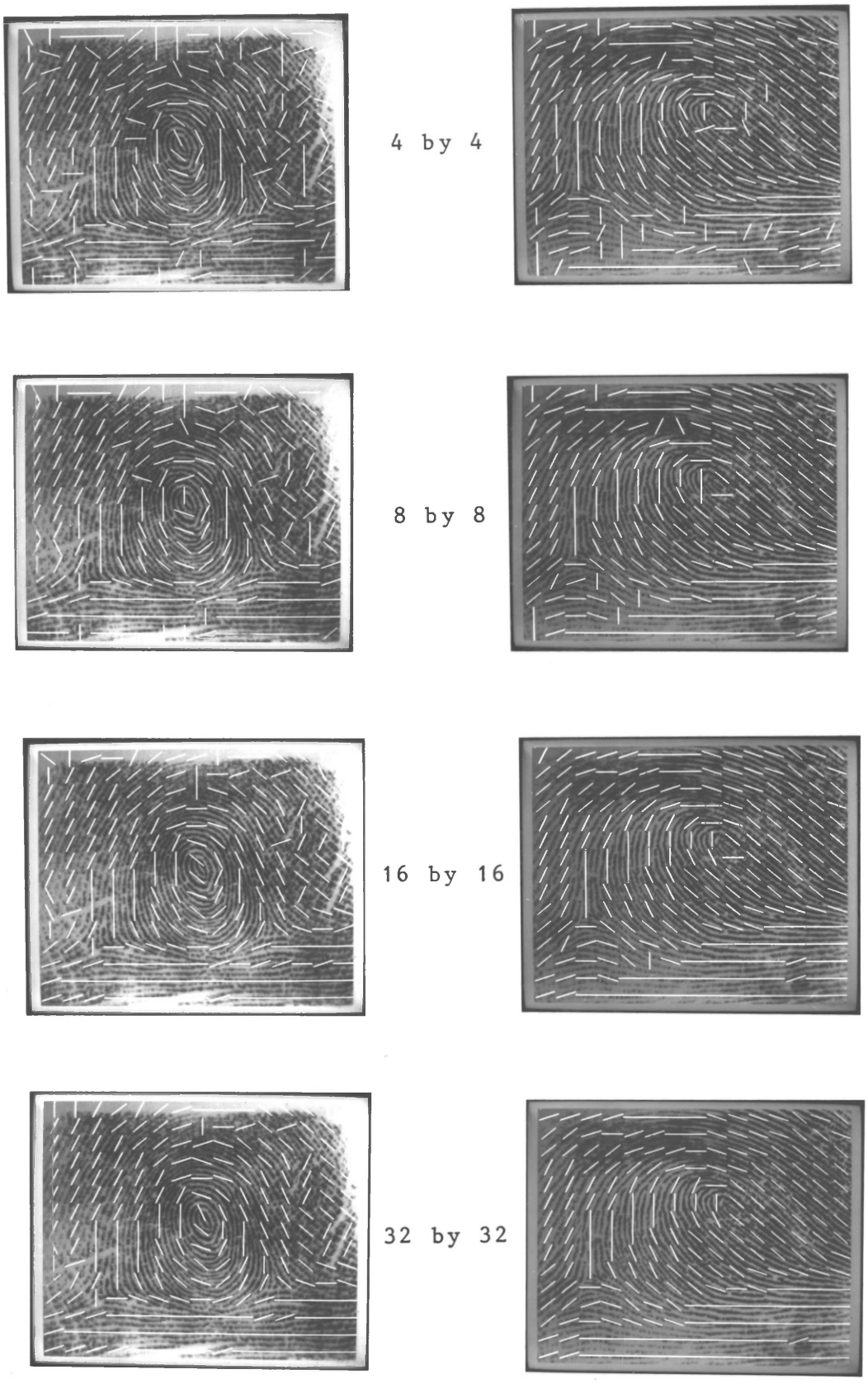


Figure 5.4. Orientations obtained using the approach of section 5.2.3 for fingerprints 5.1a (on the left) and 5.1b (on the right). The window sizes used are indicated near each pair of images.

each of eight different orientations $\theta_i = i\pi/8$ for i=0 to 7. This is illustrated in figure 5.5.

At each orientation, a projection along the y-axis of the window is taken:

$$p_i(x) = \frac{1}{32} \sum_{y=0}^{31} W_i(x,y) \qquad x=0..31$$

where $W_i(x,y)$ is the data within the window at angle θ_i. (Two dimensional linear interpolation is used to obtain a square grid of data $W_i(x,y)$ oriented at angle θ_i).

In this way a projection $p_i(x)$ in each of the eight directions θ_i is obtained.

If the x-axis of the window is aligned perpendicularly to the ridge orientation, one would expect maximum variation of the projection, since the ridges are crossed as x varies. On the other hand, alignment of the x-axis along the ridges should result in minimum variation of the projection, since no ridges will be crossed as x varies.

The idea is then to find a variation measure (possibly variance) which can be applied to the eight projections $p_i(x)$, and then determine the LRO by identifying which of the projections varies the most according to this measure.

It is advantageous, however, first to bandpass filter the $p_i(x)$ functions to remove noise. This can be done since a variation at a spatial frequency corresponding to the ridge spacing is sought,

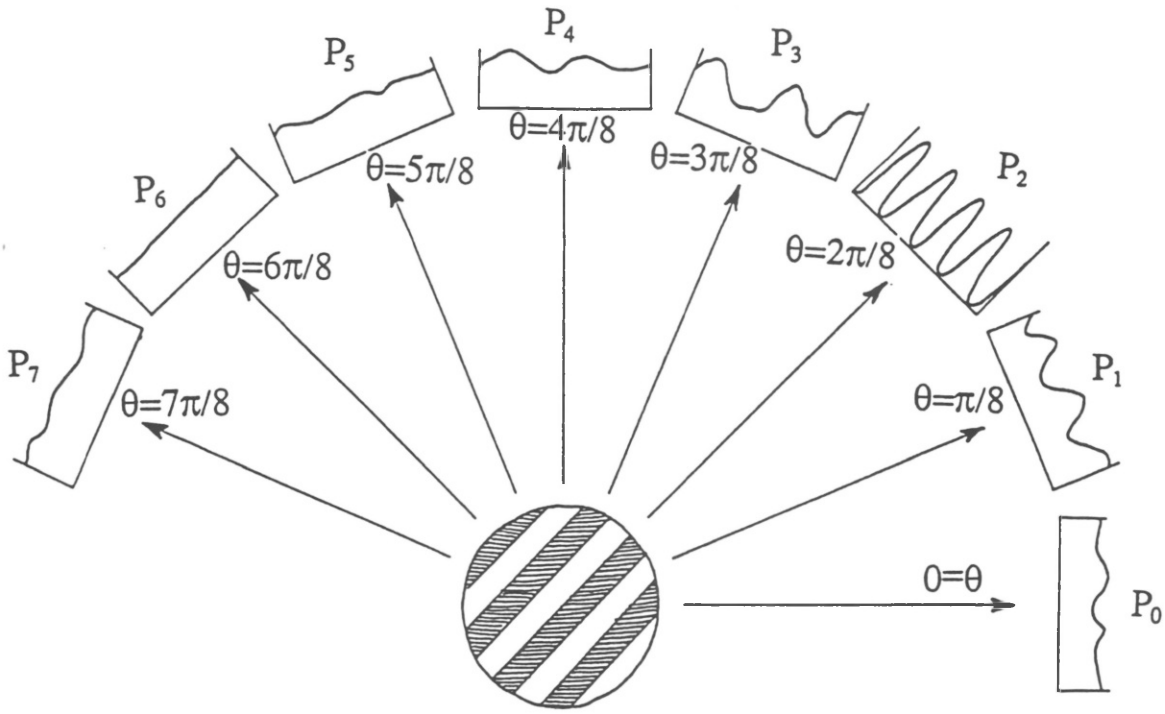


Figure 5.5. Eight projections of a window of fingerprint data. The projection which exhibits the greatest variation corresponds to the orientation of the ridges within the data window.

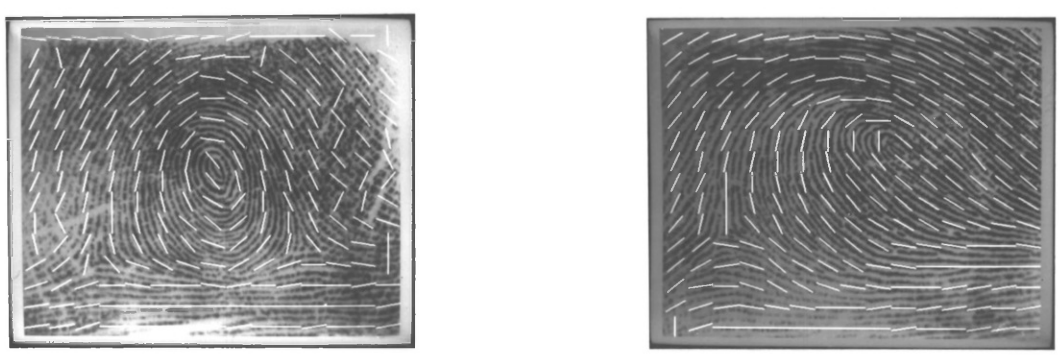


Figure 5.6. Orientations obtained using projections of local windows, for fingerprints 5.1a (on the left) and 5.1b (on the right). The window size is 32 by 32 pixels.

and components at other frequencies may be filtered out. A second-order Butterworth bandpass filter of wide enough bandwidth to allow for ridge spacing variations was chosen for this purpose.

Let the eight filtered projections be called $fp_i(x)$. One measure of how much $fp_i(x)$ varies would be its variance, given by

$$\text{Var}\ \{\ fp_i(x)\ \} = 1/32 \sum_{x=0}^{31} (fp_i(x) - \text{Mean}\ \{fp_i(x)\}\ )^2$$

where $\text{Mean}\ \{fp_i(x)\} = 1/32 \sum_{x=0}^{31} fp_i(x)$

Another measure, which gives an indication of the power of the derivative of $fp_i(x)$, is given by

$$V_i = \sum_{x=0}^{31} \left| fp_i(x+1) - fp_i(x) \right|$$

This latter measure can be expected to produce better results because it takes account of the sequence of occurrence of the samples $fp_i(x)$ as x is increased, whereas simple variance does not. When the two measures were compared empirically at 900 points on a good quality fingerprint, both produced correct values for i at all except two points. The measure V_i was chosen.

Obtaining the value of i for which V_i is the maximum of the eight values, the LRO is given by

$$\text{LRO} = i\pi/8.$$

This technique was implemented and compared against the method of

section 5.2.3 for a sample of five fingerprints. Visual inspection of these results led to the conclusion that this technique was marginally more accurate and tolerant of noise and poor quality data. Consequently this method was chosen in preference to that of section 5.2.3, even though it was found to run more slowly.

In all the prints that were processed using this method, a wrong estimate was produced only when there was noticeably poor print quality in the neighbourhood of the point. It is therefore felt that any substantial improvement over this method would have to rely upon a contextual analysis of neighbouring ridge orientations to allow neighbouring orientations to guide the algorithm in its choice. This aspect of the problem was not examined as it was felt to be beyond the scope of this research, but it is likely that the LRO model in chapter 4 would be useful to any such investigation.

5.2.5 Editing of orientation estimates

For a good quality rolled print, it is to be expected that very few of the orientations obtained will be incorrect. As an example, for the print of figure 1.5b there are only two false orientations.

In the case of poorer quality prints (such as that of figure 1.5a), one would expect a greater proportion of incorrect estimates. The effect of an isolated incorrect orientation estimate is to degrade the quality of the enhancement in the neighbourhood of the

incorrect estimate. The degradation is more severe for errors of larger magnitude.

In AFR systems, various methods have been used to reduce the effects caused by incorrect LRO estimates. In the algorithm used by the Home Office, for example, the matrix of LRO samples is spatially smoothed by replacing each estimate with the average of its eight neighbouring estimates. This spreads the error due to an incorrect estimate over itself and its nearest neighbours, but has the disadvantage that the estimates are often altered even when they are correct.

When processing scene of crime marks, however, the image is usually of such poor quality that errors in LRO estimate occur too frequently for smoothing to be of use. At present, there is no reliable automated technique for determining LRO for poor quality prints and scene of crime marks. Consequently, when the Home Office AFR system was adapted for use by New Scotland Yard, one of the modifications introduced was an "angle editor" which is used to correct interactively and manually the erroneous LRO estimates from scene of crime marks. Other AFR systems expect the operator to trace each ridge manually, or to locate the minutiae manually.

The use of smoothing and similar methods to reduce the perceived degradation due to incorrect LRO estimates was not investigated during this research, and instead the approach of using an angle editor was chosen, for the following reasons:

1. If smoothing were used, it would be difficult to provide a

reliable evaluation of the enhancement and the filters used, since it would not be easy to distinguish between effects caused by incorrect LRO estimation and those caused by the properties of the filters. The editor has the advantage that its output is known to be accurate.

2. It was decided that an investigation of different means of automatically correcting or smoothing inaccurate estimates would be too time-consuming and, in view of the discussion in section 5.1, beyond the scope of this research.

3. Without the use of an editor, it would not have been possible to apply the enhancement algorithm to scene of crime marks.

For all of the prints and marks enhanced in this thesis, the LRO estimates were edited using the angle editor. For good rolled prints, such as that of figure 5.1b, only a few (typically less than half a dozen) estimates need to be changed and therefore very little time or effort on the part of the operator is required to edit the data. On the other hand, for scene of crime marks, up to 30 percent of the estimates may require correction.

Use of an angle editor is unavoidable in the case of scene of crime marks, where no alternative is available. If the results presented in the next chapter are examined, it will be observed that in this way worthwhile enhancement of scene of crime marks can be achieved.

5.3 Interpolation of LRO values

5.3.1 Problems encountered with traditional interpolation methods

Interpolation between sampled LRO values poses problems which are not encountered in classical two-dimensional interpolation.

In the classical 2-d interpolation problem, one is given a regularly-spaced grid of values, which are assumed to be samples of some continuous real-valued function $f(x,y)$. $f(x,y)$ describes a surface passing through the given samples. The values of f at points not on the sampling grid are the interpolated values required.

The difference in the case of LRO samples, is that they are not samples of a continuous real-valued function; rather, they are samples of a continuous orientation-valued function $\theta(x,y)$, i.e. a direction field. The samples are thus orientations in the circle P^1 rather than real numbers (see chapter 4). Because $\theta(x,y)$ is a direction field, it is continuous and differentiable everywhere (except at the singular points) as an orientation valued function. However, when it is regarded as a real-valued function taking values within the real interval $[0,\pi)$, this is no longer the case. For, as the value of θ increases towards π, it suddenly and discontinuously drops back to zero before continuing to increase.

The periodic nature of orientations and of P^1, leads to ambiguities during the interpolation process. Suppose for instance that values must be interpolated along the line joining points p_1 and p_2, where $\theta(p_1) = 3\pi/4$ and $\theta(p_2) = \pi/4$ as shown in

figure 5.7. There is an infinite number of ways of getting from the point $\theta_1 = 3\pi/4$ to the point $\theta_2 = \pi/4$ on the circle P^1: by travelling around P^1 any number of times either clockwise or anticlockwise. Two of these options are illustrated in figure 5.7, and it is clear that the interpolated values obtained are different.

Note that knowledge of the rate of change of θ in the direction of interpolation can assist in resolving the ambiguity. The sign of the derivative will indicate whether to travel clockwise or anticlockwise around P^1, and its magnitude will indicate how many revolutions are required.

Section 5.3.2 describes how the LRO model of chapter 4 can provide the necessary information to resolve the ambiguity in orientation values. Section 5.3.3 then describes how classical interpolation algorithms can be adapted to use this information and successfully interpolate orientation functions.

5.3.2 Use of the LRO model to resolve ambiguities

5.3.2.1 Orientation unwrapping

It would be possible to use classical interpolation if the samples of the LRO function $\theta(x,y)$, which is continuous as a direction field (i.e. an orientation-valued function), could be converted to samples of a new continuous real-valued function $\theta_u(x,y)$. This would be done by adding to each sample a suitable multiple of π in order to **unwrap** the orientations and thus remove the discontinuities described earlier. The new "unwrapped" function

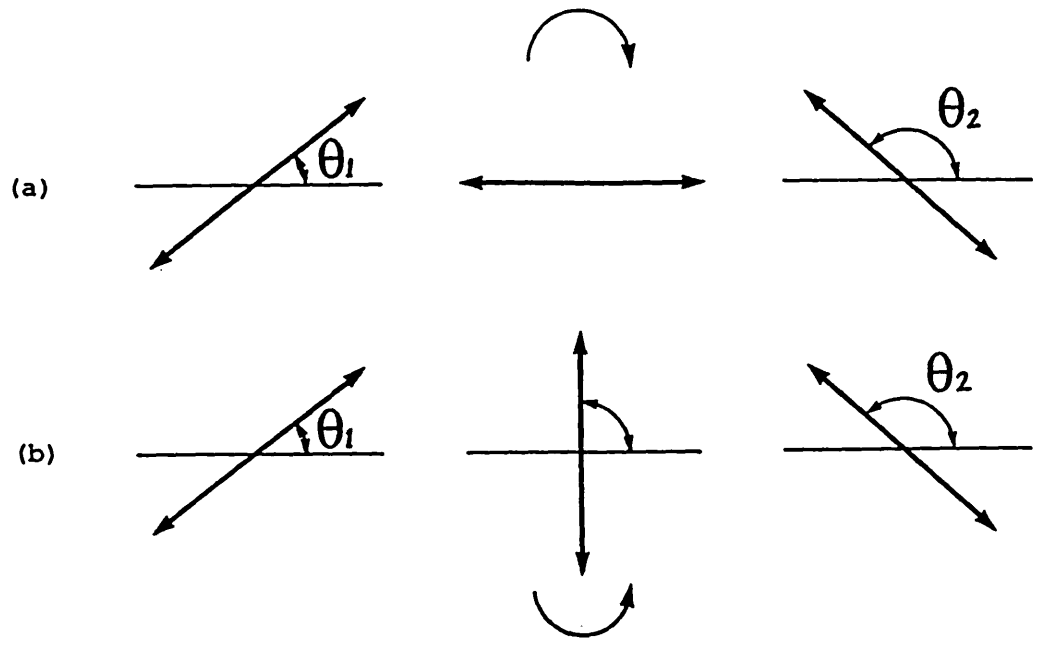


Figure 5.7. Ambiguity encountered during the interpolation of orientations. In linear interpolation between the orientations $\pi/4$ and $3\pi/4$, clockwise rotation (a) yields an orientation of 0 at the midpoint, whereas anticlockwise rotation (b) yields an orientation of $\pi/2$.

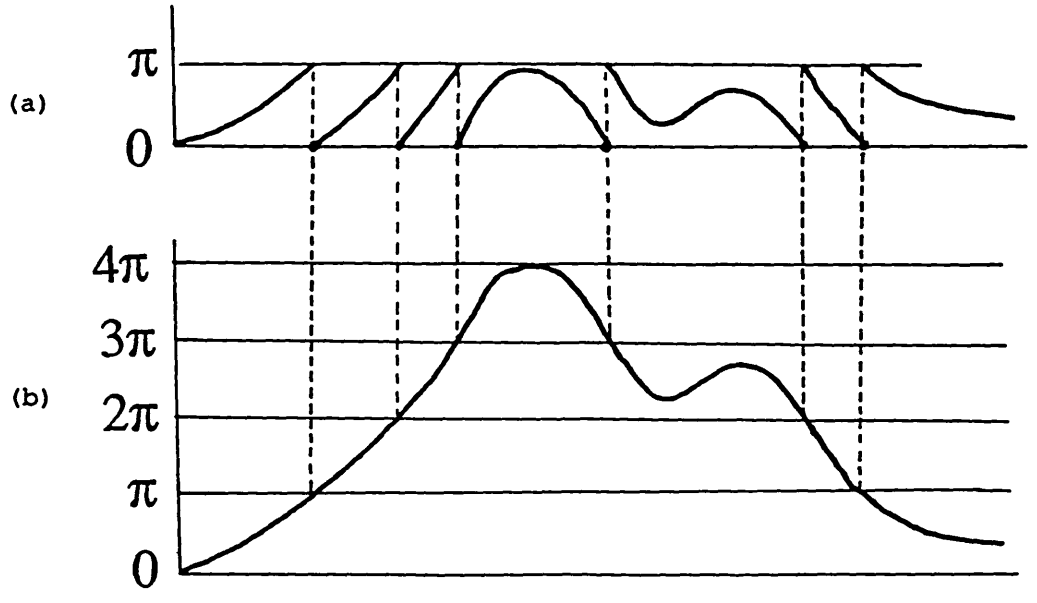


Figure 5.8. Unwrapping of a 1-dimensional orientation-valued function (a) to produce a continuous real-valued function (b).

$\theta_u(x,y)$ could then be interpolated using any classical technique.

Unfortunately, this approach is not of use as it stands because of the following result:

Theorem: A two-dimensional orientation valued function $\theta(x,y)$ cannot be unwrapped if it has a singular point.

Proof: Let θ have a singular point p. It will now be shown that any candidate "unwrapped" function θ_u will be discontinuous around any closed curve of non-zero index. To see this, suppose that a closed curve C surrounds the singular point p. Clearly, ix(p) is non-zero. Let (x,y) be any point on C. As C is traversed in an anticlockwise sense starting from (x,y), θ_u must change from its original value $\theta_u(x,y)$ through $2\pi.ix(p)$ to its final value $\theta_u(x,y) + 2\pi.ix(p)$. However, this final value is taken on at the same point (x,y), so that if θ is continuous then $\theta_u(x,y) = \theta_u(x,y) + 2\pi.ix(p)$, and therefore $ix(p)=0$. This contradicts the statement that p is singular. ■

The above result shows that 2-dimensional unwrapping of orientation functions is in general impossible. However, it is always possible to unwrap a continuous 1-dimensional orientation function $\theta(x)$, to produce a continuous real-valued unwrapped function $\theta_u(x)$. Unwrapping is performed by adding at each point x a suitable integral multiple k(x) of π chosen in such a way that the unwrapped function $\theta_u(x) = \theta(x) + k(x)\pi$ becomes continuous. Figure 5.8 gives an example of a wrapped function and its corresponding unwrapped version.

This process is closely related to the phase unwrapping problem [Oppenheim & Schafer 1975 section 10.4; Tribolet 1977] encountered in homomorphic signal processing. The phase $\theta(\omega)$ of the discrete Fourier transform of a sampled 1-d function $f(x)$ is treated as samples of a continuous function θ taking values in the set $S^1 = [0,2\pi)$ of directions. The phase unwrapping problem involves finding the continuous real-valued function θ_u where $\theta_u(\omega) = \theta(\omega) + k(\omega).2\pi$ is obtained by adding an integral multiple $k(\omega)$ of 2π at each point.

A solution by Tribolet [1977] to the phase unwrapping problem makes use of a knowledge of the inverse transform $f(x)$ in order to determine an estimate of the derivative $\theta'(\omega)$ of the phase, and hence the value of $k(\omega)$, at each ω. Phase unwrapping is therefore not directly applicable to the present orientation unwrapping problem, since there is no corresponding inverse transform data with which to work. However, the basic approach of using the derivative of the phase to guide the unwrapping process can be used to develop a technique suited to the problem of orientation unwrapping in fingerprint images. The LRO model described in chapter 4 is used instead of the inverse transform to give an estimate of the derivative of the orientation. This technique is described in section 5.3.2.2.

The inverse process to unwrapping is, naturally enough, termed "wrapping", and simply involves reducing all values modulo π so that they fall within $[0,\pi)$, i.e.

$$\theta_w(t) = \theta_u(t) \bmod \pi, \quad \text{for all } t.$$

As was shown above, it is not possible to unwrap a 2-d orientation function $\theta(x,y)$ if there are any singular points. Fortunately, however, the ability to unwrap a 1-d function is sufficient to enable interpolation between sampled values of a 2-d orientation function. For, certainly, given any curve $C(t)$ not passing through any of the singularities of $\theta(x,y)$, we can unwrap the 1-dimensional orientation function $\theta(C(t))$ along the curve to produce a continuous 1-d real-valued function $\theta_u(C(t))$ and then interpolate this along C using any classical interpolation algorithm. In particular, C could be any straight line parallel to either of the coordinate axes provided that it does not pass through a singularity. (If it did, it could be dealt with by assuming the line to be displaced slightly to the one side of the singularity).

5.3.2.2 **Application of the LRO model to orientation unwrapping**

This section describes the 1-d orientation unwrapping technique, based upon the LRO model, which is used in the interpolation algorithm described in the previous section.

In chapter 4 the following mathematical model was developed to describe local ridge orientation:

$$\theta(z) = \text{Arg}\{p(z)\} \bmod \pi,$$

where $$p(z) = \sqrt{\frac{(z-z_{c1})\ldots(z-z_{ck})}{(z-z_{d1})\ldots(z-z_{dm})}}.$$

An estimate of the derivative of the orientation can be obtained from the gradient of the LRO function given by the model:

$$\text{grad } \theta = \text{grad } (\arg(p(z)))$$

$$= \sum_{i=1}^{k} \text{grad } \theta_{ci} + \sum_{i=1}^{m} \text{grad } \theta_{di}$$

where

$$2\theta_{ci} = \arg(z-z_{ci}) = \arctan \frac{y-y_{ci}}{x-x_{ci}}$$

$$2\theta_{di} = \arg \frac{1}{z-z_{di}} = -\arctan \frac{y-y_{ci}}{x-x_{ci}}$$

so that

$$\text{grad } \theta_{ci} = \frac{1}{2r_{ci}^2} (y_{ci}-y , x-x_{ci})$$

$$\text{grad } \theta_{di} = \frac{1}{2r_{di}^2} (y-y_{di} , x_{di}-x)$$

where

$$r_{ci}^2 = (x-x_{ci})^2 + (y-y_{ci})^2$$

$$r_{di}^2 = (x-x_{di})^2 + (y-y_{di})^2$$

The interpolation algorithm interpolates in directions parallel to each of the two axes. When unwrapping, it is important that the orientation derivative be evaluated in the direction in which we are unwrapping, i.e. the x-component of grad θ should be used when interpolating parallel to the x-axis, and the y component when interpolating parallel to the y-axis.

A simplification was made by assuming that the distance between consecutive LRO values is small enough that the LRO never changes as much as π radians between samples. Under this assumption, one need only choose whether to go clockwise or anticlockwise around

P^1 in interpolating between samples, and in this case it is only necessary to know whether the LRO is increasing or decreasing in order to make this choice. Thus, knowledge only of the sign of the derivative is sufficient. This simplification was included in the unwrapping algorithm, which is illustrated by the pseudo-code below:

```
procedure unwrap (var x, xdot: array of real)
{x is the data to be unwrapped, and      }
{xdot is the array of derivative values }
const
  k = π/4
begin
  x[1] := x[1] mod π
  for i:=2 to length of x do
    Restrict x[i] to the range x[i-1]-π/2 .. x[i-1]+π/2 by
    adding or subtracting a suitable multiple of π.
    diff := x[i] - x[i-1]
    if (diff > k) and (xdot[i] < 0) then
      x[i] := x[i]-π
    elseif (diff < -k) and (xdot[i] >0) then
      x[i] := x[i]+π
    endif
  endfor
end
```

The idea is that x[i] is brought as close as possible to x[i-1] by addition of an integer multiple of π, subject to the following constraint. For large angle changes (i.e. changes larger than k) between adjacent samples, x[i] - x[i-1] must have the same sign as xdot[i]. In practice, a value for k of π/4 was found to work well.

In order to make use of the LRO model, however, it is necessary to know the positions of the cores and deltas. Locating singular points using a general purpose computer is one of the more difficult problems in fingerprint image processing, and no techniques are known which are both reliable and efficient.

On certain non-conventional computer hardware systems, locating cores and deltas is considerably easier. The GOP mentioned earlier could be used to form an image consisting of orientation estimates. These could then be smoothed and a gradient operation applied, to yield an image in which the cores and deltas are pinpointed. In chapter 8, the use of a system called WISARD to locate singular points will be investigated.

Since an algorithm such as the GOP one would be too slow if implemented on a general-purpose computer, the problem was side-stepped by providing an interactive utility allowing a human operator to specify the locations of the cores and deltas in an image. Although this solution is not ideal, it is the usual one adopted in systems which require singular point positions, and only requires a few seconds of the operator's time.

5.3.3 The orientation interpolation algorithm

The considerations in section 5.3.2 lead to the following algorithm for the interpolation, by a factor K, of an n by n array of orientation values:

```
{We have n rows and n columns of data}
for i:=1 to n do
  unwrap row i
  interpolate row i by factor K
endfor

{We now have n rows and n*K columns}
for i:=1 to n*K do
  unwrap column i
  interpolate column i by factor K
  wrap column i
endfor

{We now have n*K rows and n*K columns}
```

5.3.4 Choice of classical interpolation algorithm

Provided that the unwrapping technique does not fail, the behaviour of the orientation interpolation algorithm will be identical to that of the classical interpolation algorithm used. For this application, it is not of great importance exactly which interpolation algorithm is chosen, since interpolation was performed in order to obtain a continuous function rather than to provide accurate intermediate values.

Continuity will be ensured by any reasonable interpolation technique, and it was therefore decided to use linear interpolation because of its simplicity. Linear interpolation at a point x requires only the two samples nearest to the point x, and therefore it is not necessary to unwrap an entire line of data at once; only the two samples used need be unwrapped. This would prove an advantage over other interpolation methods in the event of failure of the unwrapping algorithm, since an unwrapping failure will then only affect the interpolated values between two samples. In practice, however, the unwrapping algorithm never did fail.

In other applications, it may be desirable to use a more sophisticated interpolation technique, and should this be necessary, one of the techniques described in Appendix B could be used.

CHAPTER 6

A DIRECTIONAL APPROACH TO FINGERPRINT ENHANCEMENT

In chapters 2 and 3 techniques of position-dependent anisotropic filtering based upon local image parameters were developed. In chapters 4 and 5 the directional properties of images were discussed, and means of obtaining estimates of the ridge orientation and curvature parameters required by the filters were developed.

The purpose of the present chapter is to bring all these results and techniques together to present an algorithm which produces high quality enhancements of fingerprint images. The enhancement algorithm is divided into three stages: the filtering stage, the thresholding stage and the (optional) thinning stage. These three stages are described in sections 6.1, 6.2, and 6.3 respectively. Results obtained using the enhancement algorithm are presented in section 6.4 and evaluated in section 6.5.

6.1 Directional filtering.

The input to this first stage consists of:

(a) A 512 by 512 pixel raw fingerprint image.

(b) A 30 by 30 array of values of local ridge orientation for this image.

(c) The positions of the cores and deltas.

As discussed in chapter 3, the number of prefiltered images to be formed is either eight or sixteen, depending upon whether one or two sets of directional components are to be used. The algorithm will be presented assuming the use of sixteen prefiltered images; the equivalent algorithm using eight is similar.

The filtering algorithm is the following:

1. Obtain the prefiltered images PF(i) from the raw fingerprint PRINT, as described by the following pseudocode:

```
WINDOW(PRINT)                  { Window the image, to reduce  }
                               { edge effects.                }

PRINTFOURIER := FFT(PRINT)     { Take the 2-d transform of the}
                               { windowed image.              }

for i:=0 to 15 do
  TEMP := PRINTFOURIER*F[i]    { Multiply this by the i'th    }
                               { filter function,             }
  PF[i] := IFFT(TEMP)          { and inverse transform this to}
                               { get the i'th prefiltered     }
                               { image PF[i].                 }
endfor
```

2. Obtain the LRO at each point by interpolation between the 30 by 30 sampled values, using the LRO model to guide the interpolation as described in chapter 4. At points near the edge of the image, extrapolate linearly beyond the 30 by 30

samples.

3. Form the output image OUT(x,y) by appropriate combination of pixel values from the prefiltered images, as follows:

```
for x:=0 to 511 do
  for y:=0 to 511 do

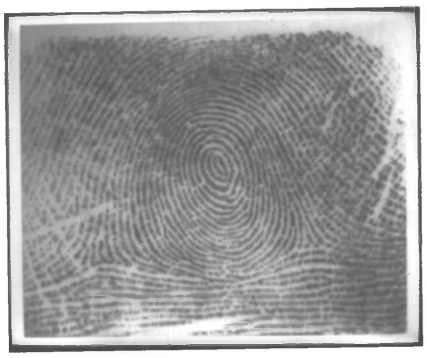
    Determine the distance r from (x,y) to the nearest
    singular point.  Using the function in figure 3.4,
    determine the angular bandwidth that is required, and
    hence the number j of prefiltered images that must be
    combined.

    Form the output pixel OUT(x,y) as a weighted average of
    the (x,y) pixels of the j prefiltered images nearest in
    orientation to the LRO (as determined in (2) above).

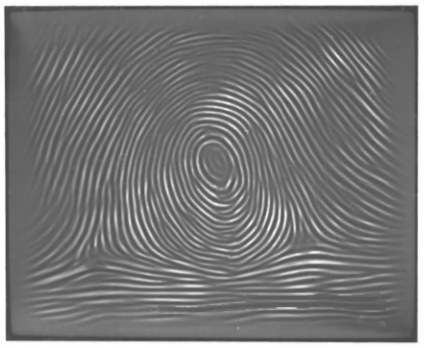
  endfor
endfor
```

The result of application of this algorithm is a filtered image which has been smoothed in the orientation of the ridges. Additionally, only wavelengths within the permissible range of ridge spacings have been passed.

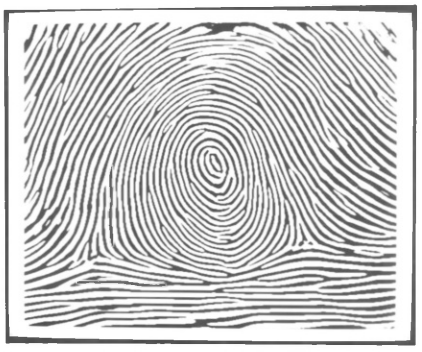
Figure 6.1a shows a fingerprint image, and figure 6.1b shows the filtered image produced by application of the filtering algorithm. The prefiltered images at 16 orientations are shown in figure 6.2.



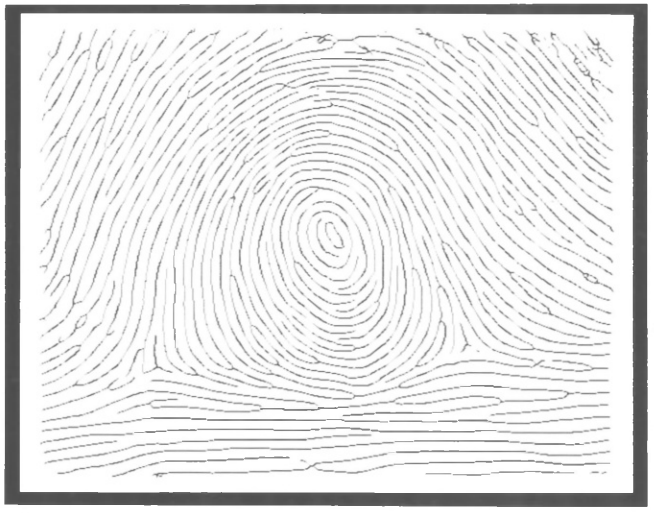
(a) raw fingerprint



(b) after filtering



(c) after thresholding



(d) after thinning

Figure 6.1. Stages of the fingerprint enhancement process

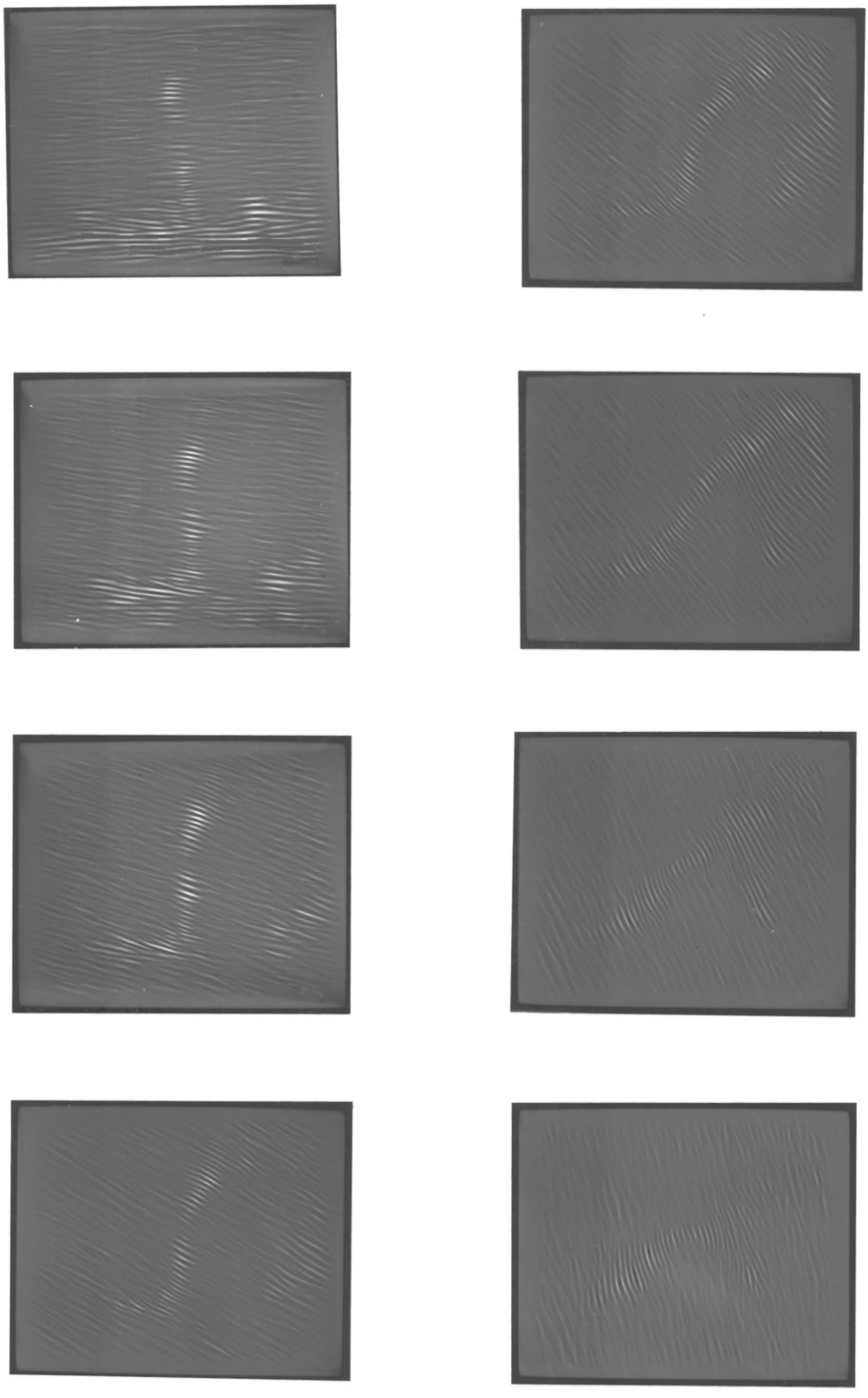


Figure 6.2. 16 directionally filtered versions of the print of figure 6.1a.

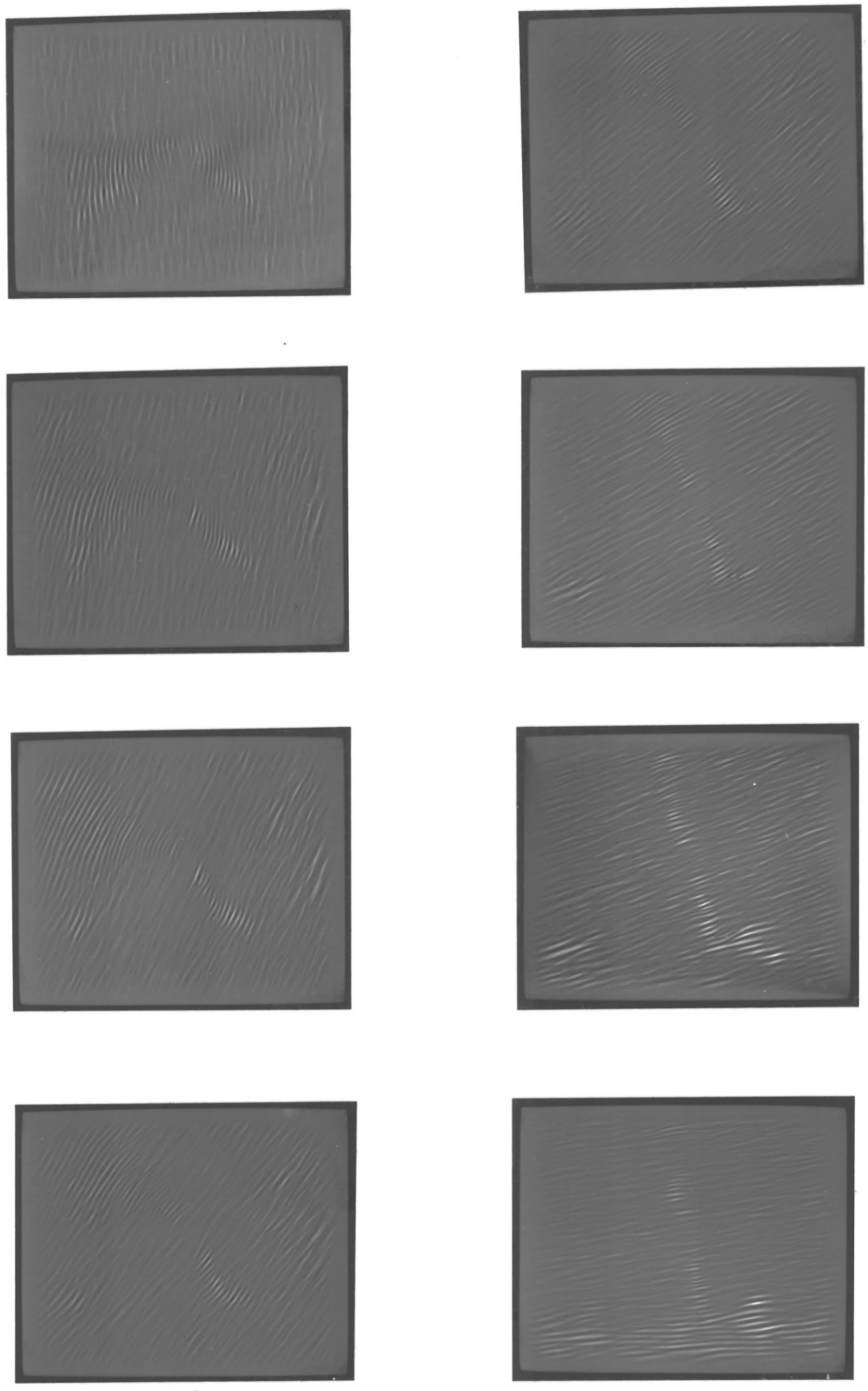


Figure 6.2 (continued)

6.2 Thresholding

After the filtering stage, most of the noise and unwanted information has been removed, but the image contains more information than is required because it is still a greyscale image. The desired enhancement is a binary image, with black representing points on ridges and white representing points not on ridges. To produce such a binary image, a thresholding operation is applied.

Thresholding is a special case of an approach known as image segmentation [Rosenfeld & Kak 1976]. In segmentation problems it is assumed that the picture represents some object against its background, and it is desired to separate the object from the background by converting the image to a binary image with one state representing pixels on the object and the other state representing pixels in the background. Very many segmentation techniques have been proposed. It is often the case that a priori knowledge about the object such as its size, shape or texture may be taken into account by a segmentation technique. However, in a class of segmentation techniques known as thresholding techniques, no a priori assumptions are made about the object, and the segmentation is performed purely on the values of the image pixels.

The simplest type of thresholding is global thresholding. In global thresholding, a greyscale value g_T (known as the threshold) is chosen and the image is thresholded against this value. That is, for each point (x,y), if $f(x,y) > g_T$ then the thresholded image is made white at (x,y), otherwise it is made

black.

Many techniques have been proposed to determine the global threshold g_T; examples are those of Weszka & Rosenfeld [1978], Kirk & Rosenfeld [1979] and Tsai [1985]. Global thresholding techniques are reviewed by Weszka [1978] and O'Connor [1988]. However, global thresholding is not usually suitable for any but the simplest images. If, for example, the lighting on the image is such that the left hand side were considerably more brightly lit than the right hand side, then the thresholded image can be expected to be mostly white on the left and mostly black on the right. For many such images, it is not possible to choose a value of g_T such that the detail is clear in all parts of the image.

An alternative to the global thresholding approach is variable thresholding [Nakagawa & Rosenfeld 1979], in which g_T is allowed to vary over the image, so that there is a different threshold value $g_T(x,y)$ for each pixel position (x,y) in the image. Then the thresholded image is white if $f(x,y) > g_T(x,y)$, and black otherwise. $g_T(x,y)$ defines a threshold surface against which the image is thresholded. Global thresholding is the special case where the threshold function is constant.

It is important to ensure that the threshold surface $g_T(x,y)$ is continuous. Any discontinuities in the surface could give rise to undesired artefacts in the resultant thresholded image. Also, it is generally desirable that the threshold surface exhibit a high order of continuity.

Thresholding of an image then becomes the problem of determining

the threshold surface $g_T(x,y)$. There is no single method for doing this because the criteria for distinguishing between background and foreground vary greatly between different applications.

The highly structured nature of fingerprints proves to be of assistance in developing a technique for thresholding the filtered fingerprint images. To threshold the image at a given point, all one needs to know is whether or not this point lies on a ridge. Considering that the image has already been smoothed by filtering, it should be clear that no more information is required for this than is contained within a window around the point, two or more ridge spacings in width. Therefore, the threshold values $g_T(x,y)$ may be calculated using the data from appropriately-sized windows centred at (x,y).

A window size of 32 by 32 pixels was chosen as always containing at least two ridges. Two methods of determining thresholds within the moving windows were investigated:

1. Local median value thresholding: The threshold $g_T(x,y)$ is taken as the median of all the pixel values within a window surrounding the point (x,y).

2. Local average value thresholding: The threshold $g_T(x,y)$ is taken as the mean of all the pixel values within a window surrounding the point (x,y), i.e.

$$g_T(x,y) = \frac{1}{32^2} \sum_{i=0}^{31} \sum_{j=0}^{31} f(x-16+i,\ y-16+j)$$

These techniques will automatically produce a smooth threshold function, since windows around adjacent points contain the same pixels except for those in one column or row. Thus the mean or median calculated can be expected not to change rapidly from point to point.

Both methods were implemented and tested on the filtered versions of three rolled fingerprint images. Both produced satisfactory results. No visually discernible difference in quality could be detected between the results obtained from the two methods. Local average thresholding was chosen because its implementation was simpler.

Figure 6.1c shows the thresholded version of the filtered image in figure 6.1b. Further examples of enhanced fingerprints may be found in section 6.4.

6.3 Thinning

The binary image produced as the output of the thresholding stage is the enhanced version of the fingerprint. This image is close to the one which is "imagined" by a trained fingerprint officer when examining a raw print.

However, a further stage of processing may be applied. In this stage the image is thinned so that the ridges are transformed into curves of single-pixel width. A thinned enhanced image is usually easier to encode by computer than one that has not been thinned. For example, a syntactical analysis of ridge structure would be simplified because of the relative ease of tracing lines of single-pixel rather than multiple-pixel width.

An early paper of Rosenfeld [1970] defined concepts of 4- and 8-connectedness for binary digital images and used these definitions to obtain results which can be used to prove the correctness of shrinking algorithms (of which thinning algorithms are a variant). Another paper by Rosenfeld [1978] described thinning algorithms in their more general context as examples of iterative or relaxation methods.

Stefanelli and Rosenfeld [1971], combining ideas from several earlier algorithms, developed two algorithms for obtaining thinned versions of objects from a binary image. The one method yields a 4-connected skeleton and the other an 8-connected skeleton. These methods involve the successive deletion of pixels from the boundary of an object, based on simple tests upon the pixel values within a 3 by 3 neighbourhood of each pixel. The

algorithms are parallel in the sense that during each iteration of the process, all pixels of the image can be processed simultaneously.

Rosenfeld [1975] derived necessary and sufficient conditions for a 3 by 3 operation to preserve connectedness and to have no effect upon arcs and curves. Rosenfeld and Davis [1976] developed a thinning algorithm similar to those of Stefanelli & Rosenfeld but satisfying these conditions.

Deutsch [1972a,1972b] developed a widely used parallel isotropic thinning algorithm similar to the above algorithms. Versions of this algorithm suited to processing images defined on hexagonal and triangular arrays were also produced. A variant of Deutsch's algorithm was proposed by Zhang and Suen [1984]. Lu and Wang [1985, 1986] suggested improvements to this algorithm. All of these algorithms are simple, fast and parallel.

Arcelli et al [1981] and Arcelli and di Baja [1985] produced thinning algorithms whereby the pixels of the object are labelled according to their distance from the background, after which an algorithm is applied to identify the skeletal points. Preprocessing of the image is necessary to ensure that objects have sufficiently smooth boundaries for the technique to work. An advantage of this algorithm for some applications is that an inverse transformation can be applied to recover the original image almost perfectly. Suzuki and Abe [1986] proposed a similar but simpler technique, which includes a postprocessing stage to ensure that two-pixel width lines are reduced to single-pixel width.

Pavlidis [1980] proposed an algorithm whose time complexity is proportional to image size rather than object width. This method is based upon peeling off successive contours of the object to be thinned.

Xia [1986] defined the "interior neighbour" IN of a point P on the edge of an object and showed that the point P can be removed if the point IN is an interior point of the object. Since only the points IN need be tested at each iteration, this algorithm is fast; however, the resultant skeleton is not guaranteed to be of single-pixel width everywhere, and therefore postprocessing would be required.

Any technique suitable for fingerprints has to satisfy the following three requirements:

1. The connectivity requirement. The thinned version of any connected object must be connected. This would ensure that the thinning process does not cause any breaks in ridges.

2. The technique should retain the endpoints of lines once they have reached single-pixel width. This is to ensure that minutiae corresponding to ridge endings do not change position during the thinning process.

3. The technique should thin uniformly inwards from the boundaries of the object. Thus the thinned version of a thick line should be a thin line situated along the centre of the original line. If this requirement were not met, and thinning were done from the one side of the object only, distortion could result.

An advantage of iterative techniques satisfying (1) and (2) above is that the termination condition is simple. Iterations may be terminated as soon as there is no further change in the image. If (2) were not satisfied, the ends of a thinned line might be "thinned" inwards by one pixel at each iteration, resulting in the line's gradual disappearance and the consequent need to find a more elaborate termination condition for the algorithm.

Ogawa and Taniguchi [1982] proposed a technique for directionally thinning handwritten Chinese characters based upon a knowledge of the orientation of each of the strokes comprising the character. One approach that was considered for the present application was to adapt this to yield a technique of "directional thinning" for fingerprints. This approach would make use of LRO information already available from the filtering stage to guide the thinning algorithm so that thinning is done only in directions perpendicular to the local ridge orientation. There is scope for further research into this approach, but in practice it was found that for a print which has been enhanced using a good technique such as the one presented above, excellent results can be obtained using standard thinning algorithms. Therefore, further pursuit of this approach was considered unnecessary.

For fingerprint images, the simple methods based upon 3 by 3 pixel neighbourhoods can be expected to run most quickly because the ridges are only a few pixels wide, and consequently only a few iterations are required. The method which was chosen was that of Zhang & Suen, and Lu & Wang. It satisfies all three of the above

requirements, is simple and fast and can be easily and efficiently implemented. A description of the method follows.

Suppose that the pixels of the objects to be thinned (i.e. the fingerprint ridges) have value 1 and pixels of the background have value 0.

At each iteration, this method removes a layer of pixels from the boundary of each object, leaving behind any that lie on the skeleton of the object. The new value given to a point depends upon its own value and the values of its eight nearest neighbours after the previous iteration.

Each iteration consists of two sub-iterations. The first sub-iteration removes points from the lower right-hand (south east) boundary of each object, and the second sub-iteration removes points from the upper left-hand (north west) boundary.

Suppose that the point under consideration is called P1 and its eight neighbours P2,P3,...,P9 as illustrated in figure 6.3.

Let A(P1) be the number of "01" patterns in the ordered set {P2,P3,...,P9,P2}. Let B(P1) be the number of neighbours of P1 that belong to the object, i.e. B(P1) = P2 + P3 + P4 + P5 + P6 + P7 + P8 + P9.

Then the first sub-iteration is expressed as

> For each point P1 in the object, remove it from the object if it satisfies the following conditions:
>
> (a) $2 <= B(P_1) <= 6$

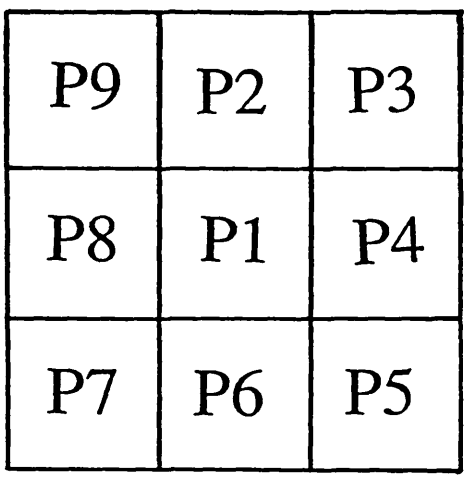


Figure 6.3. An image pixel P1 and its eight nearest neighbours P2, P3, ... P9.

(b) $A(P_1) = 1$

(c) $P_2 * P_4 * P_6 = 0$

(d) $P_4 * P_6 * P_8 = 0$

and the second sub-iteration is

For each point P1 in the object, remove it from the object if it satisfies the following conditions:

(a) $2 <= B(P_1) <= 6$

(b) $A(P_1) = 1$

(c) $P_2 * P_4 * P_8 = 0$

(d) $P_2 * P_6 * P_8 = 0$

Note that it is only necessary to consider points lying on the object boundaries as candidates for removal.

Figure 6.1d shows the thinned image that was obtained by applying the above algorithm to the enhanced image of figure 6.1b.

6.4 Results

In this section the enhancements of several rolled fingerprints and scene-of-crime marks are presented.

6.4.1 Enhancements of rolled fingerprint images

The enhancement algorithm was applied to 15 rolled fingerprint images representing various Henry classes, which were provided by the Home Office. The results obtained are presented in figure 6.4. An evaluation of these results follows in section 6.5.

6.4.2 Enhancements of scene-of-crime marks

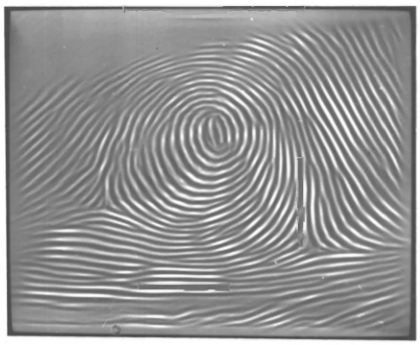
Scene-of-crime marks are generally of much poorer quality than rolled prints, and in addition are often incomplete. Figure 6.5 illustrates the effect of the enhancement algorithm applied to eight different scene-of-crime marks provided by the Home Office. A discussion of these results follows in section 6.5.



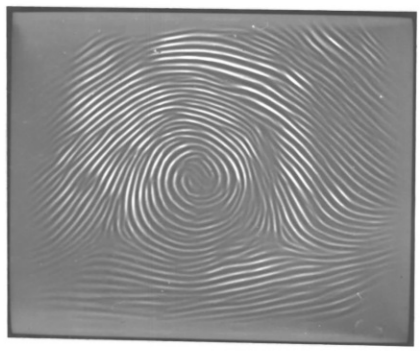
raw fingerprint



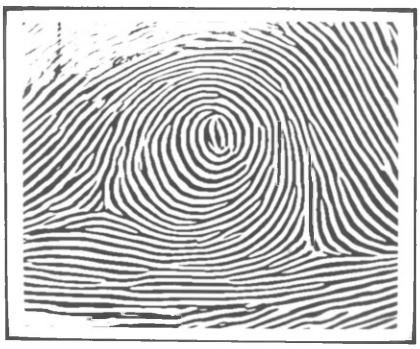
raw fingerprint



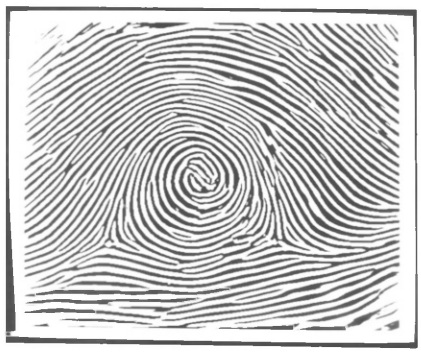
filtered image



filtered image



enhanced fingerprint



enhanced fingerprint

Figure 6.4. Enhancements of rolled fingerprint images of various Henry types. On each page, enhancements are shown for two fingerprints.



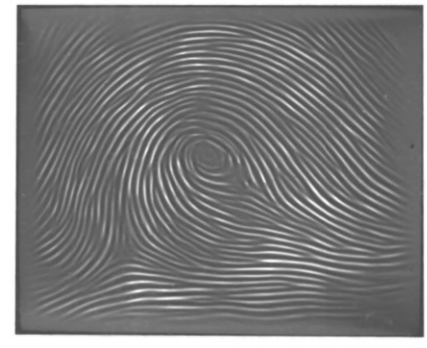
raw fingerprint



raw fingerprint



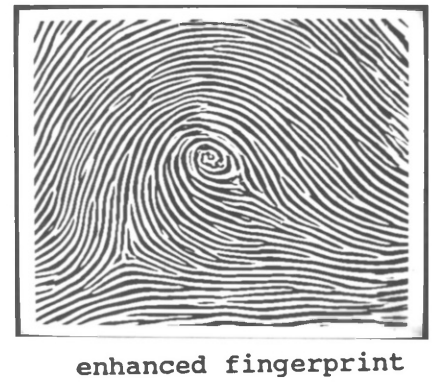
filtered image



filtered image

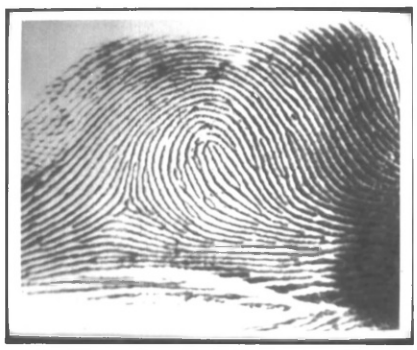


enhanced fingerprint



enhanced fingerprint

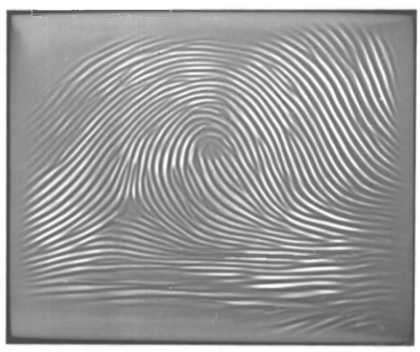
Figure 6.4. (continued)



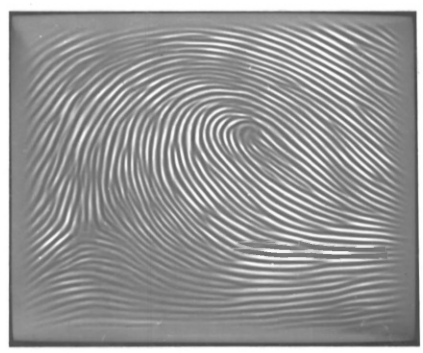
raw fingerprint



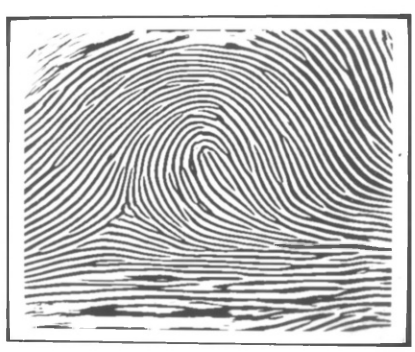
raw fingerprint



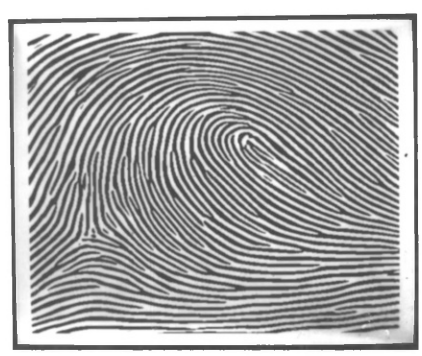
filtered image



filtered image

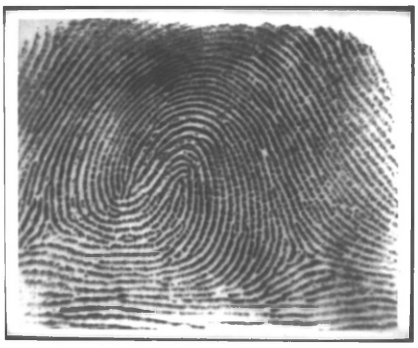


enhanced fingerprint

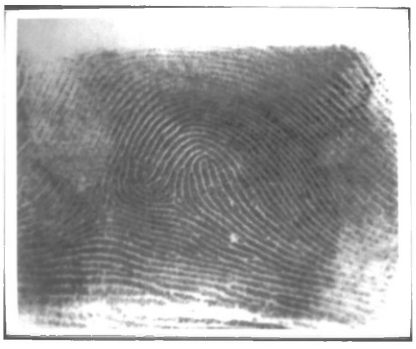


enhanced fingerprint

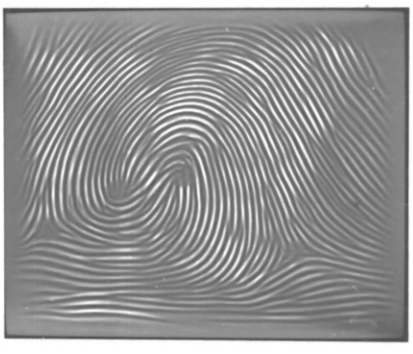
Figure 6.4. (continued)



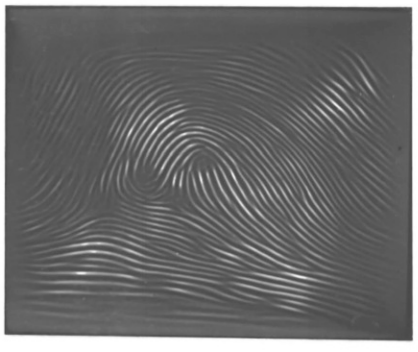
raw fingerprint



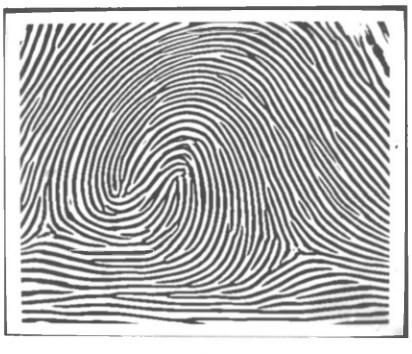
raw fingerprint



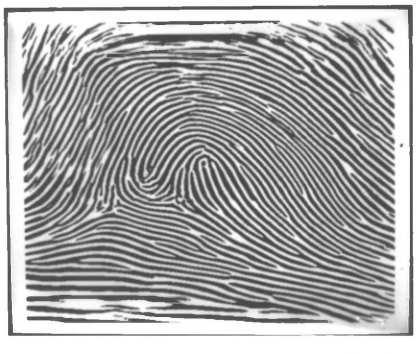
filtered image



filtered image



enhanced fingerprint



enhanced fingerprint

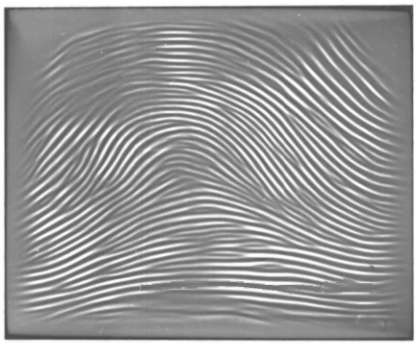
Figure 6.4. (continued)



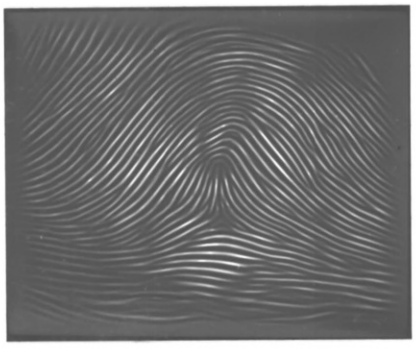
raw fingerprint



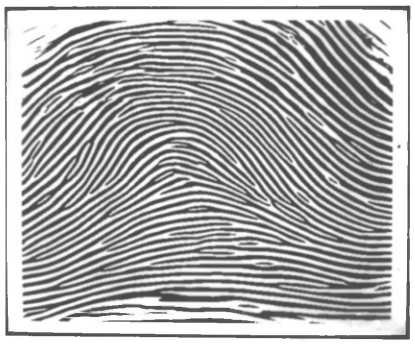
raw fingerprint



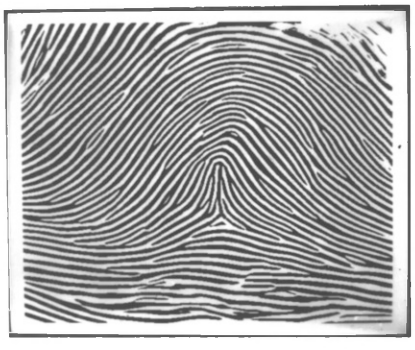
filtered image



filtered image

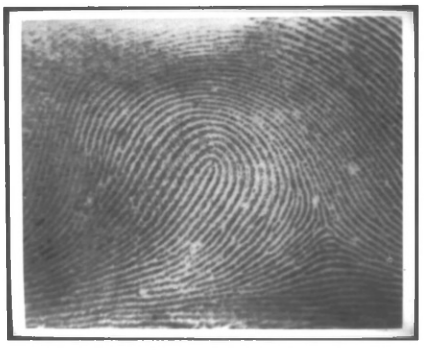


enhanced fingerprint



enhanced fingerprint

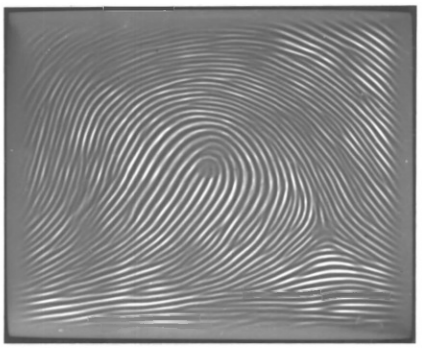
Figure 6.4. (continued)



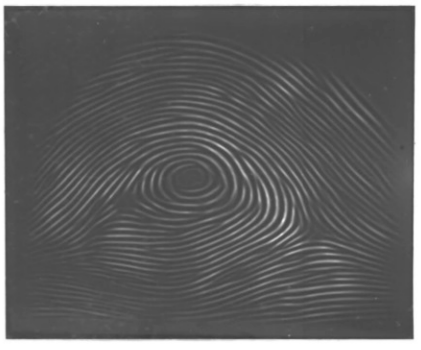
raw fingerprint



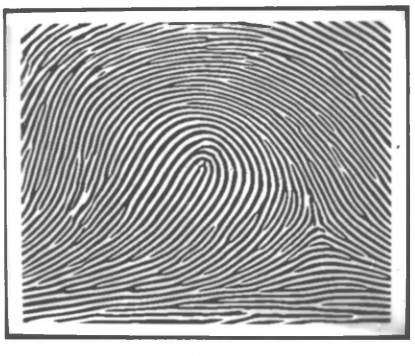
raw fingerprint



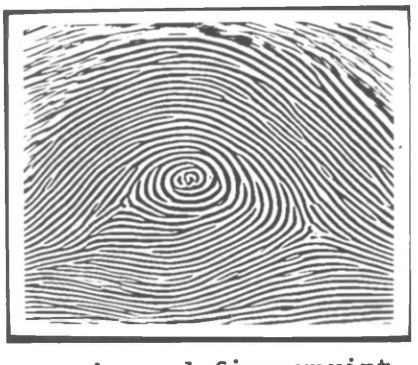
filtered image



filtered image



enhanced fingerprint



enhanced fingerprint

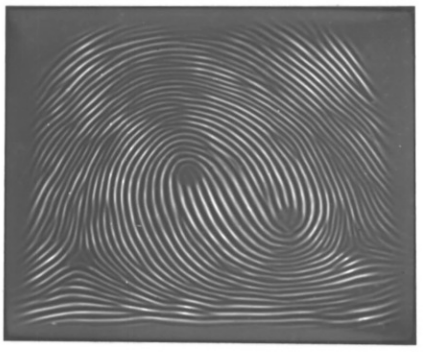
Figure 6.4. (continued)



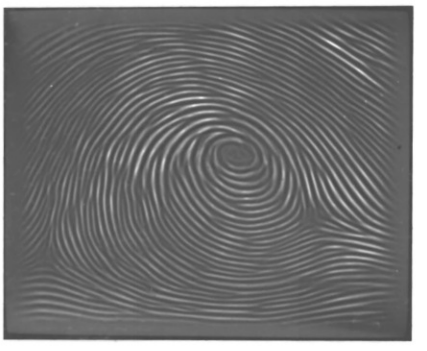
raw fingerprint



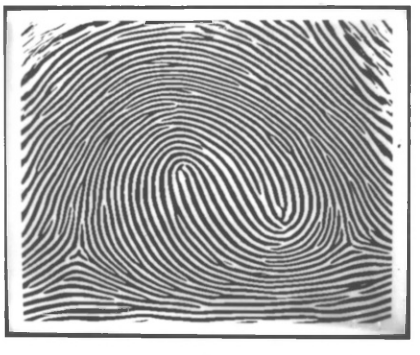
raw fingerprint



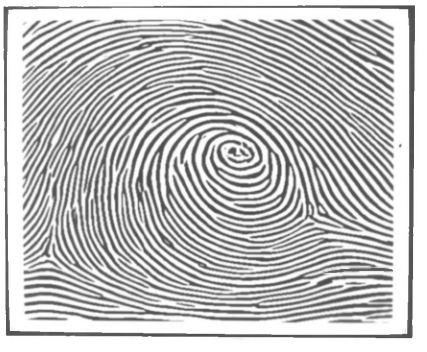
filtered image



filtered image

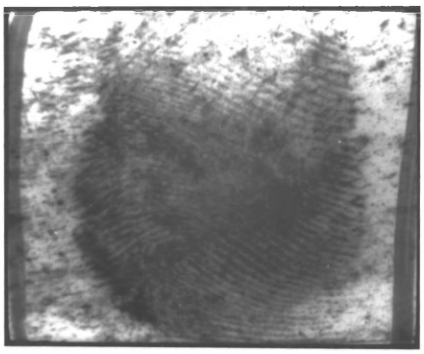


enhanced fingerprint



enhanced fingerprint

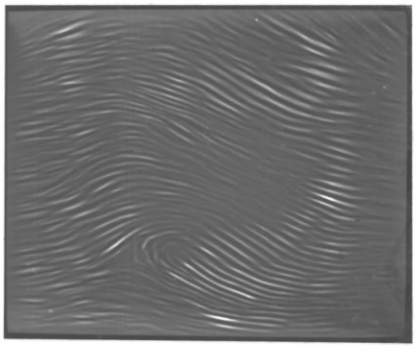
Figure 6.4. (continued)



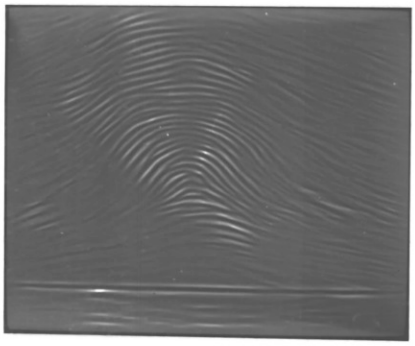
raw latent print



raw latent print



filtered image



filtered image

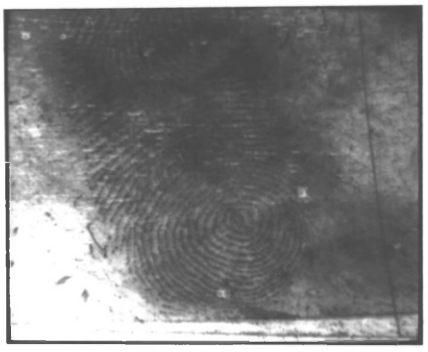


enhanced latent print

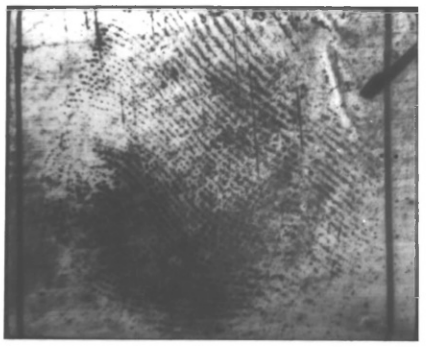


enhanced latent print

Figure 6.5. Enhancements of latent fingerprint images. On each page, enhancements are shown for two latent prints.



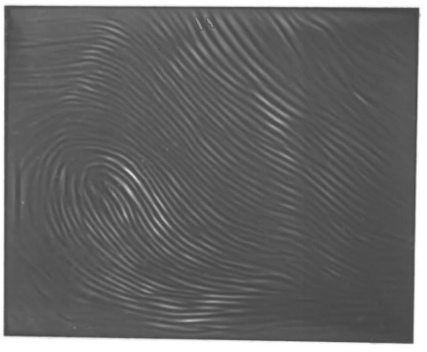
raw latent print



raw latent print



filtered image



filtered image



enhanced latent print



enhanced latent print

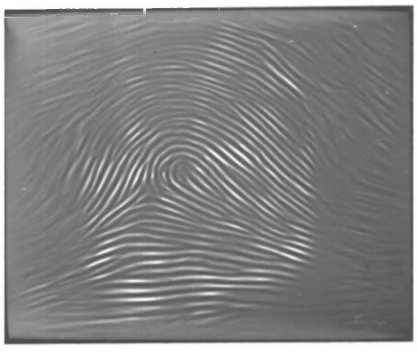
Figure 6.5. (continued)



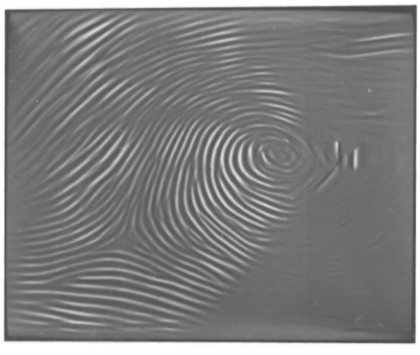
raw latent print



raw latent print



filtered image



filtered image

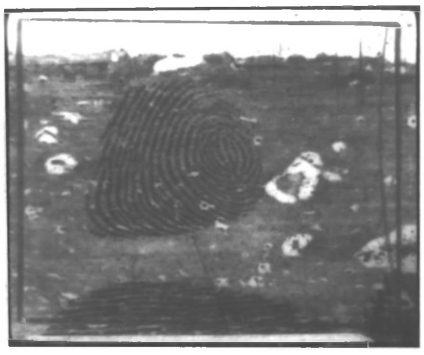


enhanced latent print



enhanced latent print

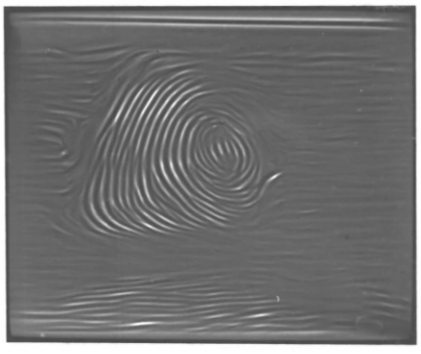
Figure 6.5. (continued)



raw latent print



raw latent print



filtered image



filtered image



enhanced latent print



enhanced latent print

Figure 6.5. (continued)

6.5 Evaluation of results

Image enhancement is the processing of an image so that the enhanced version is more suitable than the original for a specific application. The quality of this fingerprint enhancement technique must therefore be judged in terms of the suitability of the enhanced prints for the specific application of personal identification on an AFR system.

Evaluation of the enhancements of the rolled prints and scene of crime marks will be dealt with separately in the following two subsections. Conclusions will be drawn in section 6.6.

6.5.1 Evaluation of enhancements of rolled fingerprints

The method of evaluation that was adopted depends upon the use of part of the encoding stage of the Home Office AFR algorithm, which will now be introduced.

After the enhancement stage of the Home Office AFR algorithm, the enhanced print is encoded by locating the minutiae and storing in the data base a record containing the minutiae positions and orientations for the print. The process of locating the minutiae consists of two phases:

1. Minutiae extraction. Analysis of the image yields a list of candidate minutiae. Amongst this list are usually a large proportion of "false minutiae", i.e. points which have been incorrectly identified as minutiae. These may be further categorised into "soft" and "hard" false minutiae. Soft false minutiae are caused by noisy imperfections within the enhanced

image such as jagged ridge edges, isolated dots and small holes in the ridges. Hard false minutiae occur where the enhanced image possesses apparently true minutiae which are not present in the raw fingerprint image.

2. Minutiae reduction. During this phase, further processing of the image and the minutiae list removes most of the false minutiae from the list. Unfortunately, this process can also result in "missing minutiae", i.e. the removal of true minutiae from the list.

The Home Office has established that the scores achieved by its matching algorithm are particularly sensitive to the presence of false minutiae in the encoded fingerprint records, yet relatively tolerant of missing minutiae. Because of this, it is far more important to achieve a low number of false minutiae than a low number of missing minutiae. Consequently, the minutiae reduction algorithm has been tuned to be fairly strict so that, when applied to the images produced by the Home Office enhancement algorithm, it aims to remove all of the soft false minutiae leaving only the hard false minutiae, even though this tends to result in a substantial number of missing minutiae.

The enhancement method proposed in this thesis will henceforth be referred to as the "proposed method" in order to distinguish it from the Home Office method.

The proposed method was evaluated against the Home Office method by enhancing each print using both algorithms, and comparing the results obtained upon submitting each of the two enhancements to

the Home Office minutiae detection and reduction algorithms. These results are laid out in Table 6.1.

In addition, a Home Office fingerprint expert [Millard 1987] determined "ground truth" values for the true numbers of minutiae within each print, as well as the numbers of missed and false minutiae remaining after completion of the minutiae reduction algorithm. These numbers are presented in Table 6.2. The fingerprint expert also assisted with the interpretation of these results.

The results of Tables 6.1 and 6.2 yield the following comparisons between the Home Office and the proposed enhancements:

1. Total number of candidate minutiae detected. This gives an indication of the relative degree of noisiness of the two enhancements, since many more false minutiae will be detected in an image which exhibits jagged ridges, small holes, etc., than in an image which is smooth and clean. In every case, many more minutiae were detected on the Home Office images. On average 319.5 minutiae per image were detected on the Home Office enhancements, as against 133.7 minutiae for the proposed enhancements, i.e. 2.42 times as many minutiae. (The average ground truth value was 96 minutiae).

The number of minutiae generated by the Home Office enhancements was on average 250 percent in excess of the ground truth number. The corresponding excess for the proposed method was 43 percent, which compares very favourably.

Fingerprint figure number	Using Home Office enhancement algorithm			Using enhancement algorithm proposed in this thesis		
	a.	b.	c.	a.	b.	c.
6.1	402	100	125	114	84	35
6.4a	313	78	92	113	71	30
6.4b	560	122	200	190	79	53
6.4c	355	92	100	143	77	45
6.4d	409	122	128	165	88	52
6.4e	300	99	87	132	80	40
6.4f	243	118	72	132	82	44
6.4g	233	122	75	125	86	45
6.4h	467	101	164	169	78	51
6.4i	310	89	85	98	68	28
6.4j	333	82	98	95	64	31
6.4k	214	92	61	116	89	40
6.4l	226	105	75	139	85	38
6.4m	237	135	79	158	114	55
6.4n	190	92	57	116	74	36

a. = Total number of minutiae found by minutiae detection algorithm
b. = Number of minutiae remaining after minutiae reduction
c. = Time required to perform the minutiae reduction (in seconds)

Table 6.1 Results of application of Home Office minutiae detection and reduction algorithms to enhanced fingerprint images

Fingerprint figure number	Ground Truth (true number of minutiae)	Using Home Office enhancement algorithm		Using enhancement algorithm proposed in this thesis	
		Missed minutiae	False minutiae	Missed minutiae	False minutiae
6.1	87	11	24	9	8
6.4a	-	-	-	-	-
6.4b	74	4	52	13	18
6.4c	-	-	-	-	-
6.4d	102	9	29	20	6
6.4e	92	9	18	19	9
6.4f	113	10	15	27	4
6.4g	107	9	24	25	4
6.4h	84	24	41	23	17
6.4i	83	10	16	18	3
6.4j	80	14	16	23	7
6.4k	92	10	10	9	6
6.4l	108	13	10	23	0
6.4m	131	8	13	24	7
6.4n	96	15	11	22	0

Table 6.2 Analysis of fingerprint enhancement results by fingerprint expert. This Table shows the "ground truth" number of minutiae within each print, as well as the number of false and missing minutiae after completion of the minutiae reduction algorithm.

2. Time taken for minutiae reduction. Because of the far greater number of false minutiae detected on the Home Office enhancements, one would expect that the task of minutiae reduction for the Home Office enhancements is more onerous than for the proposed enhancements. This is borne out by the processing times that were observed. On average, minutiae reduction took 2.43 times as long for the Home Office enhancements as for the proposed enhancements. The average processing times were 100 and 42 seconds per image respectively, an average saving of 58 seconds per image.

3. False minutiae remaining after minutiae reduction. Since almost all of the "soft" false minutiae are removed during the minutiae reduction process, the number of false minutiae remaining after this process will provide an indication of the number of "hard" errors which have occurred during the enhancement process. This number therefore provides an important evaluation of the enhancement algorithm.

The Home Office enhancements yielded false minutiae counts ranging from 10 to 52 and averaging at 21.5. The proposed enhancements yielded counts that were a factor of approximately 3 smaller, varying from 0 to 18 and averaging at 6.9. In terms of the ground truth numbers of minutiae, the false minutiae counts for the Home Office enhancements ranged from 9.3 to 70.3 percent of ground truth values, and averaged at 23.9 percent. The corresponding range for the proposed enhancements was 0 to 24.3 percent, with an average of 7.8 percent. Clearly, the proposed enhancement compares very favourably with the Home Office enhancement.

4. Missing minutiae after minutiae reduction. On average, the

Home Office enhancements yielded 11.2 missing minutiae per image, as against 19.6 for the proposed enhancements.

These figures do not constitute a disadvantage of the proposed enhancement, since the minutiae reduction algorithm has been optimised for the Home Office enhancements, which exhibit approximately five times as many soft false minutiae as the proposed enhancements. To eliminate such large numbers of false minutiae, the reduction algorithm had to be tuned to be particularly strict. Consequently, when it is applied to the proposed enhancements a larger proportion of the true minutiae are eliminated. It is expected that a gentler tuning of the minutiae reduction algorithm would reduce the number of missed minutiae while not significantly increasing the number of false minutiae.

6.5.2. Discussion of enhancements of scene of crime marks

Unfortunately, the method of evaluation of the previous section cannot be applied to latent prints, because the Home Office algorithm is unable to enhance images of the poor quality associated with most latent prints.

Indeed, enhancement of latent images is well known to be a difficult problem and few attempts have been made to solve it. Commercially available AFR systems at present do not include an enhancement feature for latent images, but instead require manual encoding either by interactively marking the positions of minutiae on the screen (as in the De La Rue Printrak and Morpho systems), or by manually tracing each ridge in the image (as in the NEC

system).

Expert opinion [Millard 1987] suggested that in the absence of an alternative enhancement algorithm against which to compare the proposed algorithm, and in the absence of an algorithm (such as minutiae detection) for further processing of the enhanced print, no meaningful quantitative evaluation of its suitability for AFR purposes could be provided. The best that could be done would be for an expert to mark up the positions of the minutiae upon the raw latent images. Although not immediately useful, such data could assist in the evaluation of any alternative algorithm proposed by a future researcher.

Nonetheless, visual examination of the enhancements of figure 6.5 shows a considerable reduction in the noise content as compared to the raw images, and indicates a potential for processing by automated encoding algorithms.

6.6 Conclusion

This chapter has presented an algorithm for the enhancement of fingerprint images, and demonstrated its use upon rolled and latent fingerprints of various Henry classes. For rolled fingerprints, the quality of enhancement produced by this algorithm was found to compare favourably with that produced by the algorithm currently in use by the Home Office and Scotland Yard. The algorithm may also have the potential for use in an automated encoding system for latent prints.

CHAPTER 7

AN IMPLEMENTATION USING OPTICAL HARDWARE

7.1 Introduction

Chapters 2 to 6 showed how fingerprints can be enhanced using nonstationary anisotropic Fourier domain filters. In optical systems, the operation of Fourier transformation is easily achieved "instantaneously" and in parallel. This chapter describes an optical implementation of the fingerprint enhancement technique. Only analogue optical systems will be considered here. Much research is being done into the feasibility of digital optical systems, but no functioning digital optical computer systems exist at present.

The development of the techniques in Chapters 2 to 6 was strongly influenced by the requirement that they be implemented on a general purpose computer. For example, when choosing a method of determining LRO at each point, considerations of computational speed upon a general purpose computer led to the decision to use a method involving the evaluation of LRO at a grid of points, followed by interpolation between these orientations, even though alternative methods where an estimate is found at each point are potentially superior if the resultant orientations are spatially smoothed.

In an implementation involving optical hardware, however,

different factors influence the speed of processing. Optical systems have the advantage that their processing is inherently parallel. An optical image processing operation, such as forming the Fourier transform of an image or adding two images, is performed in parallel upon all points within the window or image simultaneously and can therefore be performed instantly in real time. In many applications, this can lead to an enormous speed advantage, making possible (and simple) such operations as, for example, real-time low-pass filtering of moving images. This operation could not be implemented upon a general-purpose computer, since even if additional array processing hardware were used, taking a discrete Fourier transform of a reasonably sized image would take a second or two. The major constraint on the speed of an optical system would be the amount of motion required from any mechanically moving parts that might be present within the system, rather than the number of multiplications or other computational operations that are necessary.

Set against the advantage of inherent parallelism are three main disadvantages:

1. Because of its analogue nature, optical processing is more sensitive to distortion and noise within the system and its components. The precision of the results is determined by the signal-to-noise ratio of the output image, and is usually several orders of magnitude lower than the precision available on digital systems.

2. Optical systems provide serious constraints upon the available

range of operations that can be implemented. Fourier transforms can be done using Fourier lenses. Multiplication by functions taking values between 0 and 1 can be done by obscuring the path of the light by a transparency. Therefore, convolutions can easily be performed. Averaging, shifting, rotation, stretching and simple distortions are also simple to implement.

3. Optical systems are less flexible than digital systems not only because of the limited range of operations that are available, but also because they are more difficult to modify when a change to the system is required. Modification of a system implemented on a conventional computer is easily done by editing and recompiling the source software. Making changes to an optical system may involve exchanging parts, generating new transparencies or changing the physical configuration of the parts of the system.

Despite these disadvantages, optical systems can be simple, cheap and fast provided that the processing required maps simply onto the range of operations that can be easily implemented in optical hardware.

A popular introduction to the capabilities and applications of optical computing systems is given by Stroke [1972].

7.2 Previous work on optical processing of fingerprints

The work of Marom [1967] on fingerprint classification by rotational averaging of the optically generated Fourier transform of the image, and that of Scherlowski [1974] on fingerprint enhancement by optical Fourier filtering, have already been described in chapter 1.

McMahon et al [1975] and Shizume and Hefner [1978] describe an optical approach to determining local ridge orientation from 35mm slides of police fingerprint cards. Their approach relies upon the fact that the Fourier transform of a grating is concentrated in the vicinity of two peaks along a line oriented perpendicularly to the grating (see chapter 2).

The apparatus used is illustrated in figure 7.1. Plane coherent light is shone through the transparency and the Fourier transform of the print is formed using a Fourier lens (see section 7.3.1.1). A rotating sector slit is placed in the transform plane, and a second Fourier lens inverts the transform, forming a filtered image on the optical detector array.

Consider any small region of the print. The contribution by this region to the Fourier transform will be in the form of two peaks along a line oriented perpendicularly to the LRO. When the rotating sector slit is aligned with the two peaks, the light intensity on the photodetector in the image plane which corresponds to this region will reach a maximum. Therefore, the time interval from some reference orientation of the slit to the instant the signal peaks in the detector is proportional to the

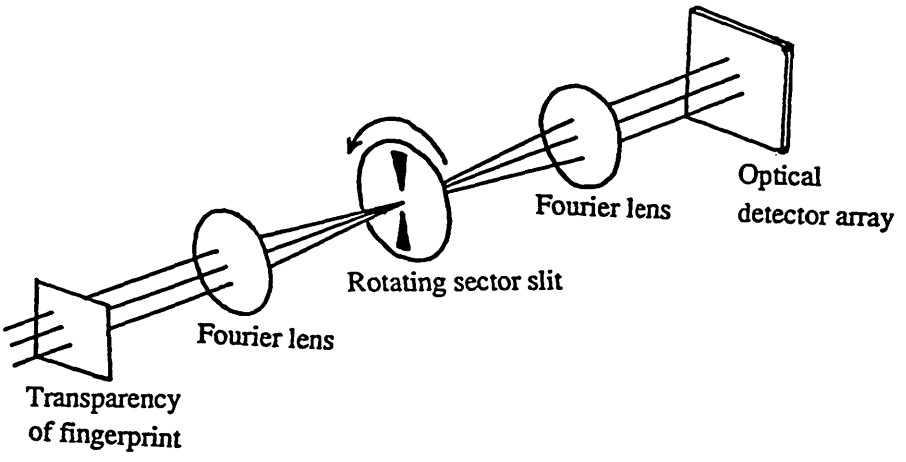


Figure 7.1. Apparatus for determining LRO optically from a transparency of a fingerprint image.

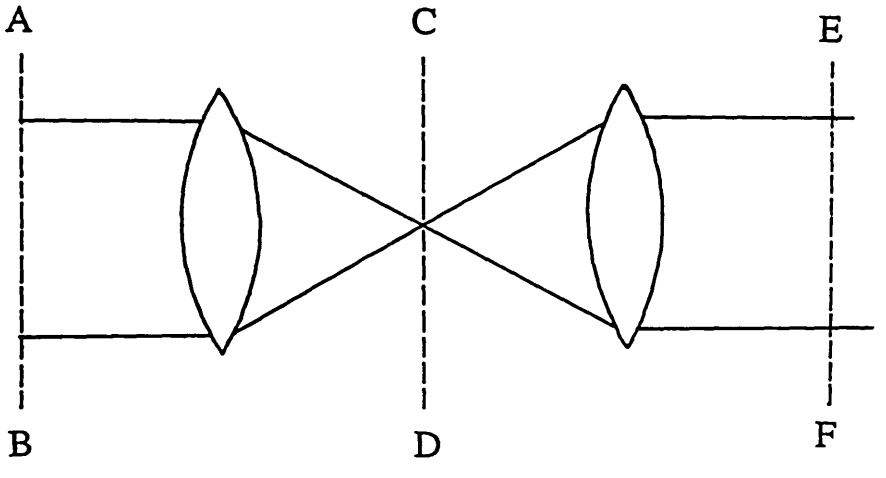


Figure 7.2. Use of a Fourier lens to form the Fourier transform of an image transparency.

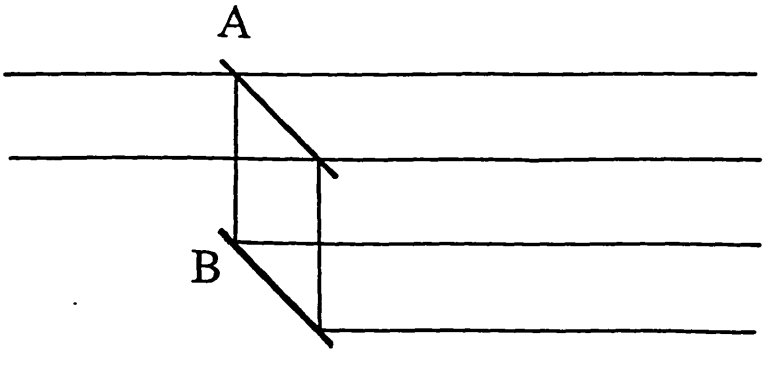


Figure 7.3. A two-way beam splitter.

LRO within that region of the print.

Horvath and Doyle [1968], and Horvath et al [1967] investigated the use of optics to perform fingerprint recognition by matched spatial filtering. Although matched filtering has the advantage that recognition can be achieved even when only part of the image is available, its disadvantage is its sensitivity to topological distortions of the print caused by skin stretching, compression and growth. This sensitivity occurs because the matching is performed upon the transform of the entire image, which will change considerably under the influence of distortions.

7.3 Optical implementation of the enhancement algorithm

7.3.1 Optical components used

This section introduces the optical components that will be needed as parts of the processing systems described later. It is not intended to be an exhaustive catalogue of available optical processing elements.

7.3.1.1 Fourier lens

If a transparency of an image is placed in the focal plane AB of a Fourier lens, and plane monochromatic light is shone through the transparency as shown in figure 7.2, then a signal whose magnitude and phase is proportional to the magnitude and phase of the Fourier transform of the image will be produced at the opposite focal plane CD of the lens. A Fourier lens can also be used to perform the inverse transform. If a second Fourier lens is placed (as in the figure) such that the transform is in the focal plane CD, the inverse transform will appear in the opposite focal plane EF.

7.3.1.2 Beam splitter

A (two-way) beam splitter will split up a beam of monochromatic light into two beams having the same magnitude and phase distribution but approximately half the intensity of the input beam. The splitter consists of a mirror and a half-silvered mirror as shown in figure 7.3. Multi-way splitters can be produced by cascading two-way splitters.

7.3.1.3 **Sector detector**

If the Fourier transform of an image is available, then an optical element known as a sector detector can be used to analyse the distribution of energy within the frequency domain. A typical configuration for a sector detector is illustrated in figure 7.4a. The detector is disc-shaped. The top half of the disc is divided into equal-sized sectors and the bottom half into half-annular regions of equal width. Sector detectors are available having from 2 to 32 sectors and half-annuli. The surface of each region consists of material whose electrical properties are photosensitive. Each region is insulated from its neighbours, and an electrical connection is provided to each region. The electrical signal produced by each region is dependent upon the amount of energy in the light impinging upon the region.

To analyse the Fourier transform of an image (or of a window within an image), the sector detector is centred upon the DC component of its transform, as shown in figure 7.4b. Because of the Hermitian symmetry of the transform, the energy distribution within the top half will be repeated in the bottom half. The signals θ_1, θ_2, ... , θ_n indicate the energy within the n orientation ranges

$$(i-1)\pi/n <= \theta <= i\pi/n \quad \text{for } i=1..n.$$

Similarly, the signals r_1, r_2, ... , r_n will indicate the radial-frequency distribution of the energy.

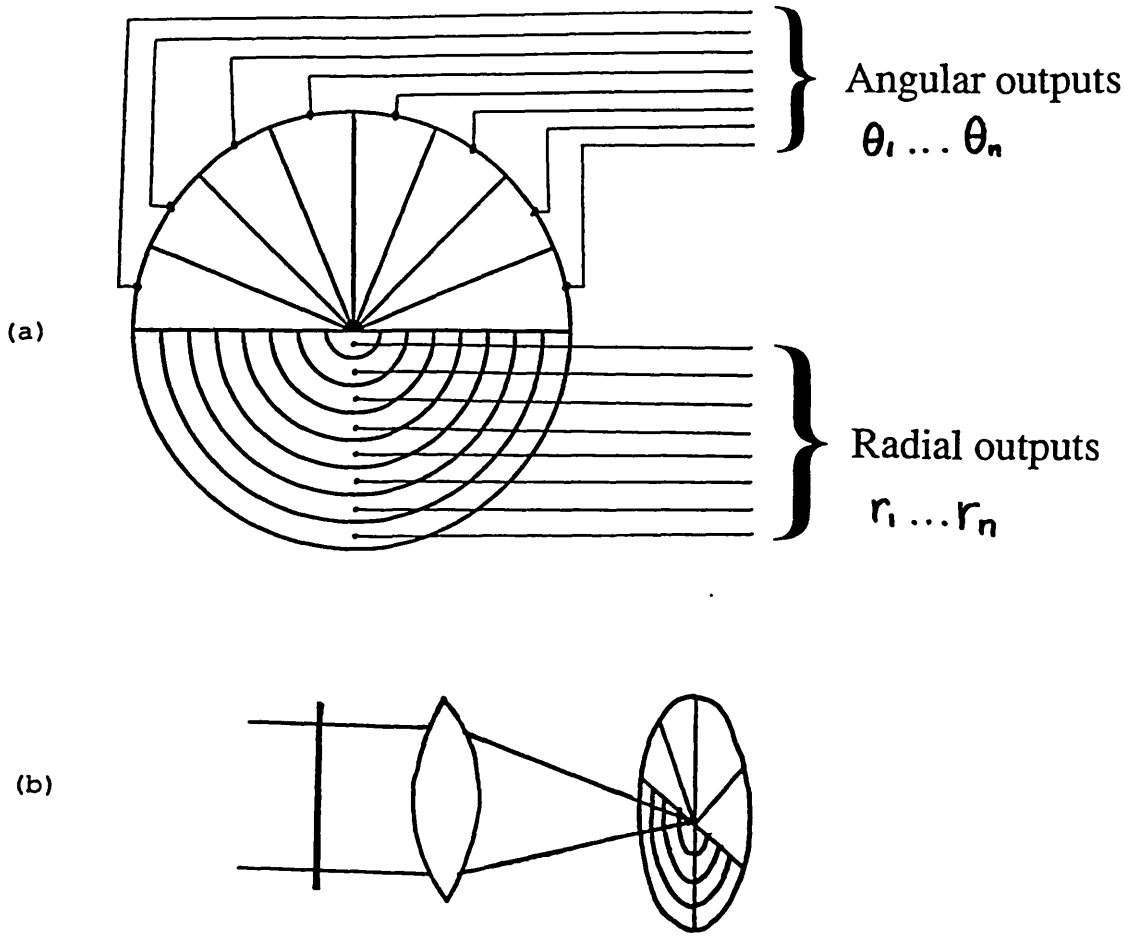


Figure 7.4. (a) A typical sector-detector.
(b) Use of a sector-detector to analyse the Fourier transform of an image.

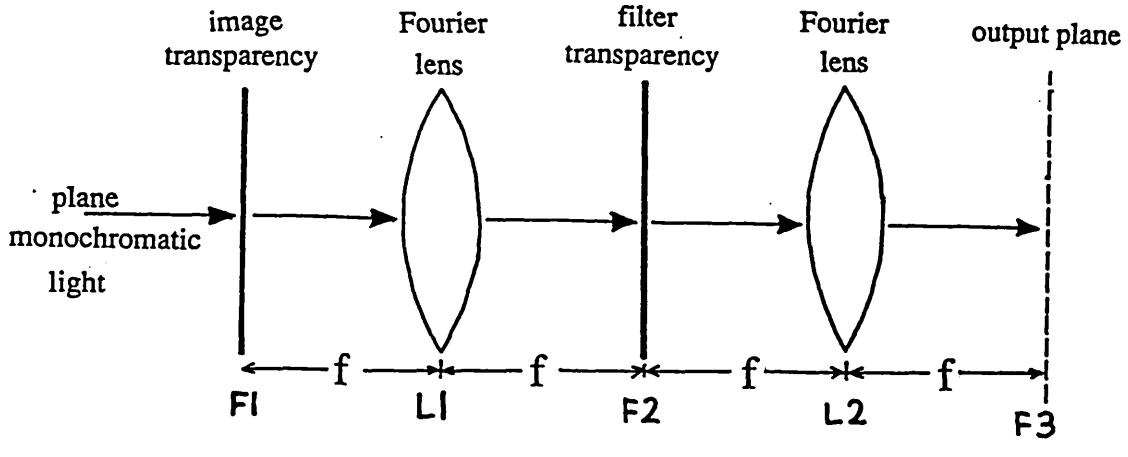


Figure 7.5. Two-dimensional stationary filtering using optical hardware.

If connected to appropriate electronics, the sector detector can provide signals indicating the approximate radial and orientation coordinates of the region in the transform having the greatest concentration of signal energy.

7.3.1.4 **Selector**

Presented with input beams $B_1,\ldots,B_n$, and an electrical selection signal S, the output will be one of $B_1,\ldots,B_n$ depending upon the value of S. Selection can be achieved either by the opening and closing of mechanical shutters or by the use of liquid crystal spatial light modulators.

7.3.2 **Fourier domain filtering of images using optical hardware**

Figure 7.5 shows how a two-dimensional filtering operation can be performed upon an image. Monochromatic light is spatially modulated by the image in the form of a transparency. This transparency is located in the focal plane F1 of a Fourier lens L1 so that its transform is formed at its opposite focal plane F2. A transparency representing the filter is also located in the plane F2. The distribution of light emerging to the right of the filter transparency is therefore the product of the image transform and the filter function. The inverse transform of this product is formed by the Fourier lens at its focal plane F3 where the filtered image is formed and recorded.

The recording process is usually photographic or electronic. Usually only the magnitude distribution will be recorded and any information in the phase distribution will be lost. Because of

this, it is important that the DC component of the signal is not removed by the filter. If the DC component were removed, then the output filtered image would consist of both positive and negative values of intensity. Where the output is negative, the phase will be shifted by π radians, but the magnitude would be the same as that of a positive signal of the same intensity. Therefore, the recording process would record the absolute value of the output rather than its true value. On the other hand, if the DC component were allowed through by including a pinhole at the origin of the filter transparency, then the output image would have a positive DC component about which the filtered values could vary, with greatly reduced risk of negative intensities being encountered.

7.3.3 Implementations

7.3.3.1 Implementation using fixed filters

Figure 7.6 shows an implementation of the filtering algorithm which is closely analogous to the implementation of Chapter 2 to 6 based upon directionally prefiltered images. Because the amount of optical hardware needed for this implementation increases linearly with the number of prefiltered images, it will be discussed assuming that a single set of eight directional components is used.

The fingerprint image is illuminated with a uniform plane monochromatic beam derived from a low power laser. A small aperture is used to define the region of the transparency to be

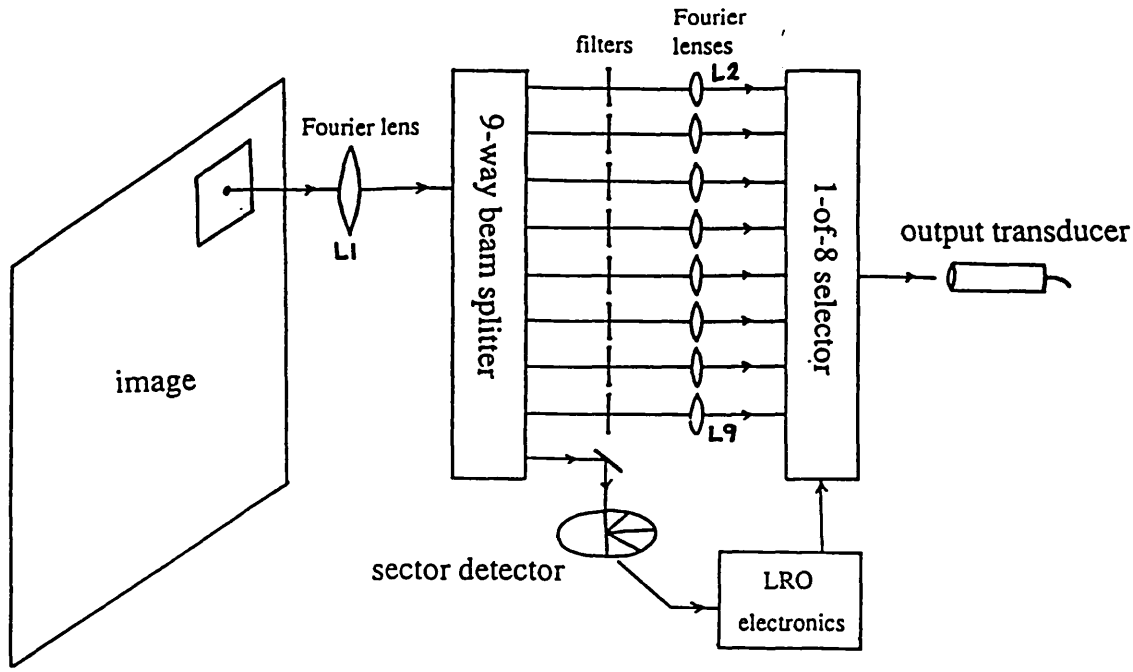


Figure 7.6. Optical implementation of the enhancement algorithm using fixed filters.

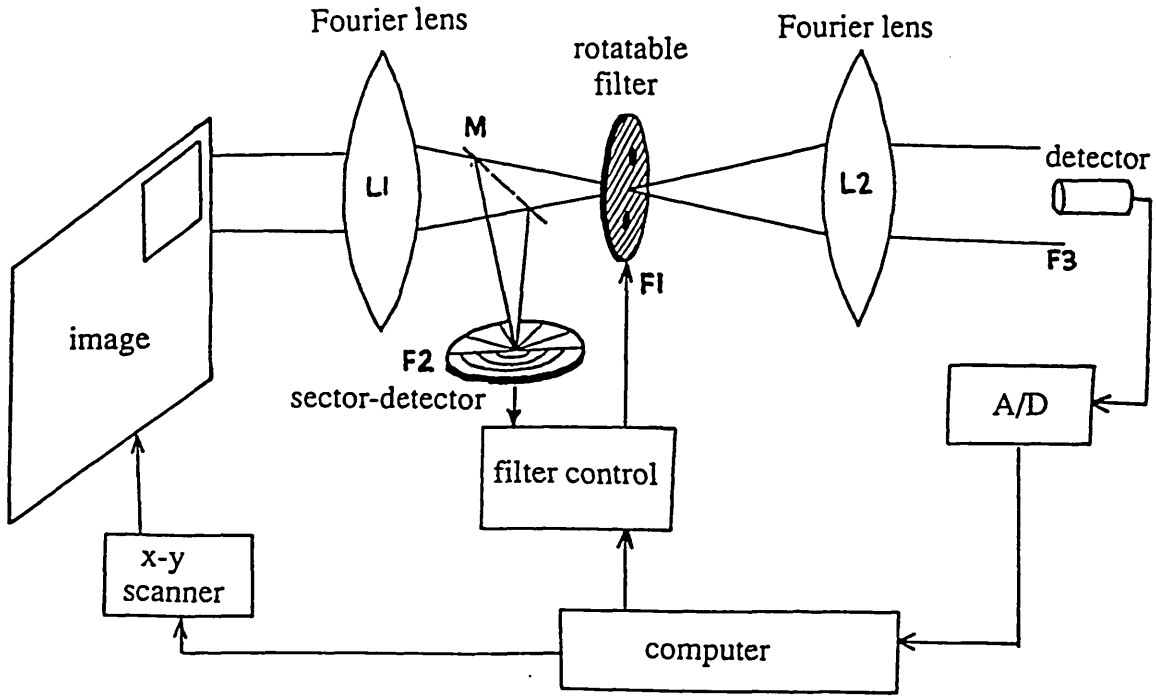


Figure 7.7. Optical implementation of the enhancement algorithm using a servo-controlled rotating filter.

interrogated. The transparency is raster-scanned through the image plane so that a window around each point of the image is processed in turn. Suppose that the region is centred about the point (x,y). The contents of the window are Fourier transformed by lens L1. This transformed signal is then split nine ways. One of the signals falls upon a sector detector. The outputs from this sector detector are connected to an electronic circuit which determines the LRO within the window by finding the angular sector with the most energy. The other eight signals pass through transparencies representing the eight filters of Chapter 3:

$$F_i(\rho,\phi) = \delta(0,0) + F_{angle}(\phi-\phi_i).F_{radial}(\rho),$$

where $\phi_i = i\pi/8$, and i=0..7.

Note that an additional term $\delta(0,0)$ representing a pinhole at the origin has been included in order that the DC component may pass through.

The eight filtered signals are then inverse transformed by Fourier lenses L2 to L9, yielding eight directionally filtered versions of the data within the window. Finally, the LRO value from the sector detector is used to select the appropriate one of the eight filtered images, and its value at (0,0) is the (x,y) contribution to the output image.

7.3.3.2 **Implementation using servo controlled rotating filter**

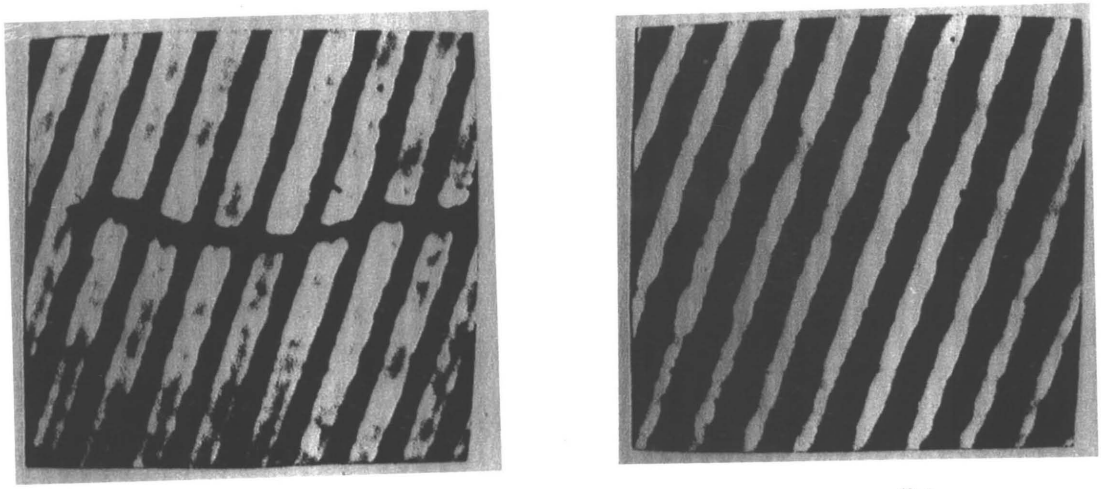
Since the eight filters used in the previous approach are simply rotated versions of one another, filtering in any desired

orientation can be achieved by physically rotating to the required orientation a transparency representing the filter.

Implementation of this approach [Monro, Sherlock & Petts 1987] leads to the system shown in figure 7.7. This system works as follows. As in the system of figure 7.6, a window must be raster scanned over the image. A uniform plane beam of light is modulated by the window contents and then passed through a Fourier lens L1. A half-silvered mirror M placed between the lens and its focal plane splits the beam into two so that the Fourier transform of the window is formed in planes F1 and F2. The transform in plane F2 falls upon a sector detector which is connected to electronic circuitry which produces a signal whose value indicates the LRO within the window. The transform in plane F1 falls upon a transparency of the filter function. The filter transparency is mounted so that it can be rotated to any desired orientation. The LRO signal from the sector detector drives a servo controller which sets the orientation of the filter transparency appropriately so that the window contents is filtered in the same orientation as the ridges. Lens L2 performs the inverse Fourier transform. The intensity at the centre of the filtered image in plane F3 is obtained using a photodetector and forms the (x,y) pixel value of the filtered fingerprint.

The system of figure 7.7 represents a considerable improvement over the system of figure 7.6, since the 9-way splitter becomes a two-way splitter, only one Fourier lens is needed for the forward transform and no selector is required at the output.

Figures 7.8a and b show results obtained from a simple optical

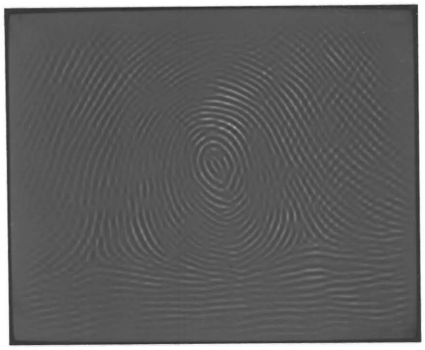


(a) (b)

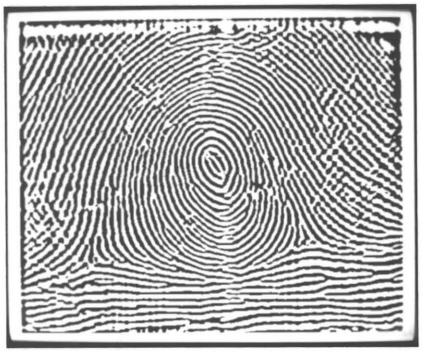
Figure 7.8. Results obtained using an optical system based on figure 7.7, showing a periodic image with defects (a), and the filtered version of this image (b).



(a)



(b)



(c)

Figure 7.9. (a) Fingerprint image.
(b) Filtered fingerprint, using the method of section 7.3.3.3.
(c) Thresholded version of (b).

filtering experiment based upon the configuration of figure 7.7. In this experiment a periodic image with defects, simulating an area of constant ridge orientation, has been processed. This processing can be regarded as the computation of a single point of an enhanced image. The effect of cleaning up the image is apparent.

Following the success of this initial experiment, the full system of figure 7.7 was implemented, but no complete enhancement was obtained because its operation was found to be too slow (because of mechanical constraints) to justify tying up the optical equipment for long enough to process an entire image. The equipment was required for other purposes by the staff at the GEC Hirst Research Centre, where the experiment was conducted.

In an attempt to obtain faster processing, another approach was investigated. This is described in the following section.

7.3.3.3 Implementation using 1-d filters

The implementation of the previous section had a severe speed constraint implied by the use of servo controlled moving parts to adjust the angular position of a filter transparency. Moreover, it was necessary to scan a window mechanically over the print. In the present approach it is sought to obviate the need for moving parts by replacing the two-dimensional filtering operation by two one-dimensional filtering operations, in the directions of the x and y axes. This leads to a method which would be suited to implementation in video electronics.

A small windowed region of a fingerprint can be approximated locally by a plane wave of frequency $\omega = 2\pi/LRS$ and orientation ϕ=LRO, where LRS and LRO are the ridge spacing and orientation within the window.

Along any line parallel to the x-axis, the projection of this plane wave is a sinewave of angular frequency

$$\omega_x = \frac{2\pi}{LRS} |\sin LRO|,$$

and along a line parallel to the y-axis, the projection has angular frequency

$$\omega_y = \frac{2\pi}{LRS} |\cos LRO|.$$

Note that because of the symmetries of the absolute values of the sine and cosine functions, the same values of ω_x and ω_y will be produced by a wave at orientation π-LRO as by one at orientation LRO. The significance of this will be seen when the results are examined.

The approach of this section is first to perform one-dimensional filtering upon the image, row by row, using a 1-d bandpass filter whose centre frequency is $\omega_x = \omega_o \cos LRO$ (where ω_o is the centre frequency of the two-dimensional filter used in the previous section). Then the image is filtered column by column in the same manner, using a 1-d bandpass filter with centre frequency $\omega_y=\omega_o \sin LRO$.

If the LRO is known everywhere or can be determined rapidly

enough, then this filtering operation can be performed at video frame rates, as follows:

1. The image is raster scanned. The pixels are therefore addressed row by row, in the correct order for the first pass using 1-d filters of centre frequencies ω_x.

2. The image is transposed by rotating it through 90 degrees, so that columns become rows. (If the image is held in a frame store, then the rotation may be accomplished simply by altering the addressing during video read-out).

3. Step 1 is repeated, using centre frequencies ω_y. Because of the rotation, the filtering is effectively done on columns rather than rows.

4. The image is rotated through -90 degrees to restore its original orientation.

It should be noted that this filtering operation is not identical to the 2-d filtering operation from which it was derived, but it was hoped that its approximation would be close enough to allow good enhancement to be achieved. A computer simulation of the system was used to obtain the filtered image of figure 7.9b.

It can be seen that the ridges have been smoothed directionally. The result has a similar appearance to the filtered images presented in chapter 6, except in one respect. Careful inspection of figure 7.9b shows that the filter has passed components at two orientations rather than one: in addition to the desired ridge structure (at orientation LRO), there are fainter ridge-like

artefacts crossing the ridges at orientation π-LRO. These artefacts are caused by the noise components oriented at angle π-LRO. They have been passed because the 1-d filters cannot distinguish between orientations LRO and π-LRO, as explained above.

Despite this disadvantage, the filtered image produced using this method is a considerable improvement over the raw image, especially considering the possibility of its running at video frame rates.

Figure 7.9c shows a thresholded version of figure 7.9b.

CHAPTER 8

AN INVESTIGATION OF THE USE OF ADAPTIVE WINDOWS IN FINGERPRINT PROCESSING

This chapter investigates the processing of fingerprints using hardware whose design has been based upon an artificial intelligence approach involving the simulation in electronics of neural cells such as are found in the retina of the eye and in the brain. Marr [1982] discusses approaches to artificial vision based upon development of models of the human visual system. The place of such approaches within the wider context of artificial intelligence is described by Aleksander [1987].

One such approach is described by the theory of adaptive windows. A commercially available device that implements the adaptive window approach to automatic recognition is called the WISARD. Adaptive windows and the WISARD will be discussed in detail in this chapter, and it will be seen that use of the WISARD gives promising results for some fingerprint processing tasks but disappointing results for others.

8.1 Adaptive windows

Models of the human visual system [Marr 1982] have often been described in terms of window-based image processing. In these models it is sought to explain visual phenomena in terms of the known characteristics of retinal and neural cells. Each such cell is expressed as operating upon a window consisting of the data serving as its input.

The theory of adaptive windows [Aleksander & Wilson 1985a] describes how windows can be trained or adapted to recognise objects within images or, when scanned over an image, to perform image transformations such as edge detection. Adaptive windows have also been applied to the problem of determining the depth of objects within stereoscopic images [Wilson 1985]. The WISARD described in the next section is an implementation of the adaptive window technique in hardware, which can perform recognition operations in real time.

For reasons that will be discussed in section 8.3.1, we shall be interested mainly in the case where the image is scanned by a window, pixel by pixel, producing output images consisting of window responses. The alternative, whereby the adaptive window is large and stationary and is trained to recognise entire images, turns out to be of less use in fingerprint processing. The theory and description given below is based upon [Aleksander & Wilson 1985a].

Suppose that a rectangular window of

$$a \cdot b = R \text{ pixels}$$

is scanned over an image of

$$c \cdot d = F \text{ pixels.}$$

Call the R window pixels

$$I = \{i1, i2, \ldots, iR\}.$$

Any ordered group of N pixels arbitrarily chosen from I is known as an N-tuple. An N-tuple can have 2^{NB} different states, where B is the number of bits per pixel. Clearly, the state of an N-tuple is determined by the values of the N pixels comprising the N-tuple.

Two methods may be used to sample pixels when forming N-tuples:

a. One-to-one. In this case, the pixels are chosen at random, but each pixel is used once only, and therefore belongs to only one N-tuple. The number of N-tuples within the window is therefore equal to R/N.

b. Random. In this case, N-tuples are chosen totally at random. Let the total number of N-tuples chosen be K. The percentage cover of the window is defined as

$$\%C = 100 \cdot \frac{K}{(R/N)}$$

Usually, 100% cover is regarded as sufficient cover, but higher percentages are sometimes used.

The process of choosing the N-tuples, though arbitrary, is done

once and for all and the N-tuples are thereafter kept fixed for the entire image processing operation. The training and use of the window are thus done using identically chosen N-tuples.

The process of training the window is now described. A set

$$T = \{T_1, T_2, \ldots, T_t\}$$

of t training images is used to train the window. Each training image is of the same dimensions as the window, i.e. a.b=R pixels, and can thus be expressed as:

$$T_j = \{i^j_1, i^j_2, \ldots, i^j_R\} \quad \text{for each } j \text{ in } \{1..t\}.$$

Each T_j implies some state s_{ji} for each N-tuple, and we can therefore represent the states of the N-tuples for the jth training image by the set

$$S^j = \{s^j_1, s^j_2, \ldots, s^j_K\}$$

where s^j_i is the state of the ith N-tuple.

For each N-tuple there exists a set of training states corresponding to the set of t training images:

$$S_k = \{s^1_k, s^2_k, \ldots, s^t_k\} \quad \text{for the kth N-tuple.}$$

It is usually the case that some of the elements of S_k are equal, and therefore S_k has a reduced form

$$S_k = \{s^1_k, s^2_k, \ldots, s^\tau_k\}$$

where $\tau = \text{Card}(S_k) <= t$ and $s^i_k \neq s^j_k$ for all i,j.

The training process consists of associating the logical value 1 with all those states of N-tuples that belong to the sets of training states, and 0 with those that do not. That is to say, for the kth N-tuple, 1 is associated with the τ states

$$s^1_k,\ s^2_k,\ \ldots,\ s^\tau_k,$$

and 0 is associated with the other $2^{NB} - \tau$ states.

Once trained, the window can be applied to an unknown image (also of dimension a.b = R pixels)

$$U = \{i^U_1,\ i^U_2,\ \ldots,\ i^U_R\},$$

thereby producing the following set of N-tuple states:

$$S^U = \{s^U_1,\ s^U_2,\ \ldots,\ s^U_K\}.$$

In the set S^U we shall consider all elements to be distinct, i.e.

$$s^U_i = s^U_j \quad \text{iff} \quad i = j,$$

and therefore $\text{Card}(S^U) = K$.

Some of the states within S^U will correspond to states which have occurred during the training process, and others will not. The response set Z^U is the subset of S^U consisting of those states which have occurred during training and is given by

$$Z^U = \{s^U_j \mid s^U_j \in S^U \text{ and } s^U_j \in S_j\}.$$

The response of the window to pattern U is defined as the cardinality of Z^U, and is an integer ranging from 0 to K.

If a high response is obtained, it means that a large number of

N-tuples were given the same input by the unknown image as they had received from at least one of the training images. It is to be expected therefore that an unknown image which gives a high response is "closer" or "more similar" to the set of training images than one which produces a low response. The process of applying the trained window to an unknown image effectively forms a correlation of its N-tuples with those of all the images in the training set.

In practical implementations of adaptive window processing, such as the WISARD, the number of bits per pixel is usually chosen to be B=1, i.e. the window acts upon binary input images. This avoids certain complications (discussed in [Aleksander & Wilson 1985a]) which may occur when using more than one bit per pixel. In order to produce a binary image from the greyscale one, it is necessary first to perform a suitable thresholding operation upon the input image.

8.2 The WISARD recognition device

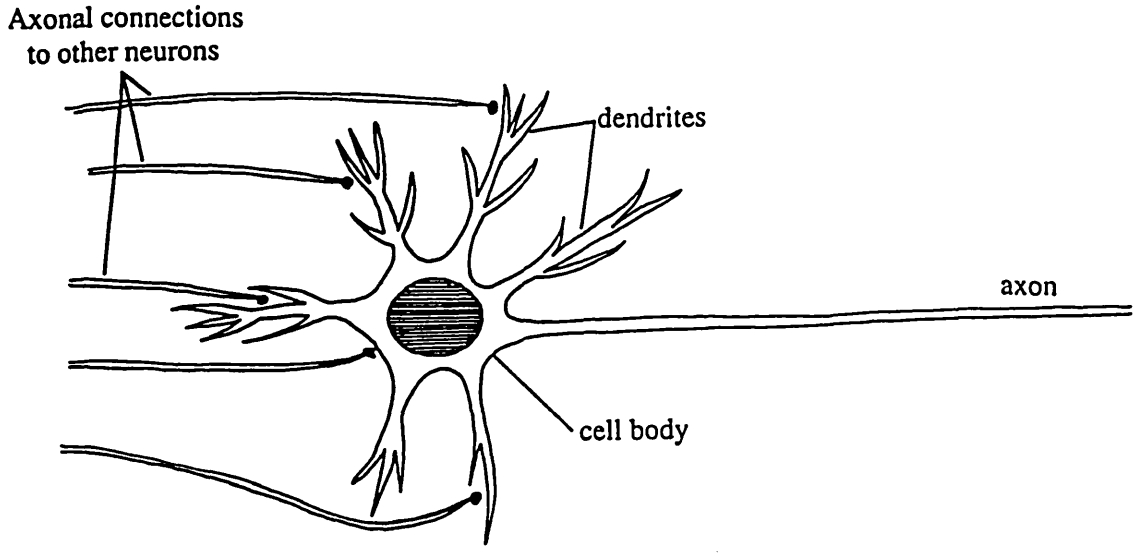
WISARD (an acronym for Wilkie, Stonham and Aleksander's Recognition Device) is a hardware implementation of the adaptive window recognition approach outlined in the previous section. This section, which describes the WISARD in some detail, is divided into three subsections, giving a conceptual description of the WISARD, a description of the actual commercially-available machine, and a discussion of a software simulation of the WISARD hardware.

Simple introductory references to the WISARD are Aleksander et al [1984], Redgers [1987] and Aleksander [1985]. WISARD is discussed in relation to the emergent "intelligent" properties of network structures by Aleksander [1983a, 1983b]. The relationship between the WISARD architecture and the theory of adaptive window processing is discussed in [Aleksander & Wilson 1985b].

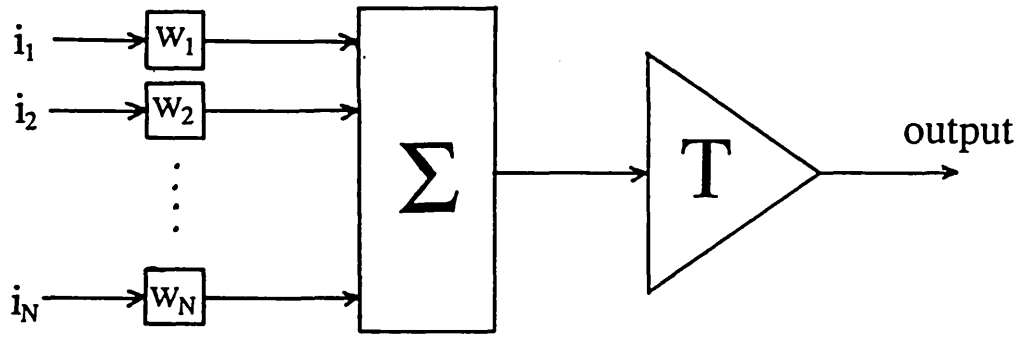
8.2.1 Conceptual introduction to WISARD

The WISARD was probably not originally conceived as an implementation of adaptive window theory. Rather, it is more likely that the theory was developed after the conceptual design of the WISARD in order to give a rigorous explanation of its already heuristically understood function. The WISARD is based upon an attempt to emulate or model the function of retinal and neural cells within the eye and brain.

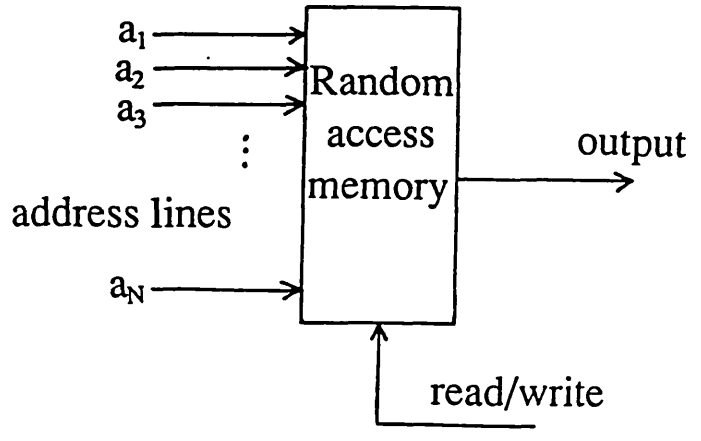
A typical nerve cell [Carpenter 1984, Kuffler & Nicholls 1976] is illustrated in figure 8.1a. It has a central cell body in which



(a) Nerve cell



(b) McCulloch & Pitts model of the nerve cell



(c) Random-access memory (RAM) circuit

Figure 8.1. Neural models

is located the cell nucleus, and protruding from the cell body is a long thin fibre called the axon. It is along the axons of nerve cells that nerve impulses are transmitted to other cells. The cell body has several receptor sites to which the axons of other cells are connected. The point of connection between an axon and a receptor site is called a synapse. A nerve cell will typically take input from the axons of a few thousand other cells.

The work of McCulloch and Pitts [1943] showed that the axonal output is a binary signal, taking one of two states according to whether or not the nerve is firing impulses along its axon. They proposed a model for the operation of the nerve cell, which has since become generally accepted. This model is illustrated in figure 8.1b. The cell bases its decision whether to fire or not upon the particular combination of signals received at its synaptic inputs. A cell may be influenced either positively or negatively by a particular synaptic input. If the presence of impulses at a particular synapse increases the likelihood of the nerve cell firing, then that synapse is said to be excitatory, otherwise it is said to be inhibitory.

In the McCulloch and Pitts neuronal model, a firing threshold is associated with each nerve cell, and a weighting factor with each synaptic input. This weight is a number in the range (-1,1) and is positive if the synapse is excitatory and negative if the synapse is inhibitory. The nerve's axon will be active (i.e. firing) if and only if the weighted sum of the inputs exceeds the cell's threshold.

Early attempts to simulate the behaviour of neurons usually

involved the construction of analogue circuitry which directly implemented the McCulloch and Pitts model, using potentiometers, summers and thresholding amplifiers. Clearly, such analogue implementations suffer from the disadvantages of mechanical complexity, unwieldy size, and expense.

Aleksander and Stonham realised that the operation of a nerve cell could be simulated digitally in a particularly convenient way. Any nerve cell having N synaptic receptor sites can be simulated using a random access memory (RAM) having N address lines and one data line. In this simulation, the address lines of the RAM correspond to the synaptic inputs of the cell and the data output line of the RAM corresponds to the cell's axonal output, as shown in figure 8.1c. Irrespective of which synapses are excitatory and what the relative weights of each are, an appropriate content for the RAM can be chosen such that the output line of the RAM always reflects the same state as the axon of the neuron, when the RAM is fed with the same combination of inputs as is the neuron.

Considerations on the above led to the publication of a paper [Aleksander & Stonham 1979] describing the use of RAMs for pattern recognition tasks.

The realisation that neurons could in full generality be simulated by RAMs was of great importance because of the simplicity, speed, reliability and cheap cost of random access memory. One simplification in the model is necessary in practice, however. A typical neuron may have thousands of synapses, whereas a RAM chip will only have a few (say four to eight) address lines. Clearly

a RAM chip with 2000 address lines, and therefore containing 2^{2000} bits (about 10^{1300} Gigabytes) of memory would not be feasible to construct! The number of synapses, or address lines, corresponds to the tuple size N as used in the development of the theory in section 8.1. Because of the abovementioned practical limitations of RAM chips, the WISARD is implemented using values of N ranging from 4 to 8.

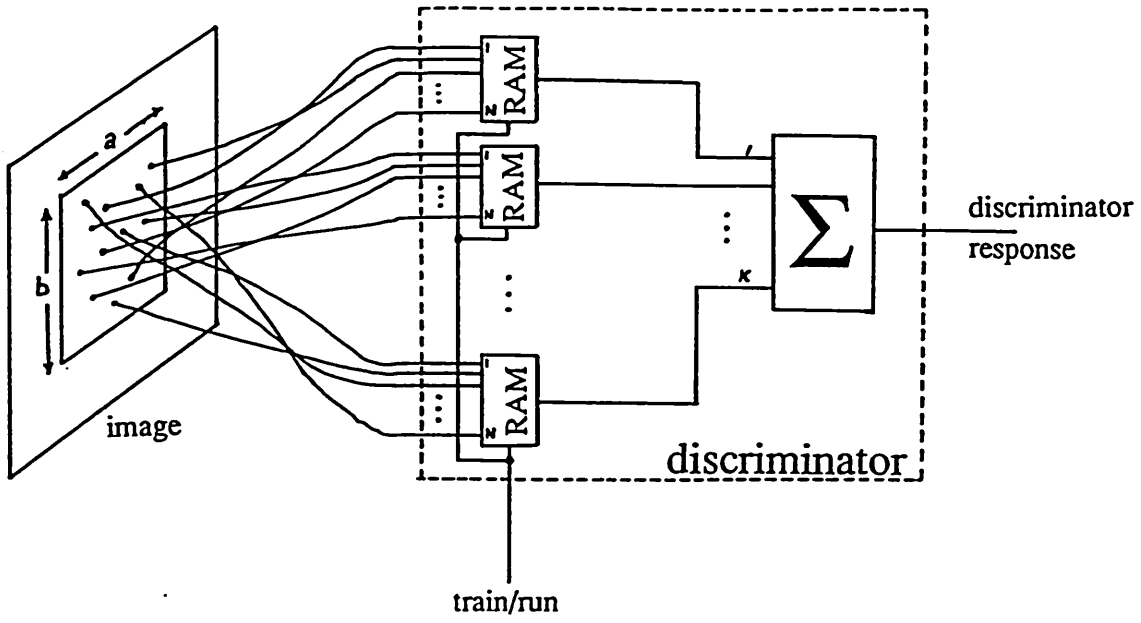
Figures 8.2a and 8.2b illustrate the functioning of the WISARD hardware. A single discriminator is shown in figure 8.2a; this circuitry is repeated identically for each discriminator as shown in figure 8.2b. The input window of size a.b is shown on the left, and is assumed to be binary, consisting of pixels having the values 0 or 1. This window may be the entire image or may be of smaller dimension so that it can be scanned around the image. The address lines of RAM chips are activated by the pixels of the window. The tuple size N is the number of address lines per RAM, and the RAM address lines are mapped to positions in the window in a random fashion, but such that one and only one address line goes to any given pixel position. The number of RAMS required is ab/N, and the total memory capacity of these RAMs is 2^N. ab/N. As an example of the sharp increase in memory requirements with tuple size, the following are the memory requirements per discriminator for a window size of 256 by 256 pixels for various values of N:

N=4 : 256 Kilobytes

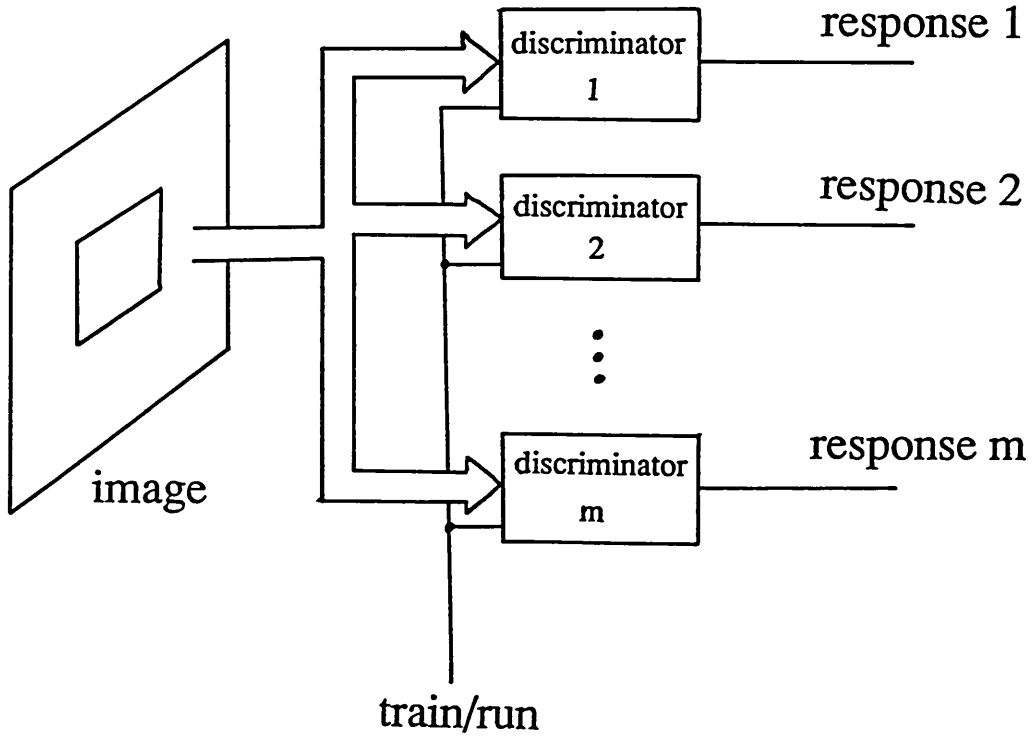
N=5 : 410 Kilobytes

N=6 : 683 Kilobytes

N=7 : 1.14 Megabytes



(a)



(b)

Figure 8.2. (a) A single discriminator.
(b) The WISARD recognition device.

N=8 : 2 Megabytes

N=16 : 256 Megabytes

For lower values of N, larger windows may be used because of the smaller memory requirements for a given window size. However, the discrimination, i.e. the ability to distinguish between different objects, improves with larger N.

The reason for using a random rather than a regular connection pattern is as follows. The WISARD system has been designed to be a general-purpose recognition device and therefore makes no assumptions about the nature of the objects which it will be trained to recognise. Any regularity in the connection pattern would correspond to assumptions about the nature of these objects. It is likely that for a given problem a carefully chosen non-random connection pattern would give better results, but flexibility and generality would then be sacrificed.

Before the training begins, the contents of the RAMs are cleared to zero. To train the WISARD, it is presented with a series of training images, each one of which possesses the particular property that it is desired the WISARD should learn to recognise. Each training image is displayed to the WISARD in turn, and the bit pattern corresponding to the image causes an address to appear on the address lines of each RAM, selecting one of the bits in each RAM. The RAMs are kept in write mode during the training process, so that for each training image one bit in each RAM is set.

Once the WISARD has been trained, it can be applied to an unknown

image. The RAMs are used in read rather than write mode, and the unknown image is put into the window. The pixels of this image will appear in the form of N-tuples at the RAM address lines, causing a value of either 0 or 1 to appear at the output of each RAM. Each RAM has been trained to have intimate knowledge of N pixel positions within the training images, and will output a one if the unknown image has the same N pixel values in these positions as any one of the training images. If, on the other hand, the RAM has not encountered the particular combination of values during training, then its output will be zero.

If the output lines of the RAMs are all fed into a summer, as shown in figure 8.2a, then the output of this summer will be a number in the range 0 to K, where K is the number of RAMs. This number is the discriminator response as described in section 8.1.

The idea is that if the unknown image is "similar" to the training set, then there will be a good chance that the RAMs will already have encountered many of the N-tuples that the unknown image produces. Several of the RAM outputs will be ones, and the output of the summer will tend to be higher than would have been the case if the unknown image were "dissimilar" from the training set. The output of the discriminator indicates a measure of the strength of correlation between the unknown image and the various training images.

The single discriminator shown in figure 8.2a can be trained to recognise one characteristic of an image. Usually it is necessary to distinguish or discriminate between two or more characteristics of the image, and for this purpose it is necessary to use more

than one discriminator. For example, to distinguish between images of oranges and of pears one would require two discriminators: one trained to recognise oranges and one trained to recognise pears.

Figure 8.2b shows a system with several discriminators. The response of each discriminator is a number ranging from 0 to K. Suppose that m discriminators have been trained, and that the WISARD is presented with an unknown image. The discriminators will produce output values $\{D_i \mid i=1..m\}$ where $0 <= D_i <= K$.

The unknown image is then classified as belonging to the class corresponding to the discriminator giving the highest response. That is to say, the unknown image belongs to the ith class, where i is chosen such that

$$D_i = \max \{ D_j \mid 1<=j<=m \}.$$

(Since the average response level increases with the number of training images used, it is important to ensure that the discriminators have been given equal amounts of training, or else to weight their responses to compensate for unequal training).

The confidence with which this classification can be given is determined by a comparison between the magnitude of the response D_i and the responses of the other discriminators.

If the discriminators have been trained to recognise pictures having various different values of some continuous parameter (for example, they may have been trained to recognise the different orientations of an object), then it is possible to determine a

more accurate value for this parameter in an unknown image by interpolating between the discriminator responses. In fingerprint processing, this could determine local ridge orientation accurately using discriminators trained to recognise a few different values of ridge orientation.

8.2.2 The CRS WISARD system [Kittler & Duff 1985]

The WISARD is available commercially from CRS Systems Ltd. The hardware consists of the following components plugged into a VMEBUS:

1. 68000 CPU with Winchester disk.
2. Boards comprising the WISARD discriminator RAM and control circuitry.
3. A board comprising the image capture and display circuitry and the image memory.

WISARD is supplied with a bootable software system which controls it and allows one to perform the basic WISARD operations such as training and running the WISARD. At extra cost, it may be purchased as a workstation unit which in addition to the above system is supplied with the Unix-like operating system Idris and ports for two serial terminals. With the workstation option, one may modify the WISARD software system or develop one's own software to control the WISARD.

The software system supplied with the WISARD was written with a particular class of applications in mind. It is geared towards

the recognition in real time of objects presented "live" to its video camera. From the point of view of its application to fingerprint processing, this software system has several disadvantages, listed below:

1. It is a separate bootable system and therefore cannot be run as a program under Idris. Instead it is necessary to reset the computer and reboot Idris every time one wishes to enter or exit from the WISARD system.

2. No provision is made for transferring images between image memory and disk file.

3. No provision is made for saving the contents of discriminator memory to file after the discriminators have been trained.

4. The system is suitable only for real-time applications where it is trained and run when presented with moving images by a video camera. No means is provided whereby training and running can be done on a still image held in the image memory.

5. It is not possible to arrange for the window to scan over the image.

6. None of the standard image processing functions such as histogram equalisation, thresholding, etc. are provided.

Because of the above disadvantages it is clear that serious research on fingerprint processing could not be done using the existing software; therefore, a program was written in C which provided the facilities missing from the supplied system, and ran

as a stand-alone task under Idris.

Unfortunately, there remained a serious problem. The CRS implementation of the WISARD hardware has a fundamental limitation which cannot be rectified in software, and which makes efficient scanning of the image by a window impossible. The problem is that the window cannot be positioned everywhere within the image, because the x and y coordinates of its top left corner can only be specified modulo 8. This means that the window can only be moved up, down, left or right by a distance equal to a multiple of 8 pixels. Therefore, only one in every 64 possible window positions in the image can be reached.

The idea of processing an image by scanning it with an adaptive window is that the window should visit each location in the image, building up an output image consisting of the discriminator response values as it travels. Because the window cannot visit each position, it is not a simple process to build up such an image. It could be done by scanning the picture, visiting all accessible locations, then shifting the picture in image store, scanning it again, and so on until all locations have been visited. However, this would be very time-consuming as the picture would have to be shifted 63 times, and the additional software would be awkward to write.

Alternatively, part of the picture could be scanned if the image were enlarged by a factor of eight and a larger window scanned over the resultant image. Some of the results presented were obtained using this approach, and others using the approach of the following section.

8.2.3 Computer simulation of WISARD

Because of the problems involved in scanning windows using the CRS WISARD, it was decided to write a software simulation of the WISARD to run on an LSI-11/73 microcomputer equipped with an image frame-store. This software simulation package was written in Pascal and is a full simulation of the operation of WISARD, with the added advantage that a full scan of the image can be done.

Because the entire program plus its data must reside within a single 64 kilobyte segment, the amount of discriminator RAM that could be simulated was limited. This restricted the usable window size to 32 by 32 pixels. This window size requires (for a tuple size of 8) 32*32/8 RAMs per discriminator, each of 256 bits. This amounts to 32 kilobytes in total if eight discriminators are provided. In cases where larger window sizes were required, the CRS WISARD and enlarged images were used.

A disadvantage with the simulation is that it runs more slowly than a hardware implementation. However, this is not as serious a problem as it might at first appear, since even the hardware implementation cannot process an entire image in real time if scanning is required. The WISARD can process the contents of the window once every two screen frame periods, i.e. 12.5 times per second. This compares with about 5 times per second which was achieved using the simulation software with a window of 32 by 32 pixels. For this window size, the hardware is thus only two and a half times as fast as the simulation. The speed advantage of

the hardware system can be expected to increase dramatically with increasing window size, however.

The N-tuple mappings, i.e. the mapping of positions within the image to RAM inputs, must be done in a random fashion. For this purpose a good pseudo random number generation algorithm [Wichmann & Hill 1982] was used. The mapping produced is random but is the same every time the system is run because the same seed values are used in the random-number generator. This means that trained discriminator memory may be stored in a file and will remain meaningful if re-loaded when the simulation is run again later.

8.3 Application to fingerprints

8.3.1 Introduction

A desirable application of the WISARD would be to discriminate one fingerprint from all others, i.e. to perform the entire AFR searching and matching operation in real time. Unfortunately, such a task is well beyond its capabilities. The effects of distortion, overinking, underinking and other defects would obscure the differences which are being sought. Even if this were not the case, one discriminator would be needed per fingerprint to be distinguished, and the amount of RAM required would rapidly become prohibitive, even on small systems.

Instead, the WISARD might be able to assist with the image processing operations that can be performed upon small windows of image data during the AFR process. In this relatively humble role it is more likely to succeed. The feasibility of using the WISARD to perform the following tasks was investigated:

- Determining the local ridge orientation as a function of position.
- Finding the position of minutiae.
- Finding the position of the cores and deltas.

8.3.2 Results

1. Local ridge orientation

For this experiment, four discriminators were trained, one each on ridges of orientation 0, $\pi/4$, $\pi/2$, and $3\pi/4$. During training, a range of different ridge spacings and displacements were used. The trained windows were then scanned over a fingerprint image, to produce the outputs shown in figure 8.3.

It can be seen that the brighter ares in each output image correspond to areas having ridges at orientations close to the orientation for which the corresponding discriminator was trained.

2. **Detection of minutiae**

A small window was used to train two discriminators, one on ridge endings and the other on bifurcations. Various orientations of training images were used. The trained window was then scanned over an image of a region of a fingerprint in an attempt to locate the minutiae within this region.

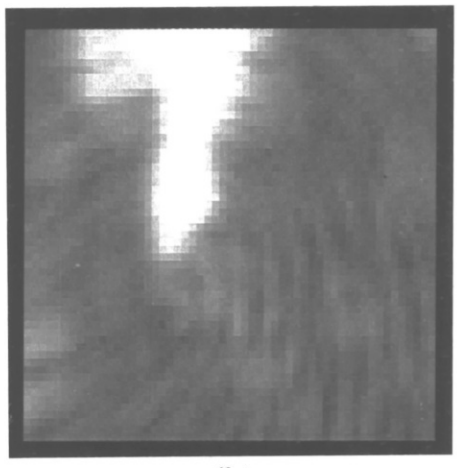
The results obtained are shown in figure 8.4. Unfortunately, the minutiae positions are not discernible above the noise.

3. **Detection of cores and deltas**

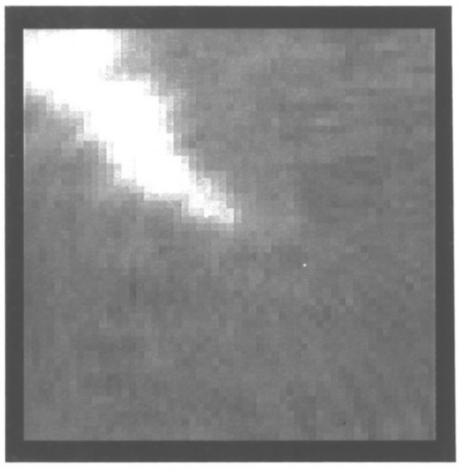
Two discriminators were trained, one on cores and the other on deltas. The training images were presented at various orientations, but were always centred within the window. The results obtained, shown in figure 8.5, are again poor, with no signal discernible above the noise.



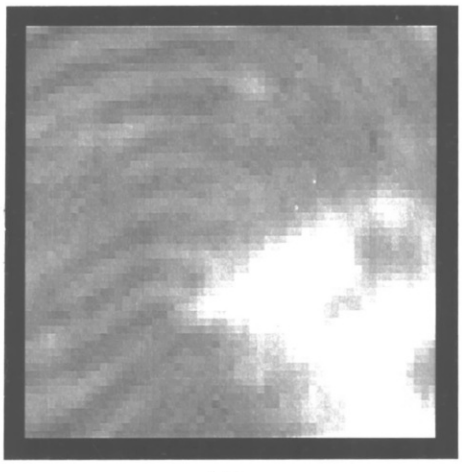
(a)



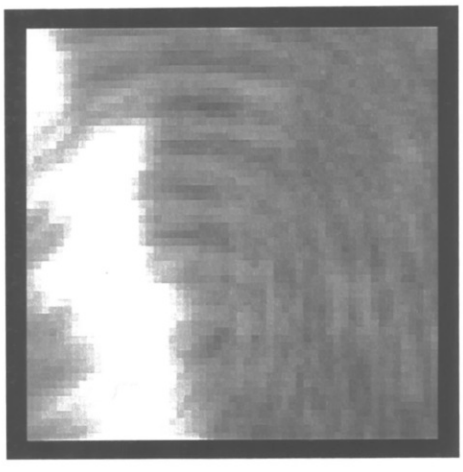
(b)



(c)

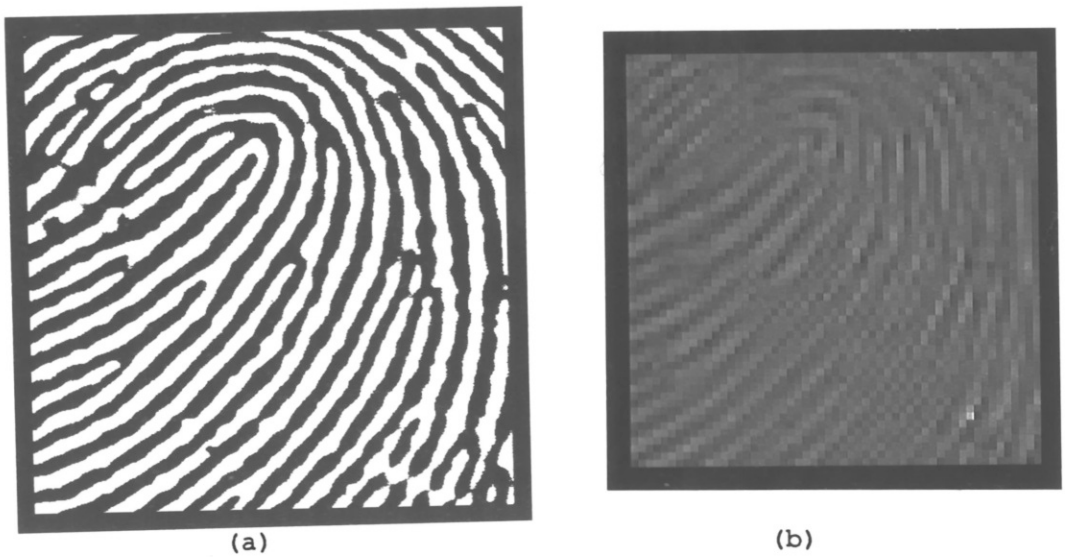


(d)



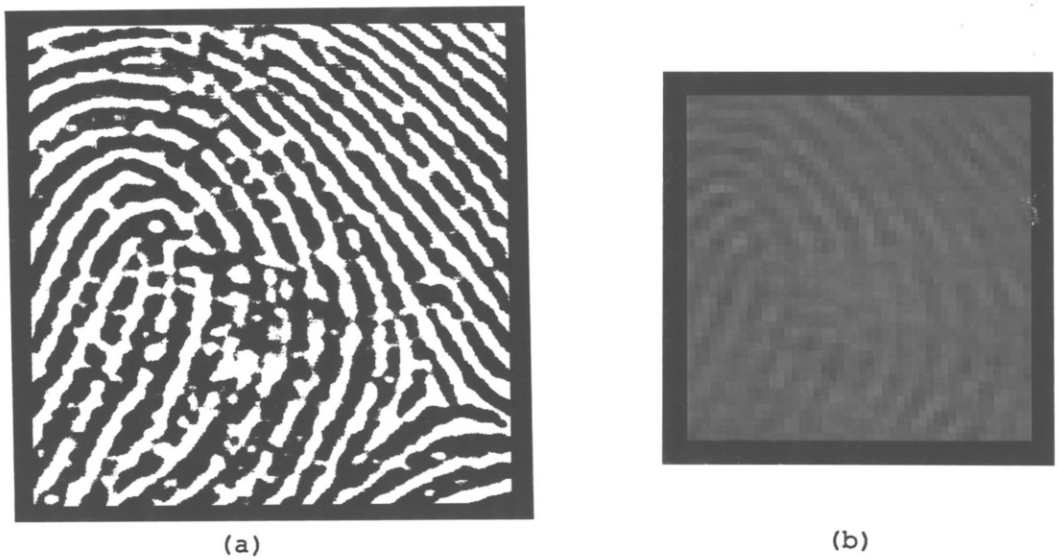
(e)

Figure 8.3. Detecting LRO using WISARD. Showing an area of fingerprint ridges (a) and the results of applying adaptive windows trained to recognise lines oriented horizontally (b), at $\pi/4$ (c), vertically (d) and at $3\pi/4$ (e).



(a) (b)

Figure 8.4. Detection of minutiae using WISARD. Showing an area of a fingerprint (a), and the result of applying a window trained on ridge endings (b). The result using a window trained on ridge bifurcations was not any clearer.



(a) (b)

Figure 8.5. Detection of singular points using WISARD. This figure shows part of a fingerprint (a), and the result of applying a window trained on cores (b). The result using a window trained on deltas was no clearer.

8.3.3 Discussion

The WISARD has performed well at detecting orientation, but poorly at detecting minutiae and singular points. Part of the reason for this is the following. The orientation within a window is a characteristic that is global within the window, so that every N-tuple potentially contributes towards the discrimination between the different orientations. On the other hand, minutiae and singular points are by their nature small in spatial extent, so that relatively few N-tuples contribute towards the discrimination.

When discriminating between different orientations, there are several positive alternatives, and it is known that at each point the "correct" answer is one of these. When detecting minutiae or singular points, however, the choice is between a positive alternative (feature present) and a negative alternative (feature absent). The existence of negative alternatives will tend to lead to less effective discrimination.

The inability to locate cores and deltas was a particular disappointment, since few good methods of locating singular points are known, and an effective method using WISARD would have been a useful contribution to fingerprint processing.

The discrimination of the WISARD is known to depend strongly upon the training set of images used, and it is sometimes found that simply retraining a window that gives poor results can lead to a considerable improvement. Therefore, it is possible that a more exhaustive evaluation in which different window sizes, tuple

sizes, coverages and training sets could lead to sets of trained windows that successfully locate minutiae or singular points. However, this research would have to be done mostly on the basis of trial and error, unless theoretical advances in the subject of optimal training of adaptive windows occur.

CHAPTER 9

CONCLUSION

9.1 General summary

Chapter 1 introduced the basic concepts involved in the study of fingerprints, and surveyed the literature in the field of computer processing of fingerprint images.

The particular suitability of the Fourier domain to the representation of fingerprint information was explained in Chapter 2, which developed position dependent Fourier filters intended for application to fingerprint images during the process of enhancement. The efficient implementation of such filters on general purpose computers was discussed in Chapter 3.

Chapter 4 and Appendix A examined the differences between the concepts of orientation and direction, concluding that the direction field of differential geometry is the mathematical concept best suited to the description of orientations in images. This was applied to produce a mathematical model giving fingerprint LRO as a spatial function in terms of the core and delta positions.

Chapter 5 described methods of determining LRO at a point. The LRO model was then applied to develop a technique for interpolating between a grid of sampled values of LRO, to yield a continuous

orientation valued function of position.

Chapter 6 combined the filtering results of Chapters 2 and 3 with the orientation results of Chapters 4 and 5, to present the enhancement algorithm which is the main result of this thesis. This algorithm was applied to several rolled and latent fingerprints, yielding enhancements which compared favourably with those produced by the successful Home Office AFR system.

Alternative implementations of the enhancement technique using optical hardware were developed in Chapter 7.

Chapter 8 described an investigation of the application of an adaptive window based hardware system known as WISARD to fingerprint processing tasks such as LRO determination and singular point location.

9.2 Contribution of this research

The primary contribution of this research is the fingerprint enhancement technique described in Chapter 6.

This enhancement technique fulfils the requirements for a good enhancement as given in Chapter 2. It produces, as it should, an image that is more suitable than the original for the purposes of identification of the individual. Most of the noise is removed by the filtering operation, as well as the ridge pore structure which is not used in the process of identification. The resulting image approximates an "ideal" representation of the structure of the ridges and their associated minutiae.

The other contributions made by this thesis were produced during the development of the enhancement algorithm, and either constitute part of it or are related to some aspect of it. The most important of these are:

1. The approach to 2-dimensional filtering which was used. This includes the following aspects:

   - Using standard classical 1-d filters to specify those aspects of filter behaviour which have 1-d analogues. This was done by expressing the filter as a separable function of the Fourier domain coordinates (ρ,ϕ), where the ρ factor is a classical filter and the ϕ factor specifies the required directional behaviour.

   - The expression of the position dependence of a nonstationary filter in terms of a set of position dependent parameters

which vary according to the local nature of the image data.

- The sampling of these parameters to produce a set of stationary filters, and hence a set of prefiltered images, from which the desired filtered image can be built up.

- The methods for reducing the number of prefiltered images required, by reduction in the number of parameters and the coarse sampling of parameters.

2. The realisation that directional information in images is projective. This means that an orientation in an image of dimension n behaves like a point within the projective space P^{n-1}. The consequent identification of the direction field of differential geometry as the appropriate concept for describing the behaviour of orientations within an image. (In particular, the use of the 2-dimensional direction field to describe LRO in fingerprints).

3. The mathematical model of fingerprint LRO, expressing the LRO at all points within the image, in terms of the singular point types and positions.

In addition to the above, this research made further contributions which are felt to be of somewhat lesser importance. These are:

1. The unwrapping technique for fingerprint LRO, making use of the model of LRO.

2. The interpolation technique for 2-dimensionally sampled orientations.

3. The methods of determining LRO at a point, which were presented in chapter 5.

4. The investigation in section 2.3.2 of the various possible functions which could describe the angular dependence of a directional filter.

5. The investigation of the use of adaptive window techniques in the processing of fingerprints.

6. The implementations of the enhancement algorithm in optical hardware, and the video electronics implementation that these suggested.

7. The Walking FFT algorithm of Appendix C.

9.3 Further work

Several potential avenues of research were not followed up because they were tangential to the main theme of the research, and it was felt that the rewards that could be expected did not justify the time required to investigate them more fully. In addition, there are several ways in which the work presented here can be extended or generalized.

There are several areas for further research, both theoretical and practical, on orientations in images.

The full implications of the theory of direction fields as applied to directionality in images have yet to be worked out. This is particularly true in the case of images of higher dimension, for which the projective spaces describing orientation become increasingly complicated geometric objects.

The theory of direction fields and the nature of the projective spaces $\mathbf{P}^n$, which are well understood by pure mathematicians, should be studied with a view to their application to image processing. This would lead to a better understanding of the concept of orientation and could lead to results of practical computational use when dealing with orientations in image processing applications.

As an example of a practical problem which might be solved, suppose that some orientation-finding algorithm has obtained orientation estimates at each pixel in a 3-dimensional image, and it is desired to smooth these orientations so that the noise caused by errors in the estimates is less objectionable. This is

essentially the problem of smoothing a 3-dimensional direction field. Vector fields can be easily smoothed by forming a vector average at each point, but this cannot be done for direction fields (except in 2 dimensions where the relationship between orientation and direction is a simple one).

Other useful results would be multidimensional analogues to the Poincare index, the associated singularity theorems and to orientation unwrapping. The latter could be applied to the problem of interpolating between sampled values of orientation in multidimensional images.

Improvements could be made to the methods of determining LRO as presented in chapter 5, by including contextual information from neighbouring estimates which have already been evaluated, and making sure that the new estimate bears an appropriate value when viewed in the context of these neighbours. The information provided by the LRO model could assist in deciding whether or not the new estimate's value is appropriate.

The LRS parameter could be included when forming the prefiltered images. This should lead to better quality enhancements, at the expense of the need to store a greater number of prefiltered images. It would be necessary to find a technique for evaluating LRS at a point. Such a technique might be based on the local Fourier transform approach of section 5.2.2.

A more thorough investigation of the use of the WISARD to process fingerprints might yield better results than those presented in chapter 8. A range of different training strategies, window

sizes, etc. should be examined. For instance, it might be possible to recognise minutiae by training windows on minutiae of fixed orientation rather than general orientation.

If a digital filter could be found which yields as its output the LRO at a point (and preferably also the LRS), this could be combined with the filters used in the enhancement method developed in this thesis. This would yield an adaptive digital filter which could enhance fingerprints directly without the need to consider the parameters separately.

This research has dealt only with the enhancement stage of fingerprint processing. The other stages, particularly encoding and matching, are also fruitful areas for research.

APPENDIX A

DIRECTIONALITY IN IMAGES

Chapter 4 discussed directionality in 2-dimensional images, with particular emphasis upon fingerprint images, and with a view to the development of a model of fingerprint LRO. In this appendix the discussion of directionality is extended to higher dimensions. The identification of the projective n-sphere and the direction field as the appropriate concepts to describe orientations is original; the mathematics itself is not. No attempt at rigour will be made, and the reader who is interested in more detail should consult the references which were given in Chapter 4.

A.1 The concept of direction

The two distinct types of direction described in chapter 4 will now be reintroduced in more general terms. We shall first introduce the concepts in two dimensions. Consider 2-dimensional Euclidean space $\mathbf{R}^2$, and suppose that an arrow or vector has been drawn with its tail at the origin, as shown in figure A.1a. This vector will form some angle θ with the Cartesian x-axis, where θ can take on any value in the range 0 to 2π radians. The angle is a periodic measure, with a period of 2π.

Suppose now that the unit circle is drawn around the origin. The unit circle is also called the 1-sphere, denoted S^1, and is described by the equation

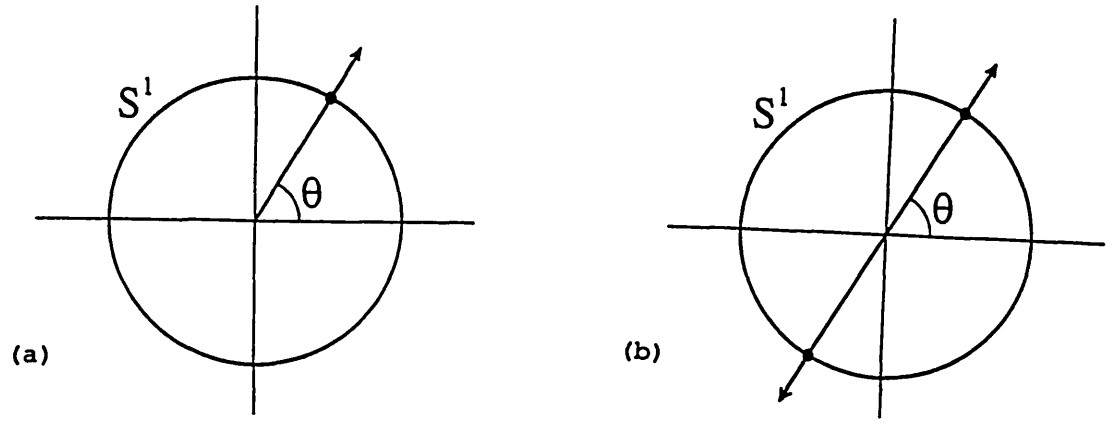


Figure A.1. Illustrating direction (a) and orientation (b) in two dimensions.

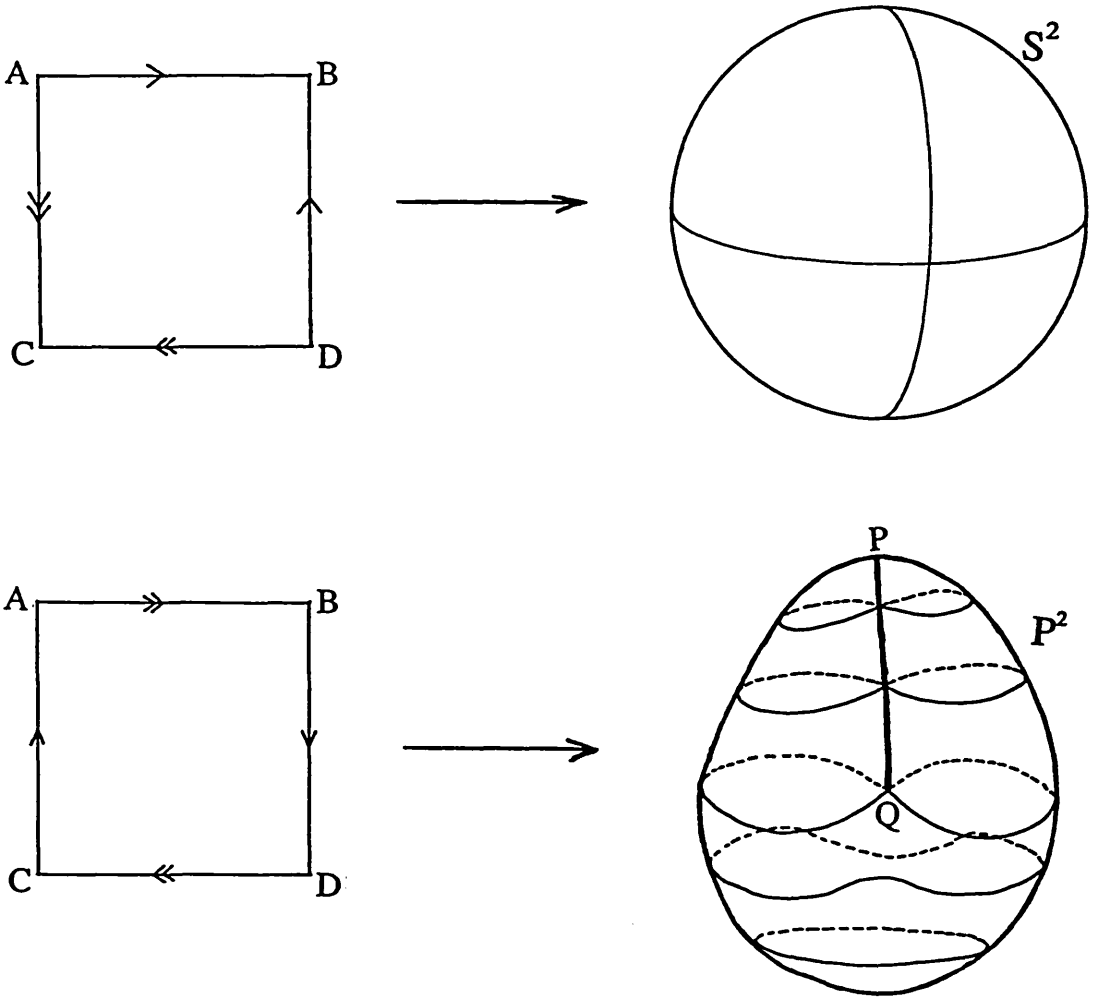


Figure A.2. The topological structure of the sphere S^2 and of the projective plane $\mathbf{P}^2$. S^2 may be produced from the square by joining ("glueing") the directed edge $\vec{AB}$ to $\vec{DB}$, and $\vec{AC}$ to $\vec{DC}$. $\mathbf{P}^2$ may be produced from the square by joining $\vec{AB}$ to $\vec{DC}$ and $\vec{CA}$ to $\vec{BD}$.

$$x^2 + y^2 = 1$$

in Cartesian coordinates. Since a vector at any angle θ will intersect the circle S^1 at one and only one point, there is a one-to-one correspondence between the different possible directions of the vector and the different elements of S^1. S^1 has circumference 2π and is the natural way to represent topologically a periodic interval of period 2π. S^1 therefore provides a natural representation for the directions of 2-dimensional vectors.

Suppose that instead of a vector, a straight line passing through the origin (see figure A.1b) is drawn. This line also forms some angle θ with the x-axis; however, this time the angle is periodic with a period of π rather than of 2π.

The line will intercept the circle S^1 at two radially-opposite points, so that a one-to-one correspondence exists between angles and pairs of radially-opposite points of S^1. The line at angle θ can therefore be represented by the pair of elements $\{\theta,\theta+\pi\}$ from S^1. A new set called the projective 1-sphere P^1 is formed by identifying the radially-opposite points of S^1. The elements of P^1 are pairs of points from S^1, each pair being regarded as an indivisible entity, i.e. a point.

The directions associated with vectors or directed lines will be termed S-directions or simply directions. The directions associated with (undirected) lines will be called P-directions or orientations.

S^1 and P^1 have identical topological structure, i.e. they are topologically equivalent or homeomorphic [Spivak 1979] to each

other. (For example, the mapping $f:P^1 \longrightarrow S^1$ given by $f(\theta) = 2\theta$ forms a topological homeomorphism between the two sets.) S^1 and P^1 are both periodic intervals, or circles, in their topology, the only difference being the period of 2π for S^1 and π for P^1. The topological equivalence between S^1 and P^1 ensures that S-directions and P-directions in two dimensions are very similar, and this has resulted in the more fundamental nature of the differences being either not noticed or ignored in much previous two-dimensional work.

Now consider the situation in 3-dimensional Euclidean space $\mathbf{R}^3$. The set of all possible directions for vectors can be given by the set of all points on the surface of a sphere, i.e. the points of the 2-sphere S^2, having equation

$$x^2 + y^2 + z^2 = 1$$

in Cartesian coordinates. A point in S^2 is often given in terms of two polar coordinates (θ,ϕ) where $0 <= \theta < 2\pi$ and $0 <= \phi <= \pi$.

On the other hand, the set of all possible orientations for lines through the origin of $\mathbf{R}^3$ is the set of all radially-opposite pairs of points from S^2. This set is called the projective 2-sphere or projective plane P^2.

The topology of S^2 is familiar as the surface of the sphere. However, the topology of P^2 is fundamentally different, and is not familiar to most non-mathematicians. P^2 is more difficult to visualize than S^2 because it can only be embedded in spaces of four or more dimensions. A representation of P^2 is shown in

figure A.2, which shows how P^2 and S^2 can be produced by joining the edges of a square region as shown. The surface representing P^2 should be regarded as "passing through" itself along PQ without self-intersection. The topology of P^2 is described in greater detail by Berger [1977].

It follows that the behaviour of P-directions (orientations) in 3 dimensions is completely different from the S-directional behaviour of vectors. In order to understand and deal computationally with orientations in 3 dimensions (e.g. to smooth a set of orientations, or to interpolate between sampled orientation values), it is necessary to gain an understanding of the properties of P^2. The detailed solution to such problems was not investigated in this thesis because the orientations encountered in fingerprints are always 2-dimensional. Because of the identical topological structures of S^1 and P^1, ways can be found to treat 2-dimensional orientations as if they were directions.

Knutsson [1985] found a continuous and distance-preserving representation of 3-dimensional orientations in terms of 5-dimensional vectors, thereby effectively defining an embedding of P^2 into $\mathbf{R}^5$. This representation is presented in the form of a mapping from $\mathbf{R}^3$ to $\mathbf{R}^5$ for which radially opposite points of $\mathbf{R}^3$ have the same image in $\mathbf{R}^5$. The mapping used is:

$$s = r \sin^2 \theta \cos 2\phi$$

$$t = r \sin^2 \theta \sin 2\phi$$

$$u = r \sin 2\theta \cos \phi$$

$$v = r \sin 2\theta \sin \phi$$

$$w = r \sqrt{3} (\cos^2 \phi - 1/3)$$

where (r,θ,ϕ) is the point in $\mathbf{R}^3$ in polar coordinates, and (s,t,u,v,w) is the point in $\mathbf{R}^5$ in Cartesian coordinates.

Although this mapping allows 3-dimensional orientations to be represented as 5-dimensional vectors, it does not permit orientation-valued computations (such as interpolation) to be done, since the mapping does not form a homeomorphism and an inverse mapping does not exist.

The situation is fully analogous in higher dimensions. In $\mathbf{R}^n$, directions are elements of the (n-1)-sphere S^{n-1} given by the equation

$$x_1^2 + x_2^2 + \ldots + x_n^2 = 1$$

and orientations are elements of the (n-1)-projective space P^{n-1} defined as the collection of all sets $\{p,-p\}$ for $p \in S^{n-1}$. Since orientations are obtained as pairs it is intuitively clear that there are "twice as many" available directions as orientations.

In one dimension, the situation is trivial. S^0 consists of two elements, -1 and +1, corresponding to forward and reverse senses along the real line $\mathbf{R}^1$. P^0 has only one element, $\{-1,+1\}$, with the result that orientations are not of any significance in one dimension.

A.2 Directions in images

In Chapter 4 it was seen that orientations rather than directions are most appropriate for the description of 2-dimensional image directionality. This is equally true in higher dimensional images. (Indeed, it is true within any manifold M (see A.3.1), upon which an image may be defined to be a real valued function $f:M \rightarrow \mathbf{R}$).

The orientation of local anisotropy within images (of any dimensionality) is given by the orientation of a line segment through the point in question. The orientation of a line segment is only determined up to a rotation of π radians about any axis perpendicular to the line segment, since such a rotation will transform the line segment into itself.

The projective nature of image directionality can be seen in the Hermite symmetry exhibited by the Fourier transform of any real-valued function (such as an n-dimensional image) $f(x_1,\ldots,x_n)$:

$$F(u_1,\ldots,u_n) = F^*(-u_1,\ldots,-u_n)$$

This shows that radially opposite points in the Fourier domain are complex conjugates of one another, such that information in the direction of the vector $\underline{u} = (u_1,\ldots,u_n)$ cannot be distinguished from that in the direction of the vector $-\underline{u}$.

The Fourier domain can be described in terms of generalised polar coordinates r and θ where $r \in \mathbf{R}$ and $\theta \in S^{n-1}$, such that

$$r = \sqrt{(u_1^2 + u_2^2 + \ldots + u_n^2)}$$

is the magnitude and $\theta \in S^{n-1}$ is the direction of the vector $\underline{u}$. Then the Hermite symmetry can be expressed as

$$F(r,\theta) = F^*(r,-\theta) \text{ for all } r \in \mathbf{R}^+ \text{ and } \theta \in S^{n-1}$$

(where $\mathbf{R}^+$ is the set of all nonnegative reals).

Since the directions θ and $-\theta$ are equivalent to one another, it follows that they could be combined together and the Fourier transform of a real-valued function be described in terms of P^{n-1} rather than S^{n-1}. The new Fourier domain obtained by this approach would be $\mathbf{R}^+ \times P^{n-1}$ topologically rather than the usual $\mathbf{R}^n$ (which is $\mathbf{R}^+ \times S^{n-1}$). However, the topology of $\mathbf{R}^+ \times P^{n-1}$ is less familiar and hence harder to understand than that of $\mathbf{R}^n$, and in any case, the normal approach would be needed when dealing with transforms of (non-image) complex-valued functions, for which directions are appropriate.

A.3 Differential geometry and the direction field.

A.3.1 The manifold

The vector and direction fields, which were presented in chapter 4 in two dimensions, are concepts of a branch of mathematics known as differential geometry. Differential geometry describes general multidimensional curved spaces called manifolds on which vector and direction fields may be defined. The curved surface of the skin of the fingers, the direction spaces S^n and P^n, the

Euclidean plane $\mathbf{R}^2$ and Euclidean 3-space $\mathbf{R}^3$ are all examples of manifolds.

An n-dimensional manifold M is a space which is locally similar to the flat Euclidean space $\mathbf{R}^n$ in the sense that for every point x in M it is possible to define a homeomorphism from a neighbourhood of x to $\mathbf{R}^n$. This means that a manifold is locally flat everywhere (within small enough neighbourhoods) even though as a whole it may be curved. A good example for visualization purposes is the surface of the earth, which was for centuries believed to be flat despite its overall spherical shape, because locally the curvature is too small to be easily noticeable.

One of the most important applications of differential geometry is General Relativity, in which manifolds of dimension 4 are used as models of the global structure of the universe [Hawking & Ellis 1973].

A two-dimensional manifold is called a surface. Discussions of directionality in image processing usually take place within the context of $\mathbf{R}^2$, $\mathbf{R}^3$ or surfaces embedded within $\mathbf{R}^3$.

An image on a manifold M may be defined as a real valued function $f:M\rightarrow\mathbf{R}$. At each point $x \in M$ the "pixel value" of the image is given by $f(x)$.

The condition that an n-dimensional manifold be locally flat is equivalent to being able to define for each point x in M a flat space, denoted T_xM, known as the tangent space at x. (In two dimensions these are known as tangent planes to the surface). This is illustrated in figure A.3. The tangent bundle of M,

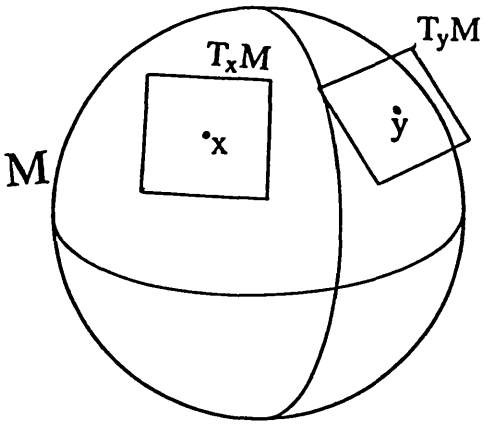


Figure A.3. A manifold M with tangent spaces T_xM and T_yM at the points x and y.

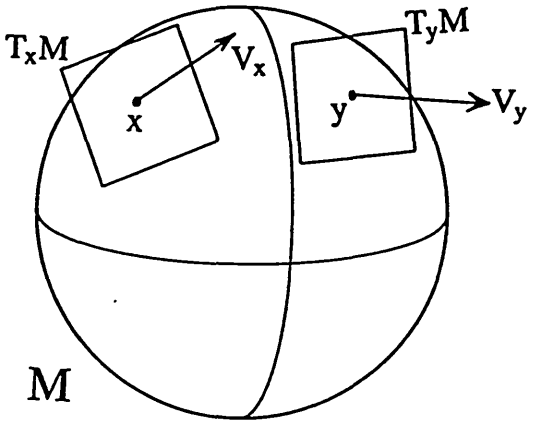


Figure A.4. A vector field on a manifold. Vectors are shown at the points x and y.

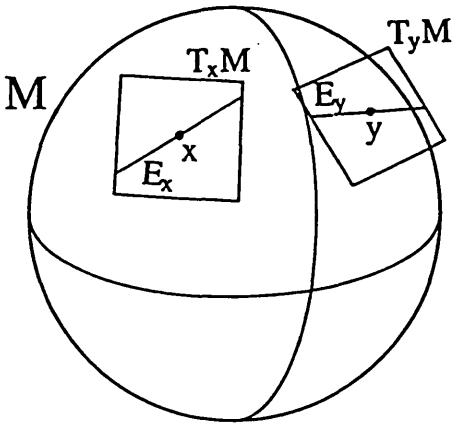


Figure A.5. A direction field on a manifold. Lines are shown at the points x and y.

denoted TM, is the collection of all tangent spaces for all points of M, and can be identified with the Cartesian product of M and R^n. It is in terms of the tangent bundle that vector and direction fields are defined.

A.3.2 **Vector and direction fields**

A vector field V on a manifold M is a subbundle of the tangent bundle defined by specifying in a continuous fashion a point V_x in T_xM for each x in M.

The vector field can be represented geometrically by an arrow in each tangent space, with the tail at the origin of T_xM and the head at the point V_x. This is one reason why it is necessary to have a tangent space to define a vector: the vector always lies in the tangent space and not in the manifold, since it "protrudes" due to the curvature of the manifold. (See figure A.4).

A vector field can be specified by its magnitude and direction. The magnitude is a function from M to R^+, which is continuous everywhere except at isolated points known as the singular points of V. The direction is a function from M to S^{n-1} which is continuous everywhere except where the magnitude is zero or where V is singular.

A direction field (or line element field) E on a manifold M is a subbundle of the tangent bundle, defined by specifying in a continuous manner a line E_x through the origin of T_xM for each x in M. (See figure A.5).

A direction field can equivalently be regarded as a continuous function from M to P^{n-1}, specifying an orientation or P-direction at each point in the manifold. Again, there may be isolated singular points at which the function is discontinuous.

As in chapter 4, an integral curve of a vector or direction field is a curve C to which the field is tangential at every point. Similarly to the 2-dimensional case, the integral curves of a vector field are always orientable, whereas those of a direction field may or may not be. In particular, the following results are true in general:

1. Any vector field V on an n-dimensional manifold M induces an orientable direction field E on M having the same integral curves as V. For each x in M the orientation of E at x is given by the element $\{\theta,-\theta\}$ of P^{n-1}, where $\theta \in S^{n-1}$ is the direction of V at x.

2. Any orientable direction field E induces two equivalence classes of vector fields having the same integral curves as E. These two classes are referred to as the orientations of E, as in chapter 4.

A.3.3 The Poincare index

The concept of the Poincare index, which was defined in Chapter 4 for directions in $\mathbf{R}^2$, can be generalized to such parts of a surface as can be homeomorphically mapped onto a subset of $\mathbf{R}^2$ (as, for example, the surface of the fingertip is "mapped" onto the paper when a rolled fingerprint impression is taken). Given such

a region A and the homeomorphism $f:A \to \mathbf{R}^2$, the Poincare index of any point p of A is defined to be the Poincare index in $\mathbf{R}^2$ of the homeomorphic image f(p) of p. Similarly, the index of a curve C in A is defined as the index in $\mathbf{R}^2$ of the image f(C) of the curve. These definitions are independent of the particular homeomorphism chosen.

APPENDIX B

ONE DIMENSIONAL INTERPOLATION

Three interpolation methods will be described here: linear interpolation, cubic spline interpolation and Monro-Chebyshev interpolation. Each of these techniques is applicable to interpolation between data values at a set $\{x_n\}$ of equally spaced points.

B.1 Linear interpolation

Given two consecutive samples $f(x_n)$ and $f(x_{n+1})$, linear interpolation finds intermediate values for $x_n < x < x_{n+1}$ by approximating $f(x)$ using a straight line joining the two sample function values. That is, it uses

$$f(x) = f(x_n) + \frac{f(x_{n+1}) - f(x_n)}{x_{n+1} - x_n} (x - x_n) \text{ for } x_n<x<x_{n+1}.$$

Linear interpolation therefore produces a continuous function which is piecewise linear, but not differentiable at the sample points $\{x_n\}$. It has the advantages of conceptual simplicity and speed, and could also cope with irregularly-spaced data if necessary. Its disadvantages are that its derivative is discontinuous at the sample points, and that it is inaccurate in regions where the function is rapidly varying, since only two data

points are used in determining the function value.

B.2 Cubic spline interpolation [Clarke 1984]

Given data values for a 1-dimensional set of equally-spaced points, the cubic spline interpolation technique fits a cubic polynomial between the data values for each pair of adjacent points, in such a manner that C^1 smoothness of the resulting function is ensured everywhere.

Consider a representative pair of adjacent data points f(0) and f(1) as shown in figure B.1.

The cubic polynomial to be fitted through these points is

$$f(x) = ax^3 + bx^2 + cx + d$$

where a,b,c and d have to be determined.

The derivative of f(x) is

$$f'(x) = 3ax^2 + 2bx + c.$$

At the endpoints x=0 and x=1, f(x) and f'(x) take on the following values:

$$f(0) = d$$

$$f(1) = a + b + c + d$$

$$f'(0) = c$$

$$f'(1) = 3a + 2b + c$$

Rearranging these four equations yields

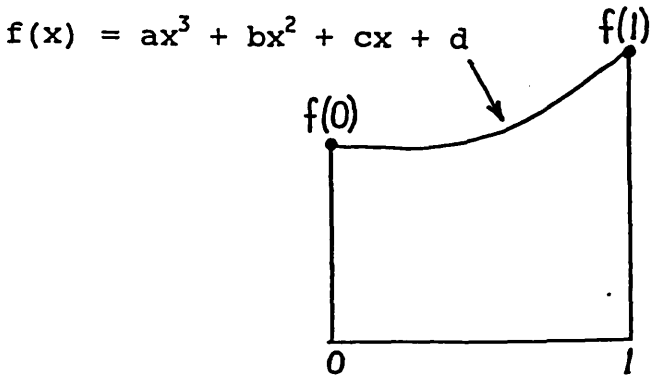


Figure B.1. Cubic spline interpolation.

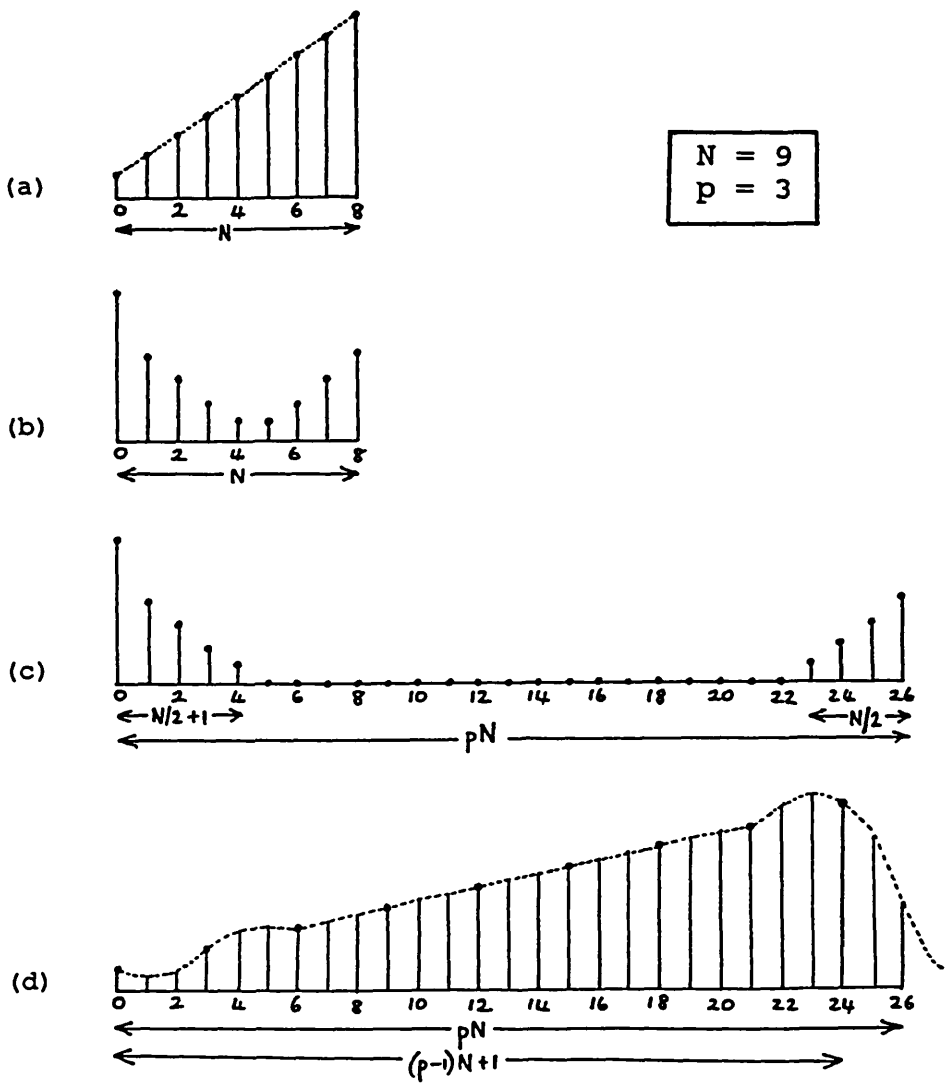


Figure B.2. Fourier interpolation

(a) Original data.
(b) Fourier transformed data.
(c) Padded with zeros.
(d) After inverse transformation. The original data points are marked with solid dots.

$$a = f'(1) + f'(0) - 2[f(1) - f(0)]$$

$$b = 3[f(1) - f(0)] - f'(1) - 2f'(0)$$

$$c = f'(0)$$

$$d = f(0),$$

expressing the coefficients of the cubic as functions of the endpoint values and derivatives.

The endpoint values are of course known, and the endpoint derivatives can be estimated from the data values using the following 5-point difference formula for numerical differentiation:

$$f'_n = 1/12 \; (\; f_{n-2} - 8f_{n-1} + 8f_{n+1} - f_{n+2} \;)$$

This technique is shown in [Clarke 1984] to be relatively insensitive to errors in the derivative estimates. It has the advantage over linear interpolation of C^1 smoothness, and is far more computationally efficient than the Monro-Chebyshev technique discussed next.

B.3 Monro-Chebyshev interpolation

This technique performs interpolation of regularly-spaced data using a signal processing approach based upon the use of the discrete Fourier transform.

Monro-Chebyshev interpolation is based upon Fourier interpolation [Schafer & Rabiner 1974]. To perform Fourier interpolation, take the DFT of the data, extend the transform by the insertion of zeros above the component of highest frequency, and then take the

inverse DFT of this extended data to yield the interpolated form of the data. This is illustrated in figure B.2. The Fourier transform interpolation technique has the disadvantage that severe ringing is encountered (particularly near the endpoints) if the data to be interpolated is undersampled.

Monro [1979,1980] demonstrated that if the data to be interpolated is first suitably extended to give it Hermite symmetry, then the resultant interpolation will have the properties of interpolation by Chebyshev series of the first or second kind. Interpolation by Chebyshev polynomial provides minimum maximum error and evenly distributed ripple with undersampled data.

Figure B.3 illustrates the Monro-Chebyshev interpolation procedure using series of the first kind. The algorithm is given below, to interpolate a set $\{f_n\}$ of N data points by a factor p:

1. Enter the given values f_n into the first N locations of an array A of length 2N. (See figure B.3a).

2. Produce an extended data set of length 2N and having Hermitian symmetry, by repeating all the data in reverse order in the upper half, as follows:

$$A_{2N-n-1} = A_n \quad \text{for } n=0 \text{ to } N-1.$$

(See figure B.3b).

3. Compute the DFT of length 2N of the array A. (See figure B.3c).

4. Transfer the coefficients of the transform to an array B of

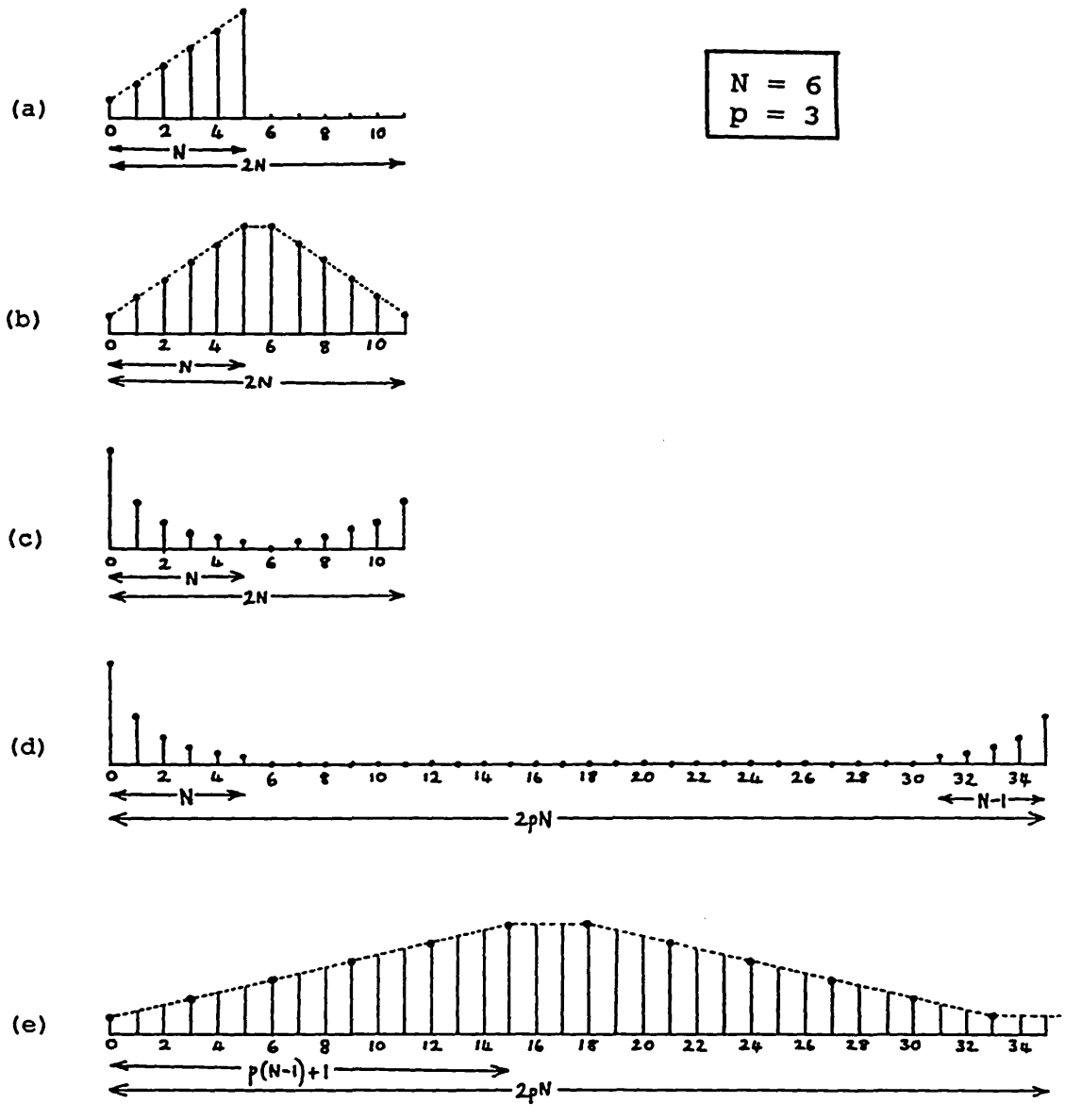


Figure B.3. Monro-Chebyshev interpolation (First Kind)

(a) Original data.
(b) Extended to length 2N with Hermitian symmetry.
(c) After DFT is taken
(d) Padded with zeros
(e) After inverse DFT is taken. The original data points are marked with solid dots.

length 2pN as illustrated in figure B.3d, such that

$$B_n = \begin{cases} A_n & \text{for } n=0 \text{ to } N-1 \\ 0 & \text{for } n=N \text{ to } (2p-1)N \\ A_{n-2(p-1)N} & \text{for } n=(2p-1)N+1 \text{ to } 2pN-1 \end{cases}$$

5. Perform the inverse 2pN point DFT of the array B. The required interpolation then consists of the first $(N-1)p + 1$ values of the array, B_0 to $B_{(N-1)p}$. The values B_{kp} for k=0 to N-1 will be the original data and the interpolated values will be situated between them in the correct order. (See figure B.3e).

The algorithm for interpolation by Chebyshev polynomial of the second kind differs in the manner in which the data values are made Hermitian. It is listed below:

1. Enter the given values f_n into the first N locations of an array A of length 2N-2. (See figure B.4a).

2. Produce an extended data set of length 2N-2 and having Hermitian symmetry, by repeating all the data except for the points A_0 and A_{N-1} in reverse order in the upper half, as follows:

$$A_{2N-2-n} = A_n \quad \text{for } n=1 \text{ to } N-2.$$

(See figure B.4b).

3. Compute the DFT of length 2N-2 of the array A. (See figure B.4c).

4. Transfer the coefficients of the transform to an array B of

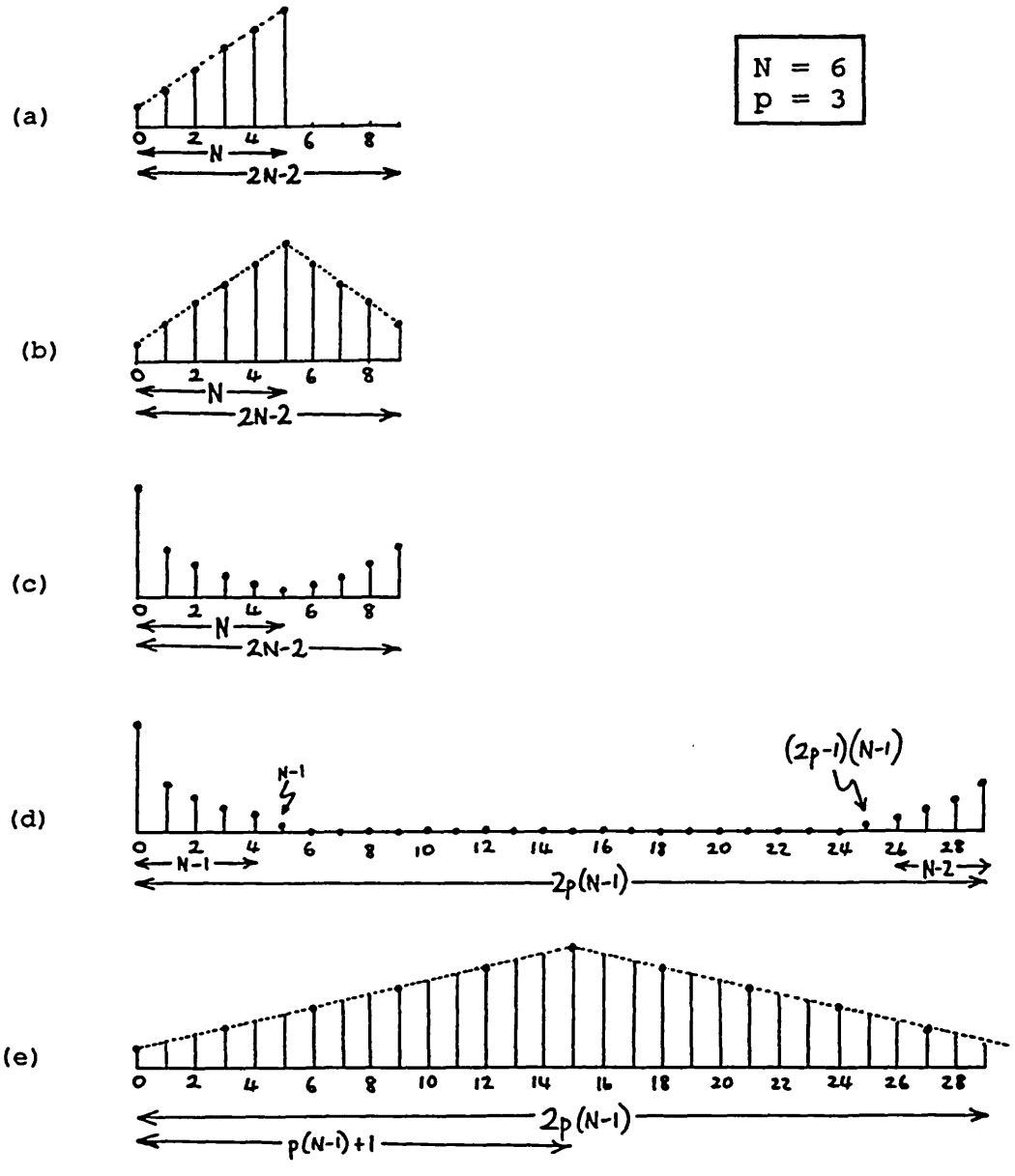


Figure B.4. Monro-Chebyshev interpolation (Second Kind)

(a) Original data.
(b) Extended to length 2N-2 with Hermitian symmetry.
(c) After DFT is taken.
(d) Padded with zeros.
(e) After inverse DFT is taken. The original data points are marked with solid dots.

length 2p(N-1) as illustrated in figure B.4d, such that

$$B_n = \begin{cases} A_n & \text{for } n=0 \text{ to } N-2 \\ \frac{1}{2}A_{N-1} & \text{for } n=N-1 \\ 0 & \text{for } n=N \text{ to } (2p-1)(N-1)-1 \\ \frac{1}{2}A_{N-1} & \text{for } n=(2p-1)(N-1) \\ A_{n-2(p-1)N} & \text{for } n=(2p-1)(N-1)+1 \text{ to } 2p(N-1)-1 \end{cases}$$

5. Perform the inverse 2p(N-1) point DFT of the array B. The required interpolation then consists of the first (N-1)p + 1 values of the array, B_0 to $B_{(N-1)p}$. The values B_{kp} for k=0 to N-1 will be the original data and the interpolated values will be situated between them in the correct order. (See figure B.4e).

In the above algorithms it is necessary to perform DFTs of lengths 2N and 2pN, or 2N-2 and 2p(n-1). These may be performed efficiently using the chirp z-transform (CZT) algorithm [Rabiner et al 1969, Monro & Branch 1977] which can obtain the DFT of data of arbitrary length n in O(n ln n) operations.

This technique can produce excellent results, but involves considerably more computation than the other two methods. It is felt on balance that the cubic spline method would usually provide the best compromise between quality and computational effort when linear interpolation is found to be inadequate.

APPENDIX C

THE WALKING FFT ALGORITHM

Because of its simplicity, it is quite possible that the Walking FFT algorithm has been thought of by other researchers. However, a literature search did not yield any reference to it, and it is therefore believed that it has not been published. For this reason it is presented below.

C.1 Introduction

The Walking FFT algorithm was developed while investigating the moving window approach to filtering, which was described in section 3.3. Provided that the contents of the window are not modified by a windowing function, then as the window moves one pixel to the right (say), the relationship between the window contents before and after the move is the following. The data within the window has shifted one column to the left, removing the leftmost column of data, and a new column of data has moved into the rightmost column. It is reasonable to ask whether there is a method whereby the Fourier transform of the data can be "updated" for each new window position to reflect the new window contents, using less computation than would be necessary to evaluate the transform of the new data directly using the FFT algorithm.

It turns out that this can be done easily, using the "Walking FFT"

technique which reduces the computational order of a k by k transform from $k^2 \log_2 k$ to k^2. The Walking FFT will for simplicity first be described for one-dimensional data.

C.2 The Walking FFT in one dimension

Suppose that we have data points f(0), f(1), ..., as shown in figure C.1a, and that the k-point Fourier transform F(u), u=0,...,k-1, of the first k points of this data has already been evaluated.

Suppose also that it is desired to "walk" the transform one point to the right in order to get the transform of the k points f(1),...,f(k). Call these points $f_{new}(0),...,f_{new}(k-1)$, where

$$f_{new}(x) = f(x+1) \ , \ x=0..k-1.$$

The desired updated transform $F_{new}(u)$ can be obtained from the original F(u) as follows.

Rotating the data f(0),...,f(k-1) one position to the left yields the data $f_{rot}(x)$ as illustrated in figures C.1b and c, where

$$\begin{aligned} f_{rot}(x) &= f(x+1) \quad \text{if } x<k-1 \\ &= f(0) \quad \text{if } x=k-1 \end{aligned}$$

The transform of this is obtained using the shift theorem

$$F_{rot}(u) = F(u).\exp(j2\pi u/k)$$

Now $f_{rot}(x)$ is equal to the desired function $f_{new}(x)$ at all points except x=k-1; therefore

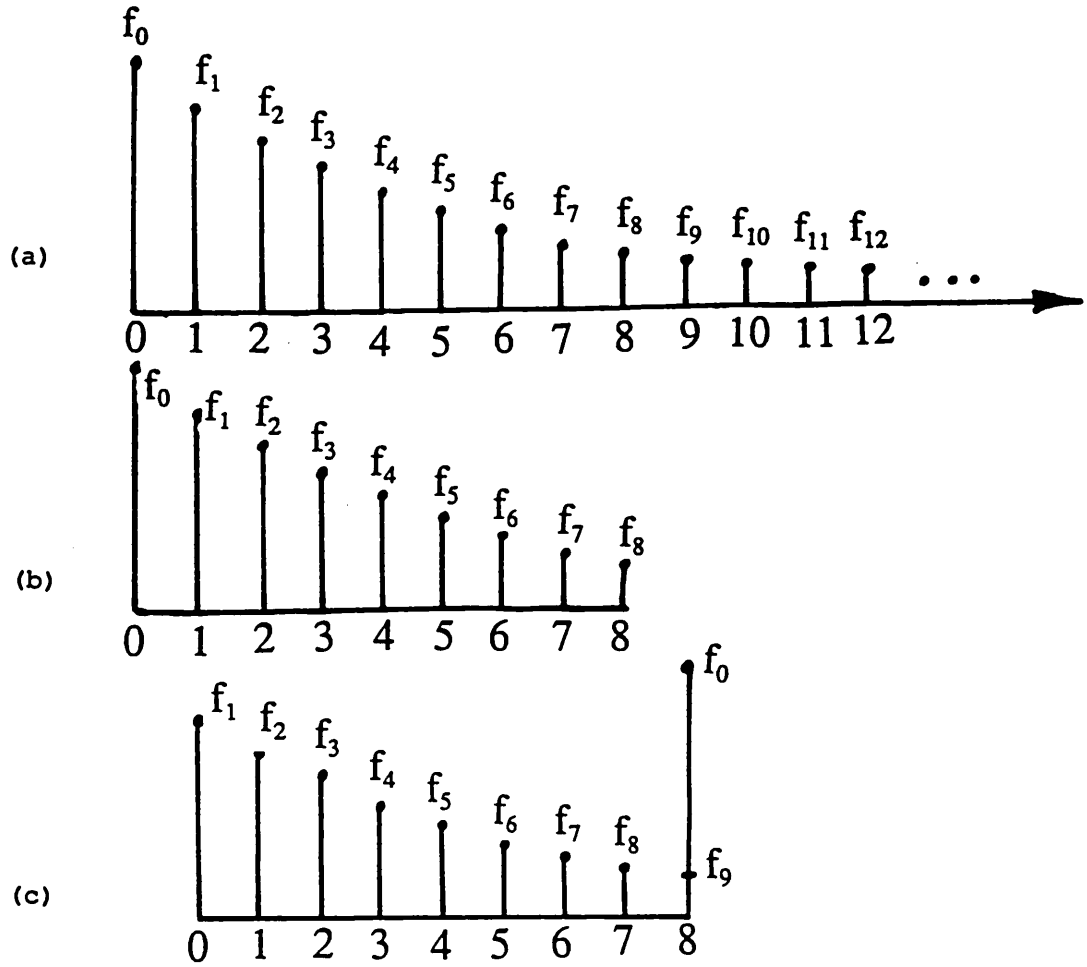


Figure C.1. The Walking FFT in one dimension. This is illustrated for a transform of length 9.

(a) A set of data points.
(b) The first 9 data points.
(c) These 9 points after rotation one position to the left.

$$f_{new}(x) = f_{rot}(x) + f_{fixup}(x)$$

$$\text{where } f_{fixup}(x) = \begin{cases} 0 & \text{if } x<k-1 \\ f(k) - f(0) & \text{if } x=k-1 \end{cases}$$

From the definition of the DFT, we have

$$F_{fixup}(u) = 1/k \sum_{x=0}^{k-1} f_{fixup}(x) \exp [-j2\pi ux/k]$$

$$= 1/k \ (f(k) - f(0)) \exp [-j2\pi u(k-1)/k]$$

$$= 1/k \ (f(k) - f(0)) \exp [j2\pi u/k]$$

Therefore

$$F_{new}(u) = \left[F(u) + \frac{f(k) - f(0)}{k} \right] \exp(j2\pi u/k)$$

This formula describes how to update a 1-d Fourier transform by the inclusion of the new point f(k) and the removal of the old f(0). Note that the number of operations required is of order k rather than k $\log_2$ k.

The following pseudocode segment illustrates how to code the algorithm on the computer, assuming that the exponentials have been precomputed.

```
for u=0 to k-1 do
  Add (f(k) - f(0))/k to F(u)
  Multiply F(u) by exp(j2πu/k)
endfor
```

C.3 The Walking FFT in two dimensions

We shall discuss the case where it is desired to modify the Fourier transform of k by k data points by shifting the window one point to the right, thereby including a new column of data. The equivalent result for including a new row rather than a column can easily be derived in a similar way.

Suppose that the Fourier transform of k by k data points

$$F(u,v) = DFT\{ f(x,y) \} \quad x,y = 0,\ldots,k-1$$

has already been evaluated and it is desired to include a new (k-1)st column, shifting the rest of the data one pixel to the left.

First, rotate the data left rather than shifting it:

$$f_{rot}(x,y) = \begin{cases} f(x+1,y) & \text{for } x<k-1 \text{ and all } y \\ f(0,y) & \text{for } x=k-1 \text{ and all } y \end{cases}$$

Therefore, using the shift theorem,

$$F_{rot}(u,v) = F(u,v)\ \exp(j2\pi u/k)$$

As for the 1-d case we have

$$f_{new}(x,y) = f_{rot}(x,y) + f_{fixup}(x,y)$$

$$\text{where } f_{fixup}(x,y) = \begin{cases} 0 & \text{for } x<k-1 \\ f(k,y) - f(0,y) & \text{for } x=k-1 \end{cases}$$

Now the Fourier transform of $f_{fixup}(x,y)$ is:

$$F_{fixup}(u,v) = 1/k \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} f_{fixup}(x,y) \exp(-j2\pi(ux+vy)/k),$$

by the definition of the 2-d DFT.

$$= 1/k \sum_{y=0}^{k-1} (f(k,y)-f(0,y)) \exp(-j2\pi(u(k-1)+vy)/k),$$

after substituting $f_{fixup}(x,y)$.

$$= 1/k \exp(j2\pi u/k) \sum_{y=0}^{k-1} f_o(y) \exp(-j2\pi vy/k),$$

where we have defined $f_o(y) = f(k,y) - f(0,y)$.

$$= \exp(j2\pi u/k) \cdot F_o(v),$$

where $F_o(v)$ is the 1-d DFT of $f_o(y)$.

Therefore,

$$F_{new}(u,v) = \exp(j2\pi u/k) \; [F(u,v) + F_o(v)]$$

where $f_o(y) = f(k,y) - f(0,y)$.

The computation involved in implementing this, assuming that the exponentials are pre-calculated, is one 1-d k-point FFT, k^2 multiplications and k^2 additions. This gives it an overall complexity of order k^2 operations, which is an improvement over the $k^2 \log_2 k$ that would be required for direct implementation.

A pseudocode algorithm for implementation is given below.

```
{Evaluate F_o(v)}
for y=0 to k-1 do
  f_o[y] := f[k] - f[0]
endfor
Take the 1-d FFT of f_o.

for u=0 to k-1 do
  Add the vector F_o(v) to the uth column of F(u,v).
  Multiply each element of the uth column of F(u,v) by the
  number exp [j2πu/k].
endfor
```

REFERENCES

ABRAMATIC J.F. & SILVERMAN L.M. 1979. Non-stationary linear restoration of noisy images. Proceedings of IEEE Conference on Decision and Control, Fort Lauderdale, Florida, pp. 92-99.

ALEKSANDER, I. 1983a. Emergent intelligent properties of progressively structured pattern recognition nets. Pattern Recognition Letters, Vol. 1, July 1983, pp. 375-384.

ALEKSANDER, I. 1983b. Emergent intelligence from adaptive processing systems. Chapter 12 of Aleksander, I.(Ed.): Artificial Vision for Robots, Chapman and Hall, pp. 215-233.

ALEKSANDER, I. 1985. WISARD: A Component for Image Understanding Architectures. Chapter 9 of Kittler, J. & Duff, M.J.B. (Eds): Image Processing System Architecture, Research Studies Press, John Wiley.

ALEKSANDER, I. 1987. Artificial intelligence for production engineering: a historical approach. Robotica, Vol. 5, pp. 99-110.

ALEKSANDER, I. & STONHAM, T.J. 1979. Guide to pattern recognition using random-access memories. IEE Computers and digital techniques, Vol. 2, No. 1, February 1979, pp. 29-40.

ALEKSANDER, I., THOMAS, W.V., & BOWDEN, P.A. 1984. WISARD - a radical step forward in image recognition. Sensor review, Vol. 4, No. 3, July 1984, pp. 120-124.

ALEKSANDER, I. & WILSON, M.J.D 1985a. Adaptive windows for image processing. IEE Proceedings, Vol. 132, Pt.E, No. 5, September 1985, pp. 233-245.

ALEKSANDER, I. & WILSON, M.J.D. 1985b. Prospects for Adaptive Window Architectures. Optical Processing Conference, Cannes 1985.

ANDERSON, G.L. & NETRAVALI, A.N. 1976. Image Restoration Based on a Subjective Criterion. IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-6, No. 12, December 1976, pp. 845-853.

ANDREWS, H.C. 1974. Digital Image Restoration: A Survey. IEEE Computer, Vol. 7, No. 5, May 1974, pp. 36-45.

ARCELLI, C. & DI BAJA, G.S. 1985. A Width-Independent Fast Thinning Algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-7, No. 4, July 1985, pp. 463-474.

ARCELLI, C., CORDELLA, L.P., & LEVIALDI, S. 1981. From Local Maxima to Connected Skeletons. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-3, No. 2, March 1981, pp. 134-143.

ARNOLD, V.I. 1973. Ordinary Differential Equations. MIT Press.

ARNOLD, V.I. 1981. Singularity theory. Cambridge University Press.

ARNOLD, V.I. 1983. Geometric methods in the theory of ordinary differential equations. Springer Verlag.

ARNOLD, V.I., GUSEIN-ZADE, S.M. & VARCHENKO, A.N. 1985. Theory of singularities of differentiable maps, Vol. 1. Birkhauser Verlag.

BATES, P.J. 1978. Fingerprints. M.Sc./D.I.C. thesis, Department of Mathematics, Imperial College, London, September 1978.

BECK, A. 1974. Continuous Flows in the Plane. Die Grundlehren der Mathematischen Wissenschaften in Einzeldarstellungen, Band 201, Springer Verlag.

BERGER, M. 1977. Geometry. Springer Verlag (2 volumes).

BEWSHER, M.R. 1967. A topological approach to fingerprint classification. National Symposium on Law Enforcement Science & Technology, pp. 479-480.

BOOTHBY, W.M. 1951a. The topology of regular curve families with multiple saddle points. American Journal of Mathematics, Vol. 73, pp. 405-438.

BOOTHBY, W.M. 1951b. The topology of the level curves of harmonic functions with critical points. American Journal of Mathematics, Vol. 73, pp. 512-538.

BRIDGES, B.C. 1942. Practical fingerprinting. Revised by C.E. O'Hara. Funk and Wagnalls 1963.

CARPENTER, R.H.S. 1984. Neurophysiology. Edward Arnold, London.

CHANG, T.L. 1980. Texture analysis of digitised fingerprints for singularity detection. IEEE Pattern Recognition conference, Miami Beach, Florida, 1-4 December 1980, pp. 478-480.

CHERRILL, F.R. 1954. The Fingerprint System at Scotland Yard. HMSO, London.

CHIRALO, R.P. & BERDAN, L.L. 1978. Adaptive digital enhancement of latent fingerprints. Proceedings of 1978 Carnahan conference on crime countermeasures, Lexington, Kentucky, pp. 131-135.

CLARKE, M. 1984. Evaluation of ECG surface maps from stress exercise tests in the diagnosis of coronary artery disease. PhD thesis, Department of Electrical Engineering, Imperial College, University of London, June 1984.

CLERICI, R. 1979. Fingerprints: A new classification scheme. Nature, Vol. 224, 22 November 1979, pp. 779-780.

CUMMINS, H. & MIDLO, C. 1943. Finger prints, palms and soles: an introduction to dermatoglyphics. Dover Publications Inc, New York 1961. (First published 1943, the Blakiston Company).

DE BRUYNE, P. & ORUM, J. 1982. Reducing storage requirements of digitised fingerprint images. 1982 Carnahan Conference on Security Technology, University of Kentucky, Lexington, Kentucky, 12-14 May 1982, pp. 1-6.

DEUTSCH, E.S. 1972a. Towards isotropic image reduction. Information Processing 71, pp. 161-172. North Holland Publishing Company.

DEUTSCH, E.S. 1972b. Thinning algorithms on Rectangular, Hexagonal, and Triangular Arrays. Communications of the ACM, Vol. 15, No. 9, September 1972, pp. 827-837.

DODSON, C.T.J. 1980. Categories, Bundles and Spacetime Topology. Shiva Publishing Company.

ELECCION, M. 1973. Automatic fingerprint identification. IEEE Spectrum, September 1973, pp. 36-45.

FAULDS, H. 1880. On the Skin-furrows of the Hand. Nature, Vol. 22, 28 October 1880, page 605.

FBI. The Science of Fingerprints: Classification and Uses. FBI, US Department of Justice. Not dated.

FERNANDO, G.M.X. 1985. Variable thresholding of images with application to ventricular angiograms. PhD thesis, Department of Electrical Engineering, Imperial College, London, April 1985.

FREEDMAN, M.D., & HIETANEN, E.D. 1967. Application of parallel neighbourhood logic to fingerprint processing. International Symposium on Law Enforcement Science and Technology, pp. 507-509.

FREEMAN, H. 1974. Computer processing of line-drawing images. Computing Surveys, Vol. 6, No. 1, March 1974, pp. 57-97.

GALTON, F. 1892. Finger Prints. MacMillan and Co., London.

GECKINLI, N.C. & YAVUZ, D. 1978. Some Novel Windows and a Concise Tutorial Comparison of Window Families. IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-26, No. 6, December 1978, pp. 501-507.

GODBILLON, C. 1983. Dynamical Systems on Surfaces. Springer.

GONZALES, R.C. & WINTZ, P. 1977. Digital Image Preocessing. Addison Wesley: Applied Mathematics and Computation Textbooks, No. 13. Revised 1979.

GRANLUND, G.H. 1978. In Search of a General Picture Processing Operator. Computer graphics and image processing, Vol. 8, pp. 155-173.

GRANLUND, G.H. 1981. The GOP Parallel Image Processor. In Bolc, L. & Kulpa, Z. (Eds): Digital Image Processing Systems. Lecture notes in computer science, Vol. 109, Springer Verlag, pp. 201-227.

GRANLUND, G.H. & ARVIDSSON, J. 1983. The GOP image computer. In Faugeras, O.D. (Ed.): Fundamentals in computer vision: an advanced course, Cambridge University Press, pp. 443-458.

GRANLUND, G.H., KNUTSSON, H. & WILSON, R. 1983. Image Enhancement. In Faugeras, O.D. (Ed.): Fundamentals in computer vision:an advanced course, Cambridge University Press, pp. 57-67.

GRASSELLI, A. 1968. On the automatic classification of fingerprints - some considerations on the linguistic interpretation of pictures. In Watanabe, S. (Ed.): Methodologies of Pattern Recognition, Academic Press,1969. (Proceedings of the International Conference on Methodologies of Pattern Recognition, Honolulu, 1968).

GREW, DR. NEHEMIAH 1684. The description and use of the pores in the skin of the hands and feet. Philosophical Transactions of the Royal Society, No. 159, Vol. 14, 20 May 1684, pp. 566-567.

HALL, E.L. 1979. Computer image processing. Academic Press.

HANKLEY, W.J. & TOU, J.T. 1968. Automatic fingerprint interpretation and classification via contextual analysis and

topological coding. In Cheng, G.C. et al (Eds): Pictorial Pattern Recognition (Proceedings of Symposium on Automatic Photointerpretation). Thompson Book Company.

HARRIS, F.J. 1978. On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. Proceedings of the IEEE, Vol. 66, No. 1, January 1978, pp. 51-83.

HAWKING, S.W. & ELLIS, G.F.R. 1973. The Large Scale Structure of Space-Time. Cambridge University Press.

HELSTROM, C.W. 1967. Image restoration by the method of least squares. Journal of the optical society of America, Vol 57, No. 3, March 1967, pp. 297-303.

HENRY, E.R. 1900. Classification and uses of finger prints. George Routledge and Sons Limited, London. (2nd edition 1901).

HERSCHEL, W.J. 1880. Skin furrows of the hand. Nature, Vol. 23, 25 November 1880, page 76.

HOME OFFICE 1974. Conference on the science of fingerprints. Home Office, London.

HOME OFFICE 1984. Conference on the science of fingerprints. Home Office, London.

HOME OFFICE 1986. Manual of Fingerprint Development Techniques. Home Office Scientific Research and Development Branch, London.

HORNER, J.L. 1969. Optical Spatial Filtering with the Least Mean Square-Error Filter. Journal of the Optical Society of America,

Vol. 59, No. 5, May 1969, pp. 553-558.

HORVATH, V.V., & DOYLE, D.D. 1968. Recent developments in fingerprint recognition using coherent optical filtering. Proceedings of the second national symposium on law enforcement science and technology, pp. 417-426.

HORVATH, V.V., HOLEMAN, J.M. & LEMMOND, C.Q. 1967. Fingerprint recognition by holographic techniques. National symposium on law enforcement science and technology, pp. 485-492.

HUANG, T.S. 1972. Two dimensional Windows. IEEE Transactions on Audio and Electroacoustics, March 1972, pp. 88-89.

HUANG, T.S., SCHREIBER, W.F., & TRETIAK, O.J. 1971. Image Processing. Proceedings of the IEEE, Vol. 59, No. 11, November 1971, pp. 1586-1609.

HUNT, B.R. 1973. The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computer. IEEE Transactions on Computers, Vol. C-22, No. 9, September 1973, pp. 805-812.

HUNT, B.R. 1975. Digital Image Processing. Proceedings of the IEEE, Vol. 63, No. 4, April 1975, pp. 693-708.

IKONOMOPOULOS, I. & KUNT, M. 1985. High Compression Image Coding via Directional Filtering. Signal Processing, Vol. 8, No. 2, April 1985, pp. 179-203.

IKONOMOPOULOS, I. & UNSER, M. 1984. A directional filtering approach to texture discrimination. Proceedings of the seventh

international conference on pattern recognition, Montreal, Canada, 30 July to 2 August 1984. IEEE Computer Society Press.

ISENOR, D.K., & ZAKY, S.G. 1986. Fingerprint identification using graph matching. Pattern Recognition, Vol. 19, No. 2, pp. 113-122.

JERRI, A.J. 1977. The Shannon sampling theorem - its various extensions and applications: a tutorial review. Proceedings of the IEEE, Vol. 65, No. 11, November 1977, pp. 1565-1596.

KANAL, L.N. 1972. Interactive Pattern Analysis and Classification Systems: A Survey and Commentary. Proceedings of the IEEE, October 1972, pp. 1200-1215.

KANAL, L.N. 1974. Patterns in Pattern Recognition: 1968-1974. IEEE Transactions on Information Theory, Vol. IT-20, No. 6, November 1974, pp. 697-722.

KAPLAN, W. 1942. The structure of a curve-family on a surface in the neighbourhood of an isolated singularity. American Journal of Mathematics, Vol. 64, pp. 1-35.

KAPLAN, W. 1955. Curve families and Riemann Surfaces. Lectures on functions of a complex variable, University of Michigan Press, pp 425-432.

KAWAGOE, M. & TOJO, A. 1984. Fingerprint pattern classification. Pattern recognition, Vol. 17, No. 3, pp. 295-303.

KIRK, R.L. & ROSENFELD, A. 1979. A Note on the Use of (Gray level, Local Average Gray Level) Space as an Aid in Threshold

Selection. IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-9, No. 12, December 1979, pp. 860-864.

KITTLER, J. & DUFF, M.J.B. 1985. Image Processing System Architecture. Research Studies Press, John Wiley.

KNUTSSON, H. 1982. Filtering and reconstruction in image processing. PhD thesis, Department of Electrical Engineering, Linkoping University, Linkoping, Sweden. (Linkopings studies in science and technology dissertations, No 88).

KNUTSSON, H. 1985. Producing a continuous and distance preserving 5-d vector representation of 3d orientation. IEEE Computer society workshop on computer architecture for pattern analysis and image database management.

KNUTSSON, H. & GRANLUND, G.H. 1980. Fourier domain design of line and edge detectors. Proceedings of fifth international conference on pattern recognition, Miami Beach, December 1-4, 1980, pp. 45-47. IEEE Publication 80CH1499-3.

KNUTSSON, H. & GRANLUND, G.H. 1983. Texture analysis using two-dimensional quadrature filters. IEEE computer society workshop on computer architecture for pattern analysis and image database management - CAPAIDM, Pasadena, California, 12-14 October 1983, pp. 206-213.

KNUTSSON, H., WILSON, R. & GRANLUND, G.H. 1981. Anisotropic filtering operations for image enhancement and their relation to the visual system. Proceedings of IEEE Computer Society Conference on Pattern Recognition and Image Processing, Dallas,

Texas, 3-5 August 1981, pp. 420-423.

KNUTSSON, H.E., WILSON, R. & GRANLUND, G.H. 1983a. Anisotropic Nonstationary Image Estimation and its Applications: PartI - Restoration of Noisy Images. IEEE Transactions on Communications, Vol. COM-31, No. 3, March 1983, pp. 388-397.

KNUTSSON, H.E., WILSON, R. & GRANLUND, G.H. 1983b. Estimating the local orientation of anisotropic 2D signals. IEEE Acoustics, Speech, and Signal Processing Spectrum Estimation Workshop II, Tampa, Folrida, 10-11 November 1983, pp. 234-239.

KUFFLER, S.W. & NICHOLLS, J.G. 1976. From neuron to brain. Sinauer Associates Inc., Sunderland, Massachusetts.

KUNT, M. 1986. Recent progress in high compression image coding. Eighth international conference on pattern recognition, Paris, France, 27-31 October 1986, pp. 1018-1025.

KUNT, M., IKONOMOPOULOS, A. & KOCHER, M. 1985. Second Generation Image Coding Techniques. Proceedings of the IEEE, Vol. 73, No. 4, April 1985, pp. 549-574.

LERNER, E.J. 1983. Sleuthing by computer. IEEE Spectrum, July 1983, pp. 44-49.

LEVI, G. & SIROVICH, F. 1972. Structural descriptions of fingerprint images. Information sciences, Vol. 4, pp. 327-355.

LIU, C.N. & SHELTON, G.L. 1970. Computer-assisted Fingerprint Encoding and Classification. IEEE Transactions on Man-Machine Systems, Vol. 11, September 1970, pp. 156-160.

LU, H.E. & WANG, P.S.P. 1985. An improved fast parallel thinning algorithm for digital patterns. Proceedings of IEEE computer society conference on computer vision and pattern recognition, San Francisco, California, 19-23 June 1985, pp. 364-367.

LU, H.E. & WANG, P.S.P. 1986. A Comment on "A Fast Parallel Algorithm for Thinning Digital Patterns". Communications of the ACM, Vol. 29, No. 3, March 1986, pp. 239-242.

McCULLOCH, W.S. & PITTS, W. 1943. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, Vol. 5, pp. 115-133.

McGINTY, L. 1974. Fingerprints - the next data in the bank. New Scientist, 31 October 1974, pp. 320-323.

McMAHON, H.H., JOHNSON, G.L., TEETER, S.L. & WHITNEY, C.G. 1975. A Hybrid Optical Computer Processing Technique for Fingerprint Identification. IEEE Transactions on Computers, Vol. C-24, No. 4, April 1975, pp. 358-369.

MARKUS, L. 1955. Line element fields and Lorentz substructures on differentiable manifolds. Annals of Mathematics, Vol. 62, No. 3, November 1955, pp. 411-417.

MAROM, E. 1967. Fingerprint classification and identification using optical methods. National Symposium on Law Enforcement Science and Technology, pp. 481-484.

MARR, D. 1982. Vision. W.H. Freeman and Co, San Francisco.

MARTELLI, A. & MONTANARI, U. 1971. Optimal smoothing in picture

processing: an application to fingerprints. Information Processing 71, Proceedings of IFIP Congress 71, Ljubljana, Yugoslavia, 23-28 August 1971, pp. 173-178.

MAVALWALA, J. 1975. Harold Cummins - and Birth, Growth and Development of Dermatoglyphics. American Journal of Physical Anthropology, Vol. 42, pp. 177-283. (Special issue devoted to dermatoglyphics).

MENZEL, E.R. 1980. Fingerprint detection with lasers. Dekker, New York.

MERSEREAU, R.M. 1981. Two-dimensional nonrecursive filter design. In Huang, T.S.(Ed.): Two-Dimensional Digital Signal Processing I: Linear Filters. Springer-Verlag.

MILLARD, K. 1974. An automatic retrieval system for scene of crime fingerprints. Conference on the science of fingerprints, London, pp. 1-14.

MILLARD, K. 1983. Developments on automatic fingerprint recognition. International Carnahan conference on security technology, Zurich, 4-6 October 1983, pp. 173-178.

MILLARD, K. 1987. Personal communication, August 1987. Home Office SRDB, Horseferry House, Dean Ryle St, London SW1.

MOAYER, B. & FU, K.S. 1973. A syntactic approach to fingerprint pattern recognition. (IEEE) International Joint Conference on Pattern Recognition, pp. 423-432.

MOAYER, B. & FU, K.S. 1975. A syntactic approach to fingerprint

pattern recognition. Pattern Recognition, Vol. 7, pp. 1-23.

MOAYER, B. & FU, K.S. 1976a. An application of stochastic languages to fingerprint pattern recognition. Pattern Recognition, Vol. 8, pp. 173-179.

MOAYER, B. & FU, K.S. 1976b. A tree system approach for fingerprint pattern recognition. IEEE Transactions on Computers, Vol. C-25, March 1976, pp. 262-274.

MOENSSENS, A.A. 1971. Fingerprint techniques. Inban law enforcement series, Chilton.

MONRO, D.M., SHERLOCK, B.G. & PETTS, C.R. 1987. Fingerprint enhancement by Fourier domain optical processing. Topical Meeting on Optical Computing, Lake Tahoe, 16-18 March 1987, pp. 245-248. Technical Digest Series Vol. 11 (Optical Society of America).

MONTAGNA, W. 1956. The structure and function of skin. Academic Press Inc., New York.

MONRO, D.M. & BRANCH, J.L. 1977. The Chirp Discrete Fourier Transform of General Length. Applied Statistics, Vol. 26, pp. 351-361. Algorithm AS117.

MONRO, D.M. 1979. Interpolation by Fast fourier and Chebyshev transforms. International Journal for Numerical Methods in Engineering, Vol. 14, pp. 1679-1692.

MONRO, D.M. 1980. Interpolation methods for surface mapping. Computer programs in biomedicine, Vol. 11, No. 2, April 1980, pp. 145-157.

NAKAGAWA, Y. & ROSENFELD, A. 1979. Some experiments on variable thresholding. Pattern Recognition, Vol. 11, pp. 191-204.

NAKAMURA, O., GOTO, K., & MINAMI, T. 1982. Fingerprint classification by directional distribution patterns. Systems • Computers • Controls, Vol. 13, No. 5, pp. 81-89.

NAKAMURA, O., OKAMOTO, N., MINAMI, T. & OKUNO, H. 1981. Structural Matching of Gray Pictures Described by Minimal Spanning Tree. Systems • Computers • Controls, Vol. 12, No. 6, pp. 77-85.

O'CONNOR, K. 1988. Image segmentation through optimal tessellation. PhD thesis, Department of Computing, Imperial College, London, January 1988.

OGAWA, H. & TANIGUCHI, K. 1982. Thinning and stroke segmentation for handwritten Chinese character recognition. Pattern Recognition, Vol. 15, No. 4, pp. 299-308.

OPPENHEIM, A.V. & SCHAFER, R.W. 1975. Digital Signal Processing. Prentice-Hall Inc.

PAPOULIS, A. 1965. Probability, Random Variables and Stochastic Processes. McGraw Hill.

PAVLIDIS, T. 1980. A Thinning Algorithm for Discrete Binary Images. Computer graphics and image processing, Vol. 13, pp. 142-157.

PENROSE, L.S. 1963. Finger-prints, palms and chromosomes. Nature, No. 4871, Vol. 197, 9 March 1963, pp. 933-938.

PENROSE, L.S. 1965a. Dermatoglyphic Topology. Nature, No.4971, Vol. 205, 6 February 1965, pp. 544-546.

PENROSE, L.S. 1965b. On the geometry of loops and deltas. Annals of Human Genetics, London, Vol. 29, pp. 207-211.

PERNUS, F., KOVACIC, S. & GYERGYEK, L. 1980. Minutiae based fingerprint registration. (IEEE) International joint conference on Pattern Recognition, pp. 1380-1382.

PRATT, W.K. 1972. Generalized Wiener Filtering Computation Techniques. IEEE Transactions on Computers, July 1972, pp. 636-641.

PRATT, W.K. 1978. Digital image processing. Wiley.

RABINER, L.R. & GOLD, B. 1975. Theory and Applications of Digital Signal Processing. Prentice Hall, Inc., Englewood Cliffs, N.J.

RABINER, L.R., COOLEY, J.W., HELMS, H.D., JACKSON, L.B., KAISER, J.F., RADER, C.M., SCHAFER, R.W., STEIGLITZ, K. & WEINSTEIN, C.J. 1972. Terminology in Digital Signal Processing. IEEE Transactions on Audio and Electroacoustics, Vol. AU-20, No. 5, December 1972, pp. 322-337.

RABINER, L.R., SCHAFER, R.W. & RADER, C.M. 1969. The Chirp z-Transform Algorithm. IEEE Transactions on Audio and Electroacoustics, Vol. AU-17, June 1969, pp. 86-92.

RABINOWITZ, A.S. 1980. Fingerprint Card Search Results with Ridge-Contour Based Classification. (IEEE) International joint conference on pattern recognition, Miami Beach, Florida, 1-4

December 1980, pp. 475-477.

RAO, T.C.M. 1976. Feature extraction for fingerprint classification. Pattern Recognition, Vol. 8, pp. 181-192.

RAO, C.V.K. 1978. On fingerprint pattern recognition. Pattern Recognition, Vol. 10, pp. 15-18.

RAO, C.V.K. & BALCK, K. 1976. Type classification of fingerprints: a syntactic approach. (IEEE) Third international joint conference on pattern recognition, pp. 778-782.

RAO, C.V.K. & BALCK, K. 1978. Finding the Core Point in a Fingerprint. IEEE Transactions on Computers, Vol. C-27, No. 1, January 1978, pp. 77-81.

REDGERS, A. 1987. Adrian's guide to understanding the WISARD. Neural Computing Group, Department of Computing, Imperial College, London, December 1987.

REID, W.T. 1971. Ordinary differential equations. John Wiley.

RIGANATI, J.P. 1977. An overview of algorithms employed in automated fingerprint processing. Proceedings of international conference on crime countermeasures - science and engineering, University of Kentucky, 25-29 July 1977, pp. 125-131.

RINO, C.L. 1969. Bandlimited Image Restoration by Linear Mean-Square Estimation. Journal of the Optical Society of America, Vol. 59, No. 5, May 1969, pp. 547-553.

ROSENFELD, A. 1970. Connectivity in Digital Pictures. Journal of

the ACM, Vol. 17, No. 1, January 1970, pp. 146-160.

ROSENFELD, A. 1975. A Characterization of Parallel Thinning Algorithms. Information and Control, Vol. 29, pp. 286-291.

ROSENFELD, A. 1978. Iterative methods in image analysis. Pattern Recognition, Vol. 10, pp. 181-187.

ROSENFELD, A. & DAVIS, L.S. 1976. A Note on Thinning. IEEE Transactions on Systems, Man, and Cybernetics, March 1976, pp. 226-228.

ROSENFELD, A. & KAK, A.C. 1976. Digital picture processing. Academic Press.

SCHAFER, R.W. & RABINER, L.R. 1974. A Digital Signal Processing Approach to Interpolation. Proceedings of the IEEE, Vol. 62, October 1974, pp. 1319-1338.

SCHERLOWSKI, L.A. 1974. Studies on the application of spatial filtering to enhance low contrast fingerprint images. Conference on the science of fingerprints, Home Office, London, pp.185-221.

SHELMAN, C.B. 1967. Machine Classification of Fingerprints. National Symposium on Law Enforcement Science and Technology, pp. 467-477.

SHELMAN, C.B. 1968. Machine extraction of ridge endings from a noisy fingerprint. Proceedings of second national symposium on law enforcement science and technology, pp. 409-416.

SHELMAN, C.B. & HODGES, D. 1973a. Fingerprint research at Argonne national laboratory. Proceedings of Carnahan conference on

electronic crime countermeasures, Lexington, Kentucky, April 1973, pp. 108-113.

SHELMAN, C.B. & HODGES, D. 1973b. A decimal Henry system. First international conference on electronic crime countermeasures, Edinburgh University, Edinburgh, Scotland, 18-20 July 1973, pp. 213-220.

SHIZUME, P.K. & HEFNER, C.G. 1978. A computer technical fingerprint search system. Proceedings of Carnahan conference on crime countermeasures, University of Kentucky, Lexington, Kentucky, pp. 121-129.

SINGH, V.K., GYERGYEK, L. & PAVESIC, N. 1977. Feature recognition and identification in fingerprint patterns. Proceedings of Carnahan conference on crime countermeasures, University of Kentucky, Lexington, Kentucky, pp. 241-248.

SLEPIAN, D. 1967. Linear Least-Squares Filtering of Distorted Images. Journal of the Optical Society of America, Vol. 57, No. 7, July 1967, pp. 918-922.

SOLOSKO, R.B. & PALEY, J.J. 1973. A semi-automated computer fingerprint encoding and matching system. Proceedings of Carnahan conference on electronic crime countermeasures, Lexington, Kentucky, April 1973, pp. 114-122.

SONDHI, M.M. 1972. Image Restoration: The Removal of Spatially Invariant Degradations. Proceedings of the IEEE, Vol. 60, No. 7, July 1972, pp. 842-853.

SPARROW, M.K. 1986. Topological Coding of Single Fingerprints. PhD Thesis, University of Kent at Canterbury.

SPARROW, M.K. & SPARROW, P.J. 1985a. A topological approach to the matching of single fingerprints: development of algorithms for use on rolled impressions. NBS SP 500-124, May 1985.

SPARROW, M.K. & SPARROW, P.J. 1985b. A topological approach to the matching of single fingerprints: development of algorithms for use on latent fingerprints. NBS SP 500-126, October 1985.

SPIVAK, M. 1979. A comprehensive introduction to differential geometry. (5 Volumes) Publish or Perish, Berkeley.

STEFANELLI, R. & ROSENFELD, A. 1971. Some Parallel Thinning Algorithms for Digital Pictures. Journal of the ACM, Vol. 18, No. 2, April 1971, pp. 255-264.

STONEY, D.A. & THORNTON, J.I. 1986. A Critical Analysis of Quantitative Fingerprint Individuality Models. Journal of Forensic Sciences, JFSCA, Vol. 31, No. 4, October 1986, pp. 1187-1216.

STOKER, J.J. 1969. Differential Geometry. Wiley-Interscience.

STROKE, G.W. 1972. Optical Computing. IEEE Spectrum, December 1972, pp. 24-41.

SUZUKI, S. & ABE, K. 1986. Sequential thinning of binary pictures using distance transformation. Eighth International Conference on Pattern Recognition, Paris, France, 27-31 October 1986, pp. 289-292.

SWONGER, C.W. 1973. Applications of fingerprint identification

technology to criminal identification and security systems. Proceedings of Carnahan conference on crime countermeasures, University of Kentucky, Lexington, Kentucky, pp. 190-212.

TRAURING, M. 1963. Automated comparison of finger-ridge patterns. Nature, No. 4871, Vol. 197, 9 March 1963, pp. 938-940.

TRIBOLET, J.M. 1977. A New Phase Unwrapping Algorithm. IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-25, No. 2, April 1977, pp. 170-177.

TSAI, W-H. 1985. Moment preserving thresholding. Computer Vision, Graphics and Image Processing, Vol. 29, pp. 377-393.

TUKEY, J.W. 1967. An introduction to the calculation of numerical spectral analysis. In Harris, B. (Ed.): Spectral Analysis of Time Series, Wiley, pp. 25-46.

VAN VALKENBURG, M.E. 1982. Analog Filter Design. Holt, Rinehart and Winston (HRW series in electrical and computer engineering).

WEGSTEIN, J.H. 1982. An Automated Fingerprint Identification System. NBS SP 500-89, National Bureau of Standards, US Department of Commerce, February 1982.

WESZKA, J.S. 1978. A Survey of Threshold Selection Techniques. Computer Graphics and Image Processing, Vol. 7, pp. 259-265.

WESZKA, J.S. & ROSENFELD, A. 1978. Threshold Evaluation Techniques. IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-8, No. 8, August 1978, pp. 622-629.

WHITNEY, H. 1933. Regular families of curves. Annals of Mathematics (2), Vol. 34, pp. 246-270.

WICHMANN, B.A. & HILL, I.D. 1982. A Pseudo-Random Number Generator. National Physical Laboratory, NPL Report DITC 6/82, June 1982.

WILSON, R., KNUTSSON, H.E. & GRANLUND, G.H. 1983. Anisotropic Nonstationary Image Estimation and its Applications: Part II - Predictive Image Coding. IEEE Transactions on Communications, Vol. COM-31, No. 3, March 1983, pp. 398-406.

WILSON, R., KNUTSSON, H.E. & GRANLUND, G.H. 1985. Image orientation estimation and its applications. IEE Electronics Division, Colloquium on Image Restoration and 2D Filtering Techniques, London, 3 May 1985, pp. 3/1 - 3/4. (Digest No. 1985/51, Professional Groups E4 and C8).

WILSON, R. & GRANLUND, G.H. 1984. The Uncertainty Principle in Image Processing. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-6, No. 6, November 1984, pp. 758-767.

WILSON, M.J.D. 1985. Stereopsis with adaptive windows. Electronics Letters, Vol. 21, No. 16, 1 August 1985, pp. 699-701.

XIA, Y. 1986. A new thinning algorithm for binary images. Eighth International Conference on Pattern Recognition, Paris, France, 27-31 October 1986, pp. 995-997.

YOUNG, T.Y & FU, K.S. 1986. Handbook of Pattern Recognition and Image Processing. Academic Press Inc.

ZHANG, T.Y. & SUEN, C.Y. 1984. A Fast Parallel Algorithm for Thinning Digital Patterns. Communications of the ACM, Vol. 27, No. 3, March 1984, pp. 236-239.

ZUCKER, S.W. 1985. Early Orientation Selection: Tangent Fields and the Dimensionality of Their Support. Computer Vision, Graphics and Image Processing, Vol. 32, pp. 74-103.

ZUCKER, S.W. 1986. Early Orientation Selection: Inferring Trace, Tangent, and Curvature Fields. Eighth International Conference on Pattern Recognition, Paris, France, 27-31 October 1986, pp. 294-301.